ABSTRACT

Title of Dissertation: AFFECTIVE HUMAN MOTION DETECTION AND SYNTHESIS

Uttaran Bhattacharya
Doctor of Philosophy, 2022

Dissertation Directed by: Dr. Dinesh Manocha
Department of Computer Science

Human emotion perception is an integral component of intelligent systems currently being designed for a wide range of socio-cultural applications, including behavior prediction, social robotics, medical therapy and rehabilitation, surveillance, and animation of digital characters in multimedia. Human observers perceive emotions from a number of cues or *modalities*, including faces, speech, and body expressions. Studies in affective computing indicate that emotions perceived from body expressions are extremely consistent across observers because humans tend to have less conscious control over their body expressions. Our research focuses on this aspect of emotion perception as we attempt to build predictive methods for automated emotion recognition from body expressions, and build generative methods for synthesizing digital characters with appropriate affective body expressions. This thesis elaborates on both components of our research in two parts, and explores how they can be applied to current problems in video understanding, specifically video highlight detection.

The first part discusses two approaches for designing and training partially supervised methods for emotion recognition from body expressions, specifically gaits. In one approach, we leverage existing gait datasets annotated with emotions to generate large-scale synthetic gaits corresponding to the emotion labels. In the other approach, we leverage large-scale unlabeled gait datasets together with smaller annotated gait datasets to learn meaningful latent representations for emotion recognition. We design an autoencoder coupled with a classifier to learn latent representations for simultaneously reconstructing all input gaits and classifying the labeled gaits into emotion classes.

The second part discusses generative methods to synthesize emotionally expressive bodily expressions, specifically gaits, gestures, and faces. The first method involves asynchronous generation, where we synthesize only one modality of the digital characters (in our case, gaits) with affective expressions. Our approach is to design an autoregression network that takes in a history of the characters' pose sequences and the intended future emotions to generate their future pose sequences with the desired affective expressions. The second method is the more challenging synchronous generation, where the affective contents of two modalities, such as body gestures and speech, need to be synchronized with each other. Our approach utilizes machine translation to translate from speech to body gestures, and adversarial discrimination to differentiate between original and synthesized gestures in terms of affective expressions, to produce state-of-the-art affective body gestures synchronized with speech. The final method takes synchronous generation a step further to three modalities, involving the synthesis of both facial expressions and body gestures synchronized with speech. This method attempts to break new ground in multimodal synthesis by simultaneously incorporating emotional expressions in

more than one modality, and does so using data from affordable, consumer-grade devices such as RGB video cameras to enable democratized usage.

Lastly, we explore the application of these approaches to industrial problems in video understanding, specifically video highlight detection. Our approach leads to state-of-the-art performance in detecting highlights in human-centric videos without requiring supervision in the form of highlight annotations. Our approach can be further fine-tuned to detect user-specific highlights at scale by automatically learning the video contents matching the users' preferences in their previously selected highlight clips.

AFFECTIVE HUMAN MOTION DETECTION AND SYNTHESIS

by

Uttaran Bhattacharya

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2022

Advisory Committee:
Dr. Dinesh Manocha, Chairperson
Dr. Jae Shim, Dean's Representative
Dr. Ming Lin
Dr. Huaishu Peng
Dr. Aniket Bera
Dr. Viswanathan Swaminathan

Dedicated To

The memory of Sri Kamalesh Chandra Chakraborty, who always believed that I could.

Acknowledgments

The Ph.D. journey has been long and intense. I thank my advisor Prof. Dinesh for all his valuable guidance and inspiration. The tireless late-nights we spent debugging and polishing our methods, revising our papers, and responding to our peer reviews, both nice and frank, are as memorable now as they were stimulating at the time. I am grateful to Aniket for his constant mentorship and companionship. He has always been ready to have a chat with me, be it about work, personal life, or something that either of us had seen on the news on any particular day. And I sincerely thank my entire thesis committee, counting in Profs. Jae, Ming and Huaishu, and Vishy, for taking the time to read and evaluate my thesis and providing their valuable feedback throughout.

Having spent close to two years interning and collaborating with Adobe Research during my Ph.D., I am thankful to my mentors Gang and Stefano for their help, guidance and support. I was a complete newcomer to the world of industrial research, and they not only helped me navigate all its nuances, but broadened my thought process to look beyond problem statements and look for impact problems.

I have also been lucky to have been supported by a number of funding sources during my Ph.D., including Intel grants, ARO grants W911NF-19-1-0069, W911NF-19-1-0315 and W911NF-21-1-0026, the Dean's fellowship from the University of Maryland, and the Adobe Research fellowship.

Any journey is incomplete without the friends who make it worth the effort, and I have had many good friends at GAMMA and others in and outside the University of Maryland. Adarsh, Senthil, Utsav and Kasun helped take away a lot of the stress of the Ph.D. grind through our mutual sharing and reassurances. Divya, Geonsun, Puneet, and Vishnu were always happy to jump into discussions about all our works, which gave me fresh perspectives and refreshing breaks from the monotony of work. Zhenyu was a year ahead of me in the doctoral program, so he became my de-facto source of guidance to find my way around as an international student in the US. Shib, Dhawal, and John made my only in-person internship experience during my Ph.D. fun and exciting, and their company was as warm as the California summer. More than just friends, I have also worked with some of the GAMMA members as co-authors, including Rohan, Trisha, Tianrui, Christian, Pooja, Niall, Nick, and Abhishek. I would also like to specially mention Tanmay, Kyra, and Prof. Kurt as co-authors who introduced me to the domain of affect detection from body expressions. I have had some great experiences collaborating with all my co-authors, discussing ideas, methods, and results feeding back into new ideas.

Wrapping all the parts of my Ph.D. journey together, I want to thank my family, including my parents and my wife Anindita, who have always believed in me, stayed with me through my lows and celebrated my highs, and encouraged me to keep trying my best.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction and Overview

Human emotion or, more generally, *affect* understanding is an integral component of intelligent systems currently being designed for a wide range of industrial, social and cultural applications, including video understanding (Bhattacharya et al. 2021c; Bhattacharya et al. 2022) behavior prediction (Allen, Machleit, and Kleine 1992; Denham et al. 2000), teaching and counseling in the digital world (Baur et al. 2013; Li et al. 2016; Liao et al. 2019; Simeone et al. 2019), social robotics (Bauer et al. 2009; Narayanan et al. 2020), medical therapy and rehabilitation (DeVault et al. 2014; Rivas et al. 2015), and animation of digital characters in multimedia (Roth et al. 2016; Heidicker, Langbehn, and Steinicke 2017; Latoschik et al. 2017). Existing research has extensively explored automated affect understanding from a number of cues, including faces (Fan et al. 2016; Hu et al. 2017; Yang, Ciftci, and Yin 2018; Zhang et al. 2018a), eye movements (Schur-

gin et al. 2014; Lu et al. 2015), speech (Rao, Koolagudi, and Vempada 2013; Jacob and Mythili 2015), written text (Santos, Nedjah, and Macedo Mourelle 2017), and even physiological signals such as heartbeats and respiration rates (Chanel et al. 2006; Zhao, Adib, and Katabi 2016). At the same time, studies in affective computing indicate that body expressions, particularly gestures and gaits, are integral sources of affect cues for humans (Montepare, Goldstein, and Clausen 1987; Meeren, Heijnsbergen, and Gelder 2005b; Michalak et al. 2009; Kleinsmith and Bianchi-Berthouze 2013), and are arguably more reliable than other cues as humans tend to have less voluntary control over them (Quigley, Lindquist, and Barrett 2014). However, automated techniques for affect understanding from these body expressions have been largely unexplored, primarily due to the lack of both systematically collected large-scale data and robust predictive and generative models for learning from that data. Our research focuses on this frontier as we attempt to build predictive models for automated affect detection from body expressions, and build generative models for synthesizing digital characters with appropriate affective body expressions to augment real-world data.

In essence, our research encompasses the two main kinds of machine learning tasks in the context of affect understanding (Figure 1.1). The first kind involves designing and training methods for automated affect detection from body expressions, specifically gaits. The second kind involves building generative methods to synthesize emotionally expressive or *affective* body expressions, specifically gaits and gestures. We discuss each of these two kinds of affect understanding tasks and how we apply them to broader video understanding problems at the industrial scale. As a use-case for our application to industrial scale video understanding, we consider the problem of highlight detection.

1.1 Partially Supervised Methods for Affect Detection from Gaits

A gait is an ordered temporal sequence of joint transformations, predominantly rotations and translations, during bipedal locomotion. In its simplest sense, a person's gait is the way that person walks. Gaits have been widely used in computer vision for many applications, including action detection (Yan, Xiong, and Lin 2018; Shi et al. 2019a; Shi et al. 2019b; Liu et al. 2020) and perceiving emotions (Randhavane et al. 2019a; Randhavane et al. 2019c; Bhattacharya et al. 2020; Mittal et al. 2020b). Moreover, studies in psychology and affective computing indicate that gaits are also useful for affect understanding. Participants of these studies were reported to have observed *affective features* – physiological features such as arm swings, head jerks, stride lengths, and upper body posture – to identify a variety of affects, including anger, happiness, pride, and sadness. (Michalak et al. 2009; Kleinsmith and Bianchi-Berthouze 2013). However, there are a few key challenges in designing automated methods for affect detection using gaits:

- Methods based on hand-crafted physiological features extracted from human gaits often suffer from low prediction accuracy (Karg, Kuhnlenz, and Buss 2010; Venture et al. 2014; Crenn et al. 2016; Wang, Enescu, and Sahli 2016; Daoudi et al. 2017).
- Trainable methods (Randhavane et al. 2019a; Bhattacharya et al. 2020) rely heavily on sufficiently large sets of annotated data. Annotations are expensive and tedious to collect due to the variations in scales and motion trajectories (Ahsan, Sun, and Essa 2018), as well as the inherent subjectivity in perceiving affects (Bhattacharya et al. 2020).
- Conditional generative methods are useful for data augmentation, but current methods can only generate data for short time periods (Holden, Saito, and Komura 2016; Khodabandeh et al. 2018; Yang et al. 2018a) or with relatively low diversity (Pavllo, Grangier,

and Auli 2018; Yan et al. 2019b; Bhattacharya et al. 2020).

On the other hand, acquiring poses from videos and motion-capture data is cheap and efficient, leading to the availability of large-scale pose-based datasets (Ionescu et al. 2014; Shahroudy et al. 2016; Carreira and Zisserman 2017; CMU-MOCAP 2018). Given the availability of these unlabeled gait datasets and the sparsity of gaits labeled with perceived affects, there is a need to develop partially supervised methods that can utilize these datasets for affect detection.

Per our research, we propose two approaches for affect detection from gaits. The first approach consists of a conditional generative network whose trained outputs are augmented to an annotated gait dataset and utilized to train a supervised classifier (Bhattacharya et al. 2020). The main contributions include:

- **A Conditional Variational Autoencoder (CVAE)**, which is trained on a sparse real-world annotated gait set and can generate annotated synthetic gaits. We add temporal constraints, gait drift and gait collapse, as a novel push-pull regularization to the loss function of the CVAE. This formulation helps to avoid over-fitting by generating more plausible gaits.
- **An end-to-end Spatial-Temporal Graph Convolution-Based Network**, which learns from peoples' gaits to predict perceived affects. This network combines deeply learned features with affective features to learn hybrid features.
- **A new dataset of human gaits annotated with affect labels**, called Emotion-Gait (E-Gait). It currently consists of $4,227$ real-world gait videos annotated with four affect labels, happy, sad, angry, and neutral.

The second approach couples the generative and the classification components of the first ap-

proach into a single network architecture that can leverage both labeled and unlabeled gait data to learn rich latent representations for affect detection (Bhattacharya et al. 2020). Specifically, it consists of:

- **A semi-supervised network**, consisting of an autoencoder and a classifier, that we train jointly to predict discrete perceived affects from 3D pose sequences of gaits of humans.
- **A hierarchical attention pooling module** on the autoencoder to learn useful embeddings for unlabeled gaits, which improves the mean average precision (mAP) in classification by 1–17% on the absolute compared to state-of-the-art methods in both affect detection and action detection from 3D gaits on the Emotion-Gait benchmark dataset.
- **Subsumption of the affective features** expressed from the input gaits in the space of learned embeddings. This improves the mAP in classification by 7–23% on the absolute compared to state-of-the-art methods.

1.2 Generative Methods for Synthesizing Affective Body Expressions

As the world increasingly uses digital and virtual platforms for everyday communication and interactions, there is a heightened need to create highly realistic virtual agents endowed with social and emotional intelligence. Virtual agents correspond to embodied digital characters that are often used as avatars to represent users and may look like real-world characters. Recent work in high-quality rendering and capturing technologies has resulted in the ability to generate agents or avatars that closely resemble humans and are widely used in VR and AR systems (Gonzalez-Franco et al. 2020; *NEON: https://www.neon.life/* 2020). Many applications, including online learning (Li et al. 2016; Liao et al. 2019; Simeone et al. 2019), virtual interviewing and counseling (Baur et al. 2013; DeVault et al. 2014), virtual social interactions (Roth et al.

2016; Heidicker, Langbehn, and Steinicke 2017; Latoschik et al. 2017), and large-scale virtual worlds (Oculus 2019), need a computationally created virtual agent or an avatar that not only looks like a real human but also behaves like one and conveys affects (Roth et al. 2016; Randhavane et al. 2019b). To this end, we have explored two sub-problems: asynchronous generation, where we synthesize only one modality of the digital characters — in our case, body expressions — with affective expressions, and synchronous generation, where we need to synchronize two or more modalities with each other, such as body expressions and speech, in terms of their affective expressions.

For asynchronous generation, we have explored the problem of generating affective gaits for virtual agents (Bhattacharya et al. 2020). The main contributions include:

- **An autoregression network** that takes in 3D pose sequences of virtual agents' gaits, the desired future trajectory, and the desired affects. It outputs the virtual agents' gaits expressing the given affect while following the given trajectory.
- **A training method that combines the agents' pose-based affective features with their trajectory-based movement** into a unified network to generate plausible, emotionally-expressive gaits.
- **A transition scheme** for the virtual agents to smoothly transition between gaits expressing different affects.
- **A web-based user study** to evaluate the benefits of the emotive gaits generated by the proposed approach. The participants reported the affects they perceived from the generated gaits, and the Likert scale (LS) values of pose affective features that contributed to their perception. The results of the user study indicate that

- there is strong statistical evidence to suggest that the participants' perceived affects are statistically similar to the corresponding intended affects of the virtual agents, thereby showing that the generated gaits are emotive,
- the participants consistently reported different LS values of the pose-based affective features for different affects, highlighting the benefits of using pose-based affective features for synthesizing affective expressiveness.

For synchronous generation, we worked on synthesizing co-speech gestures with affective expressions (Bhattacharya et al. 2021b; Bhattacharya et al. 2021a), and synthesizing affective "unified expressions" of faces and body gestures, both synced with the speech. Co-speech gestures are bodily expressions associated with a person's speech (Yoon et al. 2019). They help underline the subject matter and the context of the speech, particularly in the form of beat, deictic, iconic, or metaphoric expressions (McNeill 1992). Beat gestures are rhythmic movements following the speech, and deictic gestures point to an entity. Iconic gestures describe physical concepts, *e.g.*, spreading and contracting the arms to denote "large" and "small", and metaphoric gestures describe abstract concepts, *e.g.*, putting a hand to the heart to denote "love". Synthesizing co-speech gestures is an important task in creating socially engaging characters and virtual agents. These are useful in a variety of multimedia application such as online learning (Li et al. 2016; Liao et al. 2019; Simeone et al. 2019), interviewing and counseling (Baur et al. 2013; DeVault et al. 2014), robot assistants (Yoon et al. 2019), character designs and game development (Mascarenhas et al. 2018; Kucherenko et al. 2020), and story- and script-visualizations (Watson et al. 2019).

We explore two approaches for affective co-speech gesture synthesis. The first approach is an

end-to-end trainable generative network that produces affective body gestures aligned with natural language text transcripts corresponding to the speech. We consider only the text transcripts and not the raw speech waveforms as input in this approach. The speech waveforms can be noisy and introduce errors in the machine translation process if the spoken words are not correctly understood. On the other hand, text transcripts provide an error-free representation of the input and therefore facilitate experimenting with the translation capabilities of a machine translation network. The method we explore is aimed at interactive applications in which virtual agents are in either one-way (*e.g.*, narration) or two-way (*e.g.*, conversation) communication with real users. The main contributions include:

- **Developing a transformer-based machine translation network** that interactively takes in text one sentence at a time and generates 3D pose sequences for virtual agents corresponding to gestures aligned with that text.
- **Conditioning the generation process** to follow the intended acting task of narration or conversation and the virtual agents' intended gender and handedness.
- **Leveraging the intended affect in the text** to generate affective gestures.
- **Conducting a web-based user study** to evaluate the quality of the generated gestures compared to motion-captured sequences and the affective expressiveness of the generated gestures.

While text transcripts are error-free representations of the input content, the speech waveforms contain the beats and the speech patterns, *e.g.*, prosody and intonations, with which body gestures tend to synchronize in terms of both rhythmic movements and affective expressions. Therefore, in our second approach, we explore a method for synthesizing affective co-speech

gestures based on both the speech waveforms and the text transcripts as inputs. The main contributions include:

- **Synthesizing co-speech affective gestures** given a speaker's speech and the corresponding text transcript while maintaining the speakers' individual styles of gesticulation and following a short sequence of seed poses.
- **Designing an affective encoder for learning latent affective features** by leveraging the localized joint movements and the macroscopic body movements in the gestures. The latent affective features are useful for both synthesizing the future poses from the seed poses and adversarially guiding the synthesis based on affective expressions.
- **Designing an MFCC encoder** for leveraging the affective cues from the speech. The MFCC encoder takes in low-dimensional MFCCs containing information on the affective cues from the speech, including prosody and intonations, and transforms them into latent embeddings for affective gesture synthesis.

Lastly, we propose an approach to simultaneously synthesize co-speech, affective facial and body expressions, which we term "unified expressions". It involves learning the multimodal distribution of the speech, the faces and the body gestures, to synthesize expressions that are mutually coherent. The main contributions include:

- **Unified co-speech face and pose expression synthesis.** To the best of our knowledge, our method is the first to simultaneously synthesize affective face and upper-body pose expressions given speech audio. We also show that our unified synthesis approach provides measurable benefits over trivially combining the synthesized outputs of the individual modalities of faces and body gestures.

- **Utilization of data from affordable commodity cameras.** In contrast to facial expression synthesis using dense 3D face scans or gesture synthesis from expensive motion-captured data, our approach only relies on face landmarks and pose joints obtainable from commodity hardware such as video cameras. As a result, our scales affordably to large datasets.
- **Plausible animations and proposed evaluation metric for facial expressions.** Through quantitative evaluations and a user study, we verify that our synthesized facial expressions and body gestures have low reconstruction error and are satisfactory to human observers.
- **An extended dataset.** We extend the current TED Gesture Dataset (Yoon et al. 2019) to include 3D face landmarks that we extract from the raw videos, denoise and correct such that the faces appear front and center and are aligned with the poses. We call this the TED Gesture+Face Dataset and release it as part of our work and the necessary source code to reuse it.

1.3 Use Case: Highlight Detection Based on Human Body Expressions

The widespread availability of cameras and the popularity of video sharing platforms has led to a proliferation of unedited videos collected today, requiring automated tools to condense videos to their most interesting or highlightable moments. This is the problem of highlight detection (Sun, Farhadi, and Seitz 2014). It facilitates and expedites video indexing, browsing, previewing, sharing, and recommending at scale (Rochan et al. 2020). We explore two key challenges in highlight detection:

- Detecting highlights in human-centric videos. Human-centric videos focus on various human activities, tasks, and emotions (Zeng 2020; Vicol et al. 2018). These videos con-

stitute a significant volume of online media (Cisco 2020) coming from multiple *domains*, including amateur sports and performances, lectures, tutorials, video weblogs (vlogs), and individual or group activities. However, current highlight detection approaches understand videos in terms of their 2D image contents (Rochan, Ye, and Wang 2018; Xiong et al. 2019), which do not explicitly leverage human activities and interactions. Moreover, these approaches require supervised annotations of highlight clips to learn from, thus being limited by the availability of such annotations.

- Detecting user-specific highlights. The notion of highlightable moments in a video is ultimately subjective to the user viewing the video. Thus, to expand their practical utility, highlight detection approaches need to scalably fine-tune to individual users' highlight preferences. However, current user-specific highlight detection approaches rely on the expensive computation of shot boundaries to divide videos into highlightable and non-highlightable segments (Molino and Gygli 2018) or uniformly pool the users' selected highlights to compute their highlight preferences, *i.e.*, assuming all their selected highlights are equally relevant (Rochan et al. 2020). In practice, however, the relevance of the users' preferences may vary significantly based on the contents of both the clips from which they selected their preferences, and the videos in which their individualized highlights need to be predicted.

We propose two approaches to address each of these challenges, both centered around the idea of detecting and learning from the human face and body expressions from the videos. The first approach (Bhattacharya et al. 2021c) is an end-to-end multimodal learning network to predict per-frame highlight scores for input videos, with the following novel contributions:

- **Highlight detection with human-centric modalities.** We detect and track human-

centric modalities, such as poses and faces, in the input videos and encode their interactions both across time and across different persons.

- **Annotation-free training of highlight scores.** We do not require highlight annotations, exemplars, user-preferences, or even domain-specific knowledge. Instead, we only leverage the detected human-centric modalities to learn the per-frame highlight scores.
- **Dataset-agnostic performance.** We train our network on generic human-centric videos, and are able to achieve state-of-the-art performance in highlight detection over a diverse range of domains and user preferences, evaluated over multiple benchmark datasets.

The second approach proposes an attention mechanism to leverage both the objects and the pose-based human activities in the users' preferred highlight clips to detect user-specific highlights in different target videos. Our main contributions include:

- **User-specific highlight detection.** We learn the relative relevance between the users' preferred highlight clips and the target videos to detect the user-specific highlights in the target videos.
- **Multi-head attention mechanism.** We design a multi-head attention mechanism that learns content-based features per user from the users' preferred highlight clips. We also learn content-based features from the target videos and query those against the per-user features to detect the user-specific highlights.
- **Reliable performance improvements.** We perform extensive experiments on the benchmark personal highlight detection dataset (PHD^2) (Molino and Gygli 2018), which contains more than 6,000 testing videos, to show that our method improves the state-of-the-art performance of both highlight detection and the related problem of video summarization.

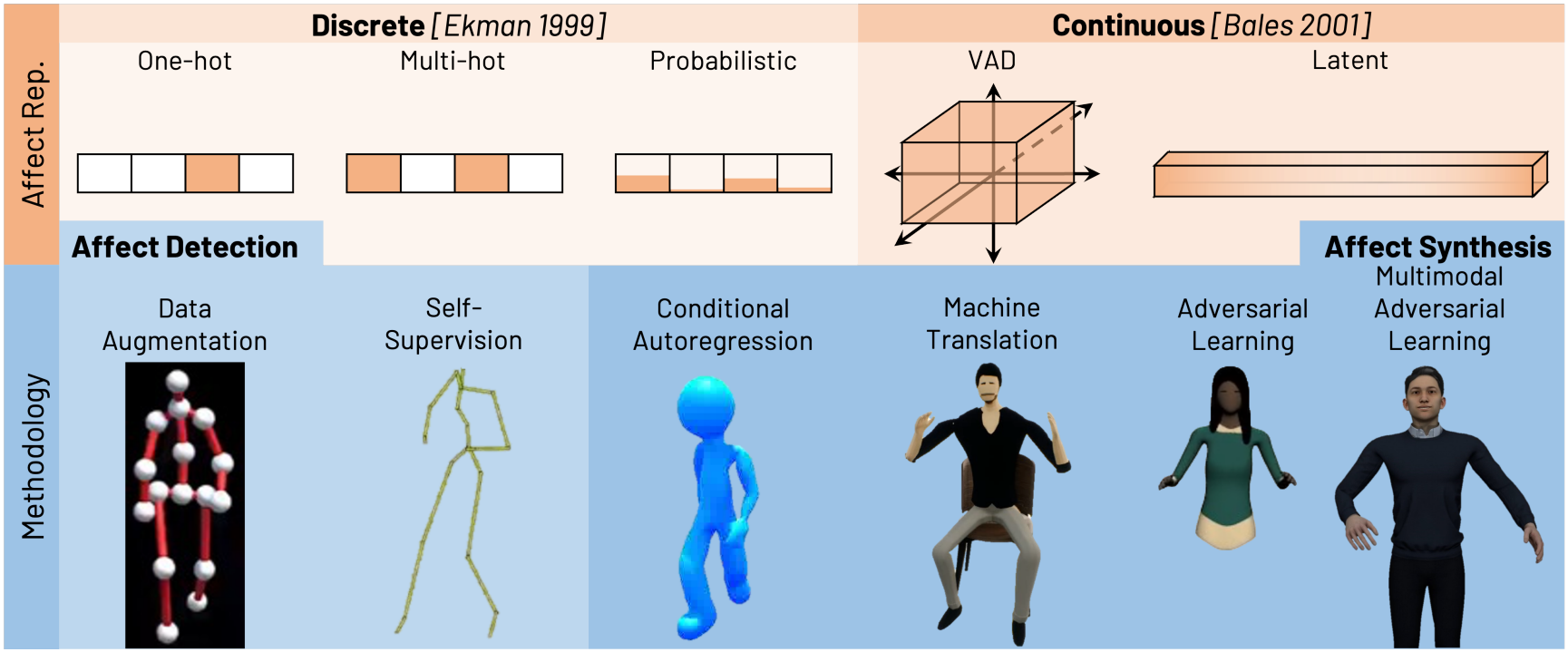


Figure 1.1: **Thesis Overview.** Our research is on designing automated techniques for affect detection and synthesis using modalities of body expressions, specifically gaits and gestures. Body expressions contain rich and reliable cues for affect understanding for human observers. However, they have been relatively unexplored for automated affect understanding tasks due to the lack of large-scale annotated data. We overcome this limitation by building predictive (lower left) and generative (lower right) models capable of learning affect-aware features from unannotated data. We also experiment with different affect representations, ranging from discrete one-hot (upper left) to continuous latent representations learned automatically from the data (upper right). Further, we synchronously combine body expressions with other modalities such as speech and facial expressions to synthesize emotionally expressive digital characters.

CHAPTER 2

Affect Detection From Gaits With Synthesized Data Augmentation

Project Website: https://gamma.umd.edu/step

Abstract

We present a novel classifier network called STEP, to classify perceived human emotion from gaits, based on a Spatial Temporal Graph Convolutional Network (ST-GCN) architecture. Given an RGB video of an individual walking, our formulation implicitly exploits the gait features to classify the perceived emotion of the human into one of four emotions: happy, sad, angry, or neutral. We train STEP on annotated real-world gait videos, augmented with annotated synthetic gaits generated using a novel generative network called STEP-Gen, built on an ST-

GCN based Conditional Variational Autoencoder (CVAE). We incorporate a novel push-pull regularization loss in the CVAE formulation of STEP-Gen to generate realistic gaits and improve the classification accuracy of STEP. We also release a novel dataset (E-Gait), which consists of 4,227 human gaits annotated with perceived emotions along with thousands of synthetic gaits. In practice, STEP can learn the affective features and exhibits classification accuracy of 88% on E-Gait, which is 14–30% more accurate over prior methods.

2.1 Introduction

Human affect understanding using intelligent systems is an important socio-behavioral task that arises in various applications, including behavior prediction (Denham et al. 2000), robotics (Bauer et al. 2009), affective computing (Yates et al. 2017; Atcheson, Sethu, and Epps 2017), etc. Current research in perceiving human emotion predominantly uses facial cues (Hu et al. 2017), speech (Jacob and Mythili 2015), or physiological signals such as heartbeats and respiration rates (Zhao, Adib, and Katabi 2016). These techniques have been used to identify and classify broad emotions including happiness, sadness, anger, disgust, fear and other combinations (Ekman and Friesen 1967a).

Understanding the perceived emotions of individuals using non-verbal cues, such as face expressions or body movement, is regarded as an important and challenging problem in both AI and psychology, especially when self-reported emotions are unreliable or misleading (Quigley, Lindquist, and Barrett 2014). Most prior work has focused on facial expressions, due to the availability of large datasets (Fabian Benitez-Quiroz, Srinivasan, and Martinez 2016). However, facial emotions can be unreliable in contexts such as referential expressions (Ekman 1993) or the presence or absence of an audience (Fernández-Dols and Ruiz-Belda 1995). Thus, we need

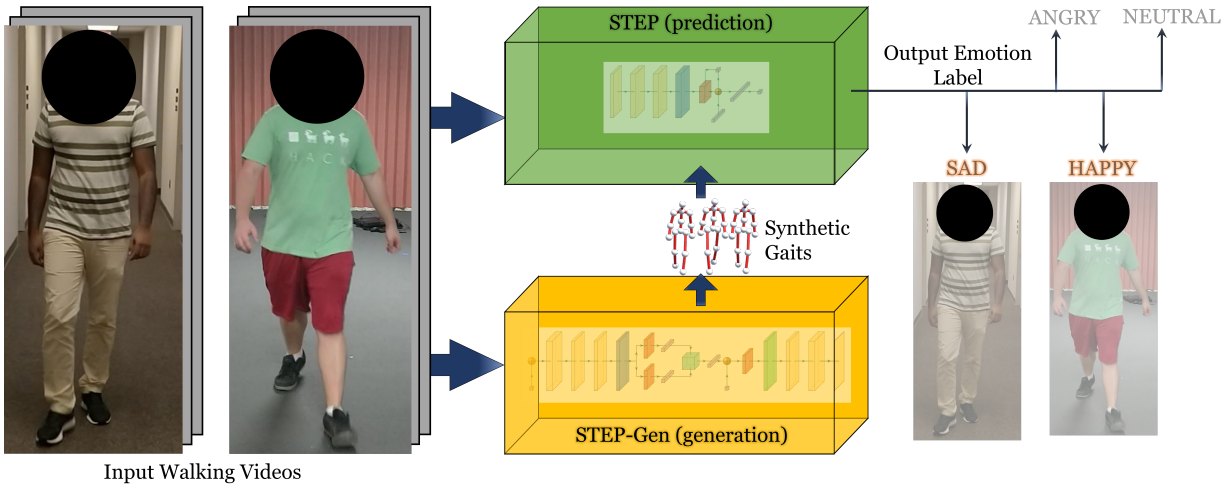


Figure 2.1: **STEP and STEP-Gen: Overview.** We present a novel classifier network (STEP) to predict perceived emotions from gaits extracted from walking videos. We also present a generator network (STEP-Gen) to generate annotated synthetic gaits to improve the accuracy of STEP.

techniques that can utilize other non-verbal cues.

In this chapter, we focus on using movement features corresponding to gaits in a walking video for affect detection. A gait as an ordered temporal sequence of body joint transformations (predominantly translations and rotations) during the course of a single walk cycle. Simply stated, a person's gait is the way the person walks. Prior works in psychology literature have reported that participants were able to identify sadness, anger, happiness, and pride by observing affective features corresponding to arm swinging, long strides, erect posture, collapsed upper body, etc. (Michalak et al. 2009; Kleinsmith and Bianchi-Berthouze 2013).

There is considerable recent work on pose or gait extraction from a walking video using deep convolutional network architectures and intricately designed loss functions (Girdhar et al. 2018; Dabral et al. 2018). Gaits have also been used for a variety of applications including action detection (Gu et al. 2010; Yan, Xiong, and Lin 2018) and person identification (Zhang and Troje 2005). However, the use of gaits for automatic affect detection has been fairly limited, primarily

due to a lack of gait data or videos annotated with emotions (Chiu, Shu, and Hui 2018). It is difficult and challenging to generate a large dataset with many thousands of annotated real-world gait videos to train a network.

Main Contributions: We present a learning-based approach to classify perceived emotions of an individual walking in a video. Our formulation consists of a novel classifier and a generative network as well as an annotated gait video dataset. The main contributions include:

1. A novel end-to-end Spatial Temporal Graph Convolution-Based Network (STEP), which implicitly extracts a person's gait from a walking video to predict their emotion. STEP combines deeply learned features with affective features to learn hybrid features.
2. A Conditional Variational Autoencoder (CVAE) called STEP-Gen, which is trained on a sparse real-world annotated gait set and can easily generate thousands of annotated synthetic gaits. We enforce the temporal constraints (*e.g.,* gait drift and gait collapse) inherent in gaits directly into the loss function of the CVAE, along with a novel push-pull regularization loss term. Our formulation helps to avoid over-fitting by generating more realistic gaits. These synthetic gaits improve the accuracy of STEP by 6% in our benchmarks.
3. We present a new dataset of human gaits annotated with emotion labels, called Emotion-Gait (E-Gait). It currently consists of $4,227$ real-world gait videos annotated with the emotion labels happy, sad, angry, and neutral.

We have evaluated the performance of STEP on E-Gait. The gaits in this dataset were extracted from videos of humans walking in both indoor and outdoor settings. In practice, STEP results in classification accuracy of 88% on E-Gait. We have compared it with prior methods and observe:

- An accuracy increase of 14% over prior learning-based method (Randhavane et al. 2019a).

This method uses LSTMs for modeling their input, but for an action detection task.

- Accuracy improvement of 21 – 30% on the absolute over prior gait-based affect detection methods reported in the psychology literature that use affective features.

2.2 Related Work

We provide a brief overview of prior work in affect detection and generative models for gait-like datasets.

Affect Detection. Face and speech data have been widely used to perceive human emotions. Prior methods that use faces as input commonly track action units on the face such as points on the eyebrow, cheeks and lips (Fabian Benitez-Quiroz, Srinivasan, and Martinez 2016), or track eye movements (Lu et al. 2015) and facial expressions (Majumder, Behera, and Subramanian 2014). Speech-based affect detection methods use either spectral features or prosodic features like loudness of voice, difference in tones and changes in pitch (Jacob and Mythili 2015). With the rising popularity of deep learning, there is considerable work on developing learned features for emotion detection from large-scale databases of faces (Yang, Ciftci, and Yin 2018; Zhang et al. 2018a) and speech signals (Deng et al. 2018). Recent methods have also looked at the cross-modality of combined face and speech data to perform affect detection (Albanie et al. 2018). In addition to faces and speech, physiological signals such as heartbeats and respiration rates (Zhao, Adib, and Katabi 2016) have also been used to increase the accuracy of affect detection. Our approach for affect detection from walking videos and gaits is complimentary to these methods and can be combined.

Different methods have also been proposed to perceive emotions from gaits. Karg, Kuhnlenz, and Buss 2010 use PCA-based classifiers, and Crenn et al. 2016 use SVMs on affective fea-

tures. Venture et al. 2014 use autocorrelation matrices between joint angles to perform similarity-based classification. Daoudi et al. 2017 represent joint movements as symmetric positive definite matrices and perform nearest neighbor classification.

Gaits have also been widely used in the related problem of action detection (Ji et al. 2013; Feichtenhofer, Pinz, and Wildes 2016; Feichtenhofer, Pinz, and Zisserman 2016; Lea et al. 2017; Yan, Xiong, and Lin 2018). In our approach, we take motivation from prior works on both affect and action detection from gaits.

Gait Generation. Collecting and compiling a large dataset of annotated gait videos is indeed a challenging task. As a result, it is important to develop generative algorithms for gaits conditioned on emotion labels. Current learning-based generation models are primarily based on Generative Adverserial Networks (GANs) or Variational Autoencoders (VAEs). MoCoGAN (Tulyakov et al. 2018) uses a GAN-based model, the latent space of which is divided into motion space (for generating temporal features) and content space (for generating spatial features). It can generate tiny videos of facial expressions corresponding to various emotions. vid2vid (Wang et al. 2018) is a state-of-the-art GAN-based network that uses a combined spatial temporal adversarial objective to generate high-resolution videos, including videos of human poses and gaits when trained on relevant real data. Other generative methods for gaits learn the initial poses and the intermediate transformations between frames in separate networks, and then combine the generated samples from both networks to develop realistic gaits (Yang et al. 2018b; Cai et al. 2018a). In this work, we model gaits as skeletal graphs and use spatial-temporal graph convolutions (Yan, Xiong, and Lin 2018) inside a VAE to generate synthetic gaits.

2.3 Background

In this section, we give a brief overview of Spatial Temporal Graph Convolutional Networks (ST-GCNs) and Conditional Variational Autoencoders (CVAE).

2.3.1 GCN and ST-GCN

The Graph Convolutional Network (GCN) was first introduced in (Bruna et al. 2013) to apply convolutional filters to arbitrarily structured graph data. Consider a graph $\mathcal{G}=\{\mathcal{V},\mathcal{E}\}$ with $N = |\mathcal{V}|$ nodes. Also consider a feature matrix $X \in \mathbb{R}^{N\times F}$, where row $x_i \in \mathbb{R}^F$ corresponds to a feature for vertex i. The propagation rule of a GCN is given as

$$Z^{(l+1)} = \sigma(AZ^{(l)}W^{(l)}), \tag{2.1}$$

where $Z^{(l)}$ and $Z^{(l+1)}$ are the inputs to the l-th and the $(l+1)$-th layers of the network, respectively. $Z^{(0)}=X$, $W^{(l)}$ is the weight matrix between the l-th and the $(l+1)$-th layers, A is the $N \times N$ adjacency matrix associated with the graph $\mathcal{G}$ and $\sigma(\cdot)$ is a non-linear activation function (*e.g.*, ReLU). Thus, a GCN takes in a feature matrix X as an input and generates another feature matrix $Z^{(L)}$ as the output, L being the number of layers in the network. In practice, each weight matrix W in a GCN represents a convolutional kernel. Multiple such kernels can be applied to the input of a particular layer to get a feature tensor as output, similar to a conventional Convolutional Neural Network (CNN). For example, if K kernels, each of dimension $F \times D$ are applied to the input X, then the output of the first layer will be an $N \times D \times K$ feature tensor.

Yan, Xiong, and Lin 2018 extended GCNs to develop the spatial temporal GCN (ST-GCN),

which can be used for action detection from human skeletal graphs. The graph in their case is the skeletal model of a human extracted from videos. Since they extract poses from each frame of a video, their input is a temporal sequence of such skeletal models. "Spatial" refers to the spatial edges in the skeletal model, which are the limbs connecting the body joints. "Temporal" refers to temporal edges which connect the positions of each joint across different time steps. Such a representation enables the gait video to be expressed as a single graph with a fixed adjacency matrix, and thus can be passed through a GCN network. The feature per vertex in their case is the 3D position of the joint represented by that vertex. In our work, we use the same representation for gaits, described later in Section 2.4.1.

2.3.2 Conditional Variational Autoencoder

The variational autoencoder (Kingma and Welling 2019) is an encoder-decoder architecture that is used for data generation based on Bayesian inference. The encoder transforms the training data into a latent lower-dimensional distribution space. The decoder draws random samples from that distribution and generates synthetic data that are as similar to the training data as possible.

In conditional VAE (Sohn, Lee, and Yan 2015), instead of generating from a single distribution space learned by the encoder, it learns separate distributions for the separate classes in the training data. Thus, given a class, the decoder produces random samples from the conditional distribution of that class, and generates synthetic data of that class from those samples. Furthermore, if we assume that the decoder generates Gaussian variables for every class, then

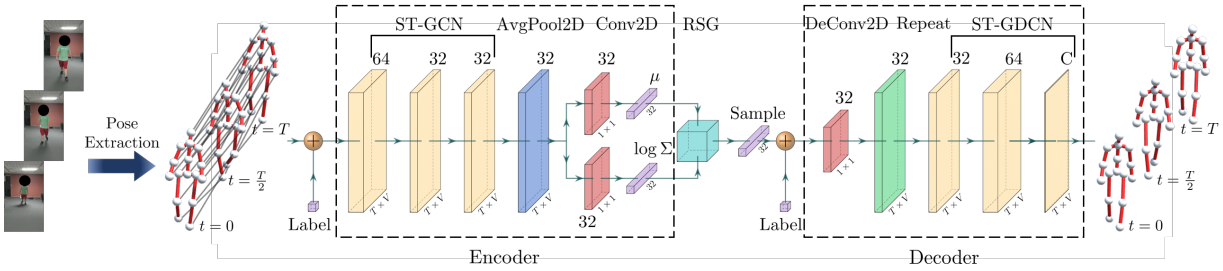


Figure 2.2: **STEP-Gen: Spatial Temporal Graph Convolutional Network Architecture.** The encoder consists of ST-GCN, Average Pool and Conv2D layers. The decoder consists of DeConv2D, Repeat and ST-GDCN layers. RSG (Random Sample Generator) is used to generate random samples from the latent space. + denotes appending; T: number of time steps (75 in our dataset); V: number of nodes (16 in our dataset); C: dimension of each node (3 in our dataset). Input: Human gaits processed from walking videos and corresponding emotion label. Spheres are nodes, thick red lines are spatial edges and thin gray lines are temporal edges. Output: Human gaits corresponding to the input label, with same T, V, and C.

the negative log likelihood for each class is given by the MSE loss

$$\mathcal{L}_o = \left|\left|x - f_{\theta_c}(z)\right|\right|^2 \tag{2.2}$$

where $f_{\theta_c}(\cdot)$ denotes the decoder function for class c, x represents the training data, and z the latent random variable. We incorporate a novel push-pull regularization loss on top of this standard CVAE loss, as described in Section 2.5.3.

2.4 STEP and STEP-Gen

Our objective is to perform affect detection from gaits. Based on prior work (Kleinsmith and Bianchi-Berthouze 2013; Karg, Kuhnlenz, and Buss 2010; Crenn et al. 2016), we assume that emotional cues are largely determined by localized variances in gaits, such as swinging speed of the arm (movement of 3 adjacent joints: shoulder, elbow and hand), stride length and speed (movement of 3 adjacent joints: hip, knee and foot), relative position of the spine joint w.r.t.

the adjacent root and neck joints and so on. Convolutional kernels are known to capture such local variances and encode them into meaningful feature representations for learning-based algorithms (Krizhevsky, Sutskever, and Hinton 2012). Additionally, since we treat gaits as a periodic motion that consists of a sequence of localized joint movements in 3D, we therefore use GCNs for our generation and classification networks to capture these local variances efficiently. In particular, we use Spatial Temporal GCNs (ST-GCNs) developed by Yan, Xiong, and Lin (2018) to build both our generation and classification networks. We now elaborate our entire approach in detail.

2.4.1 Extracting Gaits from Videos

Naturally collected human gait videos contain a wide variety of extraneous information such as attire, items carried (*e.g.*, bags or cases), background clutter, etc. We use a state-of-the art pose estimation method (Girdhar et al. 2018) to extract clean, 3D skeletal representations of the gaits from videos. Moreover, gaits in our dataset are collected from varying viewpoints and scales. To ensure that the generative network does not end up generating an extrinsic mean of the input gaits, we perform view normalization. Specifically, we transform all gaits to a common point of view in the world coordinates using the Umeyama method (Umeyama 1991). Thus, a gait in our case is a temporal sequence of view normalized skeletal graphs extracted per frame from a video. We now provide a formal definition for gait.

A gait is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{E}$ denotes the set of edges, such that

- $v_i^t \in \mathcal{V}$, $i \in \{1, \dots V\}$ represents the 3D position of the i-th joint in the skeleton at time step t and V is the total number of joints in the skeleton.

- $\mathcal{A}_i^t \subseteq \mathcal{V}$ is the set of all nodes that are adjacent to v_i^t as per the skeletal graph at time step t,
- $v_i := \{v_i^t\}_{t \in \{1,\ldots,T\}}$ denotes the set of positions of of the i-th joint across all time steps $1 \ldots T$,
- $(v_i^t, v_j^t) \in \mathcal{E}, \forall v_j^t \in \mathcal{A}_i^t \cup v_i, \forall t \in \{1, \ldots, T\}, \forall i \in \{1, \ldots, V\}$.

A key pre-requisite for using GCNs is to define the adjacency between the nodes in the graph (Bruna et al. 2013; Kipf and Welling 2016; Yan, Xiong, and Lin 2018). Note that as per our definition of gait, given fixed T and V, any pair of gaits $\mathcal{G}_x$ and $\mathcal{G}_y$ can have different sets of vertices, $\mathcal{V}_x$ and $\mathcal{V}_y$ respectively, but necessarily have the same edge set $\mathcal{E}$ and hence the same adjacency matrix A. This useful property of the definition allows us to maintain a unique notion of adjacency for all the gaits in a dataset, and thus develop ST-GCN-based networks for the dataset.

2.4.2 STEP-Gen: The Generation Network

We show our generative network in Figure 2.2. Our network architecture is based on the Conditional Variational Autoencoder (CVAE) (Sohn, Lee, and Yan 2015).

In the encoder, each $C \times T \times V$ dimensional input gait, pre-processed from a video (as per Section 2.4.1), is appended with the corresponding label, and passed through a set of 3 ST-GCN layers (yellow boxes). C=3 is the feature dimension of each node in the gait, representing the 3D position of the corresponding joint. The first ST-GCN layer has 64 kernels and the next two have 32 kernels each. The output from the last ST-GCN layer is average pooled along both the temporal and joint dimensions (blue box). Thus, the output of the pooling layer is a $32 \times 1 \times 1$ tensor. This tensor is passed through two 1×1 convolutional layers in parallel (red boxes). The

outputs of the two convolutional layers are 32 dimensional vectors, which are the mean and the log-variance of the latent space respectively (purple boxes). All ST-GCN layers are followed by the ReLU nonlinearity, and all the layers are followed by a BatchNorm layer (not shown separately in Figure 2.2).

In the decoder, we generate random samples from the 32 dimensional latent space and append them with the same label provided with the input. As commonly performed in VAEs, we use the reparametrization trick (Kingma and Welling 2019) to make the overall network differentiable. The random sample is passed through a 1×1 deconvolutional layer (red box), and the output feature is repeated ("un-pooled") along both the temporal and the joint dimension (green box) to produce a $32 \times T \times V$ dimensional tensor. This tensor is then passed through 3 spatial temporal graph deconvolutional layers (ST-GDCNs) (yellow boxes). The first ST-GDCN layer has 32 kernels, the second one has 64 channels, and the last one has C=3 channels. Hence, we finally get a $C \times T \times V$ dimensional tensor at the output, which is a synthetic gait for the provided label. As in the encoder part, all ST-GDCN layers are followed by a ReLU nonlinearity, and all layers are followed by a BatchNorm layer (not shown separately in Figure 2.2).

Once the network is trained, we can generate new synthetic gaits by drawing random samples from the 32 dimensional latent distribution space parametrized by the learned μ and Σ.

The original CVAE loss $\mathcal{L}_o$ is given by:

$$\mathcal{L}_o = \sum_{t=1}^{T} \left\| v_R^t - v_S^t \right\|^2, \tag{2.3}$$

where $v^t = \left[v_1^t \dots v_V^t \right]^\top$, where each v_i^t is assumed to be a row vector consisting of the 3D position of the joint i at frame t. The subscripts R and S stand for real and synthetic data respec-

tively.

Each gait corresponds to a temporal sequence. Therefore, for any gait representation, it is essential to incorporate such temporal information. This is even more important as temporal changes in a gait provide significant cues for affect detection (Kleinsmith and Bianchi-Berthouze 2013; Karg, Kuhnlenz, and Buss 2010; Crenn et al. 2016). But, the baseline-CVAE architecture does not take into account the temporal nature of the gaits. We therefore modify the original reconstruction loss of the CVAE by adding regularization terms that enforce the desired temporal constraints (Equation 2.8).

We propose a novel "push-pull" regularization scheme. We first make sure that sufficient movement occurs in a generated gait across the frames so that the joint configurations at different time frames do not collapse into a single configuration. This is the "push" scheme. Simultaneously, we make sure that the generated gaits do not drift too far from the real gaits over time due to excessive movement. This is the "pull" scheme.

- *Push*: We require the synthetic data to resemble the joint velocities and accelerations of the real data as closely as possible. The velocity vel_i^t of a node i at a frame t can be approximated as the difference between the positions of the node at frames t and $t-1$, *i.e.*,

$$vel_i^t = v_i^t - v_i^{t-1} \tag{2.4}$$

  Similarly, acceleration acc_i^t of a node i at a frame t can be approximated as the difference between the velocities of the node at frame t and $t-1$, *i.e.*,

$$acc_i^t = vel_i^t - vel_i^{t-1} = v_i^t - 2v_i^{t-1} + v_i^{t-2} \tag{2.5}$$

We use the following loss for gait collapse:

$$\mathcal{L}_c = \sum_{t=2}^{T} \left\| vel_R^t - vel_S^t \right\|^2 + \sum_{t=3}^{T} \left\| acc_R^t - acc_S^t \right\|^2 \tag{2.6}$$

where $vel^t = \left[vel_1^t \dots vel_V^t \right]^\top$ and $acc^t = \left[acc_1^t \dots acc_V^t \right]^\top$.

- *Pull*: When the synthetic gait nodes are enforced to have non-zero velocity and acceleration between the frames, the difference between the synthetic node positions and the corresponding real node positions tends to increase as the number of frames increases. This is commonly known as the drift error. In order to constrain this error, we use the notion of anchor frames. At the anchor frames, we impose additional penalty on the loss between the real and synthetic gaits. In order to be effective, we need to ensure that there are a high number of anchor frames and they are as far apart as possible. Based on this trade off, we choose 3 anchor frames in the temporal sequence — the first frame, the middle frame and the last frame of the gait. We use the following loss function for gait drift:

$$\mathcal{L}_d = \sum_{t=1}^{T} \sum_{\omega \in \Omega} \left\| v_R^t - v_R^\omega - (v_S^t - v_S^\omega) \right\|^2 \tag{2.7}$$

where Ω denotes the set of anchor frames.

Finally, our modified reconstruction loss $\mathcal{L}_r$ of the CVAE is given by

$$\mathcal{L}_r = \mathcal{L}_o + \lambda_c \mathcal{L}_c + \lambda_d \mathcal{L}_d \tag{2.8}$$

where λ_c and λ_d are the regularization weights. Note that this modified loss function still satisfies the ELBO bound (Kingma and Welling 2019), if we assume that the decoder generates

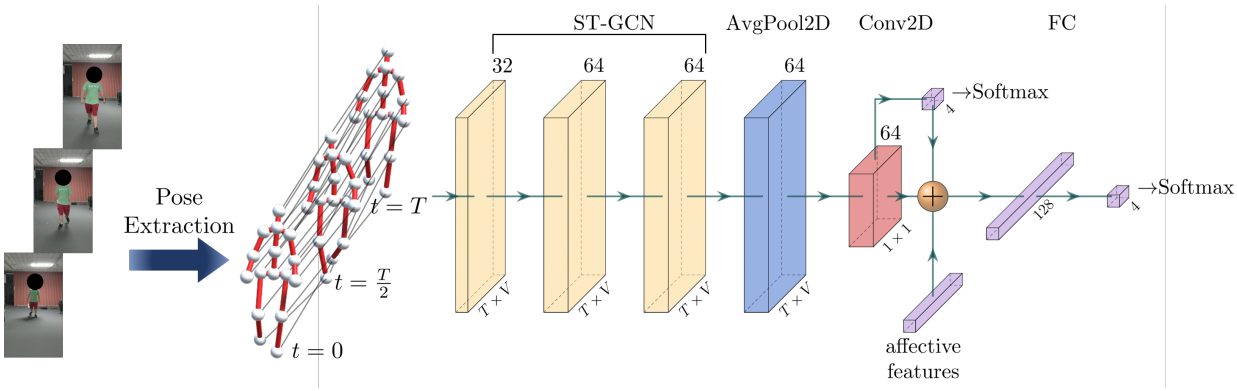


Figure 2.3: **STEP: Spatial Temporal Graph Convolutional Network Architecture.** It consists of ST-GCN, Average Pool, Conv2D and fully connected (FC) layers. + denotes appending. T: number of time steps (75 in our dataset); V: number of nodes (16 in our dataset); C: dimension of each node (3 in our dataset). Input: Human gaits processed from walking videos. Spheres are nodes, thick red lines are spatial edges and thin gray lines are temporal edges. Output: Predicted label after Softmax. The first Softmax from the left gives the output of Baseline-SETP, and the second Softmax gives the output of STEP.

variables from a mixture of Gaussian distributions for every class, with the original loss, the push loss ad the pull loss representing the 3 Gaussian distributions in the mixture.

2.4.3 STEP: The Classification Network

We show out classifier network in Figure 2.3. In the base network, each input gait is passed through a set of 3 ST-GCN layers (yellow boxes). The first ST-GCN layer has 32 kernels and the next two have 64 kernels each. The output from the last ST-GCN layer is average pooled (blue box) in both the temporal and joint dimensions and passed through a 1×1 convolutional layer (red box). The output of the convolutional layer is passed through a fully connected layer of dimension 4 (corresponding to the 4 emotion labels that we have), followed by a softmax operation to generate the class labels. All the ST-GCN layers are followed by the ReLU nonlinearity and all layers except the fully connected layer are followed by a BatchNorm layer (not shown separately in Figure 2.3). We refer to this version of the network as the *Baseline-STEP*.

Prior work in gait analysis has shown that affective features for gaits provide important information for affect detection (Kleinsmith and Bianchi-Berthouze 2013; Karg, Kuhnlenz, and Buss 2010; Crenn et al. 2016). Affective features are comprised of two types of features:

- Posture features. These include angle and distance between the joints, area of different parts of the body (*e.g.*, area of the triangle formed by the neck, the right hand and the left hand), and the bounding volume of the body.
- Movement features. These include the velocity and acceleration of individual joints in the gait.

We exploit the affective feature formulation (Kleinsmith and Bianchi-Berthouze 2013; Crenn et al. 2016) in our final network. We append the 29 dimensional affective feature (purple box) to the final layer feature vector learned by our Baseline-STEP network, thus generating hybrid feature vectors. These hybrid feature vectors are passed through two fully connected layers of dimensions 128 and 4 respectively, followed by a softmax operation to generate the final class labels. We call this combined network STEP.

2.5 Experiments and Results

We list all the parameters and hardware used in training both our generation and classification networks in Section 2.5.1. In Section 2.5.2, we give details of our new dataset. In Sections 2.5.3, we list the standard metrics used to compare generative models and classification networks and in Section 2.5.4, we list the state-of-the-art methods against which we compare our algorithms. In Section 2.5.5, we present the evaluation results. Finally, in Section 2.5.6, we analyse the robustness of our system and show that both STEP and STEP-Gen do not overfit on the E-Gait Dataset.

2.5.1 Training Parameters

For training STEP-Gen, we use a batch size of 8 and train for 150 epochs. We use the Adam optimizer (Kingma and Ba 2014) with an initial learning rate of 0.1, which decreases to $\frac{1}{10}$-th of its current value after 75, 113 and 132 epochs. We also use a momentum of 0.9 and and weight-decay of 5×10^{-4}.

For training STEP, we use a split of $7 : 2 : 1$ for training, validation and testing sets. We use a batch size of 8 and train for 500 epochs using the Adam optimizer (Kingma and Ba 2014) with an initial learning rate of 0.1. The learning rate decreases to $\frac{1}{10}$-th of its current value after 250, 375 and 438 epochs. We also use a momentum of 0.9 and and weight-decay of 5×10^{-4}. All our results were generated on an NVIDIA GeForce GTX 1080 Ti GPU.

2.5.2 Dataset: Emotion-Gait

Emotion-Gait (E-Gait) consists of 4, 227 real gaits and 1, 000 synthetic gaits each of the 4 emotion classes generated by STEP-Gen, for a total for 5, 227 gaits. We obtained the videos for the real gaits from various sources, including BML (Ma, Paterson, and Pollick 2006), Human3.6M (Ionescu et al. 2014), ICT (Narang et al. 2017), CMU-MOCAP (CMU-MOCAP 2018) and ELMD (Habibie et al. 2017), and converted all the input gaits to 21-joint skeletons following the procedure of Habibie et al. (2017) Each gait in the consolidated dataset was annotated by the same 10 annotators. The annotators were between 20 and 28 years old, with a median age of 23 years. 4 of the annotators were female and 6 were male. There was also an even mix of annotators belonging to the same culture as the subjects in the dataset as well as annotators from other, different cultures.

Based on the annotations, we found that the agreement score due to chance was 0.29, and the corresponding Fleiss' Kappa (FK) score (Fleiss 1971) was 0.45. Moreover, the FK score between only the sad and the neutral labels was -0.01 (< 0) and that between only the happy and the neutral labels was -0.03 (< 0), indicating a lack of disagreement between these two pairs of labels. This is expected, as these pairs of labels contain many marginal cases where the subjectivity of affect detection is the most prominent. In our case, we picked the class with the maximum number of votes as the emotion label for the corresponding gait. If, however, the total number of votes in the top k classes were within 20% of each other, then we add the corresponding gait as an instance of all the top k classes. For example, if a gait was annotated as happy by 40% of the annotators and neutral by 60% of the annotators, we marked that gait as an instance of both happy and neutral.

2.5.3 Evaluation Metrics

Generation: For generative models, we compute the Fréchet Inception Distance (FID) score (Heusel et al. 2017) that measures how close the generated samples are to the real inputs while maintaining diversity among the generated samples. The FID score is computed using the following formula:

$$FID = \left\|\mu_x - \mu_g\right\|_2^2 + Tr(\Sigma_x + \Sigma_g + 2(\Sigma_x\Sigma_g)^{\frac{1}{2}}) \tag{2.9}$$

2.5.3.1 Classification

For classifier models, we report the classification accuracy given by $Accuracy = (TP + TN)/TD$, where TP, TN, TD are the number of true positives, true negatives, and total data, respectively.

2.5.4 Evaluation Methods

Generation: We compare our generative network with both GAN- and VAE-based generative networks, as listed below.

- vid2vid (GAN-based) (Wang et al. 2018): This is the state-of-the-art video generation method. It can take human motion videos as input and generate high-resolution videos of the same motion.
- Baseline CVAE (VAE-based): We use a CVAE with the same network architecture as STEP-Gen, but with only the original CVAE loss given in Equation 2.3.

2.5.4.1 Classification

We compare our classifier network with both prior methods for affect detection from gaits, and prior methods for action detection from gaits, as listed below.

- **Affect Detection.** We compare with the current state-of-the-art classifiers of (Karg, Kuhnlenz, and Buss 2010; Venture et al. 2014; Crenn et al. 2016; Wang, Enescu, and Sahli 2016; Daoudi et al. 2017).
- **Action Detection.** We compare with the state-of-the-art methods using both GCNs (Yan, Xiong, and Lin 2018) and LSTMs (Randhavane et al. 2019a). The networks of both these methods were trained on our dataset before comparing the performance.

  We also perform the following ablation experiments with our classifier network:
- **Baseline-STEP.** It predicts emotions based only on the network-learned features from gaits. This network is trained on the 4,227 real gaits in E-Gait.
- **STEP.** This is our hybrid network combining affective features (Kleinsmith and Bianchi-

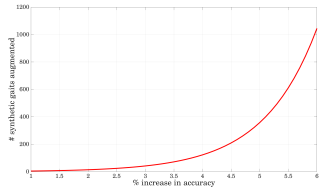


Figure 2.4: **Effect of STEP-Gen Data Augmentation.** Effect of augmenting synthetically generated data to the train and test sets of STEP+Aug. For every percent improvement in accuracy, an exponentially larger amount of data needs to be augmented.

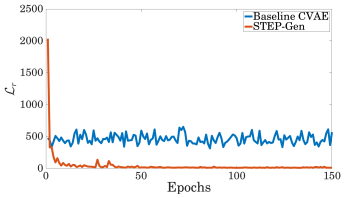


Figure 2.5: **STEP-Gen: Training Loss Convergence on the E-Gait Dataset.** Our "Push-Pull" regularization loss (Equation 2.8) as a function of training epochs, as produced by the baseline-CVAE and our STEP-Gen. The baseline-CVAE fails to converge even after 150 epochs, while STEP-Gen converges in around 28 epochs.

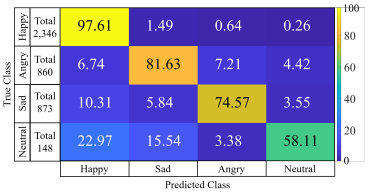


Figure 2.6: **STEP+Aug: Confusion Matrix for the E-Gait dataset.** Per class classification results over the 4, 227 gaits. We obtain a balanced accuracy score of 77.98%.

Berthouze 2013; Crenn et al. 2016) with the network-learned features of Baseline-STEP. This network is also trained on the 4, 227 real gaits in E-Gait.

- **STEP+Aug.** This is same as STEP, but trained on both the real and the synthetic gaits in E-Gait.

2.5.5 Results on E-Gait

Generation: All the generative networks are trained on the 4, 227 real data in E-Gait. We report an FID score of 11.11, while the FID score of Baseline-CVAE is 11.33. Lower FID indicates higher fidelity to the real data. However, we also note that vid2vid (Wang et al. 2018) completely memorizes the dataset and thus gives an FID score of 0. This is undesirable for our task since we require the generative network to be able to produce diverse data that can be augmented to the training set of the classifier network.

Additionally, to show that our novel "Push-Pull" regularization loss function (Equation 2.8)

Table 2.1: **STEP: Classification Accuracy Comparison.** Accuracy values are computed using the formula in Section 2.5.3 and shown as percentages. We choose methods from both psychology and computer vision literature. Base-STEP and STEP+Aug are variations of STEP.

Method	Accuracy
Venture et al. (2014)	30.83
Karg, Kuhnlenz, and Buss (2010)	39.58
Daoudi et al. (2017)	42.52
Wang, Enescu, and Sahli (2016)	53.73
Crenn et al. (2016)	66.22
ST-GCN (Yan, Xiong, and Lin 2018)	65.62
LSTM (Randhavane et al. 2019a)	74.10
Base-STEP	78.24
STEP	82.15
STEP + Aug	**88.22**

generates gaits with joint movements, we measure the decay of the value of the loss function for the baseline-CVAE and STEP-Gen with time (Figure 2.5). We add the $\mathcal{L}_c$ and $\mathcal{L}_d$ terms from equation 2.8 (without optimizing them) to the baseline-CVAE loss function (Equation 2.3). We observe that STEP-Gen converges extremely quickly to a smaller loss value in around 28 epochs. On the other hand, the base-line CVAE produces oscillations and fails to converge as it does not optimize $\mathcal{L}_c$ and $\mathcal{L}_d$.

We also perform qualitative tests of gait generated by all the methods. vid2vid (Wang et al. 2018) uses GANs to produce high-quality videos. However, in our experiments, vid2vid memorizes the dataset and does not produce diverse samples. Baseline-CVAE produces static gaits that do not move in time. Finally, our gaits are both diverse (different from input) and realistic (successfully mimics walking motion). We show all these results in our project page.

Classification: In Table 2.1, we report the mean classification accuracies of all the methods using the formula in Section 2.5.3. We observe that most of the prior methods for affect detection from gaits have less than 60% accuracy on E-Gait. Only Crenn et al. 2016, who manually

compute the same features we use in our novel "push-pull" regularization loss function (enforce *i.e.* distances between joints across time) has greater than 65% accuracy. The two prior action detection from gait methods we compare with have 65% and 75% accuracy respectively. By comparison, our Baseline-STEP has an accuracy of 78%. Combining network-learned and affective features in STEP gives an accuracy of 83%. Finally, augmenting synthetic gaits generated by STEP-Gen in STEP+Aug gives an accuracy of 88%.

To verify that our classification accuracy is statistically significant and not due to random chance, we perform two statistical tests:

- *Hypotheses Testing:* Classification as a task, depends largely on the test sample to be classified. To ensure that the classification accuracy of STEP is not achieved due to random positive examples, we determine the statistical likelihood of our results. Note that we do not test on STEP+Aug as accuracy of STEP+Aug is also dependent on the augmentation size. We generate a population of size $10,000$ accuracy values of STEP with mean 83.15 and standard deviation 6.9. We set $\mu = 83.15$, *i.e.* the reported mean accuracy of STEP as the null hypothesis, H_0. To accept our null hypothesis, we require the p-value to be greater than 0.50. We compute the p-value of this population as $0.78 > 0.50$. Therefore, we fail to reject the null hypothesis, thus corroborating our classification accuracy statistically.
- *Confidence Intervals:* This metric determines the likelihood of a value residing in an interval. For a result to be statistically significant, we require a tight interval with high probability. With a 95% likelihood, we report a confidence interval of $[81.19, 85.33]$ with a standard deviation of 1.96. Simply put, our classification accuracy will lie between 81.19 and 85.33 with a probability of 0.95.

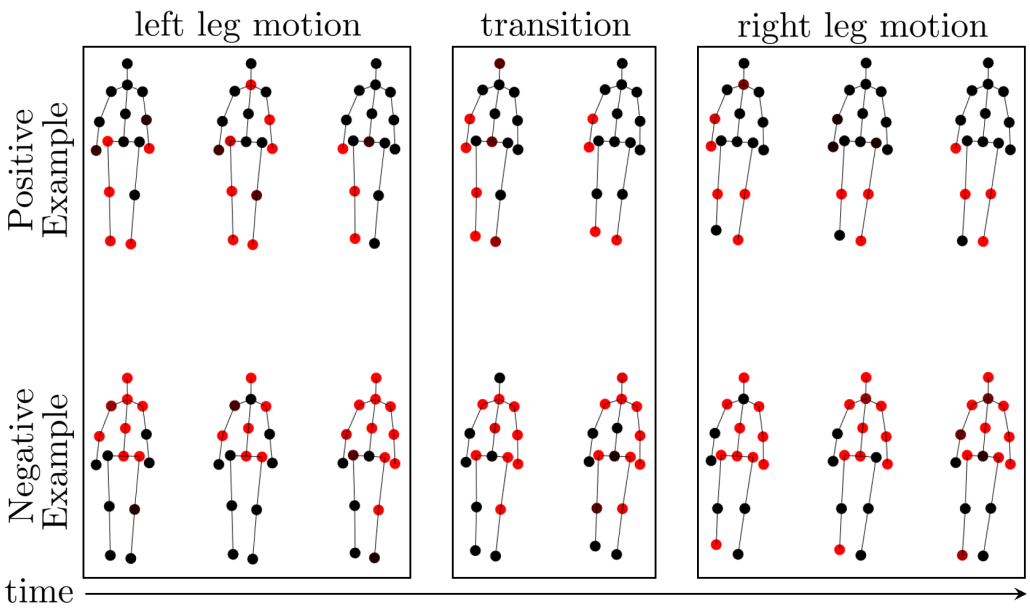


Figure 2.7: **STEP+Aug: Saliency Map on the E-Gait Dataset.** Saliency map showing the magnitude of the network gradient along the z-axis (in and out of the paper) generated by our trained network, which is the direction of walking in both the examples shown. The examples are for the 'happy' emotion. In the positive example, the network focuses on the movements of the arms and the legs. These movements contain important emotional cues, thereby confirming that our classifier is learning meaningful features to detect affects accurately.

Finally, we show the discriminatory capability of our classifier through a confusion matrix in Figure 2.6.

2.5.6 Overfitting Analysis

Effect of Generated Data on Classification: We show in Figure 2.4 that the synthetic data generated by STEP-Gen increases the classification accuracy of STEP+Aug. This, in turn, shows that STEP-Gen does not memorize the training dataset, but can produce useful diverse samples. Nevertheless, we see that to achieve every percent improvement in the accuracy of STEP+Aug, we need to generate an exponentially larger number of synthetic samples as training saturation sets in.

Saliency Maps: We show that STEP does not memorize the training dataset, but learns meaningful features, using saliency maps obtained via guided backpropagation on the learned network (Simonyan and Zisserman 2014; Springenberg et al. 2014). Saliency maps determine how the loss function output changes with respect to a small change in the input. In our case, the input consists of 3D joint positions over time, therefore, the corresponding saliency map highlights the joints that most influence the output. Intuitively, we expect the saliency map for a positively classified example to capture the joint movements that are most important for predicting the perceived emotion from a psychological point of view (Crenn et al. 2016). We show the saliency map given by our trained network for both a positively classified and a negatively classified example for the label 'happy' in Figure 2.7. The saliency map only shows magnitude of the gradient along the z-axis (in and out of the plane of the paper), which is the direction of walking in both the examples. Black represents zero magnitude, and bright red represents a high magnitude. In the positive example, we see that the network focuses on the movements of the arms and the legs. This is the expected behavior, as the movement of hands and the stride length and speed are important cues for affect detection (Crenn et al. 2016). By contrast, there is no intuitive pattern to the detected movements in the saliency map for the negative example.

2.6 Conclusion

As a first attempt towards affect detection from gaits, we have presented a classifier network called STEP to classify gaits into affect labels, using a combination of pose-based affective features and data-driven features. Our network is based on a Spatial Temporal Graph Convolutional Network (ST-GCN) architecture to explicitly leverage the spatial and temporal adjacen-

cies between the joints reprensting the body poses as it goes through different affective gaits. To improve our classification performance, we have also developed a generative network called STEP-Gen to generate synthetic gaits given affect labels, which we use to augment the training data for our classifier network. We have also released a novel dataset called E-Gait, which consists of 4,227 human gaits annotated with perceived emotions along with thousands of synthetic gaits. In terms of performance, STEP, when trained with the synthesized data provided by STEP-Gen, achieves a classification accuracy of 88% on E-Gait, which is an absolute improvement of 14–30% over prior methods.

CHAPTER 3

Affect Detection From Gaits Using Semi-Supervised Learning

Project Website: https://gamma.umd.edu/taew

Abstract

We present an autoencoder-based semi-supervised approach to classify perceived human emotions from walking styles obtained from videos or motion-captured data and represented as sequences of 3D poses. Given the motion on each joint in the pose at each time step extracted from 3D pose sequences, we hierarchically pool these joint motions in a bottom-up manner in the encoder, following the kinematic chains in the human body. We also constrain the latent embeddings of the encoder to contain the space of psychologically-motivated affective features underlying the gaits. We train the decoder to reconstruct the motions per joint per time step in

a top-down manner from the latent embeddings. For the annotated data, we also train a classifier to map the latent embeddings to emotion labels. Our semi-supervised approach achieves a mean average precision of 0.84 on the Emotion-Gait benchmark dataset, which contains both labeled and unlabeled gaits collected from multiple sources. We outperform current state-of-art algorithms for both affect and action detection from 3D gaits by 7%–23% on the absolute. More importantly, we improve the average precision by 10%–50% on the absolute on classes that each makes up less than 25% of the labeled part of the Emotion-Gait benchmark dataset.

3.1 Introduction

Humans perceive others' emotions through verbal cues such as speech (Rao, Koolagudi, and Vempada 2013; Jacob and Mythili 2015), text (Strapparava and Mihalcea 2008; Chen et al. 2018), and non-verbal cues such as eye-movements (Lu et al. 2015), facial expressions (Fabian Benitez-Quiroz, Srinivasan, and Martinez 2016), tone of voice, postures (Babu et al. 2018), and walking styles (Kleinsmith and Bianchi-Berthouze 2013). Perceiving others' emotions shapes people's interactions and experiences when performing tasks in collaborative or competitive environments (Barrett 2017). Given this importance of perceived emotions in everyday lives, there has been a steady interest in developing automated techniques for perceiving emotions from various cues, with applications in affective computing, therapy, and rehabilitation (Rivas et al. 2015), robotics (Bauer et al. 2009; Narayanan et al. 2020), audience understanding (Wu et al. 2016), and character generation (Starke et al. 2019).

While there are multiple non-verbal modalities for perceiving emotions, in our work, we only observe people's styles of walking or their gaits, extracted from videos or motion-captured data. Perceived affect detection using any non-verbal cues is considered to be a challenging

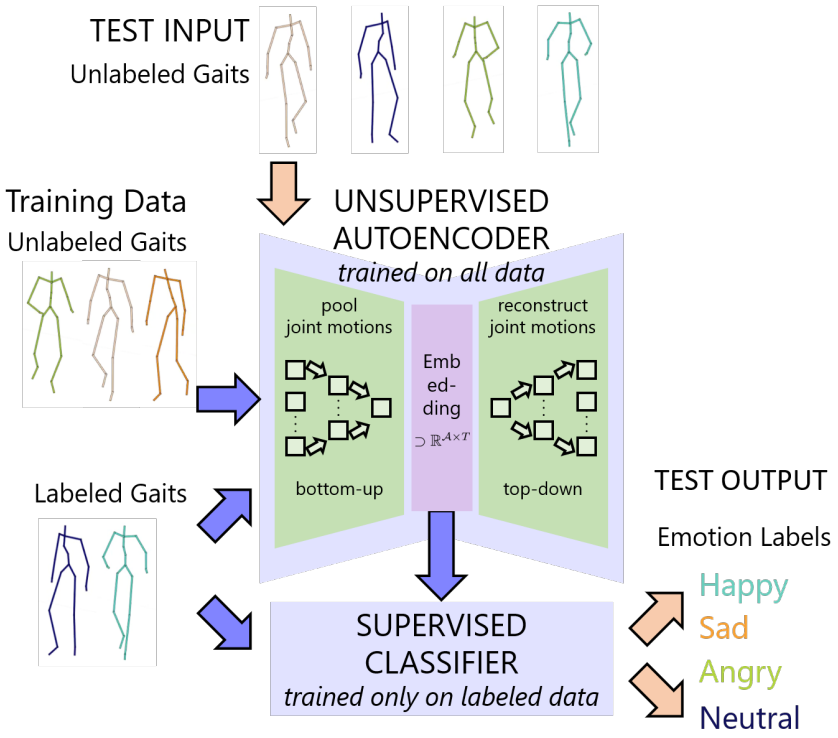


Figure 3.1: **TAEW: Overview.** We present a semi-supervised approach to predict discrete perceived emotions from 3D pose sequences of human gaits. Our unsupervised autoencoder learns latent embeddings for all the labeled and unlabeled input gaits using hierarchical pooling. Our supervised classifier learns to predict emotion labels by training only the labeled gaits. We train the autoencoder and the classifier simultaneously to learn to classify the unlabeled data. The mean average precision of our network increases linearly with the size of the unlabeled training data.

problem in both psychology and AI, primarily because of the unreliability in the cues, arising from sources such as "mock" expressions (Ekman and Friesen 1967b), expressions affected by the subject's knowledge of an observer (Fernández-Dols and Ruiz-Belda 1995), or even self-reported emotions in certain scenarios (Nisbett and Wilson 1977). However, gaits generally require less conscious initiation from the subjects and therefore tend to be more reliable cues. Moreover, studies in psychology have shown that observers were able to perceive the emotions of walking subjects by observing features such as arm swinging, stride lengths, collapsed upper body, etc. (Michalak et al. 2009; Kleinsmith and Bianchi-Berthouze 2013).

Gaits have been widely used in computer vision for many applications, including action detection (Yan, Xiong, and Lin 2018; Shi et al. 2019a; Liu et al. 2020) and perceiving emotions (Randhavane et al. 2019a; Randhavane et al. 2019c; Bhattacharya et al. 2020; Mittal et al. 2020b). However, there are a few key challenges in terms of designing machine learning methods for affect detection using gaits:

- Methods based on hand-crafted biomechanical features extracted from human gaits often suffer from low prediction accuracy (Crenn et al. 2016; Venture et al. 2014).
- Fully deep-learned methods (Randhavane et al. 2019a; Bhattacharya et al. 2020) rely heavily on sufficiently large sets of annotated data. Annotations are expensive and tedious to collect due to the variations in scales and motion trajectories (Ahsan, Sun, and Essa 2018), as well as the inherent subjectivity in perceiving emotions (Bhattacharya et al. 2020). The benchmark dataset for affect detection, Emotion-Gait (Bhattacharya et al. 2020), has around $4,000$ data points of which more than 53% are unlabeled.
- Conditional generative methods are useful for data augmentation, but current methods can only generate data for short time periods (Holden, Saito, and Komura 2016; Yang et al. 2018a) or with relatively low diversity (Pavllo, Grangier, and Auli 2018; Yan et al. 2019b; Khodabandeh et al. 2018; Bhattacharya et al. 2020).

On the other hand, acquiring poses from videos and MoCap data is cheap and efficient, leading to the availability of large-scale pose-based datasets (CMU-MOCAP 2018; Ionescu et al. 2014; Carreira and Zisserman 2017; Shahroudy et al. 2016). Given the availability of these unlabeled gait datasets and the sparsity of gaits labeled with perceived emotions, there is a need to develop automatic methods that can utilize these datasets for affect detection.

Main Contributions: We present a semi-supervised network that accepts 3D pose sequences

of human gaits extracted from videos or motion-captured data and predicts discrete perceived emotions, such as happy, angry, sad, and neutral. Our network consists of an unsupervised autoencoder coupled with a supervised classifier. The encoder in the unsupervised autoencoder hierarchically pools attentions on parts of the body. It learns separate intermediate feature representations for the motions on each of the human body parts (arms, legs, and torso) and then pools these features in a bottom-up manner to map them to the latent embeddings of the autoencoder. The decoder takes in these embeddings and reconstructs the motion on each joint of the body in a top-down manner.

We also perform affective mapping: we constrain the space of network-learned features to subsume the space of biomechanical affective features (Riggio 2017) expressed from the input gaits. These affective features contain useful information for distinguishing between different perceived emotions. Lastly, for the labeled data, our supervised classifier learns to map the encoder embeddings to the discrete emotion labels to complete the training process. To summarize, we contribute:

- **A semi-supervised network**, consisting of an autoencoder and a classifier, that are trained together to predict discrete perceived emotions from 3D pose sequences of gaits of humans.
- **A hierarchical attention pooling module** on the autoencoder to learn useful embeddings for unlabeled gaits, which improves the mean average precision (mAP) in classification by 1–17% on the absolute compared to state-of-the-art methods in both affect and action detection from 3D gaits on the Emotion-Gait benchmark dataset.
- **Subsuming the affective features** expressed from the input gaits in the space of learned embeddings. This improves the mAP in classification by 7–23% on the absolute compared

to state-of-the-art methods.

We observe the performance of our network improves linearly as more unlabeled data is used for training. More importantly, we report a 10–50% improvement on average precision on the absolute for emotion classes that have fewer than 25% labeled samples in the Emotion-Gait dataset (Bhattacharya et al. 2020).

3.2 Related Work

We briefly review prior work in classifying perceived emotions from gaits, as well as the related task of action detection and generation from gaits.

Detecting Perceived Emotions from Gaits. Experiments in psychology have shown that observers were able to identify sadness, anger, happiness, and pride by observing gait features such as arm swinging, long strides, erect posture, collapsed upper body, etc. (Montepare, Goldstein, and Clausen 1987; Meeren, Heijnsbergen, and Gelder 2005b; Michalak et al. 2009; Kleinsmith and Bianchi-Berthouze 2013). This, in turn, has led to considerable interest from both the computer vision and the affective computing communities in detecting perceived emotions from recorded gaits. Early works exploited different gait-based affective features to automatically detect perceived emotions (Karg, Kuhnlenz, and Buss 2010; Venture et al. 2014; Crenn et al. 2016; Wang, Enescu, and Sahli 2016; Daoudi et al. 2017). More recent works combined these affective features with features learned from recurrent (Randhavane et al. 2019a) or convolutional networks (Bhattacharya et al. 2020) to significantly improve classification accuracies.

Action Detection and Generation. There are large bodies of recent work on both gait-based supervised action detection (Choutas et al. 2018; Yan et al. 2019a; Yan, Xiong, and Lin 2018; Zhang et al. 2018b; Si et al. 2019; Shi et al. 2019a; Shi et al. 2019b; Liu et al. 2020), and gait-based

unsupervised action generation (Khodabandeh et al. 2018; Yan et al. 2019b; Holden, Saito, and Komura 2016; Pavllo, Grangier, and Auli 2018). These methods make use of RNNs or CNNs, including GCNs, or a combination of both, to achieve high classification accuracies on benchmark datasets such as Human3.6M (Ionescu et al. 2014), Kinetics (Carreira and Zisserman 2017), NTU RGB-D (Shahroudy et al. 2016), and more. On top of the deep-learned networks, some methods have also leveraged the kinematic dependencies between joints and bones (Shi et al. 2019a), dynamic movement-based features (Shi et al. 2019b), and long-range temporal dependencies (Liu et al. 2020), to further improve performance. A comprehensive review of recent methods in kinect-based action detection is available in (Wang, Huynh, and Koniusz 2020).

RNN and CNN-based approaches have been extended to semi-supervised classification as well (Harvey et al. 2018; Pavllo et al. 2019; Kanazawa et al. 2019; Zhang et al. 2019). These methods have also added constraints on limb proportions, movement constraints, and exploited the autoregressive nature of gait prediction to improve their generative and classification components.

Generative methods have also exploited full sequences of poses to directly generate full test sequences (Yang et al. 2018b; Cai et al. 2018a). Other approaches have used constraints on limb movements (Ahsan, Sun, and Essa 2018), action-specific trajectories (Holden, Saito, and Komura 2016), and the structure and kinematics of body joints (Pavllo, Grangier, and Auli 2018), to improve the naturalness of generated gaits.

In our work, we learn latent embeddings from gaits by exploiting the kinematic chains in the human body (Badler, Phillips, and Webber 1993) in a hierarchical fashion. Inspired by prior works in affect understanding from gaits, we also constrain our embeddings to contain the space of affective features expressed from gaits, to improve our average precision, especially

on the rarer classes.

3.3 Approach

Given both labeled and unlabeled 3D pose sequences for gaits, our goal is classify all the gaits into one or more discrete perceived emotion labels, such as happy, sad, angry, etc. We use a semi-supervised approach to achieve this, by combining an autoencoder with a classifier, as shown in Figure 3.3. We denote the set of trainable parameters in the encoder, decoder, and classifier with θ, ψ, and ϕ respectively. We first extract the rotation per joint from the first time step to the current time step in the input sequences (details in Section 3.3.2). We then pass these rotations through the encoder, denoted with $f_\theta(\cdot)$, to transform the input rotations into features in the latent embedding space. We pass these latent features through the decoder, denoted with $f_\psi(\cdot)$, to generate reconstructions of the input rotations. If training labels are available, we also pass the encoded features through the fully-connected classifier network, denoted with $f_\phi(\cdot)$, to predict the probabilities of the labels. We define our overall loss function as

$$\mathcal{C}(\theta, \phi, \psi) = \sum_{i=1}^{M} I_y^{(i)} \mathcal{C}_{CL}\left(y^{(i)}, f_{\phi\circ\theta}\left(D^{(i)}\right)\right) + \mathcal{C}_{AE}\left(D^{(i)}, f_{\psi\circ\theta}\left(D^{(i)}\right)\right), \tag{3.1}$$

where $f_{b\circ a}(\cdot) := f_b(f_a(\cdot))$ denotes the composition of functions, $I_y^{(i)}$ is an indicator variable denoting whether the i^{th} data point has an associated label $y^{(i)}$, M is the number of gait samples, $\mathcal{C}_{CL}$ denotes the classifier loss detailed in Section 3.3.3, and $\mathcal{C}_{AE}$ denotes the autoencoder loss detailed in Section 3.3.4. For brevity of notation, we will henceforth use $\hat{y}^{(i)} := f_{\phi\circ\theta}\left(D^{(i)}\right)$ and $\hat{D}^{(i)} := f_{\psi\circ\theta}\left(D^{(i)}\right)$.

Figure 3.2: **3D Pose Model for Gaits.** The names and numbering of the 21 joints in the pose follow the nomenclature in the ELMD dataset (Habibie et al. 2017).

Table 3.1: **Affective Features for Gaits – Descriptions.** List of the 18 pose affective features that we use to describe the affective feature space for our network.

Angles between	shoulders at lower back hands at root left shoulder and hand at elbow right shoulder and hand at elbow head and left shoulder at neck head and right shoulder at neck head and left knee at root head and right knee at root left toe and right toe at root left hip and toe at knee right hip and toe at knee
Distance ratios between	left hand index (LHI) to neck and LHI to root right-hand index (RHI) to neck and RHI to root LHI to RHI and neck to root left toe to right toe and neck to root
Area(Δ) ratios between	Δ shoulders to lower back and Δ shoulders to root Δ hands to lower back and Δ hands to root Δ hand indices to neck and Δ toes to root

3.3.1 Representing Emotions

The Valence-Arousal-Dominance (VAD) model (Mehrabian and Russell 1974) is used for representing emotions in a continuous space. This model assumes three independent axes for valence, arousal, and dominance values, which collectively indicate an observed emotion. Valence indicates how pleasant (vs. unpleasant) the emotion is, arousal indicates how much the emotion is tied to high (vs. low) physiological intensity, and dominance indicates how much the emotion is tied to the assertion of high (vs. low) social status. For example, discrete emotion terms such as happy indicate high valence, medium arousal, and low dominance, angry indicate low valence, high arousal, and high dominance, and sad indicate low valence, low arousal, and low dominance.

On the other hand, these discrete emotion terms are easily understood by non-expert annotators and end-users. As a result, most existing datasets for supervised emotion classification consist of discrete emotion labels, and most supervised methods report performance on pre-

dicting these discrete emotions. In fact, discrete emotions can actually be mapped back to the VAD space through various known transformations (Mehrabian 1996; Hoffmann et al. 2012). Given these factors, we choose to use discrete emotion labels in our work as well. We also note that human observers have been reported to be most consistent in perceiving emotions varying primarily on the arousal axis, such as happy, sad, and angry (Roether et al. 2009; Gross, Crane, and Fredrickson 2012). Hence we work with the four emotions, happy, sad, angry, and neutral.

3.3.2 Representing the Data

Given the 3D pose sequences for gaits, we first obtain the rotations per joint per time step. We denote a gait as $G = \left\{ \left(x_j^t, y_j^t, z_j^t \right) \right\}_{j=1,t=1}^{\mathcal{J},T}$, consisting of the 3D positions of $\mathcal{J}$ joints across T time steps. We denote the rotation of joint j from the first time step to time step t as $R_j^t \in \mathbb{SO}(3)$. We represent these rotations as unit quaternions $q_j^t \in \mathbb{H} \subset \mathbb{R}^4$, where $\mathbb{H}$ denotes the space of unit 4D quaternions. As stated in (Pavllo, Grangier, and Auli 2018), quaternions are free of the gimbal-lock problem, unlike other common representations such as Euler angles or exponential maps (Grassia 1998). We enforce the additional unit norm constraints for these quaternions when training our autoencoder. We represent the overall input to our network as $D^{(i)} := \left\{ q_j^t \right\}_{j=1,t=1}^{\mathcal{J},T} \in \mathbb{H}^{\mathcal{J} \times T}$.

3.3.3 Using Perceived Affects and Constructing Classifier Loss

Observers' perception of affects in others depends heavily influenced by their own personal, social, and cultural experiences, making affect understanding an inherently subjective task (Roether et al. 2009; Kleinsmith and Bianchi-Berthouze 2013). Consequently, we need to keep track of the differences in the perceptions of different observers. We do this by assigning multi-hot emotion

labels to each input gait.

We assume that the given labeled gait dataset consists of C discrete emotion classes. The raw label vector $L^{(i)}$ for the i^{th} gait is a probability vector where the l^{th} element denotes the probability that the corresponding gait is perceived to have the l^{th} emotion. Specifically, we assume $L^{(i)} \in [0,1]^C$ to be given as $L^{(i)} = [p_1 \dots p_C]^\top$, where p_l denotes the probability of the l^{th} emotion and $l = 1, 2, \dots C$. In practice, we compute the probability of each emotion for each labeled gait in a dataset as the fraction of annotators who labeled the gait with the corresponding emotion. To perform classification, we need to convert each element in $L^{(i)}$ to an assignment in $\{0,1\}$, resulting in the multi-hot emotion label $y^{(i)} \in \{0,1\}^C$. Taking into account the subjectivity in perceiving emotions, we set an element l in $y^{(i)}$ to 1 if $p_l > \frac{1}{C}$, *i.e.*, the l^{th} perceived emotion has more than a random chance of being reported, and 0 otherwise. Since our classification problem is multi-class (typically, $C > 2$) as well as multi-label (as we use multi-hot labels), we use the weighted multi-class cross-entropy loss

$$\mathcal{C}_{CL}\left(y^{(i)}, \hat{y}^{(i)}\right) := -\sum_{l=1}^{C} w_l \, (y_l)^{(i)} \log (\hat{y}_l)^{(i)} \tag{3.2}$$

for our classifier loss, where $(y_l)^{(i)}$ and $(\hat{y}_l)^{(i)}$ denote the l^{th} components of $y^{(i)}$ and $\hat{y}^{(i)}$, respectively. We also add per-class weights $w_l = e^{-p_l}$ to make the training more sensitive to mistakes on the rarer samples in the labeled dataset.

3.3.4 Using Affective Features and Constructing Autoencoder Loss

Our autoencoder loss consists of three constraints: affective loss, quaternion loss, and angle loss.

Affective loss. Prior studies in psychology report that a person's perceived emotions can be represented by a set of scale-independent gait-based affective features (Crenn et al. 2016). We consider the poses underlying the gaits to be made up of $\mathcal{J} = 21$ joints (Figure 3.2). Inspired by (Randhavane et al. 2019a), we categorize the affective features as follows:

- *Angles* subtended by two joints at a third joint. For example, between the head and the neck (used to compute head tilt), the neck, and the shoulders (to compute slouching), root and thighs (to compute stride lengths), etc.
- *Distance ratios* between two pairs of joints. For example, the ratio between the distance from the hand to the neck, and that from the hand to the root (to compute arm swings).
- *Area ratios* formed by two triplets of joints. For example, the ratio of the area formed between the elbows and the neck and the area formed between the elbows and the root (to compute slouching and arm swings). Area ratios can be viewed as amalgamations of the angle- and the distance ratio-based features used to supplement observations from these features.

We present the full list of the $\mathcal{A} = 18$ affective features we use in Table 3.1. We denote the set of affective features across all time steps for the i^{th} gait with $a^{(i)} \in \mathbb{R}^{\mathcal{A} \times T}$. We then constrain a subset of the embeddings learned by our encoder to map to these affective features. Specifically, we construct our embedding space to be $\mathbb{R}^{\mathcal{E} \times T}$ such that $\mathcal{E} \geq \mathcal{A}$. We then constrain the first $\mathcal{A} \times T$ dimensions of the embedding, denoted with $\hat{a}^{(i)}$ for the i^{th} gait, to match the corresponding affective features $a^{(i)}$. This gives our affective loss constraint:

$$\mathcal{L}_{\text{aff}}\left(a^{(i)}, \hat{a}^{(i)}\right) := \left\| a^{(i)} - \hat{a}^{(i)} \right\|^2. \tag{3.3}$$

We use affective constraints rather than providing affective features as input because there is no consensus on the universal set of affective features, especially due to cross-cultural differences (Ekman and Friesen 1969; Roether et al. 2009). Thus, we allow the encoder of our autoencoder to learn an embedding space using both data-driven features and our affective features, to improve generalizability.

Quaternion loss. The decoder for our autoencoder returns rotations per joint per time step as quaternions $\left(\hat{q}_j^t\right)^{(i)}$. We then constrain these quaternions to have unit norm:

$$\mathcal{L}_{\text{quat}}\left(\left(\hat{q}_j^t\right)^{(i)}\right) := \left(\left\|\left(\hat{q}_j^t\right)^{(i)}\right\| - 1\right)^2. \tag{3.4}$$

We apply this constraint instead of normalizing the decoder output, since individual rotations tend to be small, which leads the network to converge all its estimates to the unit quaternion.

Angle loss. This is the reconstruction loss for the autoencoder. We obtain it by converting the input and the output quaternions to the corresponding Euler angles and computing the mean loss between them:

$$\mathcal{L}_{\text{ang}}\left(D^{(i)}, \hat{D}^{(i)}\right) := \left\|(D_X, D_Y, D_Z)^{(i)} - \left(\hat{D}_X, \hat{D}_Y, \hat{D}_Z\right)^{(i)}\right\|_F^2 \tag{3.5}$$

where $(D_X, D_Y, D_Z)^{(i)} \in [0, 2\pi]^{3\mathcal{J}\times T}$ and $\left(\hat{D}_X, \hat{D}_Y, \hat{D}_Z\right)^{(i)} \in [0, 2\pi]^{3\mathcal{J}\times T}$ denotes the set of Euler angles for all the joints across all the time steps for input $D^{(i)}$ and output $\hat{D}^{(i)}$, respectively, and $\|\cdot\|_F$ denotes the Frobenius norm.

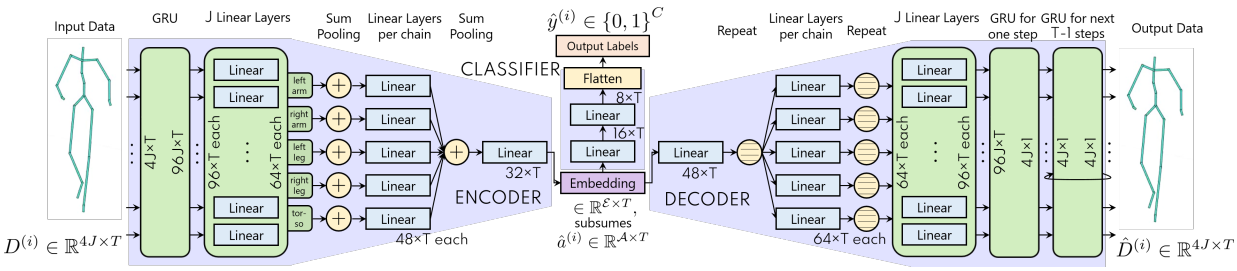


Figure 3.3: **TAEW: Semi-Supervised Network Architecture.** Inputs to the encoder are rotations on each joint at each time step, represented as 4D unit quaternions. The inputs are pooled bottom-up according to the kinematic chains of the human body. The embeddings at the end of the encoder are constrained to lie in the space of the mean affective features $\mathbb{R}^{\mathcal{A}}$. For labeled data, the embeddings are passed through the classifier to predict output labels. The linear layers in the decoder take in the embeddings and reconstruct the motion on each joint at a single time-step at the output of the first GRU. The second GRU in the decoder takes in the reconstructed joint motions at a single time step and predicts the joint motions for the next time step for $T-1$ steps.

Combining Eqs. 3.3, 3.4 and 3.5, we write the autoencoder loss $\mathcal{C}_{AE}(\cdot,\cdot)$ as

$$\mathcal{C}_{AE}\left(D^{(i)}, \hat{D}^{(i)}\right) := \mathcal{L}_{\text{ang}}\left(D^{(i)}, \hat{D}^{(i)}\right) + \lambda_{\text{quat}}\mathcal{L}_{\text{quat}} + \lambda_{\text{aff}}\mathcal{L}_{\text{aff}} \tag{3.6}$$

where λ_{quat} and λ_{aff} are the regularization weights for the quaternion loss constraint and the affective loss constraint, respectively. To keep the scales of $\mathcal{L}_{\text{quat}}$ and $\mathcal{L}_{\text{aff}}$ consistent, we also scale all the affective features to lie in $[0,1]$.

3.4 Network Architecture and Implementation

Our network for semi-supervised classification of discrete perceived emotions from gaits, shown in Figure 3.3, consists of three components, the encoder, the decoder, and the classifier. We describe each of these components and then summarize the training routine for our network.

3.4.1 Encoder with Hierarchical Attention Pooling

We first pass the sequences of joint rotations on all the joints through a two-layer Gated Recurrent Unit (GRU) to obtain feature representations for rotations at all joints at all time steps. We pass each of these representations through individual linear units. Following the kinematic chain of the human joints (Badler, Phillips, and Webber 1993), we pool the linear unit outputs for the two arms, the two legs, and the torso in five separate linear layers. Thus, each of these five linear layers learns to focus attention on a different part of the human body. We then pool the outputs from these five linear layers into another linear layer, which, by construction, focuses attention on the motions of the entire body. For pooling, we perform vector addition as a way of composing the features at the different hierarchies.

Our encoder learns the hierarchy of the joint rotations in a bottom-up manner. We map the output of the last linear layer in the hierarchy to a feature representation in the embedding space of the encoder through another linear layer. In our case, the embedding space lies in $\mathbb{R}^{\mathcal{E}\times T}$ with $\mathcal{E} = 32$, which subsumes the space of affective features $\mathbb{R}^{\mathcal{A}\times T}$ with $\mathcal{A} = 18$, as discussed in Section 3.3.4.

3.4.2 Decoder with Hierarchical Attention Un-pooling

The decoder takes in the embedding from the encoder, repeats it five times for un-pooling, and passes the repeated features through five linear layers. The outputs of these linear layers are features representing the reconstructions on the five parts, torso, two arms, and two legs. We repeat each of these features for un-pooling, and then collectively feed them into a GRU, which reconstructs the rotation on every joint at a single step. A subsequent GRU takes in the

reconstructed joint rotations at a single time step and successively predicts the joint rotations for the next $T - 1$ time steps.

3.4.3 Classifier for Labeled Data

Our classifier takes in the embeddings and passes it through a series of three linear layers, flattening the features between the second and the third linear layers. The output of the final linear layer, called "Output Labels" in Figure 3.3, provides the label probabilities. To make predictions, we set the output for a class to be 1 if the label probability for that class was more than $\frac{1}{C}$, similar to the routine for constructing input labels discussed in Section 3.3.3.

3.4.4 Training Routine

We train using the Adam optimizer (Kingma and Ba 2014) with a learning rate of 0.001, which we decay by a factor of 0.999 per epoch. We apply the ELU activation (Clevert, Unterthiner, and Hochreiter 2015) on all the linear layers except the output label layer, apply batch normalization (Ioffe and Szegedy 2015) after every layer to reduce internal covariance-shift, and apply a dropout of 0.1 to prevent overfitting. On the second GRU in the decoder, which predicts joint rotations for T successive time steps, we use a curriculum schedule (Bengio et al. 2015). We start with a teacher forcing ratio of 1 on this GRU and at every epoch E, we decay the teacher forcing ratio by $\beta = 0.995$, *i.e.*, we either provide this GRU the input joint rotations with probability β^E, or the GRU's past predicted joint rotations with probability $1 - \beta^E$. Curriculum scheduling helps the GRU to gently transition from a teacher-guided prediction routine to a self-guided prediction routine, thereby expediting the training process.

We train our network for 500 epochs, which takes around 4 hours on an Nvidia GeForce

GTX 1080Ti GPU with 12 GB memory. We use 80% of the available labeled data and all the unlabeled data for training our network, and validate its classification performance on a separate 10% of the labeled data. We keep the remaining 10% as the held-out test data. We also observed satisfactory performance when the weights λ_{quat} and λ_{aff} (in Equation 3.6) lie between 0.5 and 2.5. For our reported performances in Section 3.5.3, we used a value of 2 for both.

3.5 Results

We perform experiments with the Emotion-Gait benchmark dataset (Bhattacharya et al. 2020). It consists of 3D pose sequences of gaits collected from a variety of sources and partially labeled with perceived emotions. We provide a brief description of the dataset in Section 3.5.1. We list the methods we compare with in Section 3.5.2. We then summarize the results of the experiments we performed with this dataset on all these methods in Section 3.5.3, and describe how to interpret the results in Section 3.5.4.

3.5.1 Dataset

The Emotion-Gait dataset (Bhattacharya et al. 2020) consists of gaits collected from various sources of 3D pose sequence datasets, including BML (Ma, Paterson, and Pollick 2006), Human3.6M (Ionescu et al. 2014), ICT (Narang et al. 2017), CMU-MoCap (CMU-MOCAP 2018) and ELMD (Habibie et al. 2017). To maintain a uniform set of joints for the pose models collected from diverse sources, we converted all the models in Emotion-Gait to the 21 joint pose model used in ELMD (Habibie et al. 2017). We clipped or zero-padded all input gaits to have 240 time steps, and downsampled it to contain every 5^{th} frame. We passed the resultant 48 time steps to our network, we have *i.e.*, $T = 48$. In total, the dataset has $3,924$ gaits of which $1,835$ have

emotion labels provided by 10 annotators, and the remaining 2,089 are not annotated. Around 58% of the labeled data have happy labels, 32% have sad labels, 23% have angry labels, and only 14% have neutral labels (more details on the project webpage).

Histograms of Affective Features. We show histograms of the mean values of 6 of the 18 affective features we use in Figure 3.4. The means are taken across the $T = 48$ time steps in the input gaits and differently colored for inputs belonging to the different emotion classes as per the annotations. We count the inputs belonging to multiple classes once for every class they belong to. For different affective features, different sets of classes have a high overlap of values while values of the other classes are well-separated. For example, there is a significant overlap in the values of the distance ratio between right-hand index to the neck and right-hand index to the root (Figure 3.4, bottom left) for gaits belonging to sad and angry classes, while the values of happy and neutral are distinct from these. Again, for gaits in happy and angry classes, there is a high overlap in the ratio of the area between hands to lower back and hands to root (Figure 3.4, bottom right), while the corresponding values for gaits in neutral and sad classes are distinct from these. The affective features also support observations in psychology corresponding to perceiving emotions from gaits. For example, slouching is generally considered to be an indicator of sadness (Michalak et al. 2009). Correspondingly, we can observe that the values of the angle between the shoulders at the lower back (Figure 3.4, top left) are lowest for sad gaits, indicating slouching.

3.5.2 Comparison Methods

We compare our method with the following state-of-the-art methods for both affect and action detection from gaits. We choose to compare with action detection methods because similar to

these methods, we aim to learn a mapping from gaits to a set of labels (emotions instead of actions).

- **Affect Detection.** We compare with the network of (Randhavane et al. 2019a), which combines affective features from gaits with features learned from an LSTM-based network taking pose sequences of gaits as input, to form hybrid feature vectors for classification. We also compare with STEP (Bhattacharya et al. 2020), which trains a spatial-temporal graph convolution-based network with gait inputs and affective features obtained from the gaits, and then fine-tunes the network with data generated from a graph convolution-based variational autoencoder.
- **Action Detection.** We compare with recent state-of-the-art methods based on the spatial-temporal graph convolution network (STGCN) (Yan, Xiong, and Lin 2018), the directed graph neural network (DGNN) (Shi et al. 2019a), and the multi-scale graph convolutions with temporal skip connections (MS-G3D) (Liu et al. 2020). STGCN computes spatial neighborhoods as per the bone structure of the 3D poses and temporal neighborhoods according to the instances of the same joints across time steps and performs convolutions based on these neighborhoods. DGNN computes directed acyclic graphs of the bone structure based on kinematic dependencies and trains a convolutional network with these graphs. MS-G3D performs multi-scale graph convolutions on the spatial dimensions and adds skip connections on the temporal dimension to model long-range dependencies for various actions.

For a fair comparison, we retrained all these networks from scratch with the labeled portion of the Emotion-Gait dataset, following their respective reported training parameters, and the same data split of $8:1:1$ as our network.

Table 3.2: **TAEW: Average Precision Score Comparison.** Average precision (AP) per class and the mean average precision (mAP) over all the classes achieved by all the methods on the Emotion Gait dataset. Classes are Happy (H), Sad (S), Angry (A) and Neutral (N). Higher values are better. Bold indicates best, blue indicates second best.

Method	AP				mAP
	H	S	A	N	
STGCN (Yan, Xiong, and Lin 2018)	0.98	0.83	0.42	0.18	0.61
DGNN (Shi et al. 2019a)	0.98	0.88	0.73	0.37	0.74
MS-G3D (Shi et al. 2019a)	0.98	0.88	0.75	0.44	0.76
LSTM Network (Randhavane et al. 2019a)	0.96	0.84	0.62	0.51	0.73
STEP (Bhattacharya et al. 2020)	0.97	0.88	0.72	0.52	0.77
TAEW (Our Method)	**0.98**	**0.89**	**0.81**	**0.71**	**0.84**

3.5.2.1 Evaluation Metric

Since we deal with a multi-class, multi-label classification, we report the average precision (AP) achieved per class, which is the mean of the precision values across all values of recall between 0 and 1. We also report the mean AP, which is the mean of the APs achieved in all the classes.

3.5.3 Experiments

In our experiments, we ensured that the held-out test data were from sources different from the train and validation data in the Emotion-Gait dataset. We summarize the AP and the mean AP scores of all the methods in Table 3.2. Our method outperforms the next best method, STEP (Bhattacharya et al. 2020), by around 7% and outperforms the lowest-performing method, STGCN (Yan, Xiong, and Lin 2018), by 23%, both on the absolute. We summarize additional results, including the interpretation of the data labels and our results in the VAD dimensions (Mehrabian and Russell 1974), on our project webpage.

Both the LSTM-based network and STEP consider per-frame affective features and inter-frame features such as velocities and rotations as inputs but do not explicitly model the dependencies between these two kinds of features. Our network, on the other hand, learns to embed a part of the features learned from joint rotations in the space of affective features. These embedded features, in turn, help our network predict the output emotion labels with more precision.

The action detection methods STGCN, DGNN, and MS-G3D focus more on the movements of the leaf nodes, *i.e.*, hand indices, toes, and head. These nodes are useful for distinguishing between actions such as running and jumping but do not contain sufficient information to distinguish between perceived emotions.

Moreover, given the long-tail nature of the distribution of labels in the Emotion-Gait dataset (Section 3.5.1), all the methods we compare with have more than 0.95 AP in the happy and more than 0.80 AP in the sad classes, but perform much poorer on the angry and the neutral classes. Our method, by contrast, learns to map the joint motions to the affective features, which helps it achieve around 10–50% better AP on the absolute on the angry and the neutral class while maintaining similarly high AP in the happy and the sad classes.

3.5.3.1 Ablation Studies

We also perform ablation studies on our method to highlight the benefit of each of our three key components: using hierarchical pooling (HP) (Section 3.4.1), using the affective loss constraint (AL) (Equation 3.3), and using both labeled and unlabeled data in a semi-supervised manner (Equation 3.1). We summarize the observations of our ablation studies in Table 3.3.

First, we train our network only on the labeled dataset by removing the decoder part of our network and dropping the autoencoder loss from Equation 3.1. Without using either AL or

Table 3.3: **TAEW: Ablation Studies.** Comparing average precisions of ablated versions of our method. HP denotes Hierarchical Pooling, AL denotes the Affective Loss constraint. AP, mAP, H, S, A, N are reused from Table 3.2. Bold indicates best, blue indicates second best.

Method	AP				mAP
	H	S	A	N	
With only labeled data, no AL or HP	0.92	0.81	0.51	0.42	0.67
With only labeled data, HP and no AL	0.93	0.81	0.63	0.49	0.72
With only labeled data, AL and no HP	0.96	0.86	0.70	0.51	0.76
With only labeled data, AL and HP	0.97	0.86	0.72	0.55	0.78
With all data, no AL or HP	0.94	0.83	0.55	0.48	0.70
With all data, HP and no AL	0.96	0.85	0.70	0.60	0.78
With all data, AL and no HP	0.97	0.87	0.76	0.65	0.81
With all data, AL and HP	**0.98**	**0.89**	**0.81**	**0.71**	**0.84**

HP, the network achieves an AP of 0.51 on angry and 0.42 on neutral, the two least populous classes. We call this our baseline network. Adding only the AL increases these two APs more from the baseline than adding only the HP. This is reasonable since hierarchical pooling helps the network learn generic differences in the pose sequences of different data, while the affective loss constraint helps the network to distinguish between pose structures specific to different perceived emotions. Adding both HP and AL increases the AP from the baseline even further. From these experiments, we can confirm that using either AL or HP improves the performance from the baseline, and their collective performance is better than their individual performances.

Next, we add in the decoder and use both labeled and unlabeled data for training our network, using the loss in Equation 3.1. Without both AL and HP, the network now achieves an AP of 0.55 on angry and 0.48 on neutral, showing appreciable improvements from the baseline. Also, as earlier, adding in only the AL shows more benefit on the network's performance than adding in only the HP. Specifically, adding in only the HP produces 1% absolute improvement in

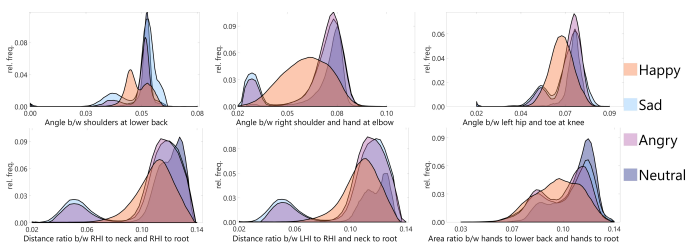


Figure 3.4: **Conditional Distributions of Mean Affective Features on the E-Gait Dataset.** Distributions of 6 of the 18 affective features, for the Emotion-Gait dataset, conditioned on the given classes Happy, Sad, Angry, and Neutral. Mean is taken across the number of time steps. We observe that the different classes have different distributions of peaks, indicating that these features are useful for distinguishing between perceived emotions.

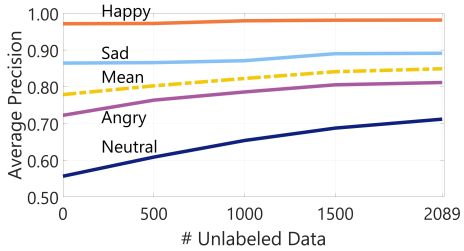


Figure 3.5: **Effect of Adding Unlabeled Data to TAEW Training.** AP achieved on each class, as well as the mean AP over the classes, increases linearly as we add more unlabeled data to train our network. The increment is most significant for the neutral class, which has the fewest labels in the dataset.

mean AP over STEP (Bhattacharya et al. 2020) (row 4 in Table 3.2) and 17% absolute improvement in mean AP over STGCN (Yan, Xiong, and Lin 2018) (row 1 in Table 3.2). Adding in only the AL produces 4% absolute improvement in mean AP over STEP (Bhattacharya et al. 2020) (row 4 in Table 3.2) and 20% absolute improvement in mean AP over STGCN (Yan, Xiong, and Lin 2018) (row 1 in Table 3.2). Adding in both, we get the final version of our network, which improves on the mean AP of STEP (Bhattacharya et al. 2020) by 7%, and the mean AP of STGCN (Yan, Xiong, and Lin 2018) by 23%.

3.5.3.2 Performance Trend with Increasing Unlabeled Data

In practice, it is relatively easy to collect unlabeled gaits from videos or using motion capture. We track the performance improvement of our network as we keep adding unlabeled data to our network, and summarize the results in Figure 3.5. We observe that the mean AP improves linearly as we add more data. The trend does not indicate a saturation in AP for the angry and the neutral classes even after adding all the 2,089 unlabeled data. This suggests that the

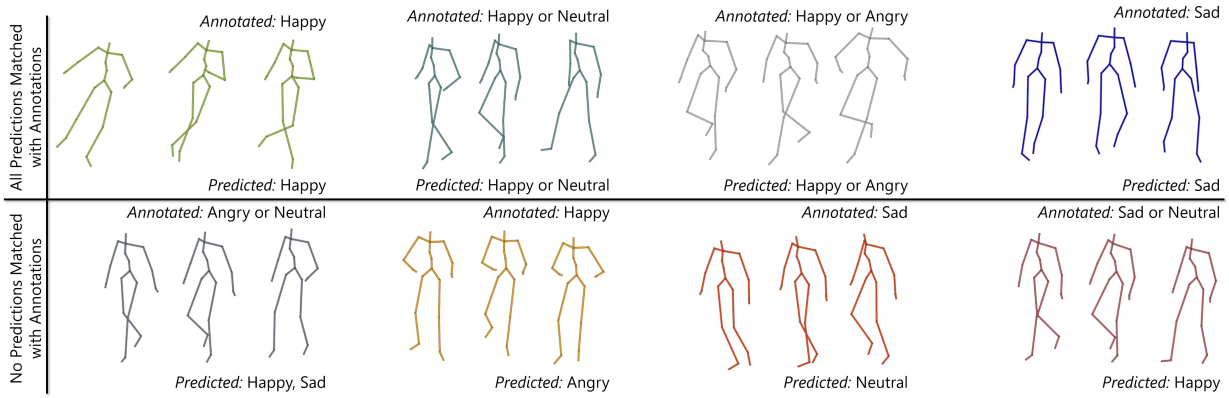


Figure 3.6: **Comparing TAEW Predictions with Human Annotations.** The top row shows 4 gaits from the Emotion-Gait dataset where the predicted labels of our network exactly matched the annotated input labels. The bottom row shows 4 gaits where the predicted labels did not match any of the input labels. Each gait is represented by 3 poses in temporal sequence from left to right. We observe that most of the disagreements are between either happy and angry or between sad and neutral, which is consistent with general observations in psychology.

performance of our approach can increase further with more unlabeled data.

3.5.4 Interpretation of the Network Predictions

We show the qualitative results of our network in Figure 3.6. The top row shows cases where the predicted labels for a gait exactly matched all the corresponding annotated labels. We observe that the gaits with happy and angry labels in the annotation have more animated joint movements compared to the gaits with sad and neutral labels, which our network was able to successfully learn from the affective features. This is in line with established studies in psychology (Mehrabian and Russell 1974), which show that both happy and angry emotions lie high on the arousal scale, whereas neutral and sad are lower on the arousal scale. The bottom row shows cases where the predicted labels for a gait did not match any of the annotated labels. We notice that most disagreements arise either between sad and neutral labels or between happy and angry labels. This again follows the observation that both happy and angry gaits, higher

on the arousal scale, often have more exaggerated joint movements, while both sad and neutral gaits, lower on the arousal scale, often have more reserved joint movements. There are also disagreements between happy and neutral labels for some gaits, where the joint movements in the happy gaits are not as exaggerated.

We also make an important distinction between the multi-hot input labels provided by human annotators and the multi-hot predictions of our network. The input labels capture the subjectivity in human perception, where different human observers perceive different emotions from the same gait based on their own biases and prior experiences (Roether et al. 2009). The network, on the other hand, indicates that the emotion perceived from a particular gait data best fits one of the labels it predicts for that data. For example, in the third result from left on the top row in Figure 3.6, five of the ten annotators perceived the gait to be happy, three perceived it to be angry, and the remaining two perceived it to be neutral. Following our annotations procedure in Section 3.4.3, we annotated this gait as an instance of both happy and angry. Given this gait, our network predicts a multi-hot label with 1's for happy and angry and 0's for neutral and sad. This indicates that the network successfully focused on the arousal in this gait, and found the emotion perceived from it to best match either happy or angry, and not match neutral and sad. We present more such results on our project webpage.

3.6 Conclusion

We have expanded affect detection from gaits to a semi-supervised method that can classify gaits by learning underlying gait patterns from both labeled and unlabeled data. Our method learns by hierarchically pooling and un-pooling the body joints following the kinematic chains in the human body, while learning data-driven latent embeddings. To better learn from pose-

based affective features, we have also trained our semi-supervised network by constraining part of its latent embeddings to match the space those affective features. Further, we have relaxed the one-hot discrete affect labels to multi-hot labels to allow for subjectivity in affect perception. Our method achieves highest improvement in performance (an absolute 10%–50% compared to the baselines) on the most rarely populated affect classes, highlighting how our design of the latent embeddings together with the use of unlabeled data for training avoids data overfitting.

CHAPTER 4

Affective Gait Synthesis Using Conditional Autoregression

Project Website: https://gamma.umd.edu/gen_emotive_gaits

Abstract

We present a novel autoregression network to generate virtual agents that convey various emotions through their walking styles or gaits. Given the 3D pose sequences of a gait, our network extracts pertinent movement features and affective features from the gait. We use these features to synthesize subsequent gaits such that the virtual agents can express and transition between emotions represented as combinations of happy, sad, angry, and neutral. We incorporate multiple regularizations in the training of our network to simultaneously enforce plausible movements and noticeable emotions on the virtual agents. We also integrate our approach with

an AR environment using a Microsoft HoloLens and can generate affective gaits at interactive rates to increase the social presence. We evaluate how human observers perceive both the naturalness and the emotions from the generated gaits of the virtual agents in a web-based study. Our results indicate around 89% of the users found the naturalness of the gaits satisfactory on a five-point Likert scale, and the emotions they perceived from the virtual agents are statistically similar to the intended emotions of the virtual agents. We also use our network to augment existing gait datasets with affective gaits and will release this augmented dataset for future research in emotion prediction and affective gait synthesis.

4.1 Introduction

Having developed state-of-the-art methods for affect detection from gaits, we now turn our attention to the harder problem of affect synthesis for intelligent virtual agents (IVAs). The generation of IVAs is important for many virtual and augmented reality systems. The virtual agents correspond to embodied digital characters that are often used as avatars to represent the users and may look like real-world characters. Recent work in photorealistic rendering and capturing technologies has resulted in generating agents or avatars that closely resemble the humans and are widely used in VR and AR systems (*NEON: https://www.neon.life/* 2020; Gonzalez-Franco et al. 2020).

Many applications, including virtual assistance, training, and AI chatbots, need a computationally created virtual agent or an avatar that not only looks like a real human but also behaves like one and conveys emotions (Latoschik et al. 2017; Randhavane et al. 2019b). Perception of such emotional expressiveness is commonly described as the ability of an observer to make decisions on a subject's emotional state by observing certain patterns or cues physically

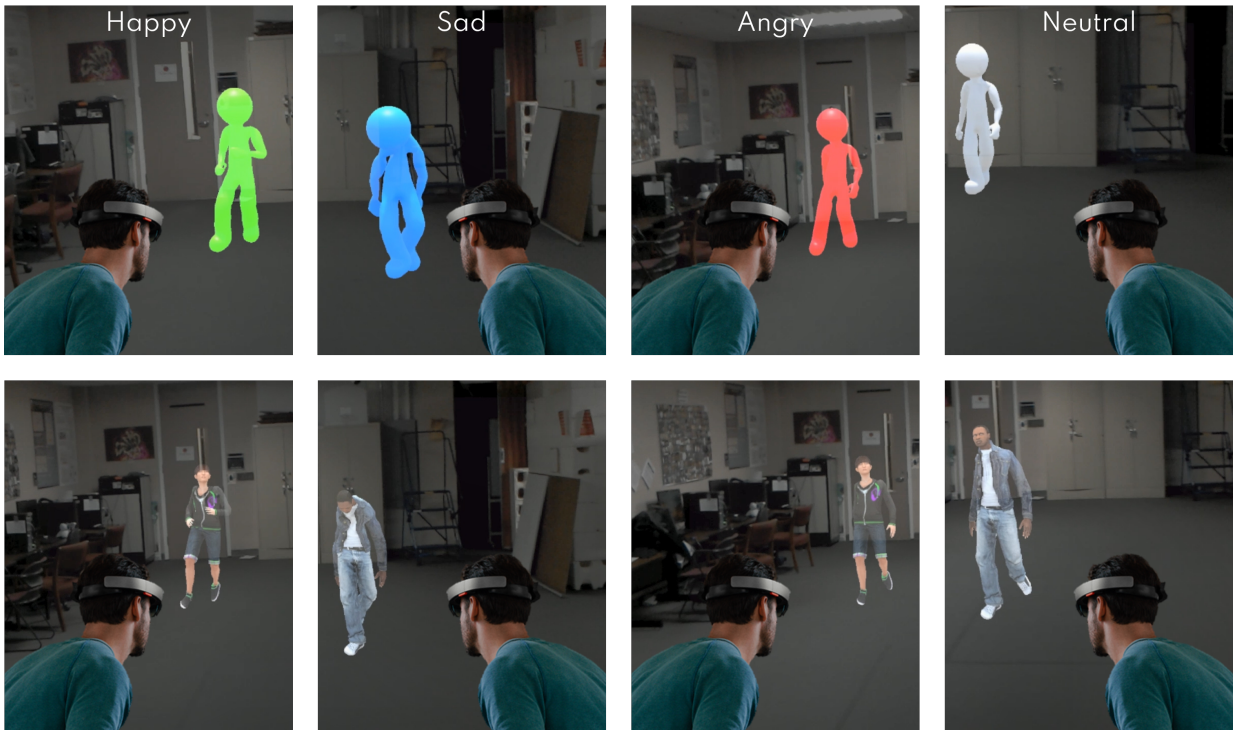


Figure 4.1: **Affective Gait Synthesis: Overview.** Samples of the affective gaits for VAs generated by our learning-based algorithm in an AR environment. The top row shows stick figures and the bottom row shows human models corresponding to the VAs. The VAs can exhibit different emotions based on their gaits, as indicated at the top of each column, while walking along user-defined trajectories at interactive rates.

expressed by the subject (Phillips et al. 2003; Roether et al. 2009; Gross, Crane, and Fredrickson 2012). These physical cues are expressed through various "modalities," including, but not limited to, facial expressions (Ekman and Friesen 1978), tones of voice (Frick 1985), gestures, body expressions (Dael, Mortillaro, and Scherer 2012), walking styles or "gaits" (Montepare, Goldstein, and Clausen 1987), etc. Emotions coming from such different modalities (Mittal et al. 2020a), in conjunction with the different underlying situational and social contexts (Kosti et al. 2019; Mittal et al. 2020b), have a significant impact on our everyday lives. They influence our social interactions and relationships and provide key insights into developing healthy social environments (Keltner and Haidt 2001). Similarly, emotions can greatly impact the perception of these virtual agents in terms of social presence and how the users behave when interacting with them

in AR and VR environments (Latoschik et al. 2019; Moustafa and Steed 2018; Randhavane et al. 2019b).

In this paper, we mainly focus on designing virtual agents that are capable of expressing different emotions through their gaits, *i.e.* virtual agents with *affective gaits*. When perceiving emotions from gaits, humans generally look at physical expressions such as arm swing, stride length, upper-body posture, head jerk, etc. (Crenn et al. 2016), collectively referred to as *affective features*. In fact, studies have shown that observers often rely on such cues from gaits and other body expressions, especially when there are mismatches with cues from more common modalities such as facial expressions (Aviezer, Trope, and Todorov 2012). As a result, it is useful to generate virtual agents with affective gaits for gaming and social VR (Osking and Doucette 2019; Stangl, Ukpabi, and Park 2020), crowd simulation, and path planning (Bauer et al. 2009; Narayanan et al. 2020), therapy, rehabilitation (Rivas et al. 2015), and psychology and neurobiology (Nestler et al. 2002; Barrett, Mesquita, and Gendron 2011; Rosenberg et al. 2019). Studies indicate that virtual agents expressing various moods, emotions, and behaviors elicit more empathy and engagement from humans interacting with them (Riek et al. 2009). However, automated methods to generate gaits that express certain emotions are challenging to design and implement. This is hard not only because of the complexity of modeling periodic and aperiodic motions constituting different gaits, but also because of the individual, social, and cultural diversities in terms of both expressing and perceiving emotions (Altarriba, Basnight, and Canary 2003; Gendron et al. 2014). These challenges are further exacerbated by the difficulty of collecting and annotating large benchmark datasets of gaits with appropriate emotion labels. Datasets collected in controlled experimental settings are often exaggerated and not always generalizable. On the other hand, datasets collected in the wild often suffer from

the long-tail nature of the distribution of emotions, with most emotion being close to neutral. Therefore, there is a need to develop automated methods to synthesize affective gaits for virtual agents that can augment the existing datasets.

There is an extensive work in AR/VR, computer graphics, vision and related areas such as biomechanics, on automated techniques for generating human characters capable of performing locomotion activities, including walking, running, leaping, and more (Randhavane et al. 2021; Randhavane et al. 2019c; Randhavane et al. 2019b; Holden, Saito, and Komura 2016; Holden, Komura, and Saito 2017; Peng et al. 2017; Yumer and Mitra 2016). These methods make use of movement features such as joint rotations, joint velocities, frequency of foot contact with the ground, and walking phases, and combine them with structural and kinematic constraints of the human body and learning techniques. While these methods can generate plausible locomotion, walking patterns, and actions, it is non-trivial to add emotional components to these techniques or use them to generate affective gaits.

Main Results: We present a novel autoregression method that takes as input 3D pose sequences of the gaits of virtual agents (VAs) and efficiently combines pose affective features such as arm swings, head jerks, body posture and more, and movement features such as stepping speed, root height and more, to generate future pose sequences of affective gaits. We present a network architecture that incorporates both spatial information and spectral information available from the input pose sequences and enables the VAs to both express and transition smoothly between different emotions while walking. We construct VAs as both stick figures and human models using our generated affective gaits and integrate these VAs in an AR environment using the Microsoft HoloLens (Figure 4.1). Our learning-based algorithm takes a few milliseconds to generate an affective gait for each agent on a pair of NVIDIA GeForce GTX 1080Ti GPUs. Our

VAs, overlayed onto a real-world room, are rendered at interactive rates to increase their sense of social presence. The novel components of our work include:

- An autoregression network that takes in 3D pose sequences of a VA's gait, the desired future trajectory, and the desired emotions. It outputs the VA's gait expressing the given emotion while following the given trajectory.
- A novel training method combining movement features and psychologically-motivated affective features into a unified network to generate plausible, emotionally-expressive gaits.
- A transition scheme for the characters to smoothly transition between gaits expressing different emotions.
- An elaborate web-based user study to evaluate the benefits of the affective gaits generated by our algorithm. We asked the observers to report the emotions they perceived from the generated gaits, as well as the Likert scale (LS) values of pose affective features that contributed to their perception. Based on the study, we conclude that
  - There is strong statistical evidence to suggest that the observers' perceived emotions are statistically similar to the corresponding intended emotions of our VAs, thereby showing that the generated gaits are emotive,
  - The observers consistently reported different LS values of the pose affective features for different emotions, making our choice of pose affective features statistically significant for perceiving emotional expressiveness.
- An augmented dataset, "Synthesized Affective Gait," which provides affective gaits generated by our method to facilitate more research in this area.

4.2 Related Work

In this section, we briefly survey prior work on representing emotions, perceiving emotions from gaits, generating and styling gaits for virtual agents, and making virtual agents emotionally expressive.

4.2.1 Modeling and Perceiving Emotions from Gaits

Various models for representing emotions have been studied in psychology, and the Valence-Arousal-Dominance (VAD) model (Mehrabian and Russell 1974) is one of the most popular. The VAD model considers a continuous 3D space, spanned by the valence, arousal, and dominance axes. Valence is a measure of the pleasantness of emotion, arousal is a measure of the intensity of expression, and dominance is the measure of how much emotion makes one feel in control. Many methods use a simpler model that is a linear combination of discrete emotions and represents a subset of VAD.

Humans perceive these emotions by observing physical features or cues expressed via different modalities. Studies conducted by Montepare, Goldstein, and Clausen (1987) concluded that observers were able to perceive emotions by only looking at the subjects' gaits. Subsequently, Roether et al. 2009 and Gross, Crane, and Fredrickson 2012 identified that observers were most consistent when looking at gaits expressing emotions that varied on the arousal axis. Follow-up studies looked more closely at the gait-based expressions observers focused on for distinguishing between different perceived emotions and identified features including arm swing, gait velocity, upper body posture, and head jerk (Karg et al. 2013; Crenn et al. 2016; Bhattacharya et al. 2020; Bhattacharya et al. 2020). In contrast to prior approaches, our goal is to

use affective features and movement features to synthesize gaits with emotions varying on the arousal axis.

4.2.2 Emotional Expressiveness in Virtual Agents

Prior works have commonly explored the generation of emotionally expressive virtual agents via modalities such as verbal communication (Chowanda et al. 2016; Sohn et al. 2018), face movements (Ferstl and McDonnell 2018a), body gestures (Jaques et al. 2016), and gaits (Randhavane et al. 2021; Randhavane et al. 2019b; Randhavane et al. 2019c). These generation techniques have had significant performance benefits when combined with concepts from affective computing. For example, Pelczer, Contreras, and Rodríguez 2007 designed a strategy to evaluate the accuracy of identifying the modalities of emotional expressiveness in a virtual agent. McHugh et al. 2010 explored how different body postures influenced the understanding of affects of individual agents in crowds. Clavel et al. 2009 studied the combined effect of faces and postures of virtual agents on affect understanding, and Liebold and Ohler 2013 generalized this to include combinations of other modalities such as verbal cues and faces. More recently, Randhavane et al. 2019c developed an empirical mapping between gait and gaze features and different emotions to generate emotionally expressive virtual agents. Our approach to generating affective gaits is complementary to these methods and can be combined with them.

4.2.3 Generating and Styling Gaits for Virtual Agents

There has been extensive prior work in computer graphics and AR/VR for generating and styling gaits for virtual agents. Early approaches used patch-based building blocks (Lee, Choi, and Lee 2006), kernel-based approaches (Wang, Fleet, and Hertzmann 2007), or modeled the motion

paths as directed graphs (Kovar, Gleicher, and Pighin 2008) to generate natural-looking movement styles. Recent approaches have leveraged large-scale datasets using deep learning-based approaches to generate diverse movement styles. These approaches include training a network on specific joint trajectories (Rokbani, Cherif, and Alimi 2009; Holden, Saito, and Komura 2016), using periodic phase-functions, which are either modeled geometrically (Holden, Komura, and Saito 2017) or learned with a neural network (Starke et al. 2019), to represent walking cycles, and exploiting transfer to reduce over-dependency on data (Mason et al. 2018). Other approaches use deep reinforcement learning to learn control policies for virtual characters exhibiting different movement styles and actions (Lee et al. 2019; Park et al. 2019; Peng et al. 2017; Peng et al. 2018). Yet other approaches model motion prediction as an autoregression problem, and have utilized recurrent networks (Pavllo, Grangier, and Auli 2018) and convolutional networks (Li et al. 2018) on motion captures pose sequences, and generative adversarial learning on dynamic pose graphs (Cui, Sun, and Yang 2020), to predict future motions. While these methods are not built for motion styling, their key concepts have been useful in developing many motion-based style transfer methods (Xia et al. 2015; Du et al. 2019; Kfir et al. 2020).

In contrast with these methods, we combine walking phases and gait-based affective features in an autoregression network to estimate future joint rotations and movement features for different affective gaits. Instead of a DRL-based control policy, our network learns a feature-based latent representation space and maps from that space to emotion-styled predicted poses. Furthermore, our emotion styles are sampled from a continuous space of emotions. Therefore, they need to be modeled differently from conventional styles, which can be viewed as one-hot labels in a discrete space. We also demonstrate that our approach can generate gaits expressing a continuous range of emotions, for AR applications.

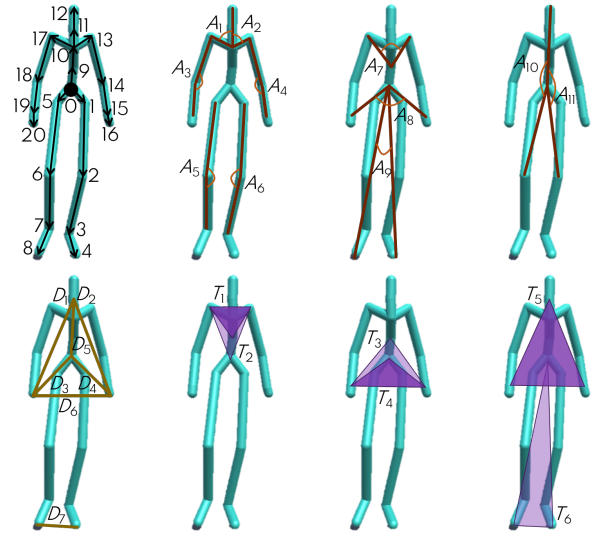


Figure 4.2: **Affective Features for Gaits.** The top left figure shows our pose graph as a directed tree, with the joints numbered 0 through 20. We use 18 affective features, counting 11 joint angles, 4 distance ratios, and 3 area ratios. The joint angles are labeled A_1 through A_{11}, and marked with red arcs on the last three figures in the top row. The leftmost figure on the bottom row shows the distances we use to compute the distance ratios. We use the ratios $\frac{D_1}{D_2}$, $\frac{D_3}{D_4}$, $\frac{D_6}{D_5}$, and $\frac{D_7}{D_5}$. The last three figures on the bottom row show the triangles we use to compute area ratios. We use the ratios $\frac{T_1}{T_2}$, $\frac{T_3}{T_4}$, $\frac{T_5}{T_6}$. These features are used by our network to generate affective gaits of the virtual agents.

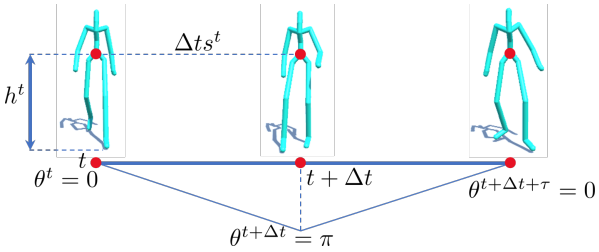


Figure 4.3: **Movement Features – Graphical.** We show the root height from the ground h^t, the root speed s^t, and the stepping phase θ^t. The root speed is the distance travelled between time steps t and $t - 1$. The stepping phase $\theta^t = 0$ when the left foot touches the ground at time step t, $\theta^{t+\Delta t} = \pi$ when the right foot touches the ground at time step $t + \Delta t$, and $\theta^{t+\Delta t+\tau} = 0$ when the left foot touches the ground again. We fill in the values for θ^t between these time steps using linear interpolation. We use these features in our autoregression network.

4.3 Generating Affective Gaits

In this section, we present our approach for generating affective gaits of VAs. The inputs to our algorithm are the sample gaits of a VA provided as motion capture data, the desired trajectory, and the desired emotion. Our goal is to generate subsequent predictions of the gait that follow the desired trajectory and express the desired emotion. Since our approach depends only on the input motion-captured gait samples and the desired trajectories, we can adapt to VAs

with different skeletal dimensions, different natural walking styles, as well as to different AR environments.

4.3.1 Emotion Model

We choose to use an emotion model consisting of linear combinations of categorical emotions varying primarily on the arousal axis (happy, angry, sad, etc.). Although this model admittedly spans a smaller set of emotions than the VAD model, prior works have reported that categorical emotion terms are more easily understood by non-experts, leading to the availability of more labeled data and the generation of a diverse range of emotions (Randhavane et al. 2019a; Bhattacharya et al. 2020).

4.3.2 Construction of the Generative Components

We present various components used by our network to perform generation: input gaits, emotions, pose affective features, and trajectory features.

4.3.2.1 Gaits

We denote a gait G as $G = \left\{ X_j^t = \left[x_j^t, y_j^t, z_j^t\right]^\top \in \mathbb{R}^3 \right\}_{j=0,t=0}^{\mathcal{J}-1,T-1}$, where $\left[x_j^t, y_j^t, z_j^t\right]^\top$ denotes the 3D positions of joint j at time step t in the world frame, $\mathcal{J}$ denotes the total number of joints, and T denotes the total number of input time steps. The input to our network are joint rotations extracted from the gait G, and control signals obtained from the gait, its trajectory, and the associated emotions.

At each time step t, we model the pose graph of the input gait as a directed tree, as shown in Figure 4.2. The root joint in the pose is the root node of the tree and the two toes, the two hand

indices, and the head, are the leaf nodes the tree. All edges in the tree are directed from the root node to the leaf nodes. We denote the parent of a joint j as $P(j)$. In our construction, each joint has a unique parent, except the root joint, which has no parent. We therefore assign $P(j) = -1$ for the root joint. For each joint j at each time step t, we consider the rotation $R_j^t \in \mathbb{SO}(3)$ that transforms the joint from its offset $o_j \in \mathbb{R}^3$ — a pre-defined initial position relative to its parent $P(j)$ — to its position at time step t relative to its parent. That is, we consider the rotation R_j^t such that the global position X_j^t of the joint j at time step t is given by $X_j^t = R_j^t o_j + X_{P(j)}^t$. For the root joint, we consider $o_0 = \mathbf{0}$ and obtain its position directly from the gait G.

Following the approach taken by Pavllo, Grangier, and Auli 2018, we represent these rotations as unit quaternions or versors $q_j^t \in \mathbb{H} \subset \mathbb{R}^4$, where $\mathbb{H}$ denotes the space of versors. We have chosen to represent rotations as versors, as they are free of the gimbal-lock problem. We enforce the additional unit norm constraints for these versors when training our network. Thus, for each gait, our input rotations are given as

$$\left\{ q^t = \left[q_1^t; \ldots; q_{\mathcal{J}-1}^t \right] \in \mathbb{H}^{\mathcal{J}-1} \right\}_{t=0}^{T-1}.$$

We do not consider q_0^t for the root joint denoted by 0 as part of the input data, since $o_0 = 0$.

4.3.2.2 Emotions

We note that the modeling of emotions is fundamentally different from modeling conventional motion styles such as strutting, zombie-like, etc. These conventional styles can be considered "discrete", such that clips or images can be categorized as belonging to a particular style. Emotions, on the other hand, span a continuous space (Mehrabian and Russell 1974), such that dif-

ferent motion clips can have different intensities of the same emotion. For example, a somewhat happy gait is expressed differently from one that is extremely happy. For conventional styles, it is generally not needed to account for such intensities, e.g., slightly strutting vs. heavily strutting. To account for this continuous nature, we assume each gait in the input dataset is associated with an emotion vector whose components are the C categorical emotion terms, *i.e.*, each emotion m is a vector in $\mathbb{R}^C$. In practice, the value of each element l in an emotion vector m is the relative count of the number of annotators who labeled the corresponding gait with the categorical emotion term l. Also, for training, and due to the practical limitations of separately annotating the emotion at each time step, we repeat the same annotated emotion m in all the time steps. In other words, we assume the emotion vector remains unchanged throughout the corresponding input gait.

4.3.2.3 Pose Affective Features

Prior studies in psychology have shown that various physically-based pose features observed per-frame during a gait, better known as *affective features*, aid the identification of perceived emotions from gaits (Karg et al. 2013; Crenn et al. 2016). Roether et al. 2009 identified such a set of necessary pose affective features for human perception. To make these features suitable for machine perception, prior works such as (Randhavane et al. 2019a; Bhattacharya et al. 2020; Bhattacharya et al. 2020) have come up with necessary sets of scale-independent pose affective features that can be computed geometrically. Scale independence is an important factor in such intra-frame affective features, as observers can identify emotions irrespective of the distance from or the physical stature of the subject. In our work, we use the following three types of scale-independent pose affective features to encode the relevant emotion information:

Angles. We use the angles subtended by a pair of joints at a third joint. For example, the angle between the two shoulder joints at the neck measures slouching, an indicator of valence and arousal.

Distance ratios. We use the ratios of the distances between two pairs of joints. For example, the ratio of the distance between the two feet joints to the distance between the neck and the root joints measures the stride, which can indicate arousal and dominance.

Area ratios. We use the ratios of areas formed by two triplets of joints. These can be considered as amalgamations of the angle- and the distance ratio-based features and they can be used to supplement observations from both these types of features. For example, the ratio of the area of the triangle formed by the hand indices and the neck to the area of the triangle formed by the toes at the root can be used to simultaneously measure arm swings and strides, which can collectively indicate the valence, arousal, and dominance.

We use 11 angles, four distance ratios and three area ratios for a total of 18 pose affective features, which we collectively denote as $a^t \in \mathbb{R}^{18}$ at each time step t. We list all these pose affective features in Figure 4.2, and direct the interested reader to (Bhattacharya et al. 2020) for a detailed analysis on choosing these features.

4.3.2.4 Movement Features

We use a trajectory followed throughout a gait to extract pertinent movement information for our network to generate gaits following given trajectories. We use two kinds of movement features: *root joint features* and *stepping features*. The former consists of root height deviation, root speed (a low-pass filtered component), and root orientation difference trajectory curvature. The

latter consists of stepping phase and foot-step frequency, which are obtained from the trajectory. Figure 4.3 illustrates some of these features. Apart from movement information, the root joint features and the foot-step frequency also provide inter-frame or dynamic affective information for emotional expressions. We define these features below.

Root joint features. The root height deviation $\left(h^t\right)$ is the signed difference of the height of the root joint from its mean height from the ground plane across the time steps. Subjects expressing emotions with higher arousal tend to have their upper bodies more upright, thus keeping the root height above the mean more often than subjects expressing emotions with lower arousal. In our case, the XZ plane is the ground plane, making the root height $h^t = y_0^t$.

The root speed $\left(s^t\right)$ is the magnitude of the difference of the 2D position of the root joint, as projected on the ground plane, between the current step t and the previous step $t-1$. Root speed helps indicate the arousal as well, with higher arousal tending to result in faster speeds more often. We represent the root speed as $s^t = \left\| \left[x_0^t, z_0^t\right]^\top - \left[x_0^{t-1}, z_0^{t-1}\right]^\top \right\|$. With root speed, we also use its loss-pass filtered component $\bar{s}^t$. This reduces the high-frequency noise in the root speed, which is especially useful when the network learns on trajectories with high curvatures.

The root orientation difference $\left(\delta^t\right)$ is the angular difference between the root orientation α^t w.r.t. the world coordinates and the tangent τ^t to the 2D root joint positions $\left[x_0^t, z_0^t\right]^\top$ on the ground plane, w.r.t. the world coordinates at each time step t. We express the tangent using forward difference, *i.e.*, $\tau^t = \left[x_0^t, z_0^t\right]^\top - \left[x_0^{t-1}, z_0^{t-1}\right]^\top$. Then we have

$$\delta^t = d_{\text{ang}}\left(\left[\sin\alpha^t, \cos\alpha^t\right]^\top, \tau^t / \left\|\tau^t\right\|\right), \tag{4.1}$$

where d_{ang} denotes the unsigned smaller angle between the unit vectors.

Stepping features. The trajectory curvature $\left(\kappa^t\right)$ is the norm of the second-order derivative of the 2D positions of the root joint on the ground plane, or equivalently, the derivative of the root joint tangents τ^t on the ground plane. We compute this using forward difference as well, *i.e.* $\kappa^t = \left\|\tau^t - \tau^{t-1}\right\|$.

The stepping phase (θ) represents the phases of the feet between the time steps where they touch the ground. We consider a half-period to be the time from the instant of one foot touching the ground, to the subsequent instant when the other foot touches the ground. Given the half-periods, we define the stepping phase θ^t at each time step t as follows. We assign a phase $\theta^t = 0$ when the left foot touches the ground and a phase $\theta^t = \pi$ when the right foot touches the ground, filling in the intermediate phases through linear interpolation.

The foot-step frequency $\left(\omega^t\right)$ is the angular velocity of the foot joints. Apart from generating realistic walk cycles (the motion between ipsilateral footsteps), this feature also supplements the root speed information to indicate the arousal in the emotions expressed by the gait. We compute the foot-step frequency at each time step t as the difference between the phase at that time step and the previous time step $t-1$, *i.e.* $\omega^t = \theta^t - \theta^{t-1}$.

4.4 Conditional Autoregression

Given a sequence of values with some information content, the overall goal of autoregression is to predict subsequent values in the sequence to maintain a similar information content (Lipton, Berkowitz, and Elkan 2015). In our work, we use an autoregression network to encode gaits with given emotions and predict subsequent gaits while maintaining the same emotions. Our

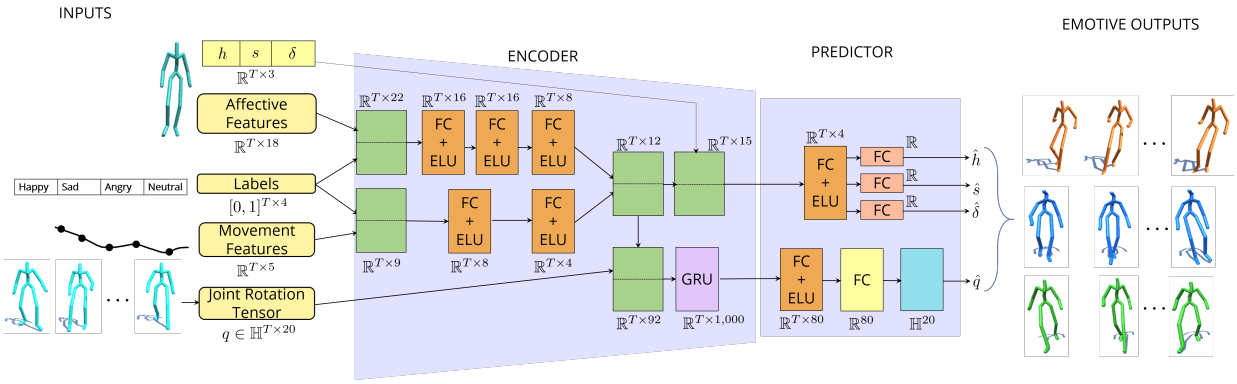


Figure 4.4: **Affective Gait Synthesis: Conditional Autoregression Network.** Our network takes in the joint rotations, input emotions as vectors consisting of probabilities for happy, sad, angry, and neutral, pose affective features, and movement features and jointly maps them to a latent representation space through the encoder. The predictor then takes in the latent representations and predicts gaits for subsequent time steps that follow the input trajectory while expressing the input emotions. The green boxes denote concatenation, and the cyan box at the end of the predictor denotes normalization of the variables to versors.

network consists of an encoder followed by a predictor. The encoder takes in the joint positions and rotations of the input gait for a number of time steps and extracts pose affective features and root joint features from the input gaits. Next, it learns the encoding functions to jointly map these extracted features, the corresponding input emotions, and the stepping features (such as curvature and foot-step frequency) to latent representations.

The predictor learns to compute the inverse mapping from the latent representations to the necessary affective and trajectory features and, by extension, the joint positions, and rotations, in the subsequent time steps.

We train the encoder and the predictor in tandem, by adding the predictor's output back to the encoder's input and advancing the temporal window of the encoder. We are able to achieve emotional expressiveness by training our network with input gaits for different emotions and forcing the network to learn to predict the corresponding pose affective features in the prediction time steps from its latent representations. We simultaneously enforce a robust

constraint on the network to adapt its trajectory features such that its predicted movements are close to the corresponding movements in the ground truth. This enables the network to take sharp turns and follow bends in the trajectory without smoothing out feet movements. Our overall approach is shown in Figure 4.4. We now elaborate on the operations of the encoder and the predictor.

4.4.1 Encoder

In the encoder, we separately combine our emotion vectors m with the pose affective features a^t, the stepping features consisting of $\left[\sin\theta^t, \cos\theta^t\right]$ and ω^t, and the root joint features $\bar{s}^t$ and κ^t, giving us input vectors $i_1^t = \left[a^t, m\right]^\top \in \mathbb{R}^{18+C}$ and $i_2^t = \left[\sin\theta^t, \cos\theta^t, \omega^t, \bar{s}^t, \kappa^t, m\right]^\top \in \mathbb{R}^{5+C}$.

We pass each of these inputs through a set of 3 fully connected layers, respectively, collectively denoted as the functions $\mathrm{FC}_{\mathrm{enc}_1}\left(\cdot, \phi_{\mathrm{FC}_{\mathrm{enc}_1}}\right) : \mathbb{R}^{T\times(18+C)} \rightarrow \mathbb{R}^{T\times H_1}$ and $\mathrm{FC}_{\mathrm{enc}_2}\left(\cdot, \phi_{\mathrm{FC}_{\mathrm{enc}_2}}\right) : \mathbb{R}^{T\times(5+C)} \rightarrow \mathbb{R}^{T\times H_2}$. Here, H_1 and H_2 denote the number of hidden units in the last fully connected layer in $\mathrm{FC}_{\mathrm{enc}_1}$ and $\mathrm{FC}_{\mathrm{enc}_2}$ respectively, and $\phi_{\mathrm{FC}_{\mathrm{enc}_1}}$ and $\phi_{\mathrm{FC}_{\mathrm{enc}_2}}$ denote the set of trainable parameters in the two sets of fully connected layers, respectively.

We combine the outputs of these fully connected networks to obtain intermediate representations γ^t, *i.e.* we have

$$\gamma^t = \left[\mathrm{FC}_{\mathrm{enc}_1}\left(i_1, \phi_{\mathrm{FC}_{\mathrm{enc}_1}}\right); \mathrm{FC}_{\mathrm{enc}_2}\left(i_2, \phi_{\mathrm{FC}_{\mathrm{enc}_2}}\right)\right] \tag{4.2}$$

where $i_1 = \left[i_1^0, \ldots, i_1^{T-1}\right]^\top$ and $i_2 = \left[i_2^0, \ldots, i_2^{T-1}\right]^\top$.

We then append γ^t separately to our input joint rotations q^t and to the remaining root joint trajectory features, h^t, s^t and δ^t. We pass the appended rotation data through a GRU to

obtain the latent representations $\widetilde{q}$, *i.e.*, we have

$$\widetilde{q} = \text{GRU}_{\text{versors}}\left([q;\gamma], \phi_{\text{GRU}_{\text{versors}}}\right) \tag{4.3}$$

where

- $q = \left[q^0, \ldots, q^{T-1}\right]^\top$,
- $\gamma = \left[i_1^0, \ldots, \gamma^{T-1}\right]^\top$,
- $\text{GRU}_{\text{versors}} : \mathbb{R}^{T\times(4(\mathcal{J}-1)+H_1+H_2)} \rightarrow \mathbb{R}^{T\times H_3}$,
- H_3 is the number of hidden units in the final layer of the GRU, and
- $\phi_{\text{GRU}_{\text{versors}}}$ denotes the trainable parameters in the GRU.

We also pass the appended root joint trajectory features through a fully connected layer $\text{FC}_{\text{root}} : \mathbb{R}^{T\times(3+H_1+H_2)\rightarrow T\times H_4}$, H_4 being the number of hidden units in the layer, to obtain the latent representations $\widetilde{h}$, $\widetilde{s}$, and $\widetilde{\delta}$. That is, we have

$$\left[\widetilde{h}, \widetilde{s}, \widetilde{\delta}\right]^\top = \text{FC}_{\text{root}}\left([h; s; \delta; \gamma], \phi_{\text{FC}_{\text{root}}}\right) \tag{4.4}$$

where

- $h = \left[h^0, \ldots, h^{T-1}\right]^\top$, $s = \left[s^0, \ldots, s^{T-1}\right]^\top$, $\delta = \left[\delta^0, \ldots, \delta^{T-1}\right]^\top$,
- $\phi_{\text{FC}_{\text{root}}}$ denotes the trainable parameters in the fully connected layer.

4.4.2 Predictor

Our predictor takes in the latent representations of the joint rotations and the root joint trajectory features from the encoder and learns to predict the same for T_{pred} subsequent time steps

such that the corresponding generated gaits follow the input trajectory while expressing the input emotion vectors. The predictor consists of a set of 2 fully connected layers, denoted as $\text{FC}_{\text{versors}} : \mathbb{R}^{T \times H_3} \rightarrow \mathbb{R}^{T_{\text{pred}} \times 4(J-1)}$ to predict the joint rotations, and three separate fully connected layers, $\text{FC}_h : \mathbb{R}^{T \times H_4} \rightarrow \mathbb{R}^{T_{\text{pred}}}$, $\text{FC}_s : \mathbb{R}^{T \times H_4} \rightarrow \mathbb{R}^{T_{\text{pred}}}$, and $\text{FC}_\delta : \mathbb{R}^{T \times H_4} \rightarrow \mathbb{R}^{T_{\text{pred}}}$ to compute the respective root joint features. Thus, we have,

$$\hat{q} = \text{FC}_{\text{versors}}\left(\widetilde{q}, \phi_{\text{FC}_{\text{versors}}}\right), \tag{4.5}$$

$$\hat{h} = \text{FC}_h\left(\widetilde{h}, \phi_{\text{FC}_h}\right), \tag{4.6}$$

$$\hat{s} = \text{FC}_s\left(\widetilde{s}, \phi_{\text{FC}_s}\right), \tag{4.7}$$

$$\hat{\delta} = \text{FC}_\delta\left(\widetilde{\delta}, \phi_{\text{FC}_\delta}\right) \tag{4.8}$$

where, as usual, $\phi_{\text{FC}_{\text{versors}}}$, ϕ_{FC_h}, ϕ_{FC_s}, and ϕ_{FC_δ} respectively denote the respective trainable parameters.

From the predicted joint rotations and root joint features at each time step t, we can also compute the predicted pose $\hat{X}^t$ and the corresponding pose affective features $\hat{a}^t$. We use these predicted variables, together with the input data, to train our network according to a curriculum schedule described in Section 4.5.3.

4.4.3 Loss Function for Training

We now describe the formulation of the loss function for training and validating our network. The loss function should accurately constrain the network both to learn the emotional expressions in the input gaits as well as to follow the gaits' trajectories in the subsequent time steps with plausible joint motions. We capture all these requirements in the loss function using four

loss terms: the motion loss, the pose loss, the pose affective features loss, and the root joint features loss.

4.4.3.1 *Motion loss* ($\mathcal{L}_{motion}$)

This loss ensures that the predicted joint motions remain plausible, *i.e.* close to the ground truth joint motions. To compute this loss, we measure the angle difference between the ground truth rotations q^t and the predicted rotations $\hat{q}^t$ on each joint at each prediction time step t. We also add the unit norm constraint on the predicted versors as regularization. Thus, we write the motion loss as

$$\mathcal{L}_{\text{motion}} := \sum_{j,t} \left\| \text{q2e}\left(q_j^t\right) - \text{q2e}\left(\hat{q}_j^t\right) \right\|^2 + \lambda_{\text{versor}} \left(\left\| \hat{q}_j^t \right\| - 1 \right)^2 \tag{4.9}$$

where $\text{q2e} : \mathbb{H} \to [0, 2\pi]^3$ maps the versors to corresponding Euler angles, and the summation is over all the joints across all the prediction time steps.

4.4.3.2 *Pose loss* $\left(\mathcal{L}_{pose}\right)$

The pose loss supplements the motion loss by adding an extra regularization to maintain plausible predicted joint motions. We require the predicted character poses $\hat{X}^t$ at each prediction time step, obtained using the predicted versors $\hat{q}_j^t$, to be as close as possible to the corresponding ground truth poses X^t. However, we do not require our predicted poses to follow the same trajectory as the ground truth poses since the desired trajectory will be provided to us at test time. We, therefore, subtract the root joint position from all the other joints at every time step

and write our pose reconstruction loss $\mathcal{L}_{\text{pose}}$ as

$$\mathcal{L}_{\text{pose}} := \sum_{t} \sum_{j=1}^{\mathcal{J}-1} \left\| \left(X_j^t - X_0^t \right) - \left(\hat{X}_j^t - \hat{X}_0^t \right) \right\|^2 . \tag{4.10}$$

4.4.3.3 *Pose affective features loss* $\left(\mathcal{L}_{aff}\right)$

This loss constrains the network to predict pose affective features similar to the ones computed from the input gaits. Therefore, it forces the network to maintain the emotional expressions in the gaits. We compute this loss by measuring the norm difference between the ground truth affective features a^t and the predicted features $\hat{a}^t$. We write it as

$$\mathcal{L}_{\text{aff}} = \sum_{t} \left\| a^t - \hat{a}^t \right\|^2 . \tag{4.11}$$

4.4.3.4 *Root joint features loss* $(\mathcal{L}_{root})$

This is a robust loss that we use to constrain the network to follow the ground truth gait trajectory at the prediction trajectory. The robustness ensures that the prediction follows sharp turns and bends in the trajectory without smoothing out the foot joint movements. We compute this loss by measuring the L_1 norm difference between the ground truth root joint features h, s, and δ, and the predicted features $\hat{h}$, $\hat{s}$, and $\hat{\delta}$ given by our network. We write it as

$$\mathcal{L}_{\text{root}} = \sum_{t} \left\| \left[h^t; s^t; \delta^t \right] - \left[\hat{h}^t; \hat{s}^t; \hat{\delta}^t \right] \right\|_1 . \tag{4.12}$$

4.4.3.5 *Foot contact loss* $\left(\mathcal{L}_{ft\_ct}\right)$

We also require the generated characters to walk naturally without any foot sliding. Therefore, we add a robust L_1 norm loss to constrain the heel and toe positions of the generated character to match the ground truth heel and toe positions. The robustness ensures that the prediction follows sharp turns and bends in the trajectory without smoothing out the foot joint movements. We write this loss as

$$\mathcal{L}_{\text{ft\_ct}} = \sum_t \left\| \left[lh^t; lt^t; rh^t; rt^t \right] - \left[\hat{lh}^t; \hat{lt}^t; \hat{rh}^t; \hat{rt}^t \right] \right\|_1 . \tag{4.13}$$

Finally, we linearly combine all these loss terms to formulate our overall loss function $\mathcal{L}$, which we write as

$$\mathcal{L} = \lambda_{\text{motion}}\mathcal{L}_{\text{motion}} + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{aff}}\mathcal{L}_{\text{aff}} + \lambda_{\text{root}}\mathcal{L}_{\text{root}} + \lambda_{\text{ft\_ct}}\mathcal{L}_{\text{ft\_ct}} \tag{4.14}$$

where λ_{motion}, λ_{pose}, λ_{aff}, λ_{root} and $\lambda_{\text{ft\_ct}}$ are the corresponding scaling terms and assign relative importance to the different loss terms.

4.5 Results

We show the performance of our autoregression network on the training dataset described below. We briefly describe the training dataset in Section 4.5.1 and discuss our augmented dataset in Section 4.5.2. We report our training routine in Section 4.5.3, elaborate on our performance benchmarks in Sections 4.5.4 and 4.5.5, and discuss the contributions of our novel components through ablation studies in Section 4.5.6. We summarize the details of integrating our setup

with the AR environment in Section 4.5.7. For a video demonstration of the results, please refer to our project website.

4.5.1 Dataset for Training

The datasets we use consist of temporal 3D pose sequences of human gaits for different types of walking, running, and other locomotion activities. This dataset was collected from various 3D pose sequence datasets, including BML (Ma, Paterson, and Pollick 2006), Human3.6M (Ionescu et al. 2014), ICT (Narang et al. 2017), CMU-MoCap (CMU-MOCAP 2018), ELMD (Habibie et al. 2017), and Emotion-Gait dataset (Bhattacharya et al. 2020). All gaits in the dataset are 240 frames long and playable at 10 fps. Due to memory constraints, we sampled every 4^{th} frame and used the resultant 60 frames as input data to our network, *i.e.* we had $T = 60$ for all our data points. In total, the dataset consists of $1,835$ gaits with corresponding emotion vectors available.

We used 80% of our gait dataset for training our network, 10% for validation, and kept the remaining 10% of the dataset for testing the emotional-expressiveness and trajectory-following performances. We performed this split randomly, and the network never sees the validation and the test sets during training.

4.5.2 Augmented Dataset: Synthesized Affective Gaits

At test time, our network is able to generate predicted gaits on trajectories it did not encounter during training. Our network is also able to transition between different emotions on the test gaits as a result of the learned inverse mapping from the latent representation space of the encoder. We, therefore, use our network to augment synthesized gaits to the Emotion-Gait benchmark datasets. We generate gaits on 20 trajectories not present in the dataset, with 100

emotions, also not present in the dataset, on each trajectory, for a total of $2,000$ new gaits. We also perform transitions between 50 pairs of emotions on each trajectory, picking a pair of emotions from the 100 novel ones without replacement, thus adding another $1,000$ new gaits, taking the total new gaits added to $5,000$.

4.5.3 Training Routine

We use the Adam optimizer (Kingma and Ba 2014) with a learning rate of 0.001, which we decay by a factor of 0.999 at every epoch. We use the ELU activation (Clevert, Unterthiner, and Hochreiter 2015) on all the fully connected layers in the network.

Similar to Pavllo, Grangier, and Auli 2018, we use a curriculum scheduling technique (Bengio et al. 2015) to train our network. We begin training by presenting various sequences of the input data and control features, each of length T, to our network, and the network predicts the rotations and translations for a single subsequent time step. This is equivalent to having a teacher forcing ratio of 1. At every subsequent epoch E, we decay the teacher forcing ratio by $\beta = 0.995$, *i.e.* with probability β^E, we supplement the data and controls at each input time step with the network's predicted data and controls at that time step. In other words, we progressively expose the network to more and more of its own predictions to make further predictions. Curriculum scheduling thus helps the network gently transition from a teacher-guided prediction routine to a self-guided prediction routine, which significantly speeds up the training process.

We train our network for 500 epochs, which takes around 18 hours on an Nvidia GeForce GTX 1080Ti GPU with 12 GB memory. We use 90% of the available data for training our network, and validate its performance on the remaining 10% of the data. We also observed that our

network performs well for any values of the scaling terms in Equation 4.14 between 0.5 and 2.0. We used a value of 1.0 for all of the performances reported in our experiments in Section 4.5.4.

4.5.4 Performance Benchmarks

We note here that methods generating motions with discrete styles test generalizability by providing their network with novel discrete-style labels not seen during training (Mason et al. 2018; Kfir et al. 2020), as opposed to ranges in the styles. Our network, on the other hand, learns the mapping between gaits and the underlying continuous space of emotions, rather than the mapping between the gaits and annotated emotions in the training samples. As a result, we test the generalizability of our network by generating gaits corresponding to continuous ranges of emotion vectors not seen during training. In order to benchmark our network on its ability to generalize to these continuous ranges of emotions, we perform experiments on emotion expressiveness and emotion transitions.

4.5.4.1 Emotion Expressiveness

We randomly pick gaits from the test set and extract the first 18 frames ($\frac{1}{3}^{\text{rd}}$ of the total data length) to provide as inputs to our network. We set the associated emotion vector of the input gait as the desired emotion, initially set a straight line as the desired trajectory, and predict for 200 time steps. Figure 4.6 (top row) shows some snapshots of the results of this evaluation. Next, we evaluate our method on trajectories with bends and sharp turns for 200 steps (Figure 4.6, middle row). We note that the generated gait maintains the emotion of the input, and follows all the trajectories. For example, the gait slows down while taking sharp turns and adjusts its stride and other joint movements such that the affective features remain similar when walking

Table 4.1: **Affective Gait Synthesis: Position and Rotation Errors.** We compute position error relative to the longest diagonal of the bounding box of the characters we test, and we compute rotation errors in degrees. The performance of our method is on par with the current state-of-the-art in motion generation.

Method	Pose Error	Rotation Error
PFNN (Holden, Komura, and Saito 2017)	0.19	0.06
QuaterNet (Pavllo, Grangier, and Auli 2018)	0.16	0.05
Affective Gaits	0.12	0.04

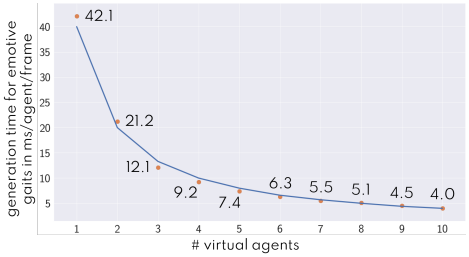


Figure 4.5: **Affective Gait Synthesis: Running Time.** We highlight the generation and rendering time for affective gaits on a pair of GPUs. The average time per agent per frame decreases, as we increase the number of virtual agents.

on a path with bends and sharp turns.

4.5.4.2 Emotion transitions

In this set of evaluations, we modify the desired emotion at each prediction time step to be different from those in the previous time steps, as well as the emotion vector associated with the input gait. To track the performance of our network, we first choose a particular emotion vector for the final prediction time step. Next, we linearly interpolate the value of each element of the emotion vector at each prediction time step separately, including everything from the input vector to the vector at the final time step. We normalize the vector at each time step to convert the values to a probability distribution that can be passed to our network. We test the results of emotion transition on trajectories with bends and turns for 200 time steps (Figure 4.6, bottom row). We observe that our network is able to smoothly transition between the different emotions, with no sharp limb movement or jarring action at any time step.

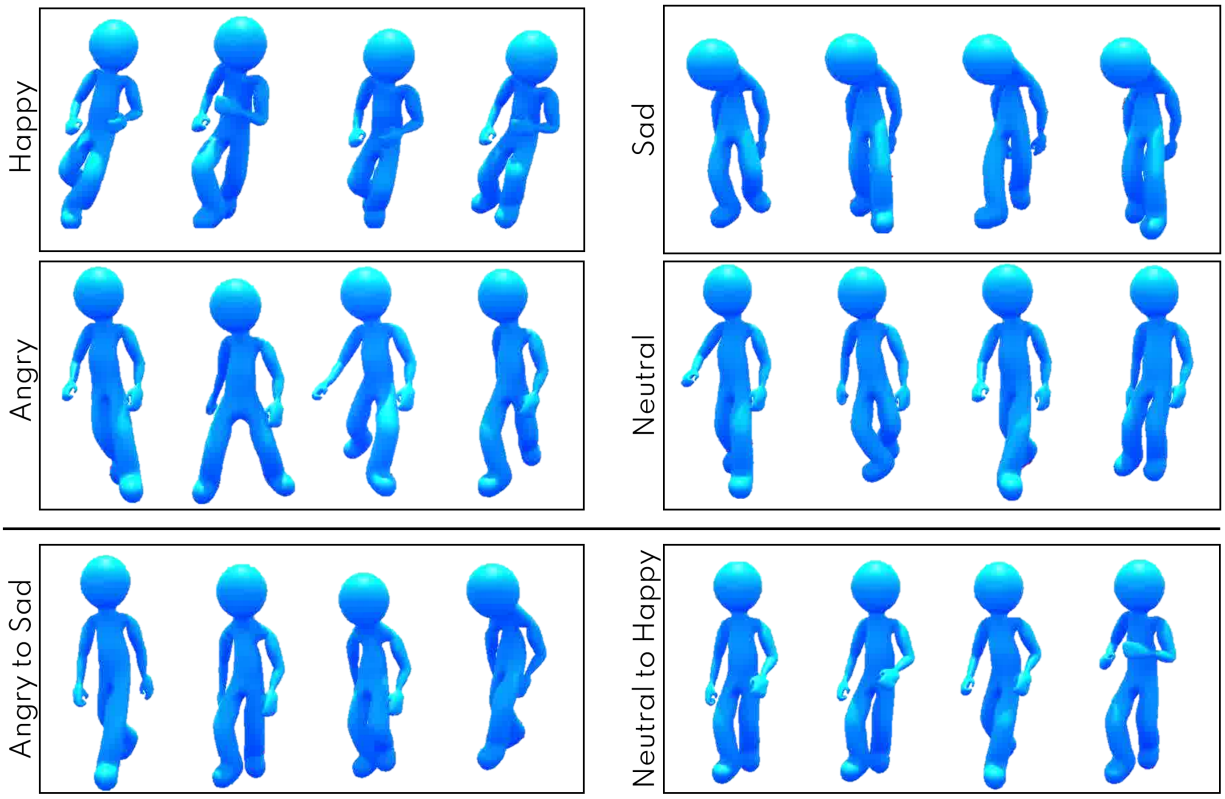


Figure 4.6: **Affective Gait Expressions and Transitions.** Each row shows four snapshots of synthesized gaits in temporal sequence from left to right. The top two rows show gaits with single emotions. The bottom row shows gaits transitioning from one emotion to another.

4.5.5 Comparisons with Motion Generation

We visually compare the performance of our autoregression network with QuaterNet developed by Pavllo, Grangier, and Auli 2018 in Figure 4.7 (rows (*a*) and (*b*)). QuaterNet is a state-of-the-art motion prediction network, and our network builds on the core prediction framework of QuaterNet. Since QuaterNet performs motion prediction but not emotional expressiveness, the gaits are not able to express the different emotions.

We also summarize the mean pose errors relative to the scale of the input data and the joint rotation errors in degrees as they are produced by our method, QuaterNet, as well as prediction networks based on alternative approaches such as the phase-functioned neural network

(PFNN) (Holden, Komura, and Saito 2017) in Table 4.1. We keep the desired emotion for our network the same as the input emotion vector to perform a fair comparison. We also require ground truth gaits to be available so the prediction time steps can actually compute these errors. Therefore, we present the first 18 frames of each data point in the test set as inputs to both the methods and compute their predicted motions on the trajectory of the ground truth data for the remaining 42 frames. We notice negligible differences between the performances of the two networks, showing that our approach is comparable to the state-of-the-art in motion prediction. However, for predicting motion beyond the 60 frames in the dataset, we noticed that the motions predicted by QuaterNet eventually reduce to no movement and the character comes to a stop, whereas both PFNN and our method can predict plausible motions for up to 200 prediction steps.

Thus, while current motion generation methods can produce highly realistic gaits for VAs, our method can additionally produce emotional expressiveness for those gaits. Therefore, our method helps improve the social presence of the VAs in an AR environment, as we observe through user evaluations in Section 4.6.

4.5.6 Ablation Studies

We have two main contributions to the design of our autoregression network. First, we provide the pose affective features as part of the input and constrain the predicted pose affective features to remain close to the corresponding ground truth during training through the affective loss $\mathcal{L}_{\text{aff}}$ (Equation 4.11). This enables our network to achieve emotional expressiveness and emotion transition on the input data. We, therefore, remove this input component and the corresponding loss function from our network and the training process and compare the results

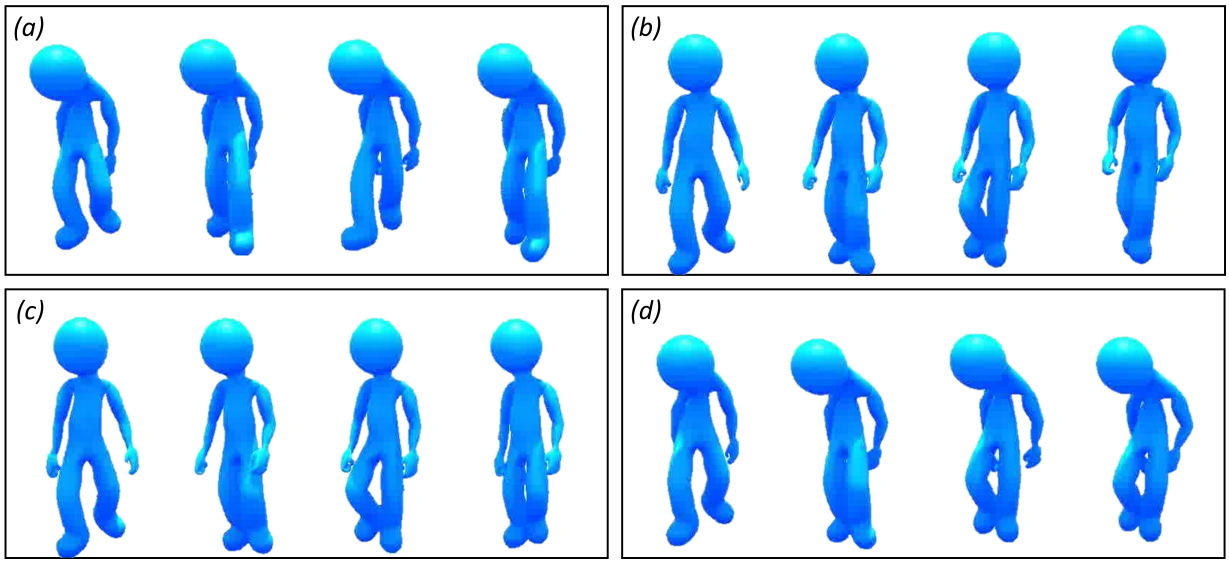


Figure 4.7: **Affective Gait Synthesis: Qualitative Comparisons and Ablation Studies.** Snapshots of gaits in temporal sequence from left to right, as generated by (*a*) our network using input emotion labels and following user-driven trajectories, (*b*) QuaterNet (Pavllo, Grangier, and Auli 2018), which has no emotive component, (*c*) our network without the affective feature component, resulting in gaits that follow the user-driven trajectories but have no emotional expressions (*e.g.*, no shoulder slouching to indicate sadness), and (*d*) our network without the movement feature component, resulting in affective gaits that do not follow the desired trajectory.

on the experiments described in Sections 4.5.4.1 and 4.5.4.2. We observe (Figure 4.7, rows (*c*) and (*d*)) that the ablated network does not maintain consistent pose affective features across the prediction time steps for the desired emotion.

Second, our network predicts the root joint features, consisting of the root height, the root speed, and the root orientation difference (as detailed in Section 4.3.2.4) alongside the predictions for the joint rotations. To ensure that the predicted motion follows the desired trajectory, we constrain these predicted root joint features to remain close to the corresponding ground truth during training through the root joint loss function $\mathcal{L}_{\text{root}}$ (Equation 4.12). To underscore the importance of this loss function, we remove it from our training and perform the experiments described in Section 4.5.4.2 on the ablated network. We notice (Figure 4.7, rows (*c*) and (*e*))

that the ablated network is not able to follow the desired trajectory. For linear trajectories, the predicted gaits often end up being oriented in arbitrary directions and not facing the direction of motion. For trajectories containing bends and turns, once the predicted gaits deviate from the desired trajectories, the ablated network is not able to reduce the deviations in subsequent prediction time steps.

4.5.7 Integration with the AR Environment

Our generative method can generate animation frames at the interactive rate of 40 ms per frame for 10 agents in an AR environment (Figure 4.5), *i.e.*, at 4 ms per agent per frame on average, when utilizing two Nvidia GeForce GTX 1080Ti GPUs. We built the realtime AR demo by rigging the generated skeletons to humanoid meshes, modifying the posed meshes to handle minor visual distortions caused by body shape mismatch, and streaming a virtual environment containing the animated characters to the Microsoft Hololens.

Rigging. Rigging the humanoid meshes to the generated skeletons requires that the rest pose of the generated skeleton is not modified to accommodate the desired mesh, as this would invalidate the rest of the animation. Thus, the rigging process must be done in reverse; the desired meshes must already contain a skeleton that allows us to repose it to meet the rest pose of the generated skeletons. We found free meshes with suitable skeletons from online 3D mesh databases. For the demo, we chose a humanoid stick figure and some humans from the Microsoft Rocketbox collection (Gonzalez-Franco et al. 2020). We performed the rigging in Blender 2.7.

Modifications To Rig. Due to the body shapes of our desired meshes not exactly matching that of the people used to generate the original dataset, we use Blender's sculpt tools to iron out any

distortions. In order to make the human meshes seem less synthetic, we used the original face bones in the meshes to create blendshapes such as blinking, breathing (mouth), and breathing (chest), which we activated at regular intervals. These blendshapes represent typical human behaviors independent of bodily animations. Our generated skeletons do not contain facial bones, thus, blendshapes are a good option for animating the face without requiring bones.

AR Implementation. We made the realtime AR demo in Unreal 4.24 due to its strong animation system allowing trivial sharing of animation files between meshes. We created An environment in which we show pairs of animations of specific emotions, human or stick figure, with the meshes approximately walking along the real ground. We used the Unreal Hololens plugin to stream the rendered images directly to the Hololens through the Hololens' Holographic Remoting player, which receives images by listening on a specific IP address. Due to the start position of the user being non-deterministic, we also provide key inputs to reposition the animated characters in front of wherever the user is when the key is pressed. The animations loop in order to make it easier for the user to determine differences in gaits and the stick figure characters are given colored materials matching their emotion.

Animation artifacts. We observe some jerkiness in the animation of the human characters in AR. The major sources of this jerkiness are (i) jerky motion of the user wearing the HoloLens, (ii) issues in the HoloLens software, e.g., frame rate clipping, and (iii) using textures instead of deformable cloth materials for the low-poly human models, which makes the jerkiness more apparent due to aliasing. We observe much-reduced jerkiness for the textureless stick figures in AR, and almost none when rendering the stick figures in a purely virtual environment with

Table 4.2: **Affective Gait Synthesis: Likert Scale Descriptions for the User Study.** The Likert scale response categories we provided users for the four broad observed pose affective features. Our goal is to evaluate if different users find the Likert scale values of these features similar for the same emotion, which would indicate these features are relevant for perceiving the emotions.

Feature	Likert Scale Response Categories				
	Value = 0	Value = 1	Value = 2	Value = 3	Value = 4
Torso	Contracted, bowed	Somewhat contracted	Neither contracted nor expanded	Somewhat expanded	Expanded, stretched
Arms	Contracted, close to the body	Somewhat contracted	Neither contracted nor expanded	Somewhat expanded	Expanded, away from the body
Gait Pace	Sustained, leisurely, slow	Somewhat sustained	Neither sustained nor hurried	Somewhat hurried	Hurried, sudden, fast
Gait Flow	Free, relaxed, uncontrolled	Somewhat free	Neither free nor bound	Somewhat bound	Bound, tense, controlled

a known ground plane.

4.6 User Evaluation

We conducted a web-based user study with our generated affective gaits to test the following **null hypothesis**:

The emotion vector used as input to generate each gait, and the emotion vector obtained by taking the arithmetic mean of the emotion vectors perceived by all the users from that generated gait, are two samples of the same statistical distribution.

In other words, the distribution of emotions we intend for a generated gait is statistically similar to the distribution of emotions perceived by the observing users. We also obtain the values of the pose affective features observed by the users from the generated gaits on a five-point Likert scale (LS) to validate our choice of pose affective features and emotional expressiveness

of the VAs in Section 4.3.2.3.

4.6.1 Procedure

The study was divided into three sections. Each section took three to four minutes to complete on average, and the entire study lasted for around ten minutes on average.

In the first section, we showed the users ten-second clips of eight randomly chosen generated gaits, one at a time, and asked them to report the emotion they perceived from each of those gaits. Users could report multiple emotions. For example, if one gait looked less happy to the user than another (but not necessarily sad), then the user could potentially mark that gait as both happy and neutral.

In the second section, we again showed the users ten-second clips of six randomly chosen generated gaits, one at a time. However, in this section, we performed emotion transitions on the generated gaits, so the final emotions were different from the initial ones. We asked the users to report the initial and the final emotions they perceived from these gaits, with the option to report multiple emotions.

In the third section, we showed the users ten-second clips of the same eight generated gaits from the first section, one at a time, and asked them to report the observed values or *intensities* of four broad pose affective features on a five-point LS. The four pose affective features we chose to ask are inspired by the critical features identified in the study by Roether et al. 2009. We summarize the scales for each of the four features in Table 4.2.

Table 4.3: **Affective Gait Synthesis: 2-Sample Anderson-Darling Test Statistics.** Based on the statistics, we are unable to reject the null hypothesis that the intended and perceived emotions of the gaits are samples from the same probability distribution, except for the case of Gait 2.

Gait #	Value of Statistic	p-value	Reject Null Hypothesis?
1	0.311	> 0.25	Not able to
2	2.111	0.04	With 96% confidence
3	0.311	> 0.25	Not able to
4	−0.615	> 0.25	Not able to
5	−0.615	> 0.25	Not able to
6	−0.611	> 0.25	Not able to
7	−0.069	> 0.25	Not able to
8	−1.081	> 0.25	Not able to

4.6.2 Participants

Since affect understanding is influenced by numerous social and cultural factors, we invited participants from diverse demographics to draw useful conclusions. We had 102 participants in total, of which 58 were male and 44 were female. 31 male and 26 female participants were in the age group of 18-24. 25 male and 14 female participants were in the age group of 25-34. 2 male and 4 female participants were above 35. Based on the overall test statistics, we did not find any noticeable difference in the emotions perceived from the generated gaits across the different sexes and age groups.

4.6.3 Analysis

We analyze the results on single emotions, emotion transition, and pose affective features. Finally, we report the perceived naturalness of the gaits by the users and miscellaneous analyses in "other feedback".

4.6.3.1 *Single Emotions*

Given the perceived emotions from the first section of the user study, we plot the normalized perceived emotions for eight randomly chosen gaits, as well as the corresponding normalized intended emotions of the generated gaits, in Figure 4.9. The emotions are denoted as four-component vectors as described in Section 4.3.2.2. We perform l_1 normalization so that each component of the emotion vector represents the intensity of the corresponding emotion.

For each gait, we perform the 2-sample Anderson-Darling test (Scholz and Stephens 1987) on the null hypothesis that the set of probability values of the perceived emotions and the set of probability values of the intended emotions are samples of the same underlying distribution. Table 4.3 summarizes the test statistic and the corresponding p-values for each of the eight gaits in Figure 4.9.

As we can observe from Table 4.3, we cannot reject our null hypothesis for seven of the eight gaits. This suggests strong statistical evidence that the intended and perceived emotions are statistically similar for those seven gaits. In Gait 2, where we reject the null hypothesis, the intended emotion was fully happy, but the observers mainly perceived it as either happy or neutral, indicating that the intensity of happiness did not come across to some of the observers.

4.6.3.2 *Emotion Transition*

We performed a similar 2-sample Anderson-Darling test (Scholz and Stephens 1987) for each of the initial and the final intended and perceived emotions, and were unable to reject the null hypothesis in 10 out of the 12 gaits we tested with. This again provides strong statistical evidence that the intended emotions for the gaits and the corresponding perceived emotions are

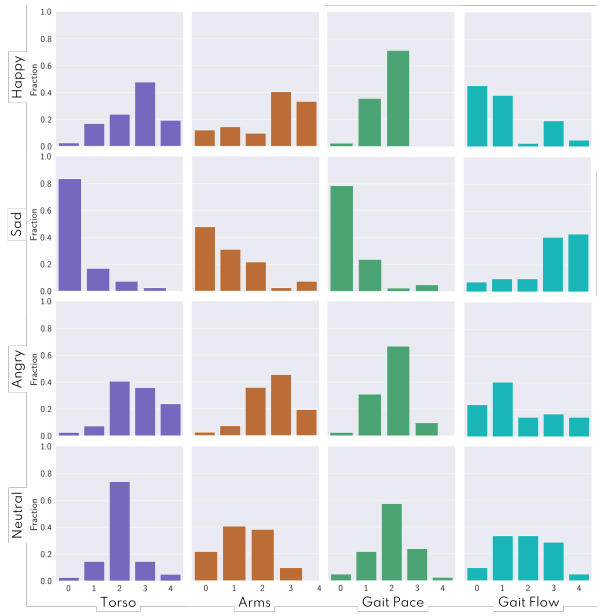


Figure 4.8: **Affective Gait Synthesis: Perceived Affective Features from the User Study.** We can observe different distinct modes for the different emotions, indicating that the pose affective features vary between different emotions and are consistent across users for a given emotion. The values in the horizontal axis correspond to the Likert scale values in Table 4.2.

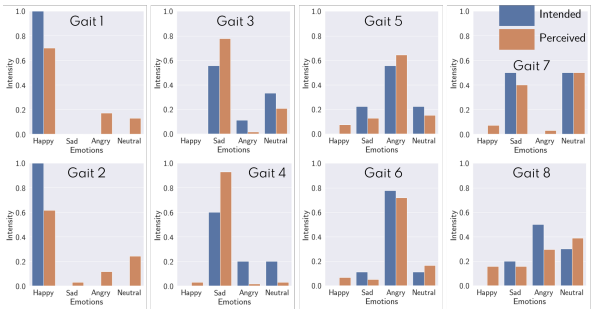


Figure 4.9: **Affective Gait Synthesis: Perceived Emotions from the User Study.** As we can observe from the plots and the statistics in Table 4.3, except for Gait 2, the intended and perceived emotions of the gaits cannot be determined to belong to separate statistical distributions.

statistically similar.

In one rejected case, the initial emotion was predominantly angry while the final was predominantly happy, but many observers indicated that both the initial and the final emotions were neutral. In the other case, the transition was from predominantly sad to predominantly happy, but many observers reported the gait to be going from sad to neutral. We hypothesize two possibilities for the mismatches:

- the intensities of the initial and final emotions did not come across in the generated gaits,
- the observers did not expect the gait to transition between extreme emotions such as angry to happy or sad to happy in the ten-second span of the clip, hence opted for choices

they found more reasonable.

4.6.3.3 Pose affective features

Our goal here is to validate the usefulness of the pose affective features we use to train our network, as well as the emotional expressiveness of the VAs through these pose affective features. However, evaluating the values of angles and ratios is out of scope for a user-study. We, therefore, opted to measure the user-observed LS values or *intensities* of the broad pose affective features that we used to formulate our geometric features. A good test is to verify if the observed intensities of these broad pose affective features are statistically consistent across different users. If this is verified, it justifies

- basing the geometric features on these broad features,
- the VAs are able to clearly express the different emotions through different LS values of the pose affective features.

We show the distribution of the fraction of users that marked each particular intensity of the four broad pose affective features for each of the single intended emotions in our study in Figure 4.8. The values in the horizontal axis correspond to the LS values in Table 4.2. From this figure, we can observe different distinct modes in the distribution for the different intended emotions. For example, the mode for the torso is at "contracted, bowed" (0) for sad, while it is concentrated more around "somewhat expanded" (3) and "expanded, stretched" (4) for happy. For angry, the users observed it to be less expanded than happy overall, but less than 10% found it to be contracted. For neutral, there is a clear mode at "neither contracted nor expanded" (2). These statistics show that the users perceived the VAs to have clear preferences of the different intensities of the pose affective features when expressing different emotions.

We also perform a k-sample Anderson-Darling test 4.3 for each gait and each of the four broad affective features (and k being the number of users) on the null hypothesis that all the user-provided values are from the same underlying distribution. We fail to reject the null hypothesis for all the four broad features in all the gaits, thus indicating strong statistical evidence that the observed intensities are consistent across different users.

4.6.3.4 Other Feedback

We asked the users to mark out of five how natural and smooth they felt the animations to be, with one indicating "not natural at all", three indicating "satisfactory", and five indicating "very natural". To establish a baseline, we asked the users to similarly marking the corresponding source motions as well. For our generated animations, 22% of the users marked five, 43% marked four, 24% marked three, 7% marked two, and 4% marked one. Thus 89% of the users marked at least three, *i.e.*, found the naturalness in the generated gaits to be satisfactory. By contrast, for the source motions, 58% of the users marked five, 35% marked four, and 7% marked three.

In the videos we sent out to the users, we used a moving camera so that the user was always looking straight at the virtual agent as it walked on different trajectories. 30% of the users reported being distracted by this moving camera during the study. Therefore, we plan to use a fixed camera in our subsequent studies.

4.7 Conclusion

To the best of our knowledge, we have presented the first method to synthesize and transition between gaits with affective expressions. Our emotion model is based on a linear combination of four widely-used categorical emotions and we present a network architecture that uses affective

features and movement features. Our algorithm can generate affective gaits that follow a given trajectory at interactive rates and develops a transition scheme to switch between gaits with different emotions. We have shown the results on gaits collected from open-source datasets and discussed our procedure for developing VAs with these gaits in an AR environment. We have also reported our observations from a web-based user study to conclude that our generated gaits looked natural, as well as had the desired emotional expressiveness. Lastly, we release an augmented dataset of affective gaits.

CHAPTER 5

Affective Co-Speech Gesture Synthesis Using Transformers

Project Website: https://gamma.umd.edu/t2g

Abstract

We present Text2Gestures, a transformer-based network that interactively generates emotive gestures for virtual agents corresponding to natural language text inputs. Our approach is designed to generate emotionally expressive gestures by utilizing the relevant biomechanical features for body expressions, also known as affective features. We also consider the intended task corresponding to the text and the target virtual agents' intended gender and handedness in our generation pipeline. We train and evaluate our network on the MPI Emotional Body Expressions Database and observe that our network produces state-of-the-art performance in generat-

ing gestures for virtual agents aligned with the text for narration or conversation. Our network can generate these gestures at interactive rates on a commodity GPU. We conduct a web-based user study and observe that around 91% of participants indicated our generated gestures to be at least plausible on a five-point Likert Scale. The emotions perceived by the participants from the gestures are also strongly positively correlated with the corresponding intended emotions, with a minimum Pearson coefficient of 0.77 in the valence dimension.

5.1 Introduction

As the world increasingly uses digital and virtual platforms for everyday communication and interactions, there is a heightened need to create highly realistic virtual agents endowed with social and emotional intelligence. Interactions between humans and virtual agents are being used to augment traditional human-human interactions in different applications, including online learning (Li et al. 2016; Liao et al. 2019; Simeone et al. 2019), virtual interviewing and counseling (Baur et al. 2013; DeVault et al. 2014), virtual social interactions (Roth et al. 2016; Heidicker, Langbehn, and Steinicke 2017; Latoschik et al. 2017), and large-scale virtual worlds (Oculus 2019). It is well-known that human-human interactions rely heavily on a combination of verbal communications (the text), inter-personal relationships between the people involved (the context), and more subtle non-verbal face and body expressions during communication (the subtext) (Matsumoto, Frank, and Hwang 2012; Knapp, Hall, and Horgan 2013). While context is often set at the beginning of interactions, it is important for virtual agents in social VR applications to be able to align their text with their subtext throughout the interaction, thereby improving the sense of presence of the human users in the virtual environment. Gesticulation is an integral component in subtext, where patterns of movement for hands, arms, heads, and

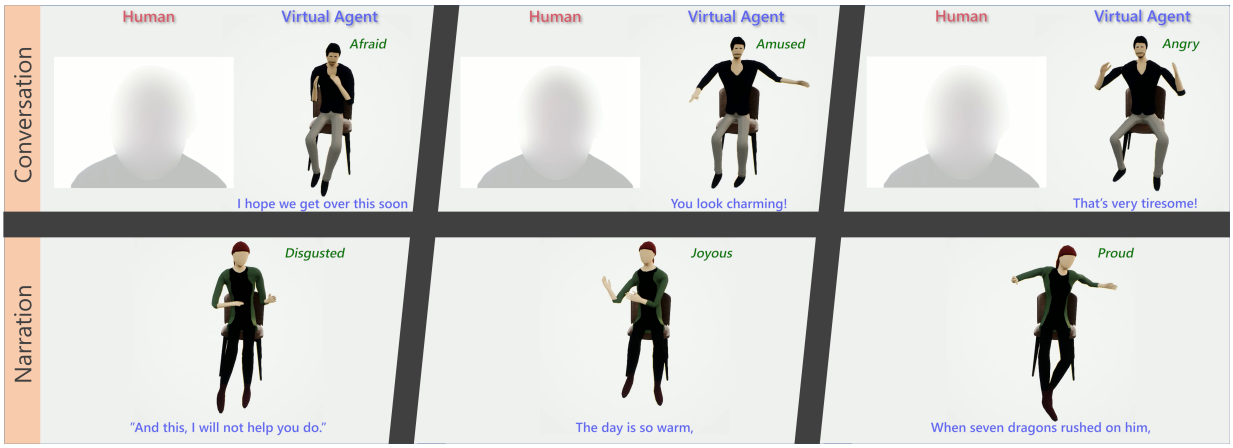


Figure 5.1: **Text2Gestures: Overview.** Our method generates emotive gestures for virtual agents that are aligned with sentences of natural language text (shown in blue). The intended emotions are labeled in green. The gestures follow the gender and handedness of the virtual agent. The top row shows gestures during a conversation in a virtual chat environment. The bottom row shows the gestures generated as the virtual agent narrates lines from a story. In each case, the columns show snapshots from a particular gesture sequence. We observe that the virtual agents express the intended emotions based on the text and the context.

torsos are used to convey a wide range of intent, behaviors, and emotions (McNeill 1992). In this work, we investigate the problem of aligning emotionally expressive gestures with the text to generate virtual agents' actions that result in natural interactions with human users.

Current game engines and animation engines can generate human-like movements for virtual agents, including head poses, hand gestures, and torso movements (Starke et al. 2019; Alexanderson et al. 2020). However, aligning these movements with a virtual agent's associated speech or text transcript is more challenging. Traditional approaches such as hand-crafting animations or collecting and transferring context-specific gestures through rotoscoping or motion capture look natural (Neff et al. 2008; Wagner, Malisz, and Kopp 2014), but need to be manually designed for every new gesture. On the other hand, virtual agents performing live social interactions with humans in VR need to adapt their gestures to their words and current social context in real-time. As a result, prior approaches based on pre-generated animations or motion

specifications are limited, and we need new interactive methods to generate plausible gestures.

The majority of the current approaches for speech-aligned gesture generation learn mappings between speech signals and gesture sequences to produce state-of-the-art gestures (Kucherenko et al. 2019; Alexanderson et al. 2020). In contrast to these speech-based methods, our goal is to align the gestures directly with the natural language text transcripts. This eliminates the need to have speeches pre-recorded by humans or machines, which have a higher production cost. Prior works on generating gestures aligned with text (Yoon et al. 2019) have leveraged the well-known sequence-to-sequence modeling network, which is efficient at performing a variety of sequence-to-sequence prediction tasks. However, these methods have only considered arms and head motions and are limited to producing small variations in simple emotions such as happy, angry, and sad, in their generated gestures.

However, it is evident that emotional expressiveness adds to the realism of virtual agents, as evidenced by works on adding emotions through facial and vocal expressions (Ferstl and McDonnell 2018a; Sohn et al. 2018), and are therefore important to generate. Studies in psychology show that body expressions also contain useful cues for perceived emotions (Argyle 2013), and often help disambiguate the emotions perceived from facial and vocal cues (Aviezer, Trope, and Todorov 2012). These body expressions are known to be composed of biomechanical features known as *affective features*. Commonly known affective features include, among others, the rate of arm swings, stride lengths, shoulder and spine postures, and head jerks (Karg et al. 2013). More recent approaches for generating virtual agents with gait-based body expressions have leveraged the relevant gait-based affective features to improve the perceived naturalness of the animations (Randhavane et al. 2019c; Bhattacharya et al. 2020).

In this paper, we aim to generate body gestures for virtual agents in social VR settings

while they follow sentences of text, to either narrate content to human participants or continue a conversation with human participants. We use affective features to make the gestures emotionally expressive, such that the human participants can perceive appropriate emotions from the virtual agents based on the natural language text.

Main Results: We present an end-to-end trainable generative network that produces emotive body gestures aligned with natural language text. Our approach is designed for interactive applications where a virtual agent narrates a line or takes part in a conversation. To this end, we make use of the transformer network (Vaswani et al. 2017), and extend current approaches to work with gestures for virtual agents in 3D. We also adapt the gestures based on narration or conversation and the intended gender and handedness (dominance of left-hand or right-hand in gesticulation) of the virtual agents. We also make the gestures emotionally expressive by utilizing the relevant gesture-based affective features of the virtual agents.

To summarize, the novel components of our work include:

- A transformer-based network that interactively takes in text one sentence at a time, and generates 3D pose sequences for virtual agents corresponding to gestures aligned with that text.
- Conditioning the generation process to follow the intended acting task of narration or conversation and the virtual agents' intended gender and handedness.
- Considering the intended emotion in the text to generate emotionally expressive gestures.
- A web study with 600 total responses to evaluate the quality of our generated gestures compared to motion-captured sequences and the emotional expressiveness of our generated gestures.

Based on our experiments, we find that our network has state-of-the-art performance for generating gestures aligned with text compared to ground truth sequences in a large-scale motion capture database. We can generate these gestures at an interactive rate of 312.5 fps using an Nvidia GeForce GTX 1080Ti GPU. Based on our user study, we also find that the emotions perceived by the participants from the gestures are strongly positively correlated with the corresponding intended emotions of the gestures, with a minimum Pearson coefficient of 0.77 in the valence dimension. Moreover, around 91% of participants found our generated gestures are plausible on a five-point Likert Scale.

5.2 Related Work

This section summarizes studies that explored how different emotions are expressed through and perceived from body gestures and how they have been utilized in previous work on generating emotive virtual agents.

We also review prior work on generating human body gestures in graphics and VR, particularly those that align the gestures with speech and text content. We focus mostly on data-driven approaches here because our work is based on similar ideas, and refer the interested reader to the extensive survey of Wagner, Malisz, and Kopp 2014 for the more classical rule-based approaches. The main limitation of such rule-based approaches is that their range of gestures is confined to the designed set of gestures. Hence, they require that gestures for every novel speech and text inputs are manually designed.

5.2.1 Expressing and Perceiving Emotions via Gestures

Studies in psychology show that body expressions, including gestures, are better suited than facial and vocal cues to express and perceive emotions varying in arousal and dominance, such as anger, relief, fear and pride (De Gelder 2006; Gross, Crane, and Fredrickson 2012). Body expressions are also useful for disambiguating between pairs of emotions such as fear or anger (Meeren, Heijnsbergen, and Gelder 2005a), and fear or happiness (Stock, Righart, and De Gelder 2007). Follow-up studies in affective computing (Kleinsmith and Bianchi-Berthouze 2013; Karg et al. 2013; Castillo and Neff 2019) have identified sets of biomechanical features, known as affective features, on which human observers focus when perceiving these different emotions from gestures. For example, rapid arm swings can indicate anger, an expanded upper body can indicate pride, and slouching shoulders can indicate fear or sadness.

In our work, we use a set of these affective features observable from gestures to improve our generated virtual agents' emotional expressiveness.

5.2.2 Generating Emotive Virtual Agents

Current approaches to endow virtual agents with emotional expressiveness make use of a number of modalities, including verbal communication (Chowanda et al. 2016; Sohn et al. 2018), face movements (Karras et al. 2017; Ferstl and McDonnell 2018a), body gestures (Jaques et al. 2016), and gaits (Randhavane et al. 2019c). In the context of generating emotional expressions aligned with speech, Chuah, Rossen, and Lok 2009 leveraged a dataset of words mapped to emotive facial expressions to generate virtual agents with basic emotions automatically. DeVault et al. 2014 developed a full-fledged virtual human counselor, using a pre-built corpus of mappings be-

tween mental states and body expressions to make their virtual agent appropriately expressive. In contrast to these approaches, we build a generalizable data-driven mapping to body gestures from a more diverse range of intended emotions associated with text transcripts, such that we can generate appropriately expressive gestures for novel, out-of-dataset text sentences.

5.2.3 Generating Gestures Aligned with Speech and Text

There has been extensive deep-learning-based work on generating human body gestures that align with speech content in the recent past (Chiu, Morency, and Marsella 2015). Levine et al. 2010 used a hidden Markov model to learn latent mappings between speech and gestures. Hasegawa et al. 2018 used recurrent neural networks to predict 3D pose sequences for gestures from input speech. More recently, Kucherenko et al. 2019 trained autoencoders to learn latent representations for the speech and the gesture data and then learned mappings between the two to generate gestures that are less sensitive to noise in the training data. In a different approach, Alexanderson et al. 2020 learned invertible sub transformations between speech and gesture spaces to stochastically generate a set of best-fitting gestures corresponding to the speech. Other approaches have also incorporated individual styles into gestures (Ginosar et al. 2019), added multiple adversarial losses to make the generated gestures look more realistic (Ferstl, Neff, and McDonnell 2019), and even added rule-based prototypical behaviors such as head nods and hand waves based on the discourse (Sadoughi and Busso 2019). This has culminated into works such as generating gestures for multiple speakers through style-transfer (Ahuja et al. 2020), and semantic-aware gesture generation from speech (Kucherenko et al. 2020).

Our approach is complementary to these approaches in that we learn mappings from the text transcripts of speech to gestures. This eliminates the noise in speech signals and helps us

focus only on the relevant content and context. Learning from the text also enables us to focus on a wider range of gestures, including iconic, deictic, and metaphoric gestures (McNeill 1992). Our work is most closely related to that of Yoon et al. 2019. They learn upper body gestures as PCA-based, low-dimensional pose features, corresponding to text transcripts from a dataset of TED-talk videos, then heuristically map these 3D gestures to an NAO robot. They have also followed up this work with generating upper-body gestures aligned with the three modalities of speech, text transcripts, and person identity (Yoon et al. 2020). On the other hand, we learn to map text transcripts to 3D pose sequences corresponding to semantic-aware, full-body gestures of more human-like virtual agents using an end-to-end trainable transformer network, while also blending in appropriate emotional expressiveness.

5.2.4 Generating Stylistic Human Body Motions

Generating speech- or text-aligned gestures with emotional expressiveness can be considered a sub-problem in generating stylistic human body motions, including facial motions, head motions, and locomotion. Existing approaches on face motions include generating lip movements and other face-muscle motions aligned with speech, using either recurrent neural networks (Suwajanakorn, Seitz, and Kemelmacher-Shlizerman 2017) or convolutional networks (Ferstl and McDonnell 2018a). Methods for generating head motions that convey the pace and intensity of speech have explored neural network architectures based on autoencoders (Greenwood, Laycock, and Matthews 2017) and generative adversarial networks (Sadoughi and Busso 2018). Methods to generate stylistic locomotion are based on convolutional networks (Holden, Saito, and Komura 2016), parametric phase functions (Holden, Komura, and Saito 2017), and deeply learned phase functions (Starke et al. 2019) for different styles of walking. Recent approaches

have also incorporated gait-based affective features to generate emotionally expressive walking (Randhavane et al. 2019a; Bhattacharya et al. 2020). Moreover, there has been considerable progress in generating person images and videos of body motions based on textual descriptions of moments and actions (Li et al. n.d.; Zhou et al. 2019).

In contrast, we aim to generate emotionally expressive gestures at interactive rates that correspond to text sentences. The space of gesture motions we explore is also different from the space of motions corresponding to locomotion, head motions, or facial muscle motions. Although there is some overlap with the space of head motions (Greenwood, Laycock, and Matthews 2017; Sadoughi and Busso 2018), the corresponding methods have not been extended to deal with full-body motions.

5.3 Transforming Text to Gestures

Given a natural language text sentence associated with an acting task of narration or conversation, an intended emotion, and attributes of the virtual agent, including gender and handedness, our goal is to generate the virtual agent's corresponding body gestures. In other words, we aim to generate a sequence of relative 3D joint rotations $\mathcal{Q}^*$ underlying the poses of a virtual agent, corresponding to a sequence of input words $\mathcal{W}$, and subject to the acting task A and the intended emotion E based on the text, and the gender G and the handedness H of the virtual agent. We therefore have

$$\mathcal{Q}^* = \arg\max_{\mathcal{Q}} \text{Prob}\left[\mathcal{Q}|\mathcal{W}; A, E, G, H\right]. \tag{5.1}$$

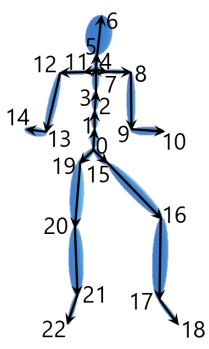


Figure 5.2: **3D Pose Model for Gestures.** Our pose graph is a directed tree consisting of 23 joints, with the root joint as the root node of the tree, and the end-effector joints (head, wrists, toes) as the leaf nodes of the tree. We manipulate the appropriate joints to generate emotive gestures.

5.3.1 Representing Text

Following standard practices in NLP tasks, we represent the word at each position s in the input sentence $\mathcal{W} = \begin{bmatrix} w_1 & \dots & w_s & \dots & w_{T_{\text{sen}}} \end{bmatrix}$, with T_{sen} being the maximum sentence length, using word embeddings $w_s \in \mathbb{R}^{300}$. We obtain the word embeddings using the GloVe model pre-trained on the Common Crawl corpus (Pennington, Socher, and Manning 2014). We opt for GloVe based on our preliminary experiments, where it marginally outperformed other similar-dimensional embedding models such as Word2Vec (Mikolov et al. 2013) and FastText (Bojanowski et al. 2017), and had similar performance as much higher dimensional embedding models, *e.g.*, BERT (Devlin et al. 2019). We demarcate the start and the end of sentences using special start of sequence (SoS) and end of sequence (EoS) vectors that are pre-defined by GloVe.

5.3.2 Representing Gestures

Following prior works on human motion generation (Pavllo, Grangier, and Auli 2018), we represent a gesture as a sequence of poses or configurations of the 3D body joints. These include body expressions as well as postures. We represent each pose with quaternions denoting 3D rotations of each joint relative to its parent in the directed pose graph (Figure 5.2). Specifically, at each time step t in the sequence $\mathcal{Q} = \begin{bmatrix} q_1 & \dots & q_t & \dots & q_{T_{\text{ges}}} \end{bmatrix}$, with T_{ges} being the

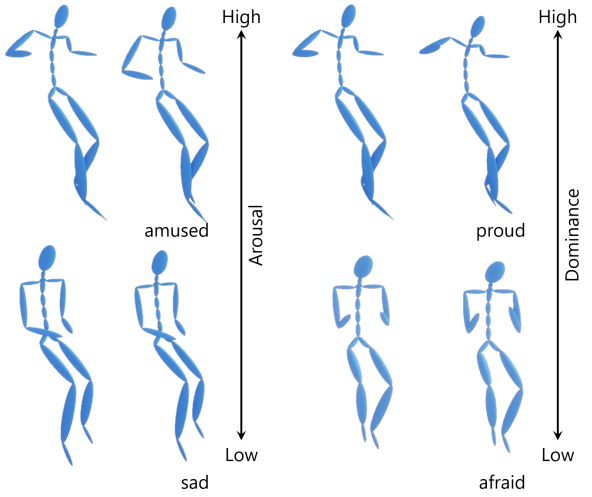


Figure 5.3: **Variance in Affective Gestures.** Emotions with high arousal (*e.g.*, amused) generally have rapid limb movements, while emotions with low arousal (*e.g.*, sad) generally have slow and subtle limb movements. Emotions with high dominance (*e.g.*, proud) generally have an expanded upper body and spread arms, while emotions with low dominance (*e.g.*, afraid) have a contracted upper body and arms close to the body. Our algorithm uses these characteristics to generate the appropriate gestures.

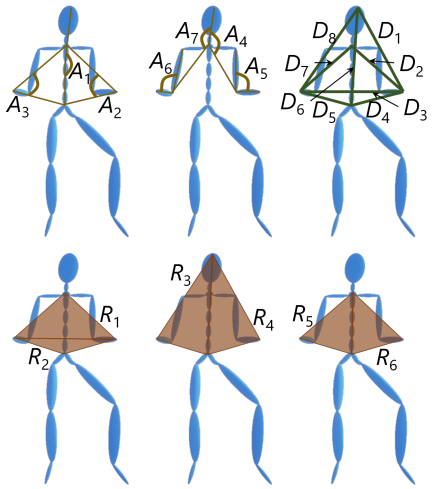


Figure 5.4: **Affective Features for Gestures.** We use a total of 15 features: 7 angles, A_1 through A_7, 5 distance ratios, $\frac{D_1}{D_4}$, $\frac{D_2}{D_4}$, $\frac{D_8}{D_5}$, $\frac{D_7}{D_5}$, and $\frac{D_3}{D_6}$, and 3 area ratios, $\frac{R_1}{R_2}$, $\frac{R_3}{R_4}$, and $\frac{R_5}{R_6}$.

maximum gesture length, we represent the pose using flattened vectors of unit quaternions $q_t = \begin{bmatrix} \dots & q_{j,t}^\top & \dots \end{bmatrix}^\top \in \mathbb{H}^\mathcal{J}$. Each set of 4 entries in the flattened vector q_t, represented as $q_{j,t}$, is the rotation on joint j relative to its parent in the directed pose graph, and $\mathcal{J}$ is the total number of joints. We choose quaternions over other representations to represent rotations as quaternions are free of the gimbal lock problem (Pavllo, Grangier, and Auli 2018). To demarcate the start and the end of each gesture sequence, we define our own start of sequence (SoS) and end of sequence (EoS) poses. Both of these are idle sitting poses with minor differences in the positions of the end-effector joints, the root, wrists and the toes.

5.3.3 Representing the Agent Attributes

We categorize the agent attributes into two types: those that depend on the text to which the gesture is to be aligned, and those that depend on the virtual agent to be animated.

5.3.3.1 Attributes Depending on Text

In this work, we consider two attributes that depend on text, the acting task, and the intended emotion.

Acting Task. We consider two acting tasks, narration and conversation. In narration, the agent narrates lines from a story to a listener. The gestures, in this case, are generally more exaggerated and theatrical. In conversation, the agent uses body gestures to supplement the words spoken in conversation with another agent or human. The gestures are subtler and more reserved. In our formulation, we represent the acting task as a two-dimensional one-hot vector $A \in \{0, 1\}^2$, to denote either narration or conversation.

Intended Emotion. We consider each text sentence to be associated with an intended emotion, given as a categorical emotion term such as joy, anger, sadness, pride, etc. While the same text sentence can be associated with multiple emotions in practice, in this work, we limit ourselves to sentences associated with only one emotion, owing primarily to the limitations in the dataset available for training. We use the NRC-VAD lexicon (Mohammad 2018) to transform these categorical emotions associated with the text to the VAD space. The VAD space (Mehrabian and Russell 1974) is a well-known representation in affective computing to model emotions. It maps an emotion as a point in a three-dimensional space spanned by valence (V), arousal (A), and

dominance (D). Valence is a measure of the pleasantness in the emotion (*e.g.*, happy vs. sad), arousal is a measure of how active or excited the subject expressing the emotion is (*e.g.*, angry vs. calm), and dominance is a measure of how much the subject expressing the emotion feels "in control" of their actions (*e.g.*, proud vs. remorseful). Thus, in our formulation, the intended emotion $E \in [0,1]^3$, where the values are coordinates in the normalized VAD space.

5.3.3.2 Attributes Depending on the Agent

We consider two attributes that depend on the agent to be animated, its gender G, and handedness H. In our work, gender $G \in \{0,1\}^2$ is limited to a one-hot representation denoting either female or male, and handedness $H \in \{0,1\}^2$ is a one-hot representation indicating whether the agent is left-hand dominant or right-hand dominant. Each agent has exactly one assigned gender and one assigned handedness.

5.3.4 Using the Transformer Network

Modeling the input text and output gestures as sequences as in Secs. 5.3.1 and 5.3.2, the optimization in Equation 5.1 then becomes a sequence transduction problem. Transformer networks (Vaswani et al. 2017) are efficient at sequence transduction tasks when both the input and the output sequences are word embeddings, generally from two different natural languages (*e.g.*, English and German). We briefly revisit the transformer network as originally introduced by Vaswani et al. 2017, and describe how we modify it to work with joint rotations as the target sequence.

The transformer network follows the traditional encoder-decoder architecture for sequence-to-sequence modeling. However, instead of using sequential chains of recurrent memory net-

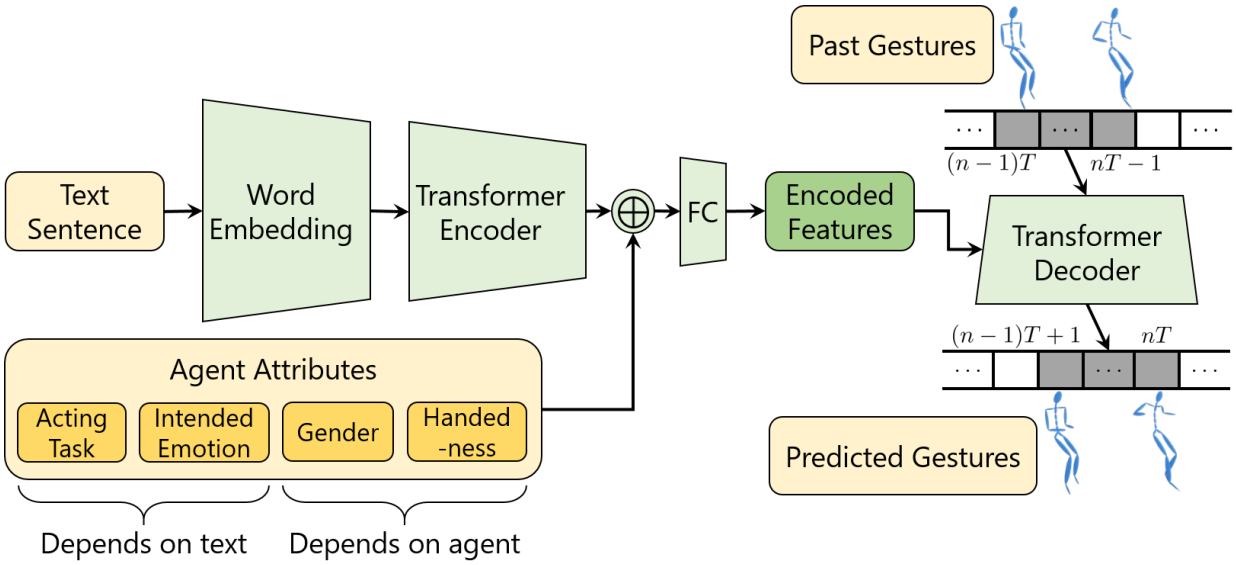


Figure 5.5: **Text2Gestures: Transformer Network Architecture.** Our network takes in sentences of natural language text and transforms them to word embeddings using the pre-trained GloVe model (Pennington, Socher, and Manning 2014). It then uses a transformer encoder to transform the word embeddings to latent representations, appends the agent attributes to these latent representations, and transforms the combined representations to encoded features. The transformer decoder takes in these encoded features and the past gesture history to predict gestures for the subsequent time steps. The gesture at each time step is represented by the set of rotations on all the body joints relative to their respective parents in the pose graph at that time step.

works, or the computationally expensive convolutional networks, the transformer uses a multi-head self-attention mechanism to model the dependencies between the elements at different temporal positions in the input and target sequences.

The attention mechanism is represented as a sum of values from a dictionary of key-value pairs, where the weight or attention on each value is determined by the relevance of the corresponding key to a given query. Thus, given a set of m queries $Q \in \mathbb{R}^{m \times k}$, a set of n keys $K \in \mathbb{R}^{n \times k}$, and the corresponding set of n values $V \in \mathbb{R}^{n \times v}$ (for some values k and v), and

using the scaled dot-product as a measure of relevance, we can write,

$$\mathrm{Att}\left(Q, K, V\right) = \mathrm{softmax}\left(\frac{QK^{\top}}{k}\right)V, \tag{5.2}$$

where the softmax is used to normalize the weights. In the case of self-attention (SA) in the transformer, Q, K, and V all come from the same sequence. In the transformer encoder, the self-attention operates on the input sequence $\mathcal{W}$. Since the attention mechanism does not respect the relative positions of the elements in the sequence, the transformer network uses a positional encoding scheme to signify the position of each element in the sequence, prior to using the attention. Also, in order to differentiate between the queries, keys, and values, it projects $\mathcal{W}$ into a common space using three independent fully-connected layers consisting of trainable parameters $W_{Q,enc}$, $W_{K,enc}$, and $W_{V,enc}$. Thus, we can write the self-attention in the encoder, SA_{enc}, as

$$\mathrm{SA}_{enc}\left(\mathcal{W}\right) = \mathrm{softmax}\left(\frac{\mathcal{W}W_{Q}W_{K}^{\top}\mathcal{W}^{\top}}{k}\right)\mathcal{W}W_{V}. \tag{5.3}$$

The multi-head (MH) mechanism enables the network to jointly attend to different projections for different parts in the sequence, *i.e.*,

$$\mathrm{MH}\left(\mathcal{W}\right) = \mathrm{concat}\left(\mathrm{SA}_{enc,1}\left(\mathcal{W}\right), \ldots, \mathrm{SA}_{enc,h}\left(\mathcal{W}\right)\right)W_{\mathrm{concat}}, \tag{5.4}$$

where h is the number of heads, W_{concat} is the set of trainable parameters associated with the concatenated representation, and each self-attention i in the concatenation consists of its own set of trainable parameters $W_{Q,i}$, $W_{K,i}$, and $W_{V,i}$.

To complete the encoder operations, the transformer network passes the MH output

through two fully-connected (FC) layers. It repeats the entire block consisting of (SA–MH–FC) N times, and uses the residuals around each layer in the blocks during backpropagation. We denote the final encoded representation of the input sequence $\mathcal{W}$ as $F_{\mathcal{W}}$.

To meet the given constraints on the acting task A, intended emotion E, gender G, and handedness H of the virtual agent, we append these variables to $F_{\mathcal{W}}$ and pass the combined representation through two fully-connected layers with trainable parameters W_{FC} to obtain feature representations

$$\bar{F_{\mathcal{W}}} = FC\left(\begin{bmatrix} F_{\mathcal{W}}^{\top} & A^{\top} & E^{\top} & G^{\top} & H^{\top} \end{bmatrix}^{\top}; W_{FC}\right). \tag{5.5}$$

The transformer decoder operates similarly using the target sequence $\mathcal{Q}$, but with some important differences. First, it uses a masked multi-head (MMH) self-attention on the sequence, such that the attention for each element covers only those elements appearing before it in the sequence, *i.e.*,

$$\mathrm{MMH}\left(\mathcal{Q}\right) = \mathrm{concat}\left(\mathrm{SA}_{dec,1}\left(\mathcal{Q}\right), \ldots, \mathrm{SA}_{dec,h}\left(\mathcal{Q}\right)\right) W_{\mathrm{concat}}. \tag{5.6}$$

This ensures that the attention mechanism is causal and therefore usable at test time, when the full target sequence is not known apriori. Second, it uses the output of the MMH operation as the key and the value, and the encoded representation $\bar{F_{\mathcal{W}}}$ as the query, in an additional multi-head self-attention layer without any masking, *i.e.*,

$$\mathrm{MH}\left(\bar{F_{\mathcal{W}}}, \mathcal{Q}\right) = \mathrm{concat}\left(\underbrace{\mathrm{Att}_{dec,1}\left(\bar{F_{\mathcal{W}}}, \mathrm{MMH}\left(\mathcal{Q}\right), \mathrm{MMH}\left(\mathcal{Q}\right)\right), \ldots}_{h \text{ entries}}\right) W_{\mathrm{concat}}. \tag{5.7}$$

It then passes the output of this multi-head self-attention through two fully-connected layers to complete the block. Thus, one block of the decoder is (SA–MMH–SA–MH–FC), and the transformer network uses N such blocks. It also uses positional encoding of the target sequence upfront and uses the residuals around each layer in the blocks during backpropagation.

5.4 Training the Network

Figure 5.5 shows the overall architecture of our transformer-based network. The word embedding layer transforms the words into feature vectors using the pre-trained GloVe model. The encoder and the decoder consist of $N = 2$ blocks of (SA–MH–FC) and (SA–MMH–SA–MH–FC), respectively. We use $h = 2$ heads in the multi-head attention. The set of FC layers in each of the blocks maps to 200-dim outputs. At the output of the decoder, we normalize the predicted values so that they represent valid rotations. We train our network using the sum of three losses: the angle loss, the pose loss, and the affective loss. We compute these losses between the gesture sequences generated by our network and the original motion-captured sequences available as ground truth in the training dataset.

5.4.1 Angle Loss for Smooth Motions

We denote the ground truth relative rotation of each joint j at time step t as the unit quaternion $q_{j,t}$, and the corresponding rotation predicted by the network as $\hat{q}_{j,t}$. If needed, we correct $\hat{q}_{j,t}$ to have the same orientation as $q_{j,t}$. Then we measure the angle loss between each such pair of rotations as the squared difference of their Euler angle representations, modulo π. We use Euler angles rather than the quaternions in the loss function as it is straightforward to compute closeness between Euler angles using Euclidean distances. To ensure that the motions look

smooth and natural, we also consider the squared difference between the derivatives of the ground truth and the predicted rotations, computed at successive time steps. We write the net angle loss $\mathcal{L}_{\text{ang}}$ as

$$\mathcal{L}_{\text{ang}} = \sum_{t} \sum_{j} \left(\text{Eul}\left(q_{j,t}\right) - \text{Eul}\left(\hat{q}_{j,t}\right)\right)^2 + \left(\text{Eul}\left(q_{j,t}\right) - \text{Eul}\left(q_{j,t-1}\right) - \text{Eul}\left(\hat{q}_{j,t}\right) + \text{Eul}\left(\hat{q}_{j,t-1}\right)\right)^2. \tag{5.8}$$

5.4.2 Pose Loss for Joint Trajectories

The angle loss only penalizes the absolute differences between the ground truth and the predicted joint rotations and does not explicitly constrain the resulting poses to follow the same trajectory as the ground truth at all time steps. To this end, we compute the squared norm difference between the ground truth and the predicted joint positions at all time steps. Given the relative joint rotations and the offset o_j of every joint j from its parent, we can easily compute all the joint positions using forward kinematics (FK). Thus, we write the pose loss $\mathcal{L}_{\text{pose}}$ as

$$\mathcal{L}_{\text{pose}} = \sum_{t} \sum_{j} \|\text{FK}\left(q_{j,t}, o_j\right) - \text{FK}\left(\hat{q}_{j,t}, o_j\right)\|^2. \tag{5.9}$$

5.4.3 Affective Loss for Emotive Gestures

To ensure that the generated gestures are emotionally expressive, we also penalize the loss between the gesture-based affective features of the ground truth and the predicted poses. Prior studies in affective computing (Gross, Crane, and Fredrickson 2012; Karg et al. 2013; Castillo and Neff 2019) show that gesture-based affective features are good indicators of emotions that

vary in arousal and dominance. Emotions with high dominance, such as pride, anger, and joy, tend to be expressed with an expanded upper body, spread arms, and upright head positions. Conversely, emotions with low dominance, such as fear and sadness, tend to be expressed with a contracted upper body, arms close to the body, and collapsed head positions. Again, emotions with high arousal, such as anger and amusement, tend to be expressed with rapid arm swings and head movements. In contrast, emotions with low arousal, such as relief and sadness, tend to be expressed with subtle, slow movements. Different valence levels are not generally associated with consistent differences in gestures and are often inferred from other cues and the context. In Figure 5.3, we show some gesture snapshots to visualize the variance of these affective features for different levels of arousal and dominance.

We define scale-independent affective features using angles, distance ratios, and area ratios for training our network, following the same rationale as in (Bhattacharya et al. 2020). Since, in our experiments, the virtual agent is sitting down, and only the upper body is expressive during the gesture sequences, only the joints at the root, neck, head, shoulders, elbows, and wrists move significantly. Therefore, we use these joints to compute our affective features. We show the complete list of affective features we use in Figure 5.4. Denoting the set of affective features computed from the ground truth and the predicted poses at time t as a_t and $\hat{a}_t$ respectively, we write the affective loss $\mathcal{L}_{\text{aff}}$ as

$$\mathcal{L}_{\text{aff}} = \sum_{t} \| a_t - \hat{a}^t \|^2. \tag{5.10}$$

Combining all the individual loss terms, we write our training loss functions $\mathcal{L}$ as

$$\mathcal{L} = \mathcal{L}_{\text{ang}} + \mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{aff}} + \lambda \| W \|, \tag{5.11}$$

where W denotes the set of all trainable parameters in the full network, and λ is the regularization factor.

5.5 Results

This section elaborates on the database we use to train, validate, and test our method. We also report our training routine, the performance of our method compared to the ground truth and the current state-of-the-art method for generating gestures aligned with text input. We also perform ablation studies to show the benefits of each of the components in our loss function: the angle loss, the pose loss, and the affective loss.

5.5.1 Data for Training, Validation, and Testing

We evaluate our method on the MPI emotional body expressions database (Volkova et al. 2014). This database consists of 1,447 motion-captured sequences of human participants performing one of three acting tasks: narrating a sentence from a story, gesticulating a scenario given as a sentence, or gesticulating while speaking a line in a conversation. Each sequence corresponds to one text sentence and the associated gestures. For each sequence, the following annotations of the intended emotion E, gender G, and handedness H, are available:

- E as the VAD representation for one of "afraid", "amused", "angry", "ashamed", "disgusted", "joyous", "neutral", "proud", "relieved", "sad", or "surprised",
- G is either female or male, and
- H is either left or right.

Each sequence is captured at 120 fps and is between 4 and 20 seconds long. We pad all the sequences with our EoS pose (Section 5.3.2) so that all the sequences are of equal length. Since

the sequences freeze at the end of the corresponding sentences, padding with the EoS pose often introduces small jumps in the joint positions and the corresponding relative rotations when any gesture sequence ends. However, our training loss function (Equation 5.11) is designed to ensure smoothness, and is able to generate gestures that transition smoothly to the EoS pose after the end of the sentence.

5.5.2 Training and Evaluation Routines

We train our network using the Adam optimizer (Kingma and Ba 2014) with a learning rate of 0.001 and a weight decay of 0.999 at every epoch. We train our network for 600 epochs, using a stochastic batch size of 16 without replacement in every iteration. We use 80% of the data for training, validate the performance on 10% of the data and test on the remaining 10% of the data that is held out. The total training takes around 8 hours using an Nvidia GeForce GTX 1080Ti GPU. At the time of evaluation, we initialize the transformer decoder with $T = 20$ (Figure 5.5) time steps of the SoS pose, and keep using the past $T = 20$ time steps to generate the gesture at every time step.

5.5.3 Comparative Performance

We compare the performance of our network with the transformer-based text-to-gesture generation network of Yoon et al. 2019 because this method is the closest to our work. To make a fair comparison, we perform the following as per their original paper:

- use the eight upper body joints (three each on the two arms, neck, and head) for their method,
- use PCA to reduce the eight upper body joints to 10 dimensional features,

Table 5.1: **Text2Gestures: Mean Pose Errors.** For each listed method, this is the mean Euclidean distance of all the joints over all the time steps from all the ground truth sequences over the entire test set. The mean error for each sequence is computed relative to the mean length of the longest diagonal of the 3D bounding box of the virtual agent in that sequence.

Method	Mean pose error
Yoon et al. 2019	1.57
Our method, no angle loss	0.07
Our method, no pose loss	0.06
Our method, no affective loss	0.06
Our method, all losses	0.05

- retrain their network on the MPI emotional body expressions database (Volkova et al. 2014), using the same data split as in our method, and the hyperparameters provided by the authors,
- compare the performances only on the eight upper body joints.

We report the mean pose error from the ground truth sequences over the entire held-out test set for both Yoon et al. 2019 and our method in Table 5.1. For each test sequence and each method, we compute the total pose error for all the joints at each time step and calculate the mean of these errors across all time steps. We then divide this mean error by the mean length of the longest diagonal of the 3D bounding box of the virtual agent to get the normalized mean error. To obtain the mean pose error for the entire test set, we compute the mean of the normalized mean errors for all the test sequences. We also plot the trajectories of the three end-effector joints in the upper body, head, left wrist, and right wrist, independently in the three coordinate directions, for two diverse sample sequences from the test set in Figure 5.6. We ensure diversity in the samples by choosing a different combination of the gender, handedness, acting task and intended emotion of the gesture for each sample.

We observe from Table 5.1 that our method reduces the mean pose error by around 97%

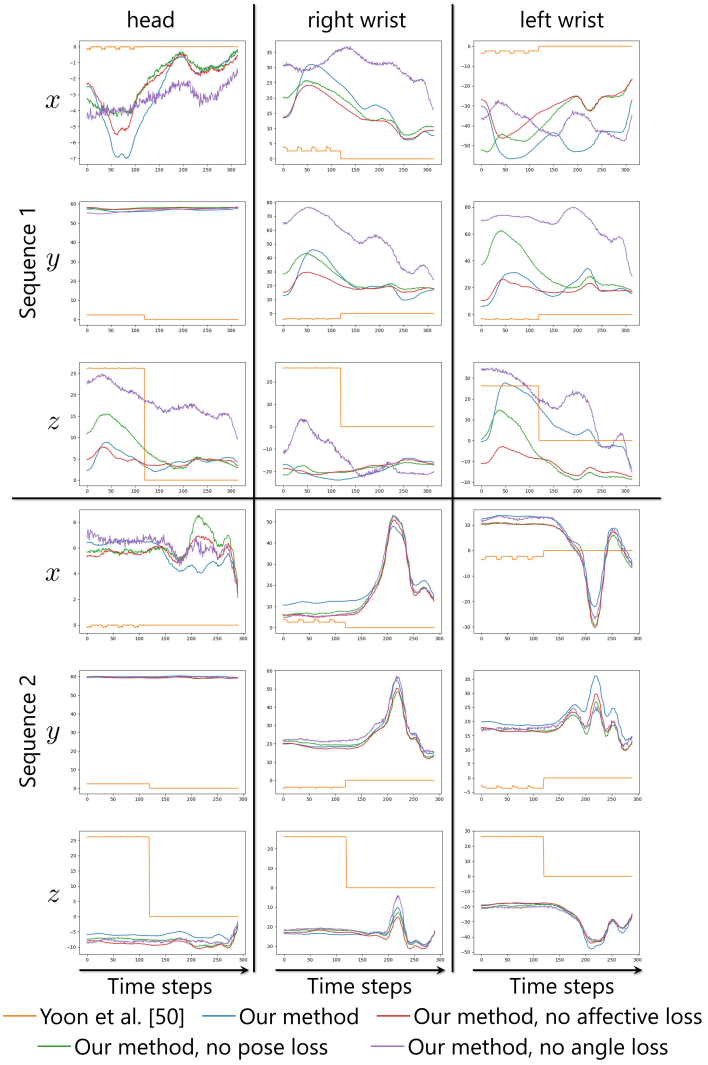


Figure 5.6: **Text2Gestures: End-Effector Trajectory Comparison.** The trajectories in each of the three coordinate directions for the head and two wrists for two sample sequences from the test set, as generated by all the methods. Removing the angle loss makes the trajectory heavily jerky. Removing the pose loss makes our method unable to follow the desired trajectory. Removing the affective loss reduces the variations corresponding to emotional expressiveness. Yoon et al. 2019 are unable to generate large amplitude variations in the trajectories because it works with a dimension-reduced representation of the sequences.

over Yoon et al. 2019. From the plots in Figure 5.6, we can observe that unlike our method, Yoon et al. 2019 are unable to generate the high amplitude oscillations in motion, leading to larger pose errors. This is because the lower-dimensional representation of pose motions considered by Yoon et al. 2019 does not sufficiently capture this detail. Moreover, the gestures generated by Yoon et al. 2019 did not produce any movements in the z-axis, rather, the movements were confined to a z-plane. The step in their method in the z-axis occurs when the gesture returns to the EoS rest pose, which is in a different z-plane.

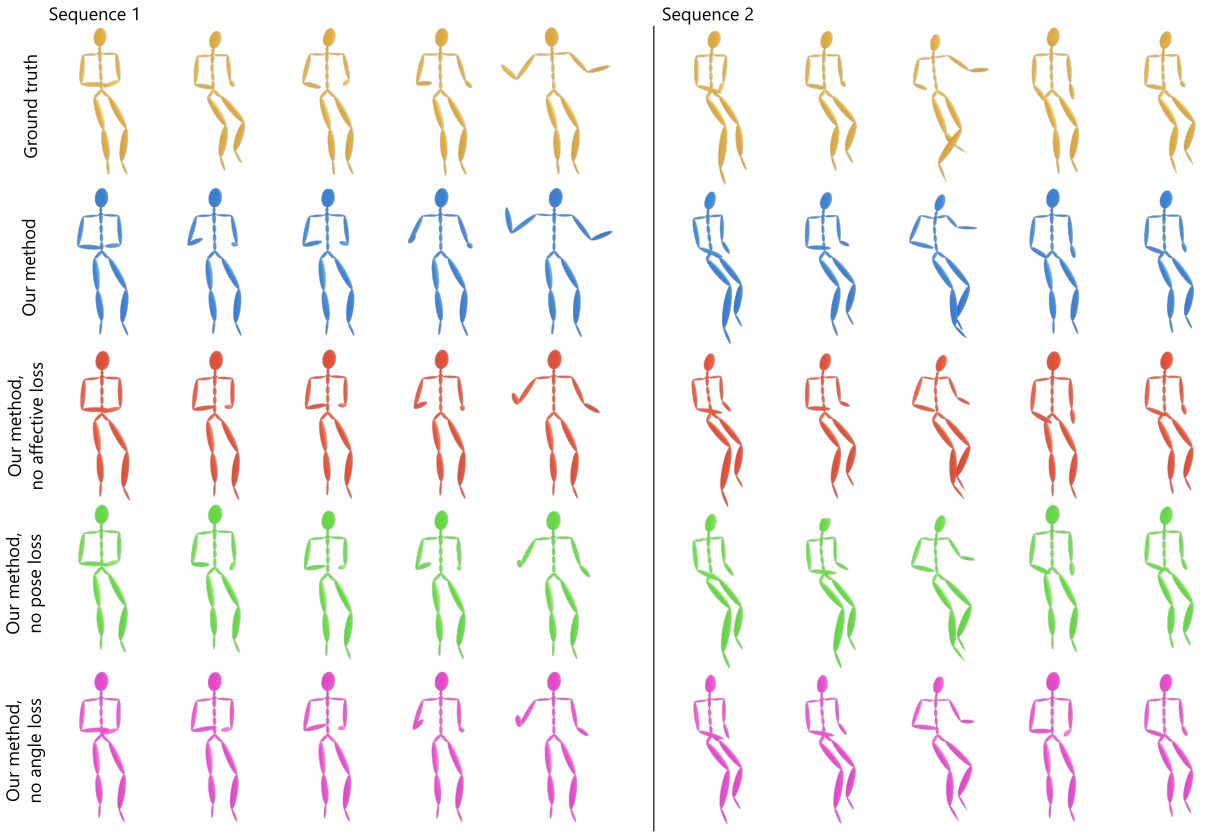


Figure 5.7: **Text2Gestures: Qualitative Ablation Studies.** Snapshots of gestures at five time steps from two sample ground truth sequences in the test set, and the gestures at the same five time steps as generated by our method and its different ablated versions. The full sequences of these gestures are available in our project website.

5.5.4 Ablation Studies

We compare the performance between different ablated versions of our method. We test the contribution of each of the three loss terms, angle loss, pose loss, and affective loss, in Equation 5.11 by removing them from the total loss one at a time, and training our network from scratch with the remaining losses. Each of these ablated versions has a higher mean pose error over the entire test set than our actual method, as we report in Table 5.1. To visualize the performance differences, we show in Figure 5.6 sample end-effector trajectories in the same setup as described in Section 5.5.3. We also show snapshots from the two sample gesture sequences

generated by all the ablated versions in Figure 5.7. We show the full gesture sequences of these and other samples in our project website.

We can observe from Figure 5.6 that the gestures become heavily jerky without the angle loss. When we add in the angle loss but remove the pose loss, the gestures become smoother, but still have some jerkiness. This shows that the pose loss also lends some robustness to the generation process. The other major drawback in removing either the angle or the pose loss is that the network can only change the gesture between time steps within some small bounds, making the overall animation sequence appear rigid and constricted.

The contribution of the affective loss is apparent when we remove it from Equation 5.11 while keeping both the angle and the pose losses. In this case, the network can generate a wide range of gestures, leading to animations that appear fluid and plausible. However, the emotional expressions in the gestures, such as spreading and contracting the arms and shaking the head, are not consistent with the intended emotions.

5.5.5 Interfacing with the VR Environment

Given a sentence of text, we can generate the gesture animation files at an interactive rate of 3.2 ms per frame, or 312.5 frames per second, on average on an Nvidia GeForce GTX 1080Ti GPU.

We use the gender and the handedness to determine the physical attributes of the virtual agent during generation of gestures. The gender impacts the pose structure. The handedness determines the hand for one-handed or longitudinally asymmetrical gestures. To create the virtual agents, we use low-poly humanoid meshes with no textures on the face. We use the pre-defined set of male and female skeletons in the MPI emotional body motion database (Volkova et al. 2014) for the gesture animations. We assign a different model to each of these skeletons, match-

ing their genders. We also manually correct any visual distortions caused by shape mismatch between the pre-defined skeletons and the low-poly meshes.

We use Blender 2.7 to rig the generated animations to the humanoid meshes. To ensure a correct rig, we modify the rest pose of the humanoid meshes to match the rest pose of our pre-defined skeletons. To make the meshes appear more life-like, we add periodic blinking and breathing movements to the generated animations using blendshapes in Blender.

We prepare our VR environment using Unreal 4.25. We place the virtual agents on a chair in the center of the scene in full focus. The users can interact with the agent in two ways. They can either select a story that the agent narrates line by line using appropriate body gestures or send lines of text as part of a conversation to which the agent responds using text and associated body gestures. We show the full demos in our project website. We use synthetic, neutral-toned audio aligned with all our generated gestures to better understand the timing of the gestures with the text. However, we do not add any facial features or emotions in the audio for the agents since these are dominant modalities of emotion expression and make a fair evaluation of emotional expressiveness of the gestures difficult. For example, if the intended emotion is happy and the agent has a smiling face, observers are more likely to respond favorably to any gesture with reasonably high valence or arousal.

5.6 User Study

We conduct a web-based user study to test two major aspects of our method: the correlation between the intended and the perceived emotions of and from the gestures, and the quality of the animations compared to the original motion-captured sequences.

Table 5.2: **Text2Gestures: Likert Scale Descriptions for the User Study.** We use the following markers in our five-point Likert scale

Very Unnatural	*e.g.*, broken arms or legs, torso at an impossible angle
Not Realistic	*e.g.*, limbs going inside the body or through the chair
Looks OK	No serious problems, but does not look very appealing
Looks good	No problems and the gestures look natural
Looks great!	The gestures look like they could be from a real person

5.6.1 Procedure

The study consisted of two sections and was about ten minutes long. In the first section, we showed the participant six clips of virtual agents sitting on a chair and performing a randomly selected gesture sequences generated by our method, one after the other. We then asked the participant to report the perceived emotion as one of multiple choics. Based on our pilot study, we understood that asking participants to choose from one of 11 categorical emotions in the EBEDB dataset (Volkova et al. 2014) was overwhelming, especially since some of the emotion terms were close to each other in the VAD space (*e.g.*, joyous and amused). We, therefore, opted for fewer choices to make it easier for the participants as well as reducing the probability of having too many emotion terms with similar VAD values in the choices. For each sequence, we, therefore, provided the participant with four choices for the perceived emotion. One of the choices was the intended emotion, and the remaining three were randomly selected. For each animation, randomly choosing three choices can unintentionally bias the participant's response (for instance, if the intended emotion is "sad" and the random options are "joyous", "amused" and "proud"). However, the probability of such a set of choices drops exponentially as we consider multiple sequences for each participant and multiple participants in the overall study.

In the second section, we showed the participant three clips of virtual agents sitting on a

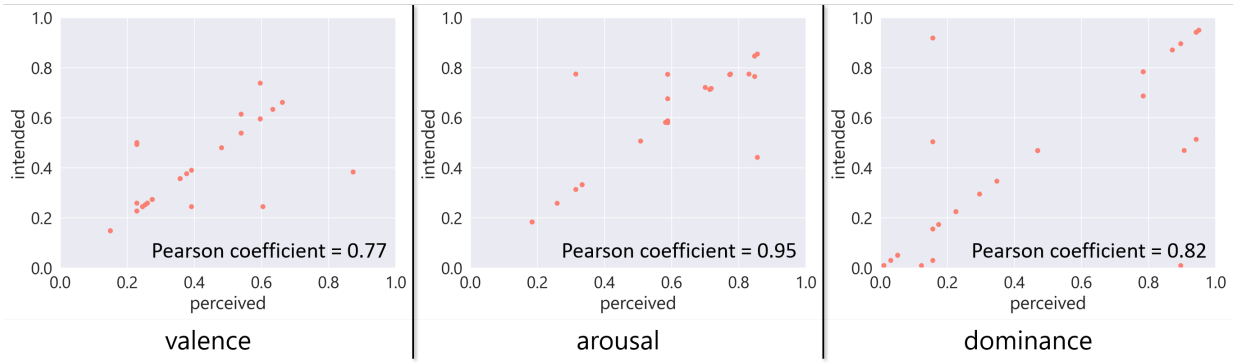


Figure 5.8: **Text2Gestures: Valence, Arousal, and Dominance Distributions in the User Study.** Distribution of values from the intended and perceived emotions in the valence, arousal, and dominance dimensions for gestures in the study. All the distributions indicate strong positive correlation between the intended and the perceived values, with the highest correlation in arousal and the lowest in valence.

chair and performing a randomly selected original motion-captured sequence and three clips of virtual agents performing a randomly selected generated gesture sequence, one after the other. We showed the participant these six sequences in random order and did not tell the participant which sequences were from the original motion-capture and which sequences were generated by our method. We asked the participant to report the naturalness of the gestures in each of these sequences on a five-point Likert scale, consisting of the markers mentioned in Table 5.2.

We had a total of 145 clips of generated gestures and 145 clips of the corresponding motion-captured gestures. For every participant, we chose all the 12 random clips across the two sections without replacement. We did not notify the participant apriori which clips had motion-captured gestures, and which clips had our generated gestures. Moreover, we ensured that in the second section, none of the three selected generated gestures corresponded to the three selected motion-captured gestures. Thus, all the clips each participant looked at were distinct. However, we did repeat clips at random across participants, to get multiple responses for each clip.

5.6.2 Participants

Fifty participants participated in our study, recruited via web advertisements. To study the demographic diversity, we asked the participants to report their gender and age group. Based on the statistics, we had 16 male and 11 female participants in the age group of 18-24, 15 male and seven female participants in the age group of 25-34, and one participant older than 35 who preferred not to disclose their gender. However, we did not observe any particular pattern of responses based on the demographics.

5.6.3 Evaluation

We analyze the correlation between the intended and the perceived emotions from the first section of the user study and the reported quality of the animations from the second section. We also summarize miscellaneous user feedback.

5.6.3.1 Correlation between Intended and Perceived Emotions

Each participant responded to six random sequences in the first section of the study, leading to a total of 300 responses. We convert the categorical emotion terms from these responses to the VAD space using the mapping of NRC-VAD (Mohammad 2018). We show the distribution of the valence, arousal, and dominance values of the intended and perceived emotions in Figure 5.8.

We compute the Pearson correlation coefficient between the intended and perceived values in each of the valence, arousal, and dominance dimensions. A Pearson coefficient of 1 indicates maximum positive linear correlation, 0 indicates no correlation, and -1 indicates maximum negative linear correlation. In practice, any coefficient larger than 0.5 indicates a strong

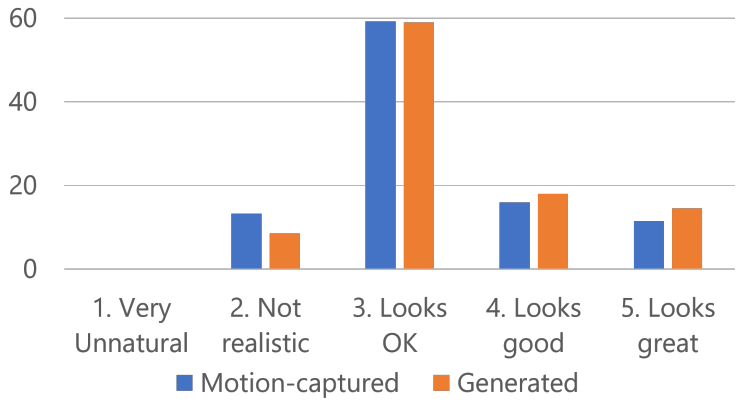


Figure 5.9: **Text2Gestures: Synthesis Quality Scores from the User Study.** A small fraction of participants responded to the few gesture sequences that had some stray self-collisions, and therefore found these sequences to not be realistic. The vast majority of the participants found both the motion-captured and generated gestures to look OK (plausible) on the virtual agents. A marginally higher percentage of participants reported that our generated gesture sequences looked better on the virtual agents that the original motion-captured gesture sequences.

positive linear correlation. We hypothesize that intended and the perceived values in all the three dimensions have such a strong positive correlation.

We observe a Pearson coefficient of 0.77, 0.95, and 0.82 between the intended and the perceived values in the valence, arousal, and dominance dimensions respectively. Thus, the values in all three dimensions are strongly positively correlated, satisfying our hypothesis. The values also indicate that the correlation is stronger in the arousal and the dominance dimensions and comparatively weaker in the valence dimension. This is in line with prior studies in affective computing (Gross, Crane, and Fredrickson 2012; Karg et al. 2013), which show that arousal and dominance are most consistently perceived from gesture-based body expressions.

5.6.3.2 Quality of Gesture Animations

Each participant responded to three random motion-captured and three random generated sequences in the second section of the study. Therefore, we have a total of 150 responses on both

the motion-captured and the generated sequences. Of these, we summarize the percentage of responses of each of the five points in the Likert scale in Figure 5.9. We consider a minimum score of 3 on our Likert scale as an indication that the participant found the corresponding gesture plausible. By this criterion, we observe that 86.67% of the responses indicated the virtual agents performing the motion-captured sequences to have plausible gestures, and 91.33% of the responses the virtual agents performing the generated sequences to have plausible gestures. In fact, we observe that a marginally higher percentage of responses scored the generated gestures 4 and 5 (2.00% and 3.33% respectively), compared to the percentage of responses that had the same score for the motion-captured gestures. This, coupled with the fact that participants did not know apriori which sequences were motion-captured and generated, indicates that our generated sequences were perceived to be as realistic as the original motion-captured sequences. One possible explanation of our generated gestures being rated marginally more plausible than the motion-captured gestures is that our generated poses return smoothly to a rest pose after the end of the sentence, whereas the motion-captured gestures freeze at the end-of-the-sentence pose.

5.6.3.3 Miscellaneous Feedback

Our virtual agents only express emotions through gestures and do not use any other modalities such as faces or voices. Therefore, we expected some participants taking the study to be distracted by the lack of emotions on the face or to be unable to determine the emotions based only on the gestures, without supporting cues from the other modalities. Indeed, 14% of the participants reported they were distracted by the lack of facial emotions, 10% were unable to conclusively determine the emotions based on only the gestures, and 8% experienced both dif-

ficulties.

5.7 Conclusion

We have expanded affect synthesis to work with two modalities, taking the text transcript of a virtual agent's speech as input and their intended affects, and synthesizing their corresponding affective body gestures. Our method takes in the natural language text one sentence at a time and generates 3D pose sequences for virtual agents corresponding to emotive gestures aligned with that text. To the best of our knowledge, this was the first method to synthesize text-based affective gestures. Our generative method also considers the intended acting task of narration or conversation, the intended emotion based on the text and the context, and the intended gender and handedness of the virtual agents to generate plausible gestures. We can generate these gestures in a few milliseconds on an Nvidia GeForce GTX 1080Ti GPU. We also conducted a web study to evaluate the naturalness and emotional expressiveness of our generated gestures. Based on the 600 total responses from 50 participants, we found a strong positive correlation between the intended emotions of the virtual agents' gestures and the emotions perceived from them by the respondents, with a minimum Pearson coefficient of 0.77 in the valence dimension. Moreover, around 91% of the respondents found our generated gestures to be at least plausible on a five-point Likert Scale.

CHAPTER 6

Affective Co-Speech Gesture Synthesis Using Adversarial Expression Learning

Project Website: https://gamma.umd.edu/s2ag

Abstract

We present a generative adversarial network to synthesize 3D pose sequences of co-speech upper-body gestures with appropriate affective expressions. Our network consists of two components: a generator to synthesize gestures from a joint embedding space of features encoded from the input speech and the seed poses, and a discriminator to distinguish between the synthesized pose sequences and real 3D pose sequences. We leverage the Mel-frequency cepstral coefficients and the text transcript computed from the input speech in separate encoders in

our generator to learn the desired sentiments and the associated affective cues. We design an affective encoder using multi-scale spatial-temporal graph convolutions to transform 3D pose sequences into latent, pose-based affective features. We use our affective encoder in both our generator, where it learns affective features from the seed poses to guide the gesture synthesis, and our discriminator, where it enforces the synthesized gestures to contain the appropriate affective expressions. We perform extensive evaluations on two benchmark datasets for gesture synthesis from the speech, the TED Gesture Dataset and the GENEA Challenge 2020 Dataset. Compared to the best baselines, we improve the mean absolute joint error by 10–33%, the mean acceleration difference by 8–58%, and the Fréchet Gesture Distance by 21–34%. We also conduct a user study and observe that compared to the best current baselines, around 15.28% of participants indicated our synthesized gestures appear more plausible, and around 16.32% of participants felt the gestures had more appropriate affective expressions aligned with the speech.

6.1 Introduction

Co-speech gestures are bodily expressions associated with a person's speech (Yoon et al. 2019). They help underline the subject matter and the context of the speech, particularly in the form of beat, deictic, iconic, or metaphoric expressions (McNeill 1992). Beat gestures are rhythmic movements following the speech, and deictic gestures point to an entity. Iconic gestures describe physical concepts, *e.g.*, spreading and contracting the arms to denote "large" and "small", and metaphoric gestures describe abstract concepts, *e.g.*, putting a hand to the heart to denote "love". Synthesizing co-speech gestures is an important task in creating socially engaging characters and virtual agents. These are useful in a variety of multimedia application such as online learning (Li et al. 2016; Liao et al. 2019; Simeone et al. 2019), interviewing and counsel-

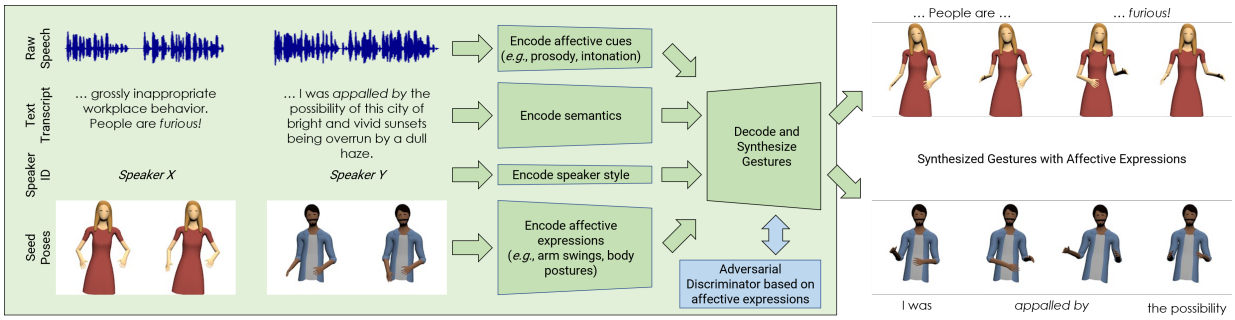


Figure 6.1: **Speech2AffectiveGestures: Overview.** We synthesize 3D pose sequences of co-speech upper-body gestures with appropriate affective expressions. We extract the affective cues from the speech, the sentiments from the corresponding text transcripts, the individual speaker styles, and the joint-based affective expressions from the seed poses (shown on the left). We train a generative adversarial network to synthesize gestures aligned with the speech by leveraging the affective information in both the generation and the discrimination phases. We show two such affective gestures on the right, with the affects *furious* and *appalled* denoted in italics.

ing (Baur et al. 2013; DeVault et al. 2014), robot assistants (Yoon et al. 2019), character designs and game development (Mascarenhas et al. 2018; Kucherenko et al. 2020), and visualizing stories and scripts (Watson et al. 2019).

In our work, we focus on synthesizing the upper-body gestures associated with speech. We consider the joints at the root, spine, head, and the two arms as part of the upper body, which are the joints most commonly used in co-speech gestures (Yoon et al. 2020; Kucherenko et al. 2020). Current state-of-the-art methods for co-speech upper-body gesture synthesis are based on an end-to-end learning approach (Ginosar et al. 2019; Yoon et al. 2020; Kucherenko et al. 2020). These methods train deep neural networks using gestures (available as videos or motion-captured datasets), raw speech waveforms and the corresponding text transcripts, and individual speaker styles. While these methods can generate different beat, deictic, iconic, and metaphoric co-speech gestures and adapt to speaker-specific styles, they do not have any mechanism to reliably incorporate affective expressions in the gestures.

Affective expressions are the modulations in gestures resulting from the emotions experienced by the speakers (Kleinsmith and Bianchi-Berthouze 2013; Karg et al. 2013). Even for a given speaker, the style of gesture expressions can change depending on the emotional context, and human observers are keenly alert to these changes (Karg et al. 2013). The combined understanding of the content of the speech and the speaker's gesture-based affective expressions are crucial to human-human interactions (Matsumoto, Frank, and Hwang 2012; Knapp, Hall, and Horgan 2013). Therefore, it is essential to incorporate affective expressions in co-speech gestures of animated characters and virtual agents to improve their plausibility in human-machine interactions.

In human-human interactions, we can break the gesture-based affective expressions down into a set of biomechanical features known as *affective features*, such as body postures, head positions, and arm motions (Kleinsmith and Bianchi-Berthouze 2013). Each affective expression is a combination of one or more affective features, *e.g.*, rapid arm swings and head jerks are often used as expressions of anger or excitement (Karg et al. 2013). A multitude of macroscopic and microscopic factors influence the affective features in a given context, including the social setting and the speaker's idiosyncrasies, making an exhaustive enumeration of affective features tedious and challenging (Bhattacharya et al. 2020). Nevertheless, it is essential to learn these affective features to understand and synthesize the desired affective expressions.

Moreover, co-speech affective gesture synthesis also requires aligning the gestures with the affective cues obtained from the speech. To this end, prior methods have either learned to map the raw speech waveforms to gestures via latent embeddings (Yoon et al. 2020) or utilized the log-Mel spectrograms to obtain a richer understanding of the affective cues, including the prosody and the intonations in the speech (Ginosar et al. 2019; Kucherenko et al. 2020). How-

ever, these are high-dimensional representations of speech that require significant computation overhead to be downscaled into convenient latent embedding spaces.

Main Contributions. We present an end-to-end learning approach for generating 3D pose sequences of co-speech gestures with appropriate affective expressions while maintaining the speakers' individual styles and following a short sequence of seed poses.[1] We leverage the Mel-frequency cepstral coefficients (MFCCs) from the speeches obtained by performing DCT on the log-Mel spectrograms. MFCCs are highly compressible representations containing sufficient information for speaker identification and also encode affective cues such as prosody and intonation for speech-based affect detection. We use separate encoders to encode the MFCCs from the raw speeches, the text transcripts obtained from the speeches, the speakers' styles, and the seed poses. We use available text- and speaker-encoders proposed by Yoon et al. 2020 to learn latent features from the text transcript and a latent style embedding space using a variational encoding of the speaker styles. We propose an encoder for the MFCCs that captures the affective cues in the speech. We also develop an "affective encoder" that transforms the 3D pose sequences to latent affective features using multi-scale spatial-temporal graph convolutions (STGCNs). We design our multi-scale STGCNs to expand attention from the local joints to the macroscopic body parts in a bottom-up manner. We use our affective encoder both in the generator to learn affective features from the seed poses to guide the gesture synthesis and in our discriminator to differentiate between the real and the synthesized gestures based on the affective expressions. To the best of our knowledge, we are the first to learn affective features directly from the gesture data to synthesize gestures with affective expressions. Our main

[1] Code and additional materials available at https://gamma.umd.edu/s2ag.

contributions include:

- **Synthesizing co-speech affective gestures.** We synthesize 3D pose sequences of gestures with appropriate affective expressions given a speaker's speech, maintaining the speakers' individual styles of gesticulation and following a short sequence of seed poses.
- **Affective encoder for learning latent affective features.** Our affective encoder leverages the localized joint movements and the macroscopic body movements in the 3D pose sequences to learn latent affective features that are used for synthesizing the future poses from the seed poses and adversarially guiding the synthesis as per affective expressions.
- **MFCC encoder for leveraging the affective cues from the speech.** Our MFCC encoder takes in low-dimensional MFCCs containing information on the affective cues from the speech, including prosody and intonations, and transforms them into latent embeddings for affective gesture synthesis.

We evaluate the quantitative performance of our network on two benchmark datasets, the TED Gesture Dataset (Yoon et al. 2019) and the GENEA Challenge 2020 Dataset (Kucherenko et al. 2021). We observe an improvement of 10–33% on the mean absolute joint error, 8–58% on the mean acceleration difference, and 21–34% on the Fréchet Gesture Distance (FGD) (Yoon et al. 2020) for our network compared to the current state-of-the-art baselines. We also conduct a user study to evaluate the plausibility of our synthesized gestures and the consistency between the affective expressions in the gestures and the speech. Around 15.28% participants indicated that our synthesized gestures are more plausible than the best current baseline of Yoon et al. 2020, and around 16.32% participants felt the gestures had more appropriate affective expressions aligned with the speech compared to the same baseline.

6.2 Related Work

We briefly summarize related prior work on how humans perceive affective body expressions and how these studies were leveraged to synthesize emotionally expressive characters. We also summarize works on synthesizing body motions, especially those aligned with a speech, a text transcript, or both.

6.2.1 Perceiving Affective Body Expressions

Affect is traditionally expressed in psychology in terms of its valence, arousal, and dominance (VAD) (Mehrabian and Russell 1974). Valence measures the level of pleasantness (*e.g.*, happy vs. sad), arousal measures how animated the person is (*e.g.*, angry vs. bored), and dominance measures the level of control over the affect (*e.g.*, admiration vs. fear). Studies in both psychology and affective computing indicate the existence of biomechanical *affective* features that provide cues to a person's perceived affect to human observers (Kleinsmith and Bianchi-Berthouze 2013; Karg et al. 2013; Castillo and Neff 2019; Banerjee, Bhattacharya, and Bera 2022; Kim et al. 2015). These affective features can be observed at different scales: they can be localized joint movements such as rapid arm swings and head jerks, indicating excitement or anger, as well as macroscopic body movements such as the upper body being expanded, indicating pride or confidence, or collapsed, indicating shame or nervousness. Subsequently, there has been work on detecting perceived emotions by leveraging known affective features either as input to a neural network (Bhattacharya et al. 2020) or to constrain the embedding space (Bhattacharya et al. 2020). In contrast, we design our neural network to explicitly attend to the body movements at these multiple scales to learn latent affective features directly from the input gesture samples.

6.2.2 Synthesizing Affective Body Expressions

There has been substantial work on synthesizing affective expressions for embodied conversation agents (Chowanda et al. 2016; Sohn et al. 2018) and other social virtual agents to interact via facial expressions (Karras et al. 2017; Ferstl and McDonnell 2018a) or gaits (Randhavane et al. 2019c). Furthermore, the synthesis of affective facial expressions has been aligned with a character's speech using data-driven techniques (Chuah, Rossen, and Lok 2009). While synthesizing speech-aligned affective facial expressions has been relatively well-studied, aligning the speech with affective body expressions has been more challenging. Some of the widely used approaches are rule-based systems such as that of DeVault et al. 2014, which has a virtual human counselor expressing appropriate affective hand and body gestures following known mappings between the emotional states and the stored animations. Recent methods utilize gait datasets annotated with categorical emotions such as happy, sad, and angry to generate emotive gaits (Randhavane et al. 2019a; Bhattacharya et al. 2020). Other techniques have extended to the VAD space of affect, where body gestures are generated given the text transcripts of speech and the corresponding intended emotion as a point in the VAD space (Bhattacharya et al. 2021b). Our approach is based on designing an end-to-end system that can synthesize body expressions by automatically understanding the affective content in the input speech.

6.2.3 Synthesizing Gestures

There is a rich body of work gesture synthesis using rule-based systems, as surveyed comprehensively by Wagner, Malisz, and Kopp 2014. However, scalability to novel scenarios remains a challenge for rule-based systems on account of manually designing new rules. Instead, we focus

on a summary of the recent data-driven approaches of automated gesture synthesis in novel scenarios (Chiu, Morency, and Marsella 2015), which are in line with our learning-based approach. Existing techniques have utilized hidden Markov models (Levine et al. 2010), recurrent neural network variants (Hasegawa et al. 2018; Yoon et al. 2019), and autoencoders (Kucherenko et al. 2019) to learn robust latent features that encode the input speech, available as either an audio or a text transcript, and can be used to decode the output gestures. Other approaches have opted to learn stochastic generation processes using tools such as invertible sub-transformations (Alexanderson et al. 2020) to map between the speech and the gesture spaces. To improve the realism of the generated speech-driven gestures, more recent works incorporate the speech semantics into the training process (Kucherenko et al. 2020), and even combined the synthesized gestures with rule-based head nods and hand waves for embodied conversation agents (Sadoughi and Busso 2019).

Our approach is complementary to these approaches in that we learn mappings from the text transcripts of speech to gestures. It eliminates the noise in speech signals and helps us focus only on the relevant content and context. Learning from the text also enables us to focus on a broader range of gestures, including iconic, deictic, and metaphoric gestures (McNeill 1992). Our work is most closely related to that of Yoon et al. 2019. They learn upper body gestures as PCA-based, low-dimensional pose features, corresponding to text transcripts from a dataset of TED-talk videos, then map these 3D gestures to an NAO robot. They have also followed up this work by generating upper-body gestures aligned with the three modalities of speech, text transcripts, and person identity (Yoon et al. 2020). On the other hand, we learn to map text transcripts to 3D pose sequences corresponding to semantic-aware, full-body gestures of more human-like virtual agents using an end-to-end trainable transformer network and blend

in emotional expressiveness.

6.2.4 Incorporating Speaker Styles

Co-speech gesture generation is intrinsically related to stylized gesture generation. There has been considerable progress on stylized generation of head motions (Greenwood, Laycock, and Matthews 2017; Sadoughi and Busso 2018), facial motions (Suwajanakorn, Seitz, and Kemelmacher-Shlizerman 2017; Ferstl and McDonnell 2018a) as well as locomotions (Holden, Saito, and Komura 2016; Holden, Komura, and Saito 2017; Starke et al. 2019). At the same time, many techniques have been proposed to generate appropriately styled body motions from textual descriptions of the actions (Li et al. n.d.; Zhou et al. 2019). Other approaches have developed separate gesture generation networks for individual speakers to adapt to their individual styles (Ginosar et al. 2019), together with adversarial losses to improve the fidelity of the generation (Ferstl, Neff, and McDonnell 2019). Recently, Yoon et al. 2020 proposed a unified architecture that considers the speech, its text transcript, and the speaker identity to generate co-speech gestures with continuously varying speaker styles. We extend such speaker-aware gesture synthesis to further incorporate the appropriate affective body expressions that align with the affective content in the speech.

6.3 Approach

Our goal is to generate 3D pose sequences of co-speech upper-body gestures with appropriate affective expressions and speaker styles, given the raw speech waveform, the speaker identity, and a short sequence of seed poses. We consider affective expressions to be specific sequences of joint movements, generally as a combination of the affective features (Bhattacharya et al.

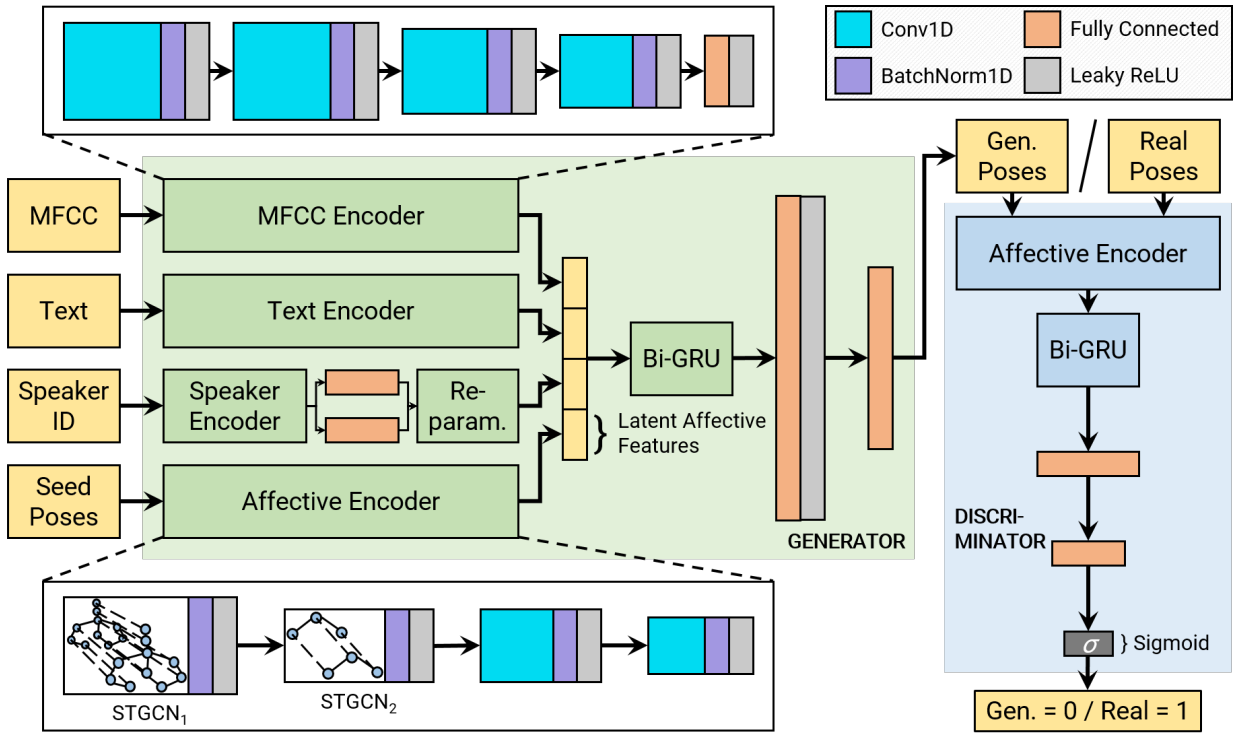


Figure 6.2: **Speech2AffectiveGestures: Generative Adversarial Expression Learning Network Architecture.** Our network consists of a generator (pale-green box) and a discriminator (pale-blue box). Our generator takes in the MFCC from the speech, the text transcript, the speaker ID, and a sequence of 3D seed poses. We use four encoders: the MFCC encoder (Section 6.3.1.1), the text encoder (Section 6.3.1.2), the speaker encoder (Section 6.3.1.3), and the affective encoder (Section 6.3.1.4). We feed the concatenation of these latent features into our Bi-GRU followed by a set of FC layers to synthesize the gestures aligned with the speech. Our discriminator learns to discriminate between the real and the synthesized gestures based on the latent affective features from the affective encoder, constraining the generator to synthesize appropriate affective expressions.

2021b). We learn these affective expressions both at the localized joint neighborhoods and the macroscopic body movements, and use them to condition the training of a generative adversarial network. We show our overall network architecture in Figure 6.2.

6.3.1 Synthesizing Co-Speech Gestures

Our generative network takes in the raw speech waveform as a 1D array, the corresponding text transcript as a sequence of words, the speaker identity as a unique number, and the seed poses

as a 3D pose sequence. Similar to Yoon et al. 2020, we encode the speech waveform, the text transcript, and the speaker identities using separate encoders. However, unlike Yoon et al. 2020, we convert the speech waveform to Mel-Frequency Cepstral Coefficients (MFCCs) to guide the encoding process based on the affective cues from speech. We also propose an affective encoder to encode the pose-based affective expressions into latent features for both gesture generation and discrimination. In the generation process, we combine the latent embeddings learned from the four encoders, speech, text, speaker, and affective, into a joint embedding for learning the upper-body gestures.

6.3.1.1 MFCC Encoder

MFCCs are known to encode signal frequencies consistent with how humans perceive sound, and are therefore particularly useful for tasks such as speech recognition (Vergin, O'Shaughnessy, and Farhat 1999), speaker identification (Murty and Yegnanarayana 2006) and speech-based affect detection (Neiberg, Elenius, and Laskowski 2006). In our case, we design our MFCC encoder to embed the speech-based affective cues such as prosody and intonations captured by the MFCCs and incorporate them in gesture synthesis. Given a raw waveform as a 1D array, we transform it to its top 14 MFCCs. These include the log-energy spectrum and 13 coefficients containing sufficient information on the speaker's pitch, intonation, prosody, and other relevant parameters (Bagher Zadeh et al. 2018; Suwajanakorn, Seitz, and Kemelmacher-Shlizerman 2017). We also append the first- and the second-order discrete forward differences of the 13 coefficients, obtaining a total of 37 values. Using a window size W on a input waveform of length L, we obtain individual MFCCs of shape $\lceil L/W \rceil$, leading to a combined feature tensor $f_m \in \mathbb{R}^{37 \times \lceil L/W \rceil}$. We pass these features through a series of 1D temporal convolutions, fol-

lowed by a single fully-connected (FC) layer, to obtain a latent feature sequence $\hat{f}_m \in \mathbb{R}^{D_m \times T}$ of sequence length T equal to 3D pose sequence length of the seed poses, as

$$\hat{f}_m = \text{Conv} \circ \text{FC}_{mfcc} \left(f_m; W_{mfcc} \right), \tag{6.1}$$

where D_m is the dimension of the latent features, $\text{Conv} \circ \text{FC}_{mfcc}$ denotes the series of 1D convolutions followed by the FC layer, and W_{mfcc} its set of trainable parameters.

6.3.1.2 Text Encoder

Given the text transcript corresponding to the speech, we first pad the transcript with padding tokens following the approach of Yoon et al. 2020, to ensure that the text transcript has the same sequence length T as the seed poses. We then use the pre-trained FastText (Bojanowski et al. 2017) word embedding model to transform the word sequence into 300-dimensional features, leading to a feature tensor $f_x \in \mathbb{R}^{300 \times T}$. We use FastText for its memory efficiency and its usefulness in sentiment analysis (Santos, Nedjah, and Macedo Mourelle 2017), which is important in understanding text-based affect. We pass the FastText features through a series of temporal 1D convolutions to obtain a latent feature sequence $\hat{f}_x \in \mathbb{R}^{D_x \times T}$, as

$$\hat{f}_x = \text{Conv}_{text} \left(f_x; W_{text} \right), \tag{6.2}$$

where D_x is the dimension of the latent features, Conv_{text} denotes the series of 1D convolutions with trainable parameters W_{text}.

6.3.1.3 *Speaker Encoder*

For the speaker IDs, we use one-hot vectors $f_s \in \{0,1\}^{\mathcal{S}}$, assuming $\mathcal{S}$ is the number of available speakers. Following Yoon et al. 2020, we use two sets of FC layers to learn an embedding space capturing the mean $\mu_s \in \mathbb{R}^{D_s}$ and the variance $\Sigma_s \in \mathbb{R}_+^{D_s \times D_s}$ of the latent distribution of the speaker styles as

$$\mu_s = \mathrm{FC}_\mu\left(f_s; w_\mu\right) \tag{6.3}$$

$$\log \Sigma_s = \mathrm{FC}_\Sigma\left(f_s; w_\Sigma\right), \tag{6.4}$$

where D_s is the dimension of the latent distribution space, FC_μ and FC_Σ denote the two sets of FC layers, and W_μ and W_Σ denote the corresponding sets of trainable parameters. Intuitively, this latent distribution space consists of all the available speakers, plus speakers that can be "constructed" by linear combinations of those speakers in the latent space. As a result, we can pick a random point from the latent space to use in the synthesis, resulting in some variability in the synthesized gestures even when the speech remains the same. We term this variability as having "speaker-aware" styles. Given the parameters μ_s and Σ_s of latent distribution space, we use the re-parametrization trick (Kingma and Welling 2019) to generate a random speaker-aware style sample $\hat{f}_s \in \mathbb{R}^{D_s}$ and repeat it for all the T time steps of the input pose sequence.

6.3.1.4 *Affective Encoder*

We propose an encoding mechanism that transforms the pose-based affective expressions into a latent embedding. Since gestures typically consist of movements in the trunk, arms, and head,

we only consider ten joints corresponding to these parts of the body: root, spine, neck, head, left and right shoulders, left and right elbows, and left and right wrists. We consider a directed graph for the pose, where the joints are the vertices, and the edges are directed from the root towards the extremities. We assume the edge lengths are known for each input and train our encoder only on the directions of the edges. We consider nine unit-vector sequences $U = [u_1; \ldots; u_9]$, each of sequence length T, to denote the edge directions at the corresponding T time steps of the input pose sequence.

We employ a hierarchical encoding strategy using spatial-temporal graph convolutions (STGCNs) (Yan, Xiong, and Lin 2018). STGCNs are adapted to leverage localized dependencies in generalized graph-structured data, and are therefore suitable for our pose graph sequences. We use two levels of hierarchy, the first at the level of individual bones and the second at the level of the three body parts, the trunk and the two arms. At the first level, our unweighted adjacency matrix $A_1 \in \{0,1\}^{9\times 9\times T}$ captures the temporal counterparts of each edge at the four nearest time steps (past two and future two), and spatially adjacent edges with a maximum hop of two, *i.e.*, we consider two edges to be spatially adjacent if they either share a vertex or are connected to the two ends of a third edge. This size of the adjacent neighborhood sufficiently groups the edges influenced by typical affective expressions such as arm swings, head jerks, and upper-body collapse. Consequently, the convolution filters can learn a latent feature sequence $\hat{f}_{a_1} \in \mathbb{R}^{D_{a_1}\times 9\times T}$ from the edges based on the variations in the affective expressions, obtained as

$$\hat{f}_{a_1} = \text{STGCN}_1\left(U, A_1; W_{a_1}\right), \tag{6.5}$$

where D_{a_1} is the dimension of the per-edge latent features, STGCN_1 denotes the first-level

STGCN with trainable parameters W_{a_1}. At the second level, the three body parts, the trunk and the two arms, capture the macroscopic body movements such as raising or crossing the arms, and bending or straightening the trunk. In the second-level adjacency matrix $A_2 \in \{0,1\}^{3\times3\times T}$, we assume both the arms to be adjacent to the torso but not to each other, since the movements on one arm need not influence the other. We again consider the temporal counterparts of each body part in the four nearest time steps in the temporal adjacency. We reshape the latent features $\hat{f}_{a_1}$ to $3D_{a_1} \times 3 \times T$, to collect the per-edge features corresponding to the three body parts in the feature dimension. Our second-level STGCN then operates on these reshaped features to produce the second-level latent features $\hat{f}_{a_2} \in \mathbb{R}^{D_{a_2}\times3\times T}$ as

$$\hat{f}_{a_2} = \text{STGCN}_2\left(\hat{f}_{a_1}, A_2; W_{a_2}\right), \tag{6.6}$$

where D_{a_2} is the dimension of the per-edge latent features, STGCN_2 denotes the second-level STGCN with trainable parameters W_{a_2}. We then apply a series of 1D convolutions on the reshaped second-level features $\hat{f}_{a_2} \in \mathbb{R}^{3D_{a_2}\times T}$ to obtain the latent affective feature sequence $\hat{f}_a \in \mathbb{R}^{D_a\times T}$, as

$$\hat{f}_a = \text{Conv}_{aff}\left(\hat{f}_{a_2}; W_a\right), \tag{6.7}$$

where D_a is the dimension of the latent affective features, and Conv_{aff} denotes the series of 1D convolutions with trainable parameters W_a.

6.3.1.5 Gesture Generator

Given the latent feature sequences $\hat{f}_m$, $\hat{f}_x$, $\hat{f}_s$, and $\hat{f}_a$, we concatenate them, pass them through a bidirectional gated recurrent unit (Bi-GRU), and sum the bidirectional outputs to obtain the

predicted edge embeddings sequence $\hat{u}_e \in \mathbb{R}^{D_e \times T}$, as

$$out_{frw}, out_{bkw} = \text{GRU}_e\left(\left[\hat{f}_m; \hat{f}_x; \hat{f}_s; \hat{f}_a\right]; W_e\right), \tag{6.8}$$

$$\hat{u}_e = out_{frw} + out_{bkw}, \tag{6.9}$$

where D_e is the dimension of the predicted edge embeddings, GRU_e denotes the bidirectional GRU with the corresponding set of trainable parameters W_e, and out_{frw} and out_{bkw} respectively denote the outputs of the forward and the backward channels of the GRU. As in Yoon et al. 2020, we then transform the predicted edge embeddings to predicted edge vector sequences $\hat{U} = [\hat{u}_1; \ldots; \hat{u}_9]$, each of sequence length T, using a set of FC layers as

$$\hat{U} = \text{FC}_{gen}\left(\hat{u}_e; W_{gen}\right), \tag{6.10}$$

where FC_{gen} denotes the set of FC layers with the trainable parameters W_{gen}. Thus, our generator is designed to take in a sequence of seed poses of length T and predicts a pose sequence of gestures for the next T time steps. Finally, we scale each predicted edge vector $\hat{u}_i$ to have the corresponding bone length b_i, $i = 1, \ldots, 9$. We add it to the 3D position $pos_{s(i)}$ of the source joint $s(i)$ of that edge vector to obtain 3D position $pos_{d(i)}$ of the destination joint $d(i)$ of the same edge vector, as

$$pos_{d(i)} = pos_{s(i)} + b_i \cdot \frac{\hat{u}_i}{\|\hat{u}_i\|}. \tag{6.11}$$

6.3.2 Discriminating Gestures

Our discriminator takes in a gesture of sequence length T and computes its latent affective feature sequence $\hat{f}_a \in \mathbb{R}^{D_a \times T}$ using our affective encoder (Section 6.3.1.4). We pass this feature sequence through another bidirectional GRU, and sum the bidirectional outputs to obtain the discriminator embeddings sequence $\hat{d} \in \mathbb{R}^{h \times T}$, as

$$out_{frw}, out_{bkw} = \text{GRU}_{disc}\left(\hat{f}_a; W_{GRU,disc}\right), \tag{6.12}$$

$$\hat{d} = out_{frw} + out_{bkw}, \tag{6.13}$$

where D_d is the dimension of the predicted discriminator embeddings, GRU_{disc} denotes the bidirectional GRU with trainable parameters W_{GRU_disc}, and out_{frw} and out_{bkw} respectively denote the outputs of the forward and the backward channels of the GRU. We then transform the discriminator embeddings to a probability vector $c \in [0, 1]$ using a set of FC layers as

$$c = \text{FC}_{disc}\left(\hat{d}; W_{FCdisc}\right), \tag{6.14}$$

where FC_{disc} denotes the set of FC layers with trainable parameters W_{FCdisc}, and c is such that $c \geq 0.5$ implies the discriminator predicts the input gesture to be real, and generated otherwise.

6.4 Dataset and Training

We train our network on the TED Gesture Dataset (Yoon et al. 2019), which consists of videos of English-language speakers at TED Talks. It provides 3D pose sequences of the upper-body gestures of the speakers, their speech audio, and the associated text transcripts. Each data

Table 6.1: **Speech2AffectiveGestures: Network Hyperparameters.** We chose all the values via empirical search.

HP	Description	Value
D_m	Latent feature from the MFCC encoder	32
D_x	Latent feature from the text encoder	32
D_s	Latent distribution space of speaker styles	16
D_{a_1}	Per-edge latent features after STGCN_1 in the affective encoder	16
D_{a_2}	Per-edge latent features after STGCN_2 in the affective encoder	16
D_a	Latent affective features from the affective encoder	16
D_e	Predicted edge embeddings from the GRU in the generator	150
D_d	Predicted embeddings from the GRU in the discriminator	150

sample has a sequence length of $T = 34$ time steps at a rate of 15 fps. There are 200,038 training samples in total, constituting around 80% of the dataset. The evaluation set consists of 26,903 samples or around 10% of the dataset. The test set consists of 26,245 samples, making up the remaining 10% of the dataset.

We use loss functions $\mathcal{L}_G$ and $\mathcal{L}_D$ identical to Yoon et al. 2020 to train our generator and discriminator respectively:

$$\mathcal{L}_G = \lambda_{Hub}\mathcal{L}_{Hub} + \lambda_{gen}\mathcal{L}_{gen} + \lambda_{stl}\mathcal{L}_{stl} + \lambda_{KLD}\mathcal{L}_{KLD}, \tag{6.15}$$

$$\mathcal{L}_D = -\mathbb{E}\left[\log\left(\text{Disc}\left(U\right)\right)\right] - \mathbb{E}\left[\log\left(1 - \text{Disc}\left(\hat{U}\right)\right)\right], \tag{6.16}$$

where Disc denotes the discriminator network (Section 6.3.2), λ_* are the weights of the corresponding loss terms with the same values as in Yoon et al. 2020, and the individual loss terms of the generator are:

- **Huber loss** (Huber 1965) between the ground truth and predicted edge vectors,

- **generative adversarial loss** on the output of the discriminator,

$$\mathcal{L}_{gen} = -\mathbb{E}\left[\log\left(\text{Disc}\left(\hat{U}\right)\right)\right], \tag{6.17}$$

- **diversity regularization** between the synthesized gestures and other gestures in the dataset to ensure that the styles of different speakers appear visually different,
- **Kullback-Leibler (KL) divergence** between the latent distribution space of the styles defined by μ_s and Σ_s, and the normal distribution $\mathcal{N}(0, I)$.

Table 6.1 lists the latent dimensions we use for training our network. We use the Adam optimizer (Kingma and Ba 2014) with $\beta_1 = 0.5$, $\beta_2 = 0.999$, batch size of 512, and learning rate of 5E^{-4} for the generator and 1E^{-4} for the discriminator with no warm-up epochs (*i.e.*, $\lambda_{gen} > 0$ starting from the first epoch). We train our network for 300 epochs, which took close to 45 hours on an NVIDIA GeForce GTX 1080 Ti GPU.

6.5 Experiments

We describe the objective evaluation of our method compared to current baseline methods. We highlight the benefit of our proposed components via ablation studies on the objective evaluation metrics. We also show the qualitative performance of our method on selected samples from the TED Gesture Dataset (Yoon et al. 2019) and the perceived quality of our synthesized gestures through a user study.

6.5.1 Baseline Methods

We compare our method with the baseline methods on two benchmark datasets, the TED Gesture Dataset (Yoon et al. 2019), and the GENEA Challenge 2020 Dataset (Ferstl and McDonnell 2018b; Kucherenko et al. 2021).

On the TED Gesture Dataset, we compare with the methods of Seq2Seq (Yoon et al. 2019), Speech to Gestures with Individual Styles (S2G-IS) (Ginosar et al. 2019), Joint Embedding Model (JEM) (Ahuja and Morency 2019), and Gestures from Trimodal Context (GTC) (Yoon et al. 2020). Seq2Seq and JEM generate gestures based only on the text transcript of the speech, whereas S2G-IS uses only the speech to generate the gestures. GTC uses the speech, the corresponding text transcript, and the speaker styles to generate gestures. Seq2Seq follows an encoder-decoder architecture, where the authors transform the text to latent features and predict gestures based on both the latent features and a short gesture history. The authors of S2G-IS employ a generative adversarial network that generates gestures from a latent space obtained from the input log-Mel spectrograms. JEM maps both the text and the target gesture into a common latent embedding space and uses a decoder to reconstruct the gestures from the embedding space. The authors train the model to learn to align the text-based and the gesture-based embeddings for the same input and decode gestures from only the text-based embeddings. For Seq2Seq, S2G-IS, and JEM, we follow the training routine and the hyperparameters used by Yoon et al. 2020. For GTC, we directly use the pre-trained model provided by Yoon et al. 2020.

The GENEA Challenge 2020 Dataset is the publicly available version of the Trinity Gesture Dataset (Ferstl and McDonnell 2018b; Kucherenko et al. 2021). It consists of the speech and the full-body motion capture of a male actor talking unrestrained about various topics over multiple

Table 6.2: **Speech2AffectiveGestures: Quantitative Comparisons.** Evaluation of our method with baselines and ablated versions of our method on two benchmark dataset, using the objective metrics of mean absolute joint error (MAJE), mean acceleration difference (MAD), and the Fréchet Gesture Distance (FGD). Bold indicates best.

Dataset	Method	MAJE (mm)	MAD (mm/s^2)	FGD
TED Gesture (Yoon et al. 2019)	Seq2Seq (Yoon et al. 2019)	45.62	6.33	6.62
	S2G-IS (Ginosar et al. 2019)	45.11	7.22	6.73
	JEM (Ahuja and Morency 2019)	48.56	4.31	5.88
	GTC (Yoon et al. 2020)	27.30	3.20	4.49
	Ours w/o MFCC Enc.	27.84	3.02	4.21
	Ours w/o Aff. Enc.	25.38	3.51	4.84
	Ours	**24.49**	**2.93**	**3.54**
GENEA Challenge 2020 (Kucherenko et al. 2021)	Gesticulator (Kucherenko et al. 2020)	82.41	3.62	31.04
	Ours w/o MFCC Enc.	105.71	1.57	23.03
	Ours w/o Aff. Enc.	92.90	2.81	24.28
	Ours	**54.93**	**1.49**	**20.36**

recording sessions. The full dataset is about 242 minutes long, of which 221 minutes are used as training data, and the remaining 21 minutes are kept for testing. We do not fine-tune our network on this dataset and evaluate our network on the test partition. Since we consider only upper-body gestures, we consider the ten relevant upper-body joints at the root, the spine, the head, and the two arms for evaluating our performance. On this dataset, we compare with the method of Gesticulator (Kucherenko et al. 2020), which leverages the acoustics and the semantics of the speech to generate semantically consistent beat, deictic, metaphoric, and iconic gestures. For a fair comparison, we use the pre-trained model provided by the authors and compare the performance on the same ten joints that we use for our method.

6.5.2 Objective Evaluation

While evaluation metrics for gesture synthesis are not standardized, we evaluate on the commonly used metrics of mean absolute joint error (MAJE), mean acceleration difference (MAD), and the Fréchet Gesture Distance (FGD) proposed by Yoon et al. 2020. MAJE measures the mean of the absolute differences between the ground truth and the predicted joint positions over all the time steps, joints, and samples. MAD measures the mean ℓ_2-norm error between the ground truth and predicted joint accelerations over all the time steps, joints, and samples. FGD measures the difference between the distributions of the latent features of the ground truth and the predicted gestures. The latent features are computed from an autoencoder network trained on the well-known Human 3.6M dataset (Ionescu et al. 2014) of human motions using the ten joints in the TED Gesture Dataset (Yoon et al. 2019). MAJE indicates how closely the predicted joint positions follow the ground truth joint positions. MAD indicates how closely the ground truth and predicted joint movements match. Since affective expressions are based on joint movements, a lower MAD is especially desirable for our stated aim of generating gestures with appropriate affective expressions. FGD is shown to align well with the perceived plausibility of the synthesized gestures to human users (Yoon et al. 2020); therefore, a lower FGD is equally desirable to gauge the quality of our synthesized gestures.

Table 6.2 summarizes the performance of all the methods on all these evaluation metrics. Our method consistently has the lowest MAJE, MAD, and FGD on both the benchmark datasets. On the TED Gesture Dataset, we observe improvements of 10.29%, 8.44%, and 21.16% on MAJE, MAD, and FGD, respectively, over the best current baseline of GTC. On the GENEA Challenge 2020 Dataset, we observe improvements of 33.34%, 58.84%, and 34.41% on MAJE, MAD, and

FGD, respectively, over the baseline of Gesticulator. We note that the absolute FGD values are significantly higher on the GENEA Challenge 2020 Dataset than on the TED Gesture Dataset. We hypothesize that this is because the gestures in the GENEA Challenge 2020 Dataset are more abstract and unscripted compared to the well-defined actions in the Human 3.6M Dataset or polished speeches in the TED Gesture Dataset. As a result, the pre-trained latent embeddings used for FGD are not as good at reconstructing the joints movements in the GENEA Challenge 2020 Dataset.

6.5.3 Ablation Studies

We perform ablation studies on our two proposed components: the MFCC encoder (Section 6.3.1.1) and the affective encoder (Section 6.3.1.4). In one study, we replace only our MFCC encoder with an encoder for the raw audio waveform, identical to that of GTC (Yoon et al. 2020), and train the resultant network. In the other study, we remove the affective encoder from our network. Our generator takes in the raw seed poses instead of the latent affective features. Our discriminator uses a convolution filter identical to GTC (Yoon et al. 2020) to transform the input gestures to latent features for the bidirectional GRU.

Without the MFCC encoder, our generator cannot take the speech-based affective cues into account. It results in a degradation of the synthesis, leading to higher MAJE, MAD, and FGD. However, without the affective encoder, our network is severely limited in understanding both the affective expressions in the seed poses and the affective expressions of the synthesized poses. It results in more severe performance degradation on all the evaluation metrics, and the synthesized gestures appear less diverse and plausible.

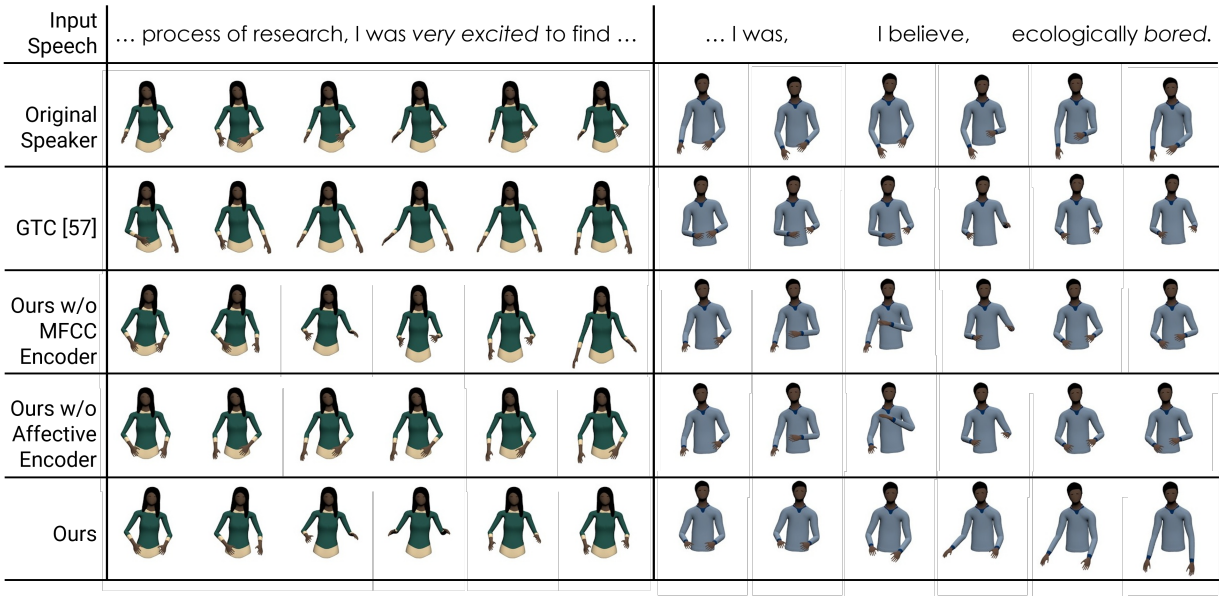
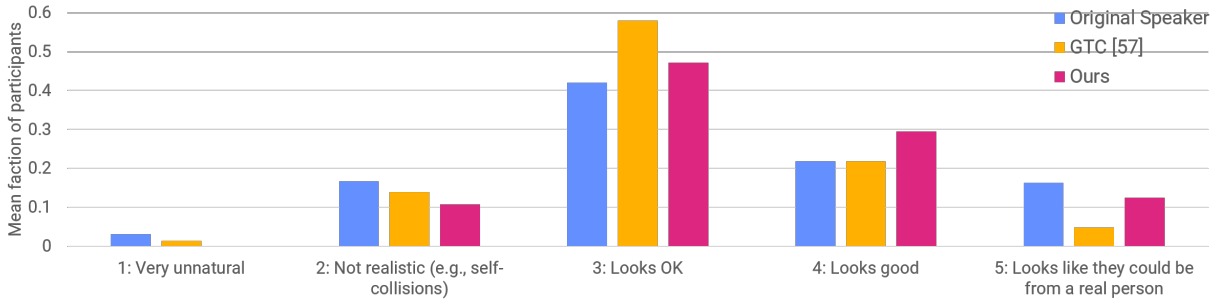


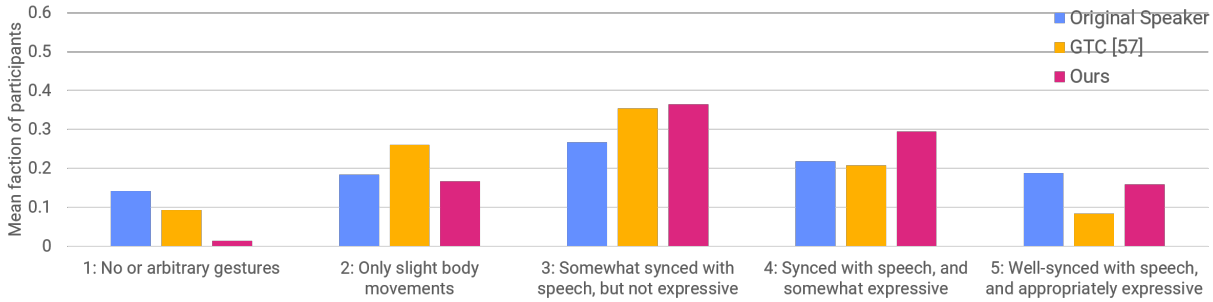
Figure 6.3: **Speech2AffectiveGestures: Qualitative Comparisons and Ablation Studies.** Qualitative results on the gestures synthesized by our method for two sample speech excerpts from the TED Gesture Dataset (Yoon et al. 2019). The italicized words *very excited* and *bored* indicate the primary affect in the corresponding speeches. We compare with the corresponding gestures of the original speakers, the output of GTC (Yoon et al. 2020), and that of the two ablated versions of our network (Section 6.5.3). See Section 6.5.4 for a detailed discussion of the results.

6.5.4 Qualitative Results

We show qualitative results on two sample speech excerpts from the TED Gesture Dataset (Yoon et al. 2019) in Figure 6.3. It has five rows of gestures: those of the original speakers', those synthesized by GTC (Yoon et al. 2020) (the current state-of-the-art), those by the two ablated versions of our network: one without our MFCC encoder (Section 6.3.1.1) and the other without our affective encoder (Section 6.3.1.4), and those by our proposed network with all the encoders. We observe a diversity of speaker styles in the synthesized gestures compared to the original speaker, which results from using a variational embedding of speaker styles using the speaker encoder (Section 6.3.1.3). GTC, however, cannot generate affective expressions except for a few words with strong intonations in the speech, such as "excited" (second row, left column).



(a) Plausibility of the different types of gestures.



(b) Synchronization of the movements and the affective expressions of the different types of gestures with the speech.

Figure 6.4: **Speech2AffectiveGestures: Synthesis Quality Scores from the User Study.** Mean fraction of participant responses on each point of the Likert scales across the 12 speech excerpts from the TED Gesture Dataset (Yoon et al. 2019) and the corresponding gestures in our user study. See Section 6.5.5 for details.

Without our MFCC encoder, our network can still match the speech content but cannot align the gestures with the affective cues from the speech. For example, it can match the words "I was, I believe" with a deictic gesture pointing to the speaker himself (third row, right column) but cannot generate any expressions for "bored". Without our affective encoder, we observe only slight body movements but no appreciable affective expressions in the synthesized gestures. With all our encoders in place, we observe appropriate affective expressions that align well with the speech. For example, we observe rapid arm movements when saying "excited" (fifth row, left column) and dropping of the arms and shoulders when saying "bored" (fifth row, right column).

6.5.5 User Study

We conducted a user study to evaluate the perceptual quality of our synthesized gestures in terms of how plausible they appear and how well-aligned are their affective expressions with the corresponding speeches. 24 participants took part in our study, of which 20 were male, and 4 were female. 10 participants were between 18 and 24 years of age, 13 were between 25 and 34, and one was above 35. Each participant observed gestures corresponding to the same 12 speech excerpts, each taken from a different TED Talk in the TED Gesture Dataset (Yoon et al. 2019). For each speech excerpt, the participants observed three different types of gestures: that of the original speaker as a 3D pose sequence (provided in the dataset), those synthesized by GTC (Yoon et al. 2020), the current state-of-the-art, and those synthesized by our network. The order of the gestures was unknown to and randomized for each participant. We then asked the participants to answer two questions. The first question was how plausible the gestures appeared on a five-point Likert scale ranging from "very unnatural" (1) to "look like they could be from a real person" (5). The second question was how well the gestures synchronized with the corresponding speeches on a five-point Likert scale, ranging from "no or arbitrary gestures" (1) to "well-synchronized with the speech, and are appropriately emotionally expressive" (5). Intuitively, our Likert-scale points for both questions reflect the participants' individual assessments of quality, with 1 being the worst, 3 being average, and 5 being the best. The entire study took around 20 minutes on average for each participant.

We summarize the participants' responses in Figure 6.4. When adjudging the plausibility of the gestures (Figure 6.4a), we observe that 15.28% more participants marked our synthesized gestures either 4 or 5 compared to the gestures synthesized by GTC (Yoon et al. 2020). Further,

3.82% more participants marked our synthesized gestures 4 or 5 than the original speakers' gestures, indicating that the participants found our synthesized gestures to have visual quality comparable to that of the original data. When adjudging the synchronization of the movements and the affective expressions of different types of gestures with speech (Figure 6.4b), we observed that 16.32% more participants marked our synchronization quality either 4 or 5 compared to that of GTC (Yoon et al. 2020). Also, 4.86% more participants marked out synchronization quality 4 or 5 than that of the original speakers, indicating that the participants perceived our synthesized gestures to be as well-synchronized and expressive as the original data.

6.6 Conclusion

We have presented an end-to-end learning approach to generate 3D pose sequences of co-speech gestures with appropriate affective expressions. Our contributions include an MFCC encoder to guide the gesture synthesis based on the speech-based affective cues such as prosody and intonation, and an affective encoder to learn joint-based affective features from the gesture data. Using these encoders in a generative adversarial learning framework, we have synthesized affective gestures that advance the state-of-the-art on co-speech gesture synthesis on multiple evaluation metrics. Our synthesized gestures also appeared more plausible and well-synced with the corresponding speeches to participants in a user study.

CHAPTER 7

Affective Synchronous Co-Speech Face and Gesture Synthesis Using Adversarial Multimodal Expression Learning

Project Website: https://gamma.umd.edu/s2ue

Abstract

We present a multimodal learning-based method to simultaneously synthesize co-speech facial expressions and upper-body gestures for virtual agents using RGB video data captured using commodity cameras. Our approach learns from sparse face landmarks and upper-body joints, estimated directly from video data, to generate plausible emotive character motions. Given a speech audio waveform and a token sequence of the speaker's face landmark motion and body-joint motion computed from a video, our method synthesizes the full sequence of motions for

the speaker's face landmarks and body joints that match the content and the affect of the speech. To this end, we design a generator consisting of a set of encoders to transform all the inputs into a multimodal embedding space capturing their correlations, followed by a pair of decoders to synthesize the desired face and pose motions. To enhance the plausibility of our synthesized motions, we also use an adversarial discriminator that learns to differentiate between the face and pose motions computed from the original videos and our synthesized motions based on their affective expressions. To evaluate our approach, we extend the TED Gesture Dataset to include view-normalized, co-speech face landmarks in addition to body gestures. Our experimental results demonstrate that our method results in low reconstruction error and produces synthesized samples with diverse facial expressions and body gestures for virtual agents. We conduct a web-based user study to evaluate the visual quality of our synthesized samples and observe that around 88% of participants reported that our virtual agent motions are plausible and around 62% reported that the face and pose expressions in our samples were collectively synchronized with the speech. We will release the extended dataset as the TED Gesture+Face Dataset consisting of 250K samples and the relevant source code.

7.1 Introduction

Spoken communications are a significant component of human-human interactions in everyday life. Further, human communications through shared digital platforms and virtual spaces has been steadily rising in many applications, including online learning (Li et al. 2016; Liao et al. 2019; Simeone et al. 2019), virtual interviewing (Baur et al. 2013), counseling (DeVault et al. 2014), social robotics (Yoon et al. 2019), automated character designing (Mascarenhas et al. 2018), storyboard visualizing for consumer media (Kucherenko et al. 2020; Watson et al. 2019),

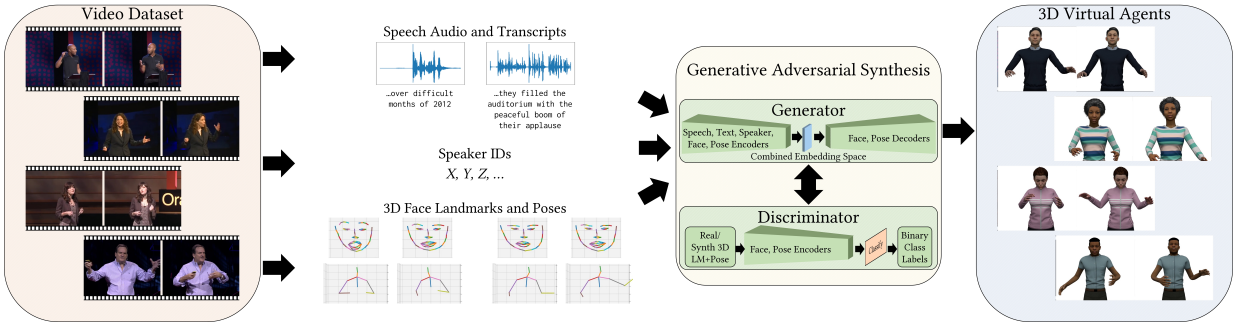


Figure 7.1: **Speech2UnifiedExpressions: Overview.** Our method leverages the individual distributions of the speech audio, the corresponding text transcripts, the speaker's unique IDs, and their sparse 3D face landmarks and pose sequences computed from RGB video data. It learns a multimodal embedding space that captures the correlations between all these inputs and leverages them to generate expressive face and pose motions for virtual agents that are synchronized with each other and with the speech.

and creating large-scale VR worlds such as the metaverse (Oculus 2019) and industry-grade virtual character synthesis platforms such as the Omniverse (NVIDIA n.d.). Simulating immersive experiences in such digital and virtual applications necessitates the development of plausible virtual human surrogates with expressive face and body motions. This is a challenging problem to approach at scale, given the diversity in human expressions and how central these expressions are in human-human interactions (Parkinson, Fischer, and Manstead 2005; Mesquita and Boiger 2014). The problem becomes even harder when we consider the fact that humans communicate by expressing emotions simultaneously through multiple cues or *modalities*, such as their speech, facial expressions, and body gestures (Mittal et al. 2020a). The emotional expressions from these different modalities are also synchronous, *i.e.*, they follow the same rhythm of communication and complement each other to fully convey a sense of presence (Laban and Ullmann 1971).

In this paper, we consider the problem of synthesizing 3D digital human motions with synchronous facial expressions and upper-body gestures aligned with given speech audio in-

puts. Given the speech audio, current approaches in computer graphics, virtual reality (VR), and AI tackle the sub-problems of "talking heads" (Karras et al. 2017) – synthesizing lip movements and facial expressions given the speech audio, and co-speech gesture synthesis (Yoon et al. 2020) – synthesizing poses for upper-body gestures, including head motions. However, these approaches synthesize only one modality, either facial expressions or body gestures. They do not consider any synchronization between the synthesized modalities, which is crucial for creating plausible virtual agents (Habibie et al. 2021). The inherent difficulty in synthesizing expressions synchronized across the different modalities is that such expressions are correlated (Ambady and Rosenthal 1992; Gunes and Piccardi 2007), *e.g.*, stretching arms and widening eyes in sync when expressing surprise. In other words, not only is the combined space of the multimodal expressions very high-dimensional, but only a small fraction of that space corresponds to *valid* expressions. Moreover, current approaches generally require specialized data such dense 3D face scans (Cudeiro et al. 2019) and motion-captured gestures (Busso et al. 2008; Bhattacharya et al. 2021b) to provide meaningful results. By contrast, our goal is to leverage large-scale video datasets (Yoon et al. 2019) to develop synchronous co-speech face and pose expressions, with the aim of synthesizing fully expressive 3D digital humans for democratized use in social VR environments such as virtual conferences and other metaverse applications.

Apart from the challenges of synthesizing synchronous 3D face and pose expressions from only video data, a post-hoc or *asynchronous* combination of separately synthesized expressions of these modalities at the output is also insufficient as it does not consider their mutual correlation. Instead, we need to develop methods that ensure that these two modalities remain in sync with each other as well as with the speech audio input.

Plausible synthesis also involves learning the appropriate affect-sensitive representations

for both the face and the pose modalities. Facial expressions are efficiently captured by facial landmarks called *action units* developed from the facial action coding system (FACS) (Ekman and Friesen 1978). Pose expressions are typically expressed using body movements such as arm swings, spine posture, and head motions (Kleinsmith and Bianchi-Berthouze 2013). Therefore, a joint synthesis of the face and pose expressions requires learning the unified, multimodal distribution of their affect representations. We also need an automated mapping from the input speech audio to the synchronous facial expressions and body gestures. In practice, a plausible multimodal mapping needs to be not only one-to-many but also stochastic (Henter, Alexanderson, and Beskow 2020), to account for the subtle variations among human speakers over time even when delivering the same speech.

Main Contributions. We present a multimodal learning algorithm to synthesize animated 3D digital humans with synchronous face and upper-body pose expressions given the corresponding speech audio inputs. We leverage the affective information in the speech audio using features learned from the Mel-frequency cepstral coefficients (MFCCs), and semantic embeddings learned from the corresponding text transcripts. We consider both intra- and inter-speaker variability by leveraging a set of unique speakers and incorporating stochasticity via random sampling on a latent space for speakers. We learn latent spaces of affect-aware features for both the faces and the poses by representing the face landmark and the pose sequences as multi-level graphs based on the anatomical components (ACs) of the face and the upper body. Our AC graphs efficiently capture the correlations between the individual face landmarks and body joints corresponding to different affective expressions of the speakers. We also learn a multimodal embedding space that enables us to represent the synchronous face and pose expressions.

We decode variables from this space to synthesize our desired synchronous face landmark and pose sequences. We use a discriminator to further improve the plausibility of our synthesizes sequences based on their synchronous, affective expressions. We perform quantitative and qualitative evaluations on the benchmark TED Gesture Dataset (Yoon et al. 2019) that consists of videos of TED-Talk speakers. The current version of the dataset only consists of 3D joint poses extracted from videos. Therefore, we append the 3D face landmarks we extract from the videos to this dataset to perform our experiments as well as enable future research in co-speech face and pose synthesis. To summarize, our main contributions include:

- **Synchronous co-speech face and pose expression synthesis.** Our method simultaneously synthesizes face and upper-body pose expressions given speech audio as a result of learning a generative multimodal embedding space and training a discriminator network. Our synchronous synthesis approach reduces the mean absolute errors on the face landmarks and the gesture poses by 30% and 21%, respectively compared to the corresponding baseline talking head and co-speech gesture syntheses models (Yoon et al. 2019; Ginosar et al. 2019; Ahuja and Morency 2019; Yoon et al. 2020; Bhattacharya et al. 2021a), thereby indicating measurable benefits over asynchronously combining the synthesized outputs of the two modalities. We also validate the benefits of synchronous synthesis qualitatively through a user study.
- **Using data from affordable commodity cameras.** In contrast to facial expression synthesis using dense 3D face scans or gesture synthesis from expensive motion-captured data, our approach only relies on face landmarks and pose joints obtainable from commodity hardware such as video cameras. As a result, our method scales affordably to large

datasets and is applicable in large-scale social VR applications such as virtual conferences.

- **Plausible virtual agent motions and proposed evaluation metric for facial expressions.** Through quantitative evaluations and user studies, we verify that our synthesized facial expressions and body gestures have low reconstruction errors and are satisfactory to human observers. Our synthesized data has a mean absolute error of 1.8% on the face landmarks and 0.9% on the poses relative to the scale of the data, and a Fréchet Gesture Distance of 1.79 on the poses. We also propose the Fréchet Landmark Distance to evaluate the quality of the synthesized landmarks and observe a value of 15.02. Based on a five-point Likert scale of responses, 88.89% participants in our user study indicated the quality of our gestures was at least satisfactory, and 62.87% participants indicated the face and pose expressions were in sync given the speech.
- **TED Gesture+Face Dataset.** We extend the TED Gesture Dataset to include 3D face landmarks that we extract from the raw videos, denoise, and correct such that the faces appear front and center and are aligned with the poses. We release this multimodal dataset of speech audio, 3D face landmarks, and 3D body pose joints as part of our work and the necessary source code to reuse it.

7.2 Related Work

We briefly review the body of work on perceiving affective expressions from multiple modalities, specifically from faces, speech, and gestures, and also the synthesis of virtual agents with co-speech face and pose expressions. The literature on synchronous co-speech face and pose synthesis is extremely sparse, however. We note the work of Habibie et al. 2021, who synthesize co-speech facial movements and upper-body gestures by training a CNN-based encoder-decoder

architecture coupled with a discriminator on a video dataset of talk-show hosts. In our work, we additionally consider affective expressions of the speech and explicitly map them to affective expressions on the faces and the gestures to improve the plausibility of our synthesis. We also note that there is a wider body of work on synthesizing standalone emotive virtual agents in the absence of any input modalities such as speech, and refer the reader to Bhattacharya et al. 2020 for a detailed discourse.

7.2.1 Perceiving Multimodal Affective Expressions

Studies in psychology and affective computing indicate that humans express emotions simultaneously through multiple modalities, including facial expressions, prosody and intonations of the voice, and body gestures (Soleymani, Pantic, and Pun 2012; Mittal et al. 2020a). Methods for detecting facial expressions (Giannopoulos, Perikos, and Hatzilygeroudis 2018) depend primarily on facial action units, based on FACS (Ekman and Friesen 1978). Method for detecting various affective vocal patterns commonly use Mel-Frequency Cepstral Coefficients (MFCCs) (Neiberg, Elenius, and Laskowski 2006). In order to detect emotions from body gestures, current approaches use physiological features, such as arm swings, spine posture, and head motions that are either pre-defined (Bhattacharya et al. 2020; Banerjee, Bhattacharya, and Bera 2022) or learned automatically from the gestures (Bhattacharya et al. 2020). The emotions themselves can be represented either as discrete categories such as the Ekman emotions (Ekman and Keltner 1970) or as combinations of continuous dimensions, such as the Valence-Arousal-Dominance (VAD) model (Mehrabian and Russell 1974). In our work, we leverage the current approaches for detecting facial, vocal, and pose expressions to design our co-speech face and gesture synthesis method. While we do not explicitly consider specific emotions, our representation implicitly

considers emotions in the continuous VAD space, leading to appropriately expressive face and pose synthesis.

7.2.2 Synthesizing Co-Speech Virtual Agent Expressions

Existing methods for co-speech virtual agent synthesis have largely explored synthesizing virtual agents with only one modality of emotional expressions, primarily faces and upper-body gestures.

Co-Speech Facial Expressions. Wang and Soong 2015 compute controllable parameters for synthesizing talking heads with desired facial expressions using a Hidden Markov Model and MFCCs of the speech audio. Recent techniques aim to automate the facial motions for large-scale synthesis, using generative paradigms such as variational autoencoders (Greenwood, Laycock, and Matthews 2017) and generative adversarial networks (Sadoughi and Busso 2018). Karras et al. 2017 train a deep neural network to map speech audio to 3D face vertices conditioned on learned latent features corresponding to different facial expressions. Zhou et al. 2018, on the other hand, learn sequences of predefined visemes using LSTM networks from the speech audio. Cudeiro et al. 2019 propose a dataset of 4D face scans and learn per-vertex offsets to synthesize the face motions given the speech audio. Richard et al. 2021 propose a method to learn co-speech facial motions using dense face meshes by disentangling speech-correlated and speech-uncorrelated facial features. Lahiri et al. 2021 focus mainly on the accuracy of the lip movements and use an auto-regressive learning-based approach to synthesize 3D vertex sequences for the lips that are synced with the speech audio. In contrast to these approaches, our facial expression synthesis method uses much sparser 3D face landmarks detected from real-world videos with arbitrary

orientations and lighting conditions of the faces w.r.t. the cameras, and synthesizes facial and pose expressions that are mutually coherent.

Co-Speech Gestures. We can consider co-speech gesture synthesis to be a specialized version of gesture stylization, where the style refers to the pose expressions that are inferred from and aligned with the speech. Ginosar et al. 2019 propose a method to synthesize speaker-specific co-speech gestures by training a neural network given their identities and individual gesticulation patterns. Ferstl, Neff, and McDonnell 2019 additionally propose using adversarial losses in the training process to improve the fidelity of the synthesized gestures. Yoon et al. 2020 extend the concept of individualized gestures to a continuous space of speakers to incorporate natural variability in the synthesized gestures even for the same speaker. Bhattacharya et al. 2021a build on top of Yoon et al. 2020 to improve the affective expressions in the co-speech gestures. Our method further extends the approach of Bhattacharya et al. 2021a to condition the gesture synthesis on both the input speech and the synthesized facial expressions.

7.3 Synchronous Co-Speech Face and Pose Synthesis

Given a speech audio waveform a, the corresponding text transcript w, the speaker's unique ID k in a set of speakers K, and the associated seed face landmark deltas $f_{1:T_s}$ and seed pose unit vectors $u_{1:T_s}$, T_s being the number of seed time steps, we synthesize the sequences of face landmark deltas $f_{1:T}$ and pose unit vectors $u_{1:T}$ for the speaker for the T prediction time steps ($T \gg T_s$), matching the content and the affect in their speech and appearing in sync. We now describe our end-to-end pipeline, including a detailed description of our inputs and outputs and their usage.

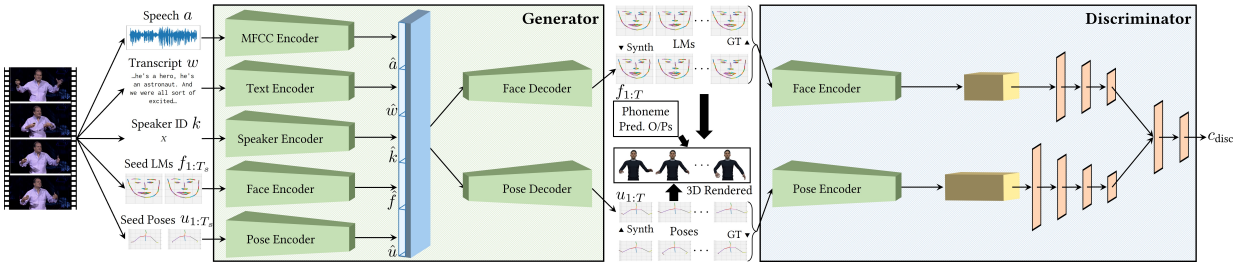
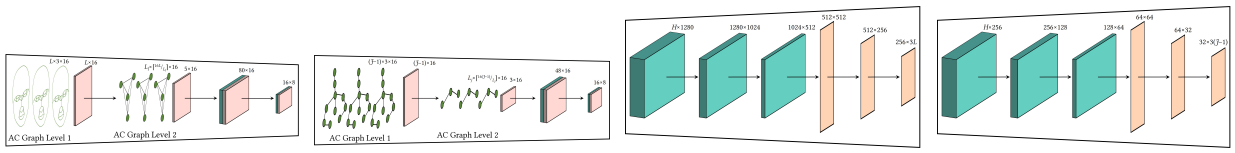


Figure 7.2: **Speech2UnifiedExpressions: Network Architecture for Synchronous Synthesis of Co-Speech Face and Pose Expressions.** Our generator encodes all the inputs: the speech audio, the corresponding test transcript, the speaker ID, and the seed 3D face landmarks and the seed 3D poses into a multimodal embedding space. It decodes variables from this space to produce the synchronized sequences of co-speech 3D face landmarks and poses. Our discriminator classifies these synthesized sequences and the corresponding ground-truths (3D motions of the original speakers), computed directly from the videos, into two different classes based both on their plausibility and their synchronous expressions. To obtain our rendered 3D virtual agent motions, we combine the outputs of our generator with our phoneme detector network and map them to 3D meshes.



(a) Face encoder (b) Pose Encoder (c) Face Decoder (d) Pose Decoder

Figure 7.3: **Speech2UnifiedExpressions: Face and Pose Encoders and Decoders.** We show their architectures with the layer sizes denoted (details in Sec. 7.3.3.3). Our architectures depend on the hierarchical anatomical component (AC) graphs for both faces and poses that efficiently learn their corresponding affect representations using spatial-temporal graph convolutions (green nodes and edges), 2D convolutions (teal blocks), 2D batch normalizations (pink blocks), and fully-connected layers (orange planes).

7.3.1 Face and Pose Preprocessing from Video

Given a video, we use Multi-Task Cascaded CNNs (Zhang et al. 2016a) to extract the 3D face landmarks. Since the faces can be arbitrarily oriented w.r.t. the camera, we eliminate the relative camera motion by rigidly transforming the face landmarks per frame to a reference frame in the normalized view, where the face looks towards the camera. We choose such a frontal view for the reference frame to maximize the likelihood of all the face landmarks being visible. For

each frame in the input video, we use the rotation and the translation given by the Umeyama method (Umeyama 1991) to map the face landmarks in that frame to the face landmarks in the reference frame. We also use similarly view-normalized 3D poses. View normalization is useful for two key reasons. First, it eliminates relative camera movements across the frames in the videos and prevents a learning-based method from confusing camera movements with changes in the face and pose expressions. Second, because a frontal view offers full and undistorted visibility of the faces and the poses, it minimizes errors in detecting the 3D face landmarks and body joints.

7.3.2 Computing Face and Pose Expressions

We consider a reference neutral expression $\mathcal{F} \in \mathbb{R}^{L\times 3}$ for each user, L being the number of face landmarks. To synthesize facial expressions, we compute the relative motion of each landmark w.r.t. the reference expression. Specifically, we obtain the configuration $\mathcal{F}_t$ at time step t as

$$\mathcal{F}_t = \mathcal{F} + f_t, \tag{7.1}$$

where $f_t \in \mathbb{R}^L$ denotes the set of relative motions of the landmarks w.r.t. $\mathcal{F}$ at time step t.

On the other hand, we assume the body joints are rigidly connected by the bones. We represent each user's body joints as 3D point vectors $\mathcal{P} \in \mathbb{R}^{\mathcal{J}\times 3}$ in a global coordinate space, where $\mathcal{J}$ is the number of joints. We consider directed line vectors connecting adjacent joints. The direction is along the path from the root (pelvis) joint to the end effectors (such as wrists). These 3D point vectors and line vectors collectively form a directed tree with $\mathcal{J}$ nodes and $\mathcal{J} - 1$ edges. We assume that the magnitudes of these line vectors correspond to the bone lengths

and that these magnitudes are known and fixed. To synthesize the users' body gestures, we compute the orientations of these line vectors at each time step t in the reference frame of the global coordinate space. Specifically, for each bone b with bone length (magnitude) $\|b\|$ and connecting the source joint $s_b(t)$ to the destination joint $d_b(t)$ at time step t, we compute a unit vector u_t such that

$$d_b = s_b + \frac{\|b\|}{\|u_t\|} u_t. \tag{7.2}$$

We do not assume any locomotion, *i.e.*, we consider the root joint is fixed at the global origin at all the time steps.

7.3.3 Synthesizing Faces and Poses

Our network architecture (Fig. 7.2) consists of a phoneme predictor to predict the lip shapes corresponding to the audio and a generator-discriminator pair to synthesize plausible co-speech face and pose expressions. Our generator follows a multimodal learning strategy. It consists of separate encoders to transform the speech audio, the text transcript, the speaker ID, the seed face landmark deltas, and the seed pose unit vectors into a latent embedding space representing their correlations. It subsequently synthesizes the appropriate face and pose motions from this multimodal embedding space. Our discriminator enforces our generator to synthesize plausible face and pose motions in terms of their affective expressions. To this end, we use the same encoder architecture for the faces and the poses as in our generator, but learned separately. We describe each of the components of our generator and discriminator.

7.3.3.1 *Phoneme Predictor*

We train a separate network to learn the positions of the lip landmarks for the different phonemes in the audio. Our synthesis network separately learns the motions of the lip corners denoting the different facial expressions, and we superpose them to the phoneme-based lip shapes to complete the lip motions. Our phoneme predictor predicts the 3D positions of all the landmarks on the inner and the boundaries of the lips over all the T prediction time steps, which we denote as $p_{1:T}\ in \mathbb{R}^{T \times L_{\text{lip}} \times 3}$. Following prior approaches for synthesizing 3D lip motions (Lahiri et al. 2021), we design a CNN backbone connected to fully-connected blocks to predict the lip landmarks from the spectrograms of the speech inputs. Specifically, given the speech audio waveform a, we compute

$$p_{1:T} = \text{PhonemePred}\left(a; \theta_{\text{PhonemePred}}\right), \tag{7.3}$$

where $\theta_{\text{PhonemePred}}$ represents the trainable parameters.

7.3.3.2 *Encoding Speech, Text, and Speaker IDs*

We use the Mel-Frequency Cepstral Coefficients (MFCCs) for the speech audio to accurately capture the affective intonations in the speech, and use an MFCC encoder (Bhattacharya et al. 2021a) to obtain speech-based latent embeddings $\hat{a} \in \mathbb{R}^{T \times D_a}$ of dimension D_a as

$$\hat{a} = \text{MFCCEncoder}\left(a; \theta_{\text{MFCC}}\right), \tag{7.4}$$

where θ_{MFCC} represents the trainable parameters.

Similarly, we use the sentiment-aware FastText (Santos, Nedjah, and Macedo Mourelle 2017) embeddings of the words in the transcript and a convolution-based text encoder to obtain the text-based latent embeddings $\hat{w} \in \mathbb{R}^{T \times D_w}$ of dimensions D_w as

$$\hat{w} = \text{TextEncoder}\left(w; \theta_{\text{text}}\right), \tag{7.5}$$

where θ_{text} represents the trainable parameters.

We also represent the speaker IDs $k \in \{0, 1\}^K$ as one-hot vectors for a total of K speakers and use a speaker encoder to obtain the parameters $\mu_k \in \mathbb{R}^{D_k}$ and $\Sigma_k \in \mathbb{R}_+^{D_k \times D_k}$ of a latent distribution space of dimension D_k as

$$\mu_k, \Sigma_k = \text{SpeakerEncoder}\left(k; \theta_{\text{speaker}}\right), \tag{7.6}$$

where θ_{speaker} represents the trainable parameters. The latent distribution space enables us to sample a random vector $\hat{k}$ representing a speaker who is an arbitrary combination of the K speakers in the dataset. This allows for variations in the synthesized motions even for the same original speaker by slightly perturbing their speaker IDs in the latent distribution space, leading to more plausible results on multiple runs of our network. To learn faces and poses with appropriate expressions, we represent them as multi-scale graphs and encode them using graph convolutional networks.

7.3.3.3 *Encoding Affective Expressions*

The face landmarks we use are based on action units developed from the facial action coding system (FACS). We represent the sequence of 3D landmarks $f_{1:T_s} \in \mathbb{R}^{T_s \times L \times 3}$ as a spatial-temporal graph. Spatially, we consider landmarks belonging to the same anatomical component (Sec. 7.3.2) and nearest landmarks across different anatomical components to be adjacent. Temporally, all landmarks are adjacent to their temporal counterparts (same nodes at different time steps) within a predetermined time window. We consider the eyes, the nose, the lips, and the lower jaw as the anatomical components. We show the face landmarks graph in Fig. 7.3a with all the intra- and inter-anatomical-component adjacencies marked with lines. We apply a sequence of spatial-temporal graph convolutions on this graph to learn from the localized motions of the landmarks and obtain embeddings $\tilde{f} \in \mathbb{R}^{T_s \times L \times D_f}$ of feature dimension D_f as

$$\tilde{f} = \text{STGCN}_f \left(f_{1:T_s}; \theta_{\text{STGCN}_f} \right), \tag{7.7}$$

where θ_{STGCN_f} represents the trainable parameters. From the landmarks graph, we obtain a face anatomy graph, where we consider the nodes to represent entire anatomical components and the graph to be fully connected. To compute such a graph, we append the features of intra-anatomical-component nodes in the graph into collated features $l \in \mathbb{R}^{T_s \times L_l \times n_l D_f}$, where L_l denotes the number of anatomical components and n_l denotes the number of landmark nodes within each anatomical component. We take n_l to be the number of nodes in the anatomical component with the most landmarks and perform zero padding as appropriate to obtain the full collated features for the other components. This hierarchically pooled representation provides

a "higher-level" view of the face and helps our network learn from the correlations between the motions of the different anatomical components. Specifically, we use another set of spatial-temporal graph convolutions to obtain the embeddings $\tilde{l} \in \mathbb{R}^{T_s \times L_l \times D_l}$ of feature dimension D_l as

$$\tilde{l} = \text{STGCN}_l \left(l; \theta_{\text{STGCN}_l} \right), \tag{7.8}$$

where θ_{STGCN_l} represents the trainable parameters. Collectively, the landmarks graph and the face anatomy graph provide complementary information to our network to encode and synthesize the required facial expressions at both the macro (anatomy) and the micro (landmark) levels. To complete our encoding, we flatten out the features of all the anatomical components in $\tilde{l}$, *i.e.*, reshaping such that $\tilde{l} \in \mathbb{R}^{T_s \times L_l D_l}$, and transform them using standard convolutional layers on the flattened feature channel and the temporal channel separately. This gives us our latent space embeddings $\hat{l} \in \mathbb{R}^{T \times D_{\tilde{l}}}$ as

$$\hat{l} = \text{ConvT}_{\tilde{l}} \left(\text{ConvS}_{\tilde{l}} \left(\tilde{l}; \theta_{\text{ConvS}_{\tilde{l}}} \right); \theta_{\text{ConvT}_{\tilde{l}}} \right), \tag{7.9}$$

where $\theta_{\text{ConvS}_{\tilde{l}}}$ and $\theta_{\text{ConvT}_{\tilde{l}}}$ represent the trainable parameters.

For the pose representation, we consider a pose graph of the upper body with $\mathcal{J} - 1$ bones represented with line vectors $u_{1:T_s}$ (Fig. 7.3b). We consider bones connected to each other or connected through a third bone to be adjacent. We use a set of spatial-temporal graph convolutions to leverage the localized motions of these bones and obtain embeddings $\tilde{u} \in \mathbb{R}^{T_s \times D_u}$ of feature dimension D_u as

$$\tilde{u} = \text{STGCN}_u \left(u_{1:T_s}; \theta_{\text{STGCN}_u} \right), \tag{7.10}$$

where θ_{STGCN_u} represents the trainable parameters. Similar to the face landmarks, we also consider a hierarchically pooled representation of the bones $v \in \mathbb{R}^{T_s \times L_j \times n_j D_u}$, where $L_j = 3$ are the three anatomical components, the torso and the two arms, represented as single nodes each consisting of n_j nodes from the pose graph. In the pose anatomy graph, we consider the two arms to be adjacent to the torso but not to each other, as they can move independently. We apply a second set of spatial-temporal graph convolutions on the collated features v to obtain the embeddings $\tilde{v} \in \mathbb{R}^{T_s \times L_j \times D_v}$ as

$$\tilde{v} = \text{STGCN}_v \left(v; \theta_{\text{STGCN}_v} \right) \tag{7.11}$$

where θ_{STGCN_v} represents the trainable parameters. To subsequently obtain the latent space embeddings $\hat{v} \in \mathbb{R}^{T \times D_{\tilde{v}}}$, we apply separate spatial and temporal convolutions on the flattened graph-convolved features $\tilde{v} \in \mathbb{R}^{T_s \times L_j D_v}$, as

$$\hat{v} = \text{ConvT}_{\tilde{v}} \left(\text{ConvS}_{\tilde{v}} \left(\tilde{v}; \theta_{\text{ConvS}_{\tilde{v}}} \right); \theta_{\text{ConvT}_{\tilde{v}}} \right), \tag{7.12}$$

where $\theta_{\text{ConvS}_{\tilde{v}}}$ and $\theta_{\text{ConvT}_{\tilde{v}}}$ represent the trainable parameters.

7.3.3.4 Synthesizing Synchronous Face and Pose Motions

Our synchronous synthesis relies on learning the multimodal distributions of the individual modalities of audio, text, speaker ID, face expressions, and pose expressions given their individual distributions. To this end, we append all the latent space embeddings — $\hat{a}$ for the audio, $\hat{w}$ for the text, $\hat{k}$ for the random speaker representation, repeated over all the T time steps, $\hat{l}$ for the

seed landmarks and $\hat{v}$ for the seed poses — into a vector $\hat{e} \in \mathbb{R}^{T \times H}$ representing a multimodal embedding space of all the inputs. Here, $H = D_a + D_w + D_k + D_{\tilde{l}} + D_{\tilde{v}}$ denotes the latent space dimension. On training, our network learns the correlations between the different inputs in this multimodal embedding space. To synthesize our face landmark motions $f_{1:T} \in \mathbb{R}^{T \times L \times 3}$, we apply separate spatial and temporal convolutions on the multimodal embeddings $\hat{e}$ to capture localized dependencies between the feature values followed by fully-connected layers capturing all the dependencies between the feature values (Fig. 7.3c), as

$$f_{1:T} = \mathrm{FC}_{f\hat{e}} \left(\mathrm{ConvS}_{f\hat{e}} \left(\mathrm{ConvT}_{f\hat{e}} \left(\hat{e}; \theta_{\mathrm{ConvT}_{f\hat{e}}} \right); \theta_{\mathrm{ConvS}_{f\hat{e}}} \right); \theta_{\mathrm{FC}_{f\hat{e}}} \right), \tag{7.13}$$

where $\theta_{\mathrm{ConvT}_{f\hat{e}}}$, $\theta_{\mathrm{ConvS}_{f\hat{e}}}$, and $\theta_{\mathrm{FC}_{f\hat{e}}}$ represent the trainable parameters. The output $f_{1:T}$ from the fully-connected layers has shape $T \times 3L$, which we reshape into $T \times L \times 3$ to get our desired 3D face landmark sequences.

We similarly synthesize the line vectors $u_{1:T} \in \mathbb{R}^{T \times (\mathcal{J}-1) \times 3}$ using separate spatial and temporal convolutions on the multimodal embeddings $\hat{e}$, followed by fully-connected layers (Fig. 7.3d), as

$$u_{1:T} = \mathrm{FC}_{u\hat{e}} \left(\mathrm{ConvS}_{u\hat{e}} \left(\mathrm{ConvT}_{u\hat{e}} \left(\hat{e}; \theta_{\mathrm{ConvT}_{u\hat{e}}} \right); \theta_{\mathrm{ConvS}_{u\hat{e}}} \right); \theta_{\mathrm{FC}_{u\hat{e}}} \right), \tag{7.14}$$

where $\theta_{\mathrm{ConvT}_{u\hat{e}}}$, $\theta_{\mathrm{ConvS}_{u\hat{e}}}$, and $\theta_{\mathrm{FC}_{u\hat{e}}}$ represent the trainable parameters. Given the synthesized face and pose motions, we use our discriminator to determine how well their affective expressions match that of the corresponding ground-truths in the training data. We obtain our ground-truths as the 3D face landmarks and the 3D pose sequences computed from the full training

video data.

7.3.3.5 Determining Plausibility Using Discriminator

Our discriminator takes in the synchronously synthesized face motions $f_{1:T}$ and pose motions $u_{1:T}$, and encodes them using encoders with the same architecture as our generator (Sec. 7.3.3.3), with only the number of input time steps being T instead of T_s. This gives us the corresponding latent space embeddings $\hat{l}$ and $\hat{v}$. Similar to our generator, we concatenate these embeddings into a multimodal embedding vector $\hat{e} \in \mathbb{R}^{T \times \left(D_{\hat{l}} + D_{\hat{v}}\right)}$. But different from our generator, we pass these multimodal embeddings through a fully-connected classifier network FC_{disc} to obtain class probabilities $c_{\text{disc}} \in [0, 1]$ per sample, as

$$c_{\text{disc}} = \text{FC}_{\text{disc}}\left(\hat{e}; \theta_{\text{FC}_{\text{disc}}}\right), \tag{7.15}$$

where $\theta_{\text{FC}_{\text{disc}}}$ represents the trainable parameters. We train our discriminator to perform unweighted binary classification between the synthesized face and pose motions and their corresponding ground-truths. By contrast, our generator attempts to make our discriminator predict the same class for both the synthesized and the ground-truth samples, *i.e.*, synthesizing them to be as close as possible to the ground-truth samples in terms of their synchronous affective expressions. We provide the details of our training loss functions and our overall training and testing procedures in Sec. 7.5.

7.4 TED Gesture+Face Dataset

We present our TED Gesture+Face Dataset that we use to train and test our network. We elaborate on collecting and processing our dataset for training and testing.

Dataset Collection. The original TED Gesture Dataset (Yoon et al. 2019) consists of videos of TED talk speakers together with text transcripts of their speeches, and their 3D body poses extracted in a global frame of reference. The topics range from personal and professional experiences to discourses on educational topics and instructional and motivational storytelling. The speakers themselves come from a wide variety of social, cultural, and economic backgrounds, and are diverse in age, gender, and physical abilities.

Dataset Processing. The 3D poses in the original TED Gesture Dataset (Yoon et al. 2019) are view-normalized to face front and center at all time steps. We compute similarly view-normalized 3D face landmarks of the speakers (Sec. 7.3.1). Similar to the original TED dataset, we divide the 3D pose and face landmark sequences into equally-sized chunks of size $T = 34$ time steps at a rate of 15 fps. Additionally, to reduce the jitter in the predicted 3D face landmarks and pose joints from each video, we sample a set of "anchor" frames at a rate of 5 fps and perform bicubic interpolation to compute the face landmark and pose joint values in the remaining frames. We use the first 4 time steps of pose and face landmarks as our seed values (Sec. 7.3.3), and predict the next 30 time steps. The processed dataset consists of 200,038 training samples, 26,903 validation samples, and 26,245 test samples, following a split of 80%-10%-10%.

7.5 Training, Testing, and VR Interfacing

We train our phoneme predictor network using reconstruction losses for the lip shapes. We train our synthesis network using a combination of reconstruction losses for the face and the pose motions, the cross-speaker diversity loss to enforce visual differences in expressions across speakers, and the generative adversarial loss for added regularization. We describe these loss functions, and our training and testing procedures.

7.5.1 Phoneme Predictor Losses

We represent our phoneme predictor loss as the robust ℓ_1-norm reconstruction loss between the ground-truth and the synthesized lip landmark positions and velocities over the prediction time steps T, as

$$\mathcal{L}_{\text{ph}} = \sum_{t=1}^{T} \left\| p_t^{(\text{GT})} - p_t^{(\text{sn})} \right\|_1 + \left\| \Delta_t p_t^{(\text{GT})} - \Delta_t p_t^{(\text{sn})} \right\|_1, \tag{7.16}$$

where the superscripts (GT) and (sn), respectively, denote the ground-truth and the synthesized data. Δ_t denotes the discrete forward difference between adjacent time steps t and $t-1$.

7.5.2 Synchronous Synthesis Network Losses

We use reconstruction losses to robustly align the outputs of our generator with the corresponding ground-truth face and pose motions. We use the generative adversarial loss to ensure that the synthesized motions are plausible, the affective expressions match the corresponding ground-truths, and prevent the mode collapse of only synthesizing singular expressions.

Table 7.1: **Speech2UnifiedExpressions: Quantitative Evaluations.** Comparison with existing co-speech gesture synthesis methods and our ablated versions (Sec. 7.6.1) on the metrics MALE (in mm), MAJE (in mm), MAcE for landmarks (MAcE-LM) (in mm/s^2), MAcE for poses (MAcE-P) (in mm/s^2), FLD, and FGD (Sec. 7.6.2). Lower values are better, bold indicates **best**, and underline indicates second-best.

Method	MALE	MAJE	MAcE-LM	MAcE-P	FLD	FGD
Seq2Seq (Yoon et al. 2019)	–	45.62	–	6.33	–	6.62
S2G-IS (Ginosar et al. 2019)	–	45.11	–	7.22	–	6.73
JEM (Ahuja and Morency 2019)	–	48.56	–	4.31	–	5.88
GTC (Yoon et al. 2020)	–	27.30	–	3.20	–	4.49
Speech2AffectiveGestures (Bhattacharya et al. 2021a)	–	24.49	–	2.93	–	3.54
Ours w/o Face Synthesis	–	28.32	–	3.89	–	4.01
Ours w/o Pose Synthesis	11.76	–	9.38	–	22.65	–
Ours w/o Vel.+Acc. Losses	26.33	24.41	21.69	7.58	27.54	7.72
Ours w/o Discriminator	14.62	27.40	13.44	11.60	31.93	8.79
Ours w/o Face AC Graph	13.05	25.97	14.24	2.74	25.61	2.25
Ours w/o Pose AC Graph	11.84	25.46	8.12	13.88	19.23	6.94
Ours w/o Synchronous Synthesis	10.72	25.03	7.83	3.22	18.03	3.92
Ours	**9.00**	**18.36**	**6.34**	**2.52**	**15.02**	**1.79**

7.5.2.1 Reconstruction Losses

We write our reconstruction losses as the ℓ_1-norm difference between the ground-truth and the synthesized face and pose positions and motions over the T prediction time steps, as

$$\begin{aligned}\mathcal{L}_{\text{Rec}} = &\sum_{t=1}^{T} \left\| \mathcal{F}_t^{(\text{GT})} - \mathcal{F}_t^{(\text{sn})} \right\|_1 + \left\| \mathcal{P}_t^{(\text{GT})} - \mathcal{P}_t^{(\text{sn})} \right\|_1 \\ &+ \lambda_{\text{vel}} \left(\left\| f_t^{(\text{GT})} - f_t^{(\text{sn})} \right\|_1 + \left\| u_t^{(\text{GT})} - u_t^{(\text{sn})} \right\|_1 \right) \\ &+ \lambda_{\text{acc}} \left(\left\| \Delta_t f_t^{(\text{GT})} - \Delta_t f_t^{(\text{sn})} \right\|_1 + \left\| \Delta_t u_t^{(\text{GT})} - \Delta_t u_t^{(\text{sn})} \right\|_1 \right), \end{aligned} \tag{7.17}$$

where λ_{vel} and λ_{acc} are the relative weighting factors. We use the velocity and the acceleration losses to enforce smoothness in the synthesized motions by reducing jitters.

7.5.2.2 *Cross-Speaker Diversity Loss*

Our cross-speaker diversity loss $\mathcal{L}_{\text{CSD}}$ follows that of (Yoon et al. 2020), consisting of a ranking loss between the ground-truth face and pose motions, and the synthesized face and pose motions using the same speaker as the ground-truth (positive example) and a randomly chosen different speaker (negative example).

7.5.2.3 *Generative Adversarial Loss*

The generative adversarial loss consists of opposing losses $\mathcal{L}_{\text{Gen}}$ for the generator and $\mathcal{L}_{\text{Dis}}$ for the discriminator, following a min-max optimization strategy (Goodfellow et al. 2014). We write these losses as

$$\mathcal{L}_{Gen} = -\mathbb{E}\left[\log\left(c_{\text{disc}}^{\text{GT}}\right)\right], \tag{7.18}$$

$$\mathcal{L}_{Dis} = -\mathbb{E}\left[\log\left(c_{\text{disc}}^{\text{GT}}\right)\right] - \mathbb{E}\left[\log\left(1 - c_{\text{disc}}^{\text{sn}}\right)\right], \tag{7.19}$$

where c_{disc} denotes the output of our discriminator network (Eq. 15). This loss adds plausibility to our synthesized samples by enforcing them to have similar affective expressions as the corresponding ground-truth samples.

7.5.3 Training Procedure

We train our phoneme predictor network using the Adam optimizer (Kingma and Ba 2014) with $\beta_1 = 0.5$, $\beta_2 = 0.999$, a batch size of 1024, and a learning rate of 10^{-3} for 500 epochs. We train our synthesis network using the Adam optimizer (Kingma and Ba 2014) with $\beta_1 = 0.5$, $\beta_2 = 0.999$, a batch size of 256, and learning rates of 10^{-4} for our generator and 5×10^{-5} for

Figure 7.4: **Speech2UnifiedExpressions: Qualitative Results.** We show snapshots from two of our synthesized samples, showing the text transcript of the speech and the corresponding face and pose expressions (row 1). We also zoom in on the eyebrow (row 2) and lip (row 3) expressions for better visualization. We observe a smile, raised eyebrows, and stretched arms (left) for the word 'excited'. By contrast, we observe frowns on the eyebrows and lips (right) for the words 'very sorry'.

our discriminator, both decayed by a factor of 0.999 per epoch, for 1000 epochs. We train both our phoneme detector network and our synthesis network on an NVIDIA GeForce RTX 2080 Ti GPU, which takes 3 seconds and 7 seconds per epoch respectively.

7.5.4 Testing Procedure and Mapping to Virtual Agents

Each test sample for our network consists of a speech audio waveform, the corresponding text transcript, a speaker ID, and the speaker's seed face and pose motions. Our phoneme predictor network provides the lip sync for the given speech audio and the generator of our synthesis network provides the required face and pose motions. We superpose the lip landmarks given by our phoneme predictor network with the lip corner landmarks given by our generator at each prediction time step to obtain the complete lip motions of the speaker. We map these motions to a rigged 3D human upper-body mesh in Blender. For mapping the face motions, we set a one-to-one mapping between our face landmarks and the landmarks on the face of the human mesh, and use them as control points for the facial motions of the mesh. For mapping the pose

motions, we use FABRIK (Aristidou and Lasenby 2011) to obtain the joint rotations given our predicted joint positions and use those rotations to animate the rigged human mesh.

7.5.5 Interfacing with the VR Environment

Given an input speech audio, we can synthesize the motions for our pre-rigged virtual agents at an interactive rate of about 250 frames per second on an NVIDIA GeForce RTX 2080 Ti GPU. We design our VR environment using Blender. For each of our virtual agents, we place them on a stage and position the camera such that it looks front and center at the agent. As the virtual agent narrates the input speech audio using our synthesized face and upper-body expressions, we slowly pan the camera in to get a more focused view of those expressions. Since we do not synthesize any lower-body motions, our virtual agents stay standing at their initial positions during the entire narration. The full video demos are available in our project website.

7.6 Experiments and Results

We run quantitative experiments using ablated versions of our approach as baselines. However, we are unable to report a direct comparison with Habibie et al. 2021 as their code is currently unavailable.

7.6.1 Baselines

We use seven ablated versions of our approach as baselines. The first two ablations correspondingly remove the entire face (Figs. 7.3a,7.3c) and pose components (Figs. 7.3b,7.3d) from our network, leading to our network learning only talking head and only co-speech gesture syntheses. The third ablation removes the velocity and the acceleration losses from our reconstruction loss

(Eqn. (7.17)), leading to jittery motions. The fourth ablation removes the discriminator and its associated losses (Eqn. (7.19)) from our training pipeline, leading to less stable motions without appreciable expressions. The fifth and the sixth ablations correspondingly remove the "higher-level" anatomical component (AC) graphs of the faces (Eqn. (7.8)) and the poses (Eqn. (7.11)), leading to reduced movements. The final ablation trains the face and the pose expressions separately, learning marginal embeddings for the two modalities based on the speech but not attending to their mutual synchronization. This ablation is a direct evaluation of the co-speech motions when combining separately synthesized face and pose expressions. For completeness, we also compare with co-speech gesture synthesis methods that use similar upper-body pose representations as ours. We evaluate all the methods on our TED Gesture+Face Dataset.

7.6.2 Evaluation Metrics

Inspired by the prior work of Yoon et al. 2020, we evaluate using four *reconstruction errors* and two *plausibility errors* (PEs). Our reconstruction errors include the mean absolute landmark error (MALE) for the faces, the mean absolute joint error (MAJE) for the poses, and their respective mean acceleration errors (MAcEs). MALE and MAJE indicate the overall fidelity of the synthesized samples w.r.t. the corresponding ground-truths, and the MAcEs indicate whether or not the synthesized landmarks and poses have regressed to their mean absolute positions. To report these metrics, we multiply our ground-truth and synthesized samples by a constant scaling factor such that they all lie inside a bounding box of diagonal length 1 m. For our PE, we use the Fréchet Gesture Distance (FGD) designed by Yoon et al. 2020 to indicate the perceived plausibility of the synthesized poses. To similarly indicate the perceived plausibility of the synthesized face landmarks, we also design the Fréchet Landmark Distance (FLD). We train

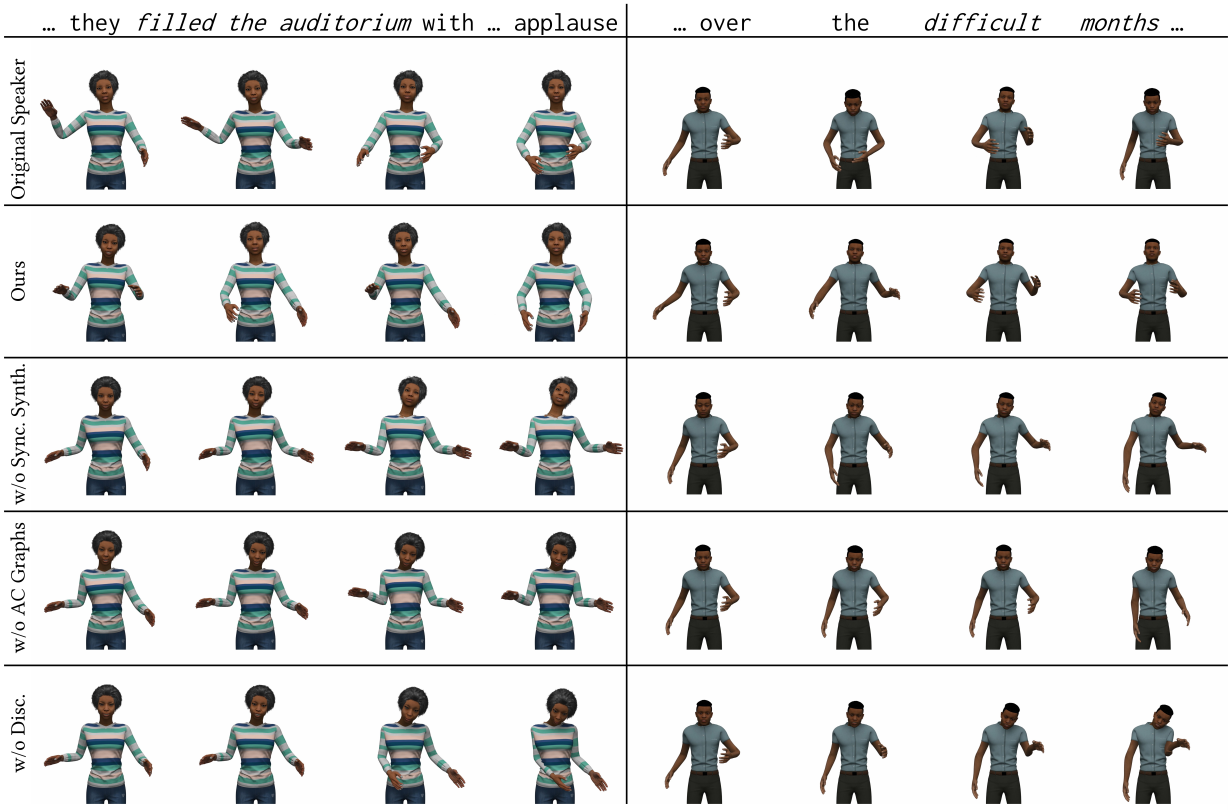


Figure 7.5: **Speech2UnifiedExpressions: Qualitative Comparisons.** For the same input speech, represented by the text transcript at the top, we compare the visual quality of our synthesized virtual agent motions with the original speaker motions and three of our ablated versions: one without synchronous face and pose synthesis, one without our anatomical component (AC) graphs for faces and poses, and one without our discriminator. We observe that our synthesized motions are visually the closest to the original speaker motions compared to the ablated versions. We elaborate on their visual qualities in Sec. 7.6.4.

an autoencoder network to reconstruct the full set of face landmarks at all time steps for all the samples in the training set of our TED Gesture+Face Dataset. To compute FLD, we then obtain the Fréchet Inception Distance (Heusel et al. 2017) between the encoded features of the ground-truth and the synthesized samples.

7.6.3 Quantitative Evaluations

We show our quantitative evaluations in Table 7.1.

Comparison with Co-Speech Gesture Synthesis. Since co-speech gesture synthesis methods do not synthesize face expressions, we leave their corresponding numbers blank. For these methods, we have taken the numbers reported by Bhattacharya et al. 2021a. However, we were unable to perform similar comparative evaluations with co-speech face synthesis methods as existing methods synthesize dense landmarks (Karras et al. 2017) or blendshape-like features (Cudeiro et al. 2019), which cannot be mapped one-to-one with our sparser face landmarks.

Comparison with Ablated Versions Removing either the face or the gesture components of our network leads to poorer values across the board compared to using both of them. Without the velocity and acceleration losses, the motions are jittery, and the MAcE losses are higher, especially MAcE for the face landmarks. Without the discriminator, the synthesized samples suffer from mode collapse and often produce implausible motions, leading to higher values across the board. Without the AC graphs, there are fewer movements in the synthesized and the reconstruction errors are higher. When synthesizing face and pose expressions separately and not synchronizing them, we observe some mismatches in when the expressions from either modality appear and how intense they are. This indicates that synchronous synthesis of facial expressions and body gestures leads to more accurate and plausible movements for both the modalities, including a 30% improvement on MALE and a 21% improvement on MAJE, compared to trivially combining synthesized outputs of the individual modalities.

7.6.4 Qualitative Comparisons

We visualize a few of our synthesized samples in Fig. 7.4, where we can observe the synchronization between the face and the pose expressions for two contrasting emotions. We also visually

Table 7.2: **Speech2UnifiedExpressions: Likert-Scale Score Statistics from User Study.** We compute the mean and the standard deviation of the Likert-scale scores across all the motions. For the mean scores, higher values are better, bold indicates **best**, and underline indicates second-best.

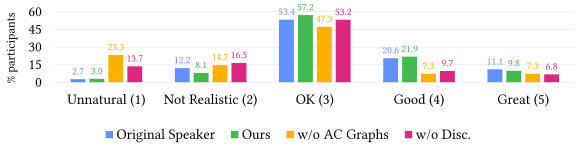
	Synthesis type	Plausibility		Synchronization	
		Mean	St. Dev.	Mean	St. Dev.
Set 1	Original Speaker	3.25	0.90	3.10	1.34
	Ours	**3.27**	0.86	**3.15**	1.32
	w/o AC Graphs	2.61	1.14	2.48	1.38
	w/o Disc.	2.79	1.02	2.02	1.30
Set 2	Original Speaker	2.99	0.80	**2.79**	1.08
	Ours	**3.01**	0.82	**2.79**	1.07
	w/o Synchronous Synthesis	2.41	0.78	1.79	0.88

compare with the original speaker motions rendered using their face landmarks and the poses extracted from the videos, and three of our ablated versions in Fig. 7.5. The original speaker motions provide an "upper bound" of our performance. The three ablated versions we compare with are: one without the synchronous synthesis, one without our face and pose AC graphs, and one without our discriminator. The ablated versions without either the face or the pose synthesis, without the velocity and acceleration losses, and without our discriminator are visually inferior in obvious ways. Without either face or pose synthesis, that modality remains static while the other one moves. Without the velocity and the acceleration losses, the overall motions regress to the mean pose. Without our discriminator, our generator often fails to understand plausible movement patterns, leading to unnatural limb and body shapes. Of these, we only keep the ablations without our discriminator as our "lower bound" baseline because, unlike the other two, this ablation has visible movements in both the face and the pose modalities.

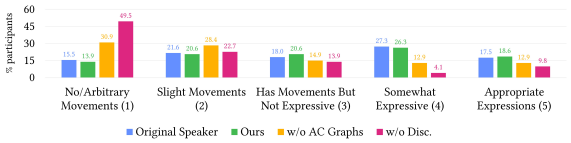
7.7 User Study

We conducted a web-based user study to evaluate the visual quality of our synthesized motions in terms of their plausibility and synchronization.

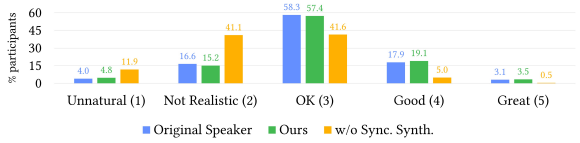
Setup. A total of 90 participants participated in our user study. All participants were aged 18 years or older, and had normal or corrected to normal vision and hearing. Each participant observed two sets of virtual agent motions. There were eight groups of motions in each set, each group having a unique input speech. In the first set, there were four types of motions in each group corresponding to the same speech: the original speaker motions rendered using their face landmarks and the poses extracted from the video, and motions rendered using the face landmarks and poses synthesized by our network and two of its ablated versions. One ablated version was without using the face and pose anatomical component (AC) graphs for training, and one without our discriminator. In the second set, there were three types of motions in each group corresponding to the same speech: the original speaker motions, motions rendered using the face landmarks and poses synthesized by our network, and the ablated version using asynchronously synthesized faces and poses. Our motivation to separately compare with the asynchronously synthesized motions was to eliminate distractors from other motions and enable our participants to focus more closely on the synchronization between the face and the pose expressions. We randomized the order of these motions in each group in each set and kept the order unknown to the participants. We did not present our other ablated versions to the participants as they did not have sufficient motion and were visually inferior in obvious ways.



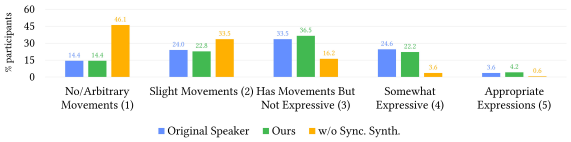
(a) **Set 1: Motion plausibility.** Compared to the ablated versions, we observe a higher distribution of "OK" or better for the motions of the original speakers and our synthesized agents. The modes of all the distributions are on "OK", implying that the corresponding participants found the visual qualities of all the motions to be reasonable.



(b) **Set 1: Synchronization between the face and the pose expressions given the speech.** Compared to the ablated versions, we obverse clear preferences for the motions of the original speakers and our synthesized agents. The modes of the distributions for these two types of motions are on "somewhat expressive" while the modes of the two ablated versions are on "no/arbitrary movements".



(c) **Set 2: Motion plausibility.** Compared to the ablated version without synchronous synthesis, we observe a higher distribution of "OK" or better for the motions of the original speakers and our synthesized agents. Similar to the motion plausibility in set 1, we observe modes of all the distributions on "OK".



(d) **Set 2: Synchronization between the face and the pose expressions given the speech.** We again obverse clear preferences for the motions of the original speakers and our synthesized agents compared to the ablated version without synchronous synthesis. However, in contrast to the same study in set 1, we notice the modes of the distributions for the first two types of motions are one point lower on the Likert scale, whereas the mode for the ablated version remains on "no/arbitrary movements". We hypothesize this to be the consequence of removing the other ablated versions from the participants' cognitive window: in the absence of other variants, the participants focused more closely on the relative qualities of asynchronous vs. synchronous motions and assessed them more critically.

Figure 7.6: **Speech2UnifiedExpressions: Synthesis Quality Scores from the User Study.** Likert-scale response distributions to the two sets of motions rendered using the five different types of face landmark and pose data (Sec. 7.7). We show the distributions of each of the five Likert-scale points for each type of motion as a percentage of the total responses across all the groups in each set.

Evaluation Process. Our aim in the user study is to evaluate our synthesized motions on two key aspects: *(i)* how plausible they appear to human observers compared to the motions of the original speakers and the ablated versions, and *(ii)* whether synchronous synthesis of face and

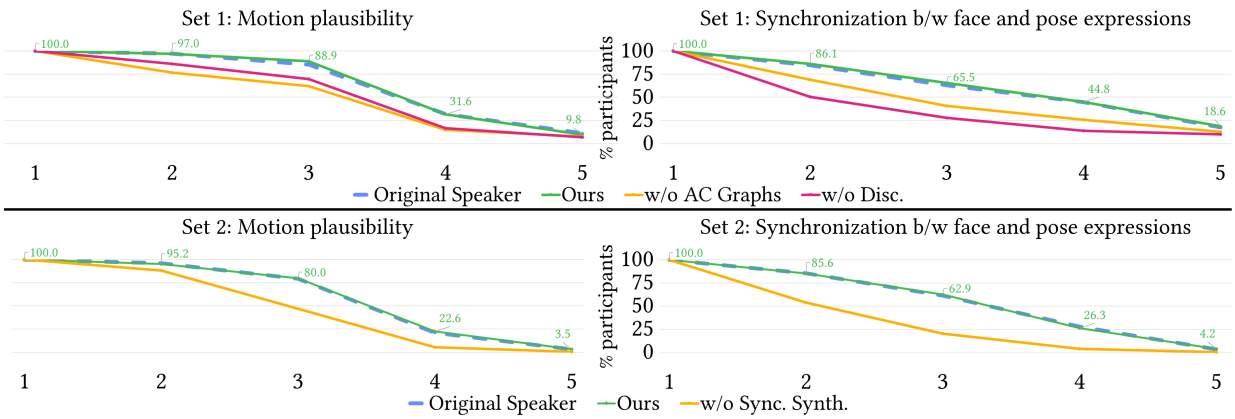


Figure 7.7: **Speech2UnifiedExpressions: Cumulative Lower-Bound of Participant Responses.** We plot the cumulative lower-bound (LB) percentage of responses across the Likert-scale scores for each type of virtual agent motion in each set. A cumulative LB percentage X for a Likert-scale score s denotes $X\%$ of responses had a score of s or higher. We observe that the curve for our synchronously synthesized motions stays at the top, indicating that the participants preferred it over the other motions.

pose expressions produces perceptible improvements over asynchronous synthesis. To evaluate plausibility, we ask the participants to rate each motion in each group in each set on "how natural the motion looks" on a five-point Likert scale, with the options "very unnatural" (worst), "not realistic", "looks OK" (average), "looks good", and "looks great" (best). To evaluate the effect of synchronous synthesis, we ask the participants to observe the face and the pose movements in each motion in each group in each set and rate them on "how the face and the pose sync with the speech" on a five-point Likert scale, with the options "no/arbitrary movements" (worst), "slight movements", "has movements, but are not expressive" (average), "somewhat expressive movements", and "have movements with appropriate expressions" (best).

Results. Since we randomly select the speech for each of the eight groups of motions each participant watched, and we also randomized the order of the motions in each group in each set, we can consider the participants' responses in each group to be independent of all the other

groups. Thus, we aggregate their responses to each type of motion across all the groups within a set to obtain the overall distributions of the Likert-scale scores of the motions for that set. We show these distributions for each of the two questions on plausibility and synchronization in each set in Fig. 7.6. We also report the Likert-scale score statistics for each type of motion on the two questions in each set in Table 7.2. For the purpose of scoring, we assign scores 1 through 5, with 1 for "worst" and 5 for "best". We observe that the scores for our synthesized samples are comparable to the corresponding original speaker motions and significantly better than the ablated versions. To further affirm this, we plot the cumulative lower bound of participant responses for each Likert-scale score for each type of motion in each set in Fig. 7.7. We note that the scores for our synchronously synthesized samples remain close to the original speaker scores and consistently above the other ablated versions, indicating a clear preference. Overall, in the two sets, 88.89% and 80.00% participants respectively marked our synchronously synthesized motions 3 or above on the first question, and 65.46% and 62.87% participants respectively marked 3 or above on the second question. This indicates that the majority of participants found the motions satisfactory.

7.8 Conclusion

We have presented a method to synthesize synchronous co-speech face and pose expressions for 3D virtual agents. Our method learns to synthesize these expressions from 3D face landmarks and 3D upper-body pose joints computed directly from videos. It is based on a generative adversarial neural network architecture consisting of a generator and a discriminator. Our generator encodes the individual speaker styles and the affective contents in the faces, poses, and speech into a multimodal embedding space and decodes variables from that space into synchronous face

and pose expressions. Our discriminator further improves the plausibility of our synthesized expressions by suppressing unnatural face and pose motions and ensuring that the synthesized expressions are similar to those of the original speakers. We evaluate our method quantitatively and qualitatively and show that it can synthesize plausible and diverse samples given the speech audio inputs. We also release an extended TED Gesture+Face Dataset to enable further research in this domain.

CHAPTER 8

Use Case 1: Detecting Highlights from Human-Centric Videos

Abstract

We present a domain- and user-preference-agnostic approach to detect highlightable excerpts from human-centric videos. Our method works on the graph-based representation of multiple observable human-centric modalities in the videos, such as poses and faces. We use an autoencoder network equipped with spatial-temporal graph convolutions to detect human activities and interactions based on these modalities. We train our network to map the activity- and interaction-based latent structural representations of the different modalities to per-frame highlight scores based on the representativeness of the frames. We use these scores to compute which frames to highlight and stitch contiguous frames to produce the excerpts. We train our network on the large-scale AVA-Kinetics action dataset and evaluate it on four benchmark

video highlight datasets: DSH, TVSum, PHD^2, and SumMe. We observe a 4–12% improvement in the mean average precision of matching the human-annotated highlights over state-of-the-art methods in these datasets, without requiring any user-provided preferences or dataset-specific fine-tuning.

8.1 Introduction

We now turn our attention to a use-case for our proposed methods for affect detection and synthesis from human body expressions in the area of video highlight detection. Specifically, we consider the problem of highlight detection from human-centric videos. Human-centric videos focus on human activities, tasks, and emotions (Zeng 2020; Vicol et al. 2018). These videos form a major part of the rapidly growing volume of online media (Cisco 2020), coming from multiple *domains*, such as amateur sports and performances, lectures, tutorials, video weblogs (vlogs), and individual or group activities, *e.g.*, cookouts and holiday trips. However, unedited human-centric videos also tend to contain large chunks of irrelevant and uninteresting content, requiring them to be edited for efficient browsing (Sun, Farhadi, and Seitz 2014).

To address this problem, researchers have developed multiple techniques for detecting highlightable excerpts and summarizing videos (Molino and Gygli 2018; Xiong et al. 2019; Rochan et al. 2020; Zhang et al. 2016b; Rochan, Ye, and Wang 2018; Zhou, Qiao, and Xiang 2018). Given unedited footage, highlight detection obtains the moments of interest, and summarization computes the most relevant and representative set of excerpts. Detecting effective highlights not only expedites browsing, but also improves the chances of those highlights being shared and recommended (Xiong et al. 2019). Current methods can learn to detect these excerpts given annotated highlights (Sun, Farhadi, and Seitz 2014; Molino and Gygli 2018), or

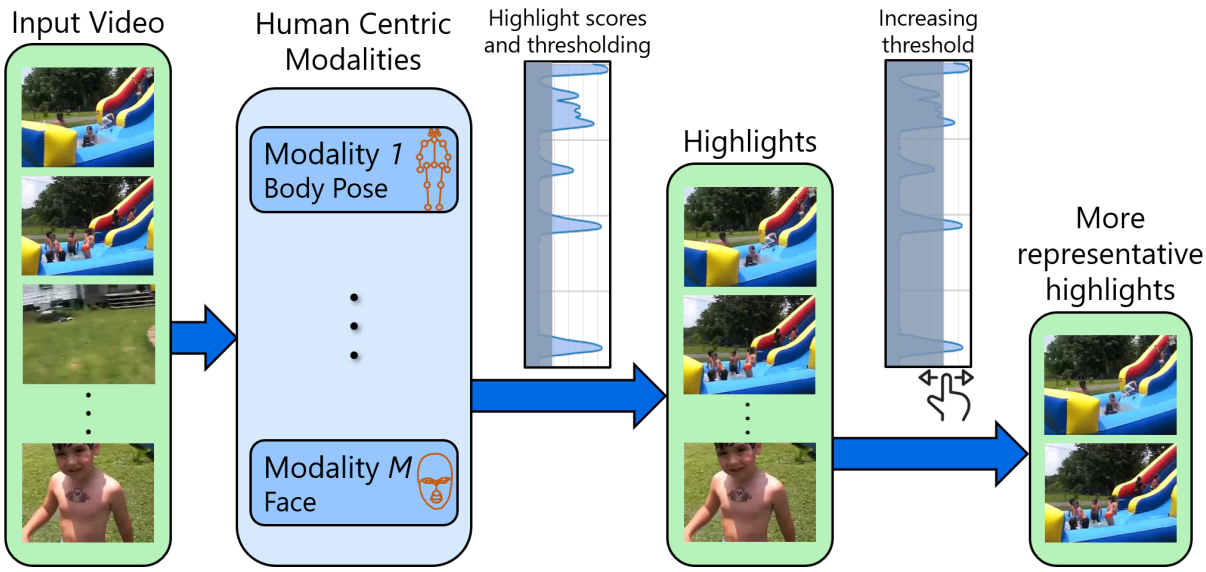


Figure 8.1: **HighlightMe: Overview.** Our method leverages multiple human-centric modalities, *e.g.*, body poses and faces, observable in videos focusing on human activities, to detect highlights. We use a 2D or 3D interconnected point representation of each modality to construct a spatial-temporal graph representation to compute the highlight scores.

sets of exemplars for different highlight categories, *e.g.*, learning from skiing images to detect skiing excerpts from videos (Kim, Sigal, and Xing 2014; Kim et al. 2018). Other methods obviate the need for supervision by learning the representativeness of each frame or shot with respect to the original video (Mahasseni, Lam, and Todorovic 2017) and exploiting video metadata such as duration (Xiong et al. 2019) and relevance of shots (Zhou, Qiao, and Xiang 2018; Zhang, Grauman, and Sha 2018). All these methods either assume or benefit from some domain-specific knowledge of the unedited footage, *e.g.*, running and jumping may be more relevant in a parkour video, whereas sliding maneuvers may be more relevant in a skiing video. Alternative methods do not consider domain-specific knowledge but consider the pre-recorded preferences of multiple users instead to detect personalized highlights (Rochan et al. 2020).

Whether they assume domain-specific knowledge or user-preferences, existing methods work in the 2D image space of the frames or shots constituting the videos. State-of-the-art

image-based networks can learn rich semantic features capturing the interrelations between the various detected objects in the images, leading to efficient highlight detection. However, these approaches do not explicitly model human activities or inter-person interactions that are the primary focus of human-centric videos. Developing methods for human-centric videos, meanwhile, has been essential for a variety of tasks, including expression and emotion recognition (Li and Deng 2020; Bhattacharya et al. 2020; Mittal et al. 2020a), activity recognition (Yan, Xiong, and Lin 2018), scene understanding (Vicol et al. 2018; Li et al. 2020), crowd analysis (Wang et al. 2020), video super-resolution (Li et al. 2020), and text-based video grounding (Tang et al. 2021). These methods show that human-centric videos need to be treated separately from generic videos, by leveraging human-centric modalities such as poses and faces. Therefore, there is both the scope and the need to bring the machineries of human-centric video understanding to the task of highlight detection as well.

Main contributions. We develop an end-to-end learning system that detects highlights from human-centric videos without requiring domain-specific knowledge, highlight annotations, or exemplars. Our approach utilizes the human activities and interactions that are expressed through multiple sensory channels or modalities, including faces, eyes, voices, body poses, and hand gestures (Aviezer, Trope, and Todorov 2012; Mittal et al. 2020a). We use graph-based representations for all the human-centric modalities to sufficiently represent how the inherent structure of each modality evolves with various activities and interactions over time. Our network learns from these graph-based representations using spatial-temporal graph convolutions and maps the per-frame modalities to *highlight scores* using an autoencoder architecture. Our highlight scores are based on the representativeness of all the frames in the videos, and we stitch together contiguous frames to produce the final excerpts. Our novel contributions

include:

- **Highlight detection with human-centric modalities.** Our method identifies the observable modalities, such as poses and faces, in each input video and encodes their interrelations, across both time and different persons, into *highlight scores* for highlight detection.
- **Annotation-free training of highlight scores.** We do not require highlight annotations, exemplars, user-preferences, or domain-specific knowledge. Instead, we only need to detect of one or more human-centric modalities using off-the-shelf modality detection techniques to train our highlight scores.
- **Domain- and user-agnostic performance.** Our trained network achieves state-of-the-art performance in highlight detection over a diverse range of domains and user preferences, evaluated over multiple benchmark datasets consisting of human-centric videos.

Our method achieves a mean average precision of 0.64 and 0.20 of matching human-annotated highlight excerpts on the benchmark domain-specific video highlight (DSH) dataset (Sun, Farhadi, and Seitz 2014) and the personal highlight detection dataset (PHD^2) (Molino and Gygli 2018) dataset, respectively, and outperform the corresponding state-of-the-art methods by 7% and 4% (absolute). We also achieve state-of-the-art performance on the smaller benchmark datasets of TVSum (Song et al. 2015) and SumMe (Gygli et al. 2014), outperforming the current state-of-the-art baselines by 12% and 4% (absolute) on the mean average precision and mean F-score, respectively. Even for domains that are not fully human-centric (*e.g.*, dog shows) or videos where human-centric modalities are sparsely detected, the performance of our method is comparable to the current state-of-the-art.

8.2 Related Work

Both highlight detection and the closely related problem of video summarization have been well-studied in computer vision, multimedia, and related fields. Early methods utilized a variety of techniques including visual-content-based clustering, scene transition graphs, temporal variance of frames (Yeung, Yeo, and Liu 1998; Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang 2003; Truong and Venkatesh 2007), and hand-crafted features representing semantic information such as facial activities (Joho et al. 2011). On the other hand, recent approaches have capitalized on an impressive range of deep learning tools and techniques to perform highlight detection and video summarization.

8.2.1 Highlight Detection

The goal of highlight detection is to detect interesting moments or excerpts from unedited videos (Truong and Venkatesh 2007; Sun, Farhadi, and Seitz 2014). A large contingent of methods pose this as a supervised ranking problem, such that the highlightable excerpts are ranked higher than all other excerpts (Sun, Farhadi, and Seitz 2014; Gygli, Song, and Cao 2016; Yao, Mei, and Rui 2016; Jiao et al. 2018; Molino and Gygli 2018; Yu et al. 2018; Jiao et al. 2019; Wei et al. 2018). These methods assume the availability of human-annotated labels of the highlightable excerpts and train networks to learn either generic or domain-specific ranking metrics that correlate with these labels. On the other hand, weakly-supervised and unsupervised highlight detection methods eliminate label dependencies by leveraging exemplars or video metadata. Exemplars include scraped web images depicting domain-specific actions such as gymnastics and skiing (Kim et al. 2018). Video metadata include information on video categories (Yang et

al. 2015), or properties useful in differentiating unedited videos from edited videos, *e.g.*, duration (Xiong et al. 2019). Some approaches also take user preferences into account to generate personalized highlights (Rochan et al. 2020). All these methods perform computations in the 2D image space of the frames and do not utilize human-centric modalities.

8.2.2 Video Summarization

Video summarization aims to provide succinct synopses of videos in a variety of formats, including storyline graphs (Kim and Xing 2014; Xiong, Kim, and Sigal 2015), keyframe sequences (Lee, Ghosh, and Grauman 2012), clips (Gygli et al. 2014; Zhang, Grauman, and Sha 2018), and their mixtures based on user requirements (Gu and Swaminathan 2018). It is commonly posed as a subsequence estimation task satisfying coherence (Lu and Grauman 2013), diversity, and representativeness (Panda and Roy-Chowdhury 2017; Zhou, Qiao, and Xiang 2018). Existing unsupervised approaches build on multiple concepts, such as visual co-occurrence (Chu, Song, and Jaimes 2015), temporal relevance between frames and shots (Kim, Sigal, and Xing 2014; Mahasseni, Lam, and Todorovic 2017; Rochan, Ye, and Wang 2018; Zhang, Grauman, and Sha 2018), learning category-aware classifiers (Potapov et al. 2014) and category-aware feature learning (Zhao and Xing 2014; Song et al. 2015). Weakly supervised approaches use exemplar web images and videos (Kim and Xing 2014; Khosla et al. 2013; Cai et al. 2018b; Rochan and Wang 2019), and category descriptions (Potapov et al. 2014; Panda and Roy-Chowdhury 2017) as priors. Other approaches use supervised learning with human-annotated summaries, using subset selection (Gong et al. 2014), visual importance scores (Lee, Ghosh, and Grauman 2012; Gygli et al. 2014), submodular mixtures (Gygli, Grabner, and Van Gool 2015; Xu et al. 2015), and temporal inter-relations (Zhang et al. 2016b; Zhang, Grauman, and Sha 2018; Zhao, Li, and Lu

2017). While our objective is highlight detection, our approach is inspired by these summarization methods. Particularly, we ensure that our highlight score captures the representativeness in the videos and satisfies robust feature reconstruction.

8.2.3 Multimodal Learning

A wide body of work has focused on multimodal action recognition (Chen, Jafari, and Kehtarnavaz 2015; Shahroudy et al. 2018; Li et al. 2020; Franco, Magnani, and Maio 2020) and emotion recognition (Busso et al. 2008; Kim, Lee, and Provost 2013; Bagher Zadeh et al. 2018; Mittal et al. 2020a; Mittal et al. 2020b). These methods observe and combine cues from multiple modalities of human expression, including faces, poses, vocal tones, eye movements hand and body gestures, and gaits. Existing methods commonly model observed modalities using points and graphs (Li et al. 2020; Busso et al. 2008; Mittal et al. 2020a), making them suitable for learning action- and emotion-specific features. In our work, we utilize the fact that highlightable excerpts of human-centric videos can be determined based on the modalities. Following recent trends in multimodal action and emotion recognition (Li et al. 2020; Mittal et al. 2020a), we also model the modalities observed across the frames in videos as spatial temporal graphs, and leverage them to learn our highlight scores.

8.3 Multimodal Highlight Detection

Given human-centric videos, our goal is to detect the moments of interest or *highlights* from the videos. This section elaborates on how we detect such highlights by leveraging the human-centric modalities observed from the videos.

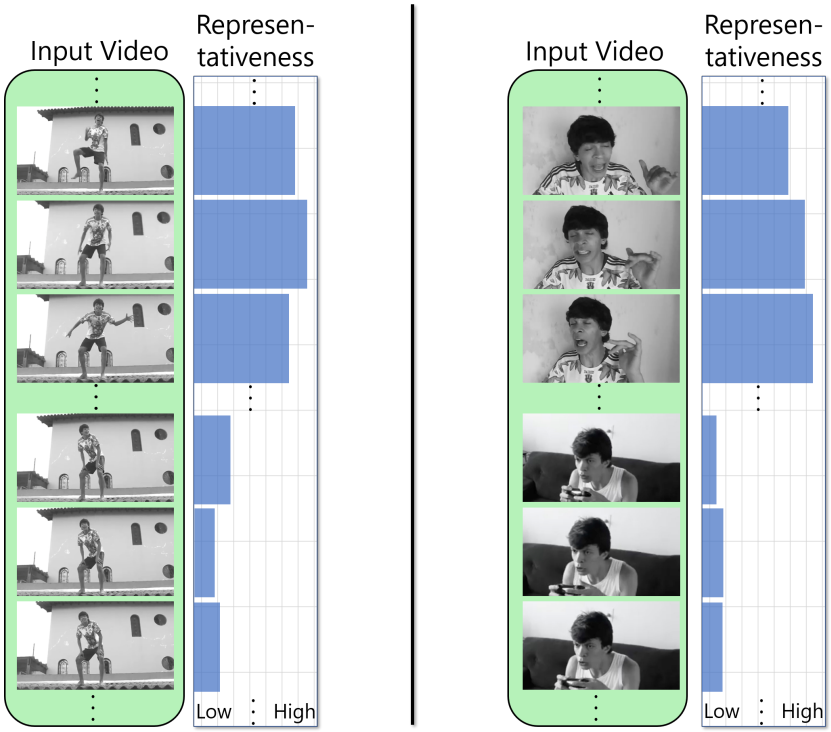


Figure 8.2: **Human-Centric Representativeness of Video Frames.** We show frames with different values of representativeness calculated in the space of poses *(left)* and face landmarks *(right)*. We learn highlight scores based on the representativeness.

8.3.1 Human-Centric Modalities

In our work, we use the term *modalities* to imply the channels of human expression sensitive to human activities and interactions, *e.g.*, faces, eyes, body poses, hands, and gaits (Chen, Jafari, and Kehtarnavaz 2015; Mittal et al. 2020a; Mittal et al. 2020b). Activities constitute individual expressions and interactions occur with other humans, living beings, and inanimate objects, pertinent to a variety of actions (Yan, Xiong, and Lin 2018; Franco, Magnani, and Maio 2020) and emotions (Bhattacharya et al. 2020; Mittal et al. 2020b). We argue that the highlightable excerpts of human-centric videos preferred by human users focus on these activities and interactions. Therefore, we aim to learn from the observable human-centric modalities in our network. For each detected modality of each human, our network leverages the inter-relations at different time instances and the inter-relations between different humans to detect the most representative excerpts.

While we extract these modalities from the RGB image-space of the video frames, we note that the modalities better capture the rich semantics of the frames. Image-space representations build on variants of the intensity differences between different parts of images, without an underlying insight on how the different parts physically interact. Conversely, modalities provide insight on such interactions based on their structure, *e.g.*, the relative movements of arms and legs indicate certain actions, and the relative movements of various facial landmarks indicate certain expressions and emotions. We build our network to explicitly consider the structure of each modality and the evolution of those structures with activities and interactions over time.

We consider $M \geq 1$ observable human-centric modalities from an input video. We assume the modalities are extracted using standard detection and tracking techniques (Kocabas 2019; Geitgey 2020), and are represented using a set of interconnected points in 2D or 3D, such as a set of 2D face landmarks for the face or a set of 3D body joints for the pose.

To represent each modality $m = 1, \ldots, M$, we construct a spatial-temporal graph representation $\mathcal{G}_m = \{\mathcal{V}_m, \mathcal{E}_m\}$. The nodes in $\mathcal{V}_m$ represent the points of the corresponding modality, and the edges in $\mathcal{E}_m$ represent both the structure of the modality and how that structure evolves over time. To sufficiently capture this, we consider three edges types:

- **Intra-person edges** capturing the spatial relationships between the nodes of a single person, *e.g.*, bones between pose joints and connectors between face landmarks. These edges represent the baseline structure of the modality at every video frame.
- **Inter-person edges** connecting the identical nodes of different persons, *e.g.*, root to root, head to head, at every video frame. These edges capture how the nodes of different persons interact with each other. They form a bipartite graph for every pair of persons, and represent the inter-person interactions at every video frame.

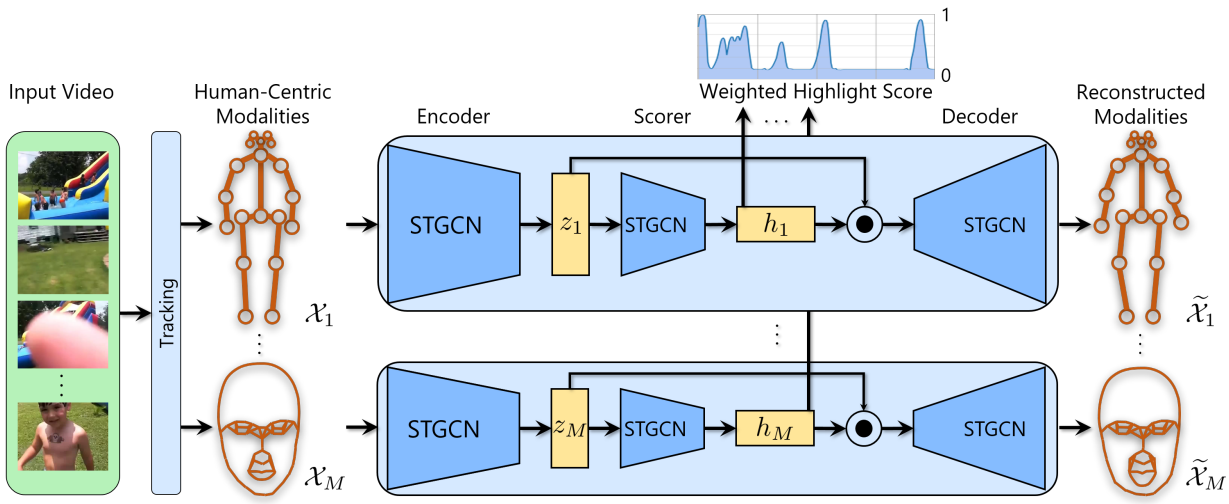


Figure 8.3: **HighlightMe: Network Architecture with Human-Centric Multimodal Learning.** Overview of our network for learning highlight scores from multiple human-centric modalities. We use standard techniques (Kocabas 2019; Geitgey 2020) to detect the human-centric modalities. We represent the modalities as sets of connected points in either 2D or 3D. We train the networks for all the modalities in parallel. The only point of interaction between the networks is their predicted highlight scores, which we combine into our weighted highlight score for training.

- **Temporal edges** connecting the identical nodes of a person, *e.g.*, root to root, head to head, over multiple video frames. These edges capture how those nodes evolve with time for every person. They form a bipartite graph for every pair of video frames, and represent the evolution of activities and interactions over time.

The spatial positions of these nodes and the combination of all these edges allow our network to learn the activities and interactions of all humans in videos and learn highlight scores accordingly, without any prior knowledge on the video domains or user-provided preferences.

8.3.2 Representativeness of the Video Frames

Since we aim to detect highlights from videos without requiring annotations or exemplars, our approach is aligned with detecting the representative frames from the videos, similar to video summarization (Mahasseni, Lam, and Todorovic 2017; Gu and Swaminathan 2018). While de-

tecting the representative frames in the image-space may or may not lead to detecting the moments of interest for highlight detection (Sun, Farhadi, and Seitz 2014), our key observation is that detecting the representative frames in the space of human-centric modalities, in fact, leads to detecting the moments of interest for highlight detection in human-centric videos.

We define the *representativeness* of a video frame as the difference, in some metric space, between the video and the video without that frame. The larger the difference, the higher the representativeness of that frame. Intuitively, the representativeness of a frame measures the fraction of information it contains in relation to the entire video. Our goal in highlight detection is to detect a *minimal* set of frames from a video with a *maximal* representativeness.

In our work, we measure the representativeness in the metric space of the observable modalities. Figure 8.2 shows examples of frames with different values of representativeness in the space of poses and face landmarks. We consider each video to consist of a total T frames and P persons (zero-padding videos with fewer frames). Therefore, for each modality m, $\mathcal{V}_m$ consists of $N \times T \times P$ nodes in total, where N is the number of nodes per person. We collate these nodes into a tensor $\mathcal{X}_m = \left[\mathcal{X}_m^{(1)}; \ldots; \mathcal{X}_m^{(T)}\right]$, where $\mathcal{X}_m^{(t)} \in \mathbb{R}^{N \times P \times D}$ for each frame t, and D is the spatial dimension of each node, most commonly 2 or 3. We can then multiply a *highlight score* $h_m^{(t)}$ of 0 or 1 to each frame t to reflect their representativeness. Thus, we can write the net difference $\mathcal{D}$ as a result of assigning the highlight scores as,

$$\mathcal{D} = \left\| \mathcal{X}_m - \left[h_m^{(1)}; \ldots; h_m^{(T)}\right] \odot \mathcal{X}_m \right\|, \tag{8.1}$$

where $\odot$ denotes the Hadamard product. We can now rewrite our goal as simultaneously minimizing $\mathcal{D}$ in Equation 8.1 and $\sum_t h_m^{(t)}$, for each modality m.

We note that a trivial solution to Equation 8.1 is to pick a threshold $0 \leq \tau \leq T$, then assign a highlight score of 1 to the top τ most representative frames from $\mathcal{X}_m$, and a highlight score of 0 to all other frames. However, the choice τ is non-trivial and needs to be learned from the data in practice. Therefore, we train an autoencoder-based deep neural network to learn the highlight scores for a wide range of data. We also allow the highlight scores to be continuous in $[0, 1]$ to keep our network differentiable. Moreover, making the highlight scores continuous also helps us understand the relative representativeness of each frame, which is an inbuilt component of modern highlight detection systems (Xiong et al. 2019; Rochan et al. 2020).

8.3.3 Network Architecture

Figure 8.3 shows our overall network architecture for predicting highlight excerpts from input videos. The goal of our network is to learn per-frame highlight scores to minimize an analogous form of Equation 8.1. Our network achieves this by taking in the per-frame graph-based representations of the observable human-centric modalities. It attempts to reconstruct all the activities in the video using as few frames of the input modalities as possible, *i.e.*, a weighted reconstruction, where the weights are the highlight scores. In this training process, our network learns to assign higher highlight scores to the frames with higher representativeness. We now describe our network architecture in detail.

Our autoencoder consists of an encoder, a scorer, and a decoder. Our encoder takes in the spatial-temporal graph $\mathcal{G}_m = \{\mathcal{V}_m, \mathcal{E}_m\}$ for each observable modality m from an input video. It uses a separate spatial temporal graph convolutional network (STGCN) (Yan, Xiong, and Lin 2018; Kipf and Welling 2016; Defferrard, Bresson, and Vandergheynst 2016) to transform $\mathcal{X}_m$ of each modality m into a latent activity-based feature $z_m \in \mathbb{R}^{N \times T \times P \times D_l}$, D_l being the dimension

of each node in the latent feature. We thus have the operation,

$$z_m = \text{STGCN}\left(\mathcal{A}_m, \mathcal{X}_m; W_m^{(\text{enc})}\right), \tag{8.2}$$

where $\mathcal{A}_m$ denotes the adjacency matrix obtained from $\mathcal{E}_m$, and $W_m^{(\text{enc})}$ consists of the set of trainable STGCN parameters in the encoder. We note here that the data $\mathcal{X}_m$ forms a full-rank tensor, therefore the STGCN avoids the degenerate solution of assigning 0's to all z_m's.

Our latent activity-based features $z_m \; \forall m$ connect to our scorer, which consists of a single layer of spatial temporal graph convolution followed by a sigmoid operation per modality. Our scorer transforms each z_m into a normalized highlight score $h_m \in [0,1]^{N\times T\times P\times 1}$ for each node, *i.e.*,

$$h_m = \sigma\left(\text{STGCN}\left(\mathcal{A}_m, z_m; W_m^{(\text{hlt})}\right)\right), \tag{8.3}$$

where $\sigma(\cdot)$ represents the sigmoid function and $W_m^{(\text{hlt})}$ consists of the set of trainable STGCN parameters.

Our decoder takes in the feature z_m and the highlight score h_m for each modality m, and produces a weighted latent feature $\widetilde{z}_m \in \mathbb{R}^{N\times T\times P\times D_l}$ by performing a Hadamard product of h_m with each node dimension of z_m, *i.e.*,

$$\widetilde{z}_m = \underbrace{[h_m; h_m; \ldots]}_{D_l \text{ times}} \odot z_m. \tag{8.4}$$

In other words, we aim to pick the latent features in z_m that correspond to the most representative frames in $\mathcal{X}_m$. While training, our scorer successfully learns to assign higher h_m values to the z_m features representing the more representative frames, and favors them in the recon-

struction process.

From the weighted latent feature $\widetilde{z}_m$, our decoder produces a reconstruction $\widetilde{\mathcal{X}}_m \in \mathbb{R}^{N\times T\times P\times D}$ of the input graph nodes using another STGCN, *i.e.*,

$$\widetilde{\mathcal{X}}_m = \text{STGCN}\left(\mathcal{A}_m, \widetilde{z}_m; W_m^{(\text{dec})}\right), \tag{8.5}$$

where $W_m^{(\text{dec})}$ consists of the set of trainable STGCN parameters in the decoder.

8.3.4 Loss Function for Training

Analogous to Equation 8.1, we train our network architecture to maximally reconstruct the input graph nodes in all the modalities while minimizing the number of frames considered for reconstruction. Our approach is based on the assumption that the frames with higher representativeness, constitute the more highlightable excerpts of the video. Therefore, in effect, we aim to suppress as many frames as possible in the reconstruction of the input video while focusing on only the frames with high representativeness.

Given the highlight scores h_m for each modality m, we perform a max-pooling of the scores across all dimensions but the time to obtain $h_m^{(\max\ T)} \in [0,1]^{T\times 1}$, the maximum highlight score per frame of the video for that modality, *i.e.*,

$$h_m^{(\max\ T)} = \max_{n\in N, p\in P} h_m. \tag{8.6}$$

We also consider a weighted contribution of $h_m^{(\max\ T)}$ for each modality m, such that the weight is proportional to the number of frames in which the modality was visible in the input

video. We define a modality to be observable in a frame if more than half the constituent points of that modality are visible in the frame. By that definition, we construct a weight α_m for each modality m as

$$\alpha_m = \frac{\#\text{ frames where modality } m \text{ is observable}}{T}. \tag{8.7}$$

We have $0 \leq \alpha_m \leq 1 \; \forall m$ since each frame can contain between no and all modalities.

We then construct weighted highlight scores $\bar{h}_m \in [0,1]^{T\times 1}$ for all the frames of the video as

$$\bar{h}_m = \alpha_m h_m^{(\max T)}. \tag{8.8}$$

Finally, given the decoder reconstructions $\widetilde{\mathcal{X}}_m$ and the weights per modality α_m, we construct our loss function $\mathcal{L}$ for training our network as

$$\mathcal{L} = \sum_m \left\| \mathcal{X}_m - \widetilde{\mathcal{X}}_m \right\| + \left\| \bar{h}_m \right\| + \lambda_m \left\| W_m \right\|, \tag{8.9}$$

where W_m collates all the trainable parameters $W_m^{(\text{enc})}$, $W_m^{(\text{hlt})}$, and $W_m^{(\text{dec})}$, λ_m are the regularization factors, and we use the smooth-ℓ_1 norm for $\|\cdot\|$. We note that $\mathcal{L}$ consists of contrasting objectives that provide the competition needed to learn the highlight scores. The subtrahend $\widetilde{X}_m$ in the first term, $\left\| \mathcal{X}_m - \widetilde{\mathcal{X}}_m \right\|$, obtained from Eqs. 8.4 and 8.5, is a stand-in for the subtrahend in Equation 8.1. Minimizing this first term would require setting all highlight scores to 1 (so all frames are highlights). Conversely, minimizing the second term $\left\| \bar{h}_m \right\|$ would require setting all highlight scores to 0 (so no frames are highlights). Consequently, our network ends up assigning high highlight scores to only the set of frames with maximal representativeness.

8.4 Implementation and Testing

We train our network on the large-scale AVA-Kinetics dataset (Li et al. 2020). This dataset consists of 235 training videos and 64 validation videos, each 15 minutes long and annotated with action labels in 1-second clips. We ignore the action labels and use the original videos to train and validate our highlight detection network. The dataset consists of a wide variety of human activities but no supervision on highlightable excerpts. Thus, it is suitable for our task of learning to detect human-specific highlight excerpts. Owing to memory constraints, we process each video in non-overlapping excerpts of 30 seconds, leading to a total of 7,050 training excerpts and 1,920 excerpts for validation.

8.4.1 Implementation

We use $M = 2$ modalities, poses and faces, which are the two most observable modalities in all the datasets we tested our method on. Other modalities, such as hand gestures and eye movements, are either rarely visible or suffer from noisy detection. We build the pose graph following the CMU panoptic model (Joo et al. 2017; Mehta et al. 2018), and the face landmarks graph following the face landmarks model of Geitgey (Geitgey 2020).

We use a multi-person tracker (Kocabas 2019) to track the persons across all the frames. We use a pose detector (Mehta et al. 2018) and a face landmark detector (Geitgey 2020), to respectively detect the coordinates of their 3D poses and 2D face landmarks. We scale all the coordinates to lie in the range $[-1, 1]$. To build our graph for each modality, we consider up to $P = 20$ persons in each frame and temporal edges to $30f$ temporally adjacent frames combining the past and the future, f being the frame rate of processing the videos. When available, we

use an equal number of frames in the past and the future for temporal adjacency. We have observed efficient performance in terms of both accuracy and memory requirements for frame rates between 2 and 5, use $f = 5$ for our experiments. For all z_m's, we use a latent dimension of $D_l = 8$.

We train using the Adam optimizer (Kingma and Ba 2014) for 200 epochs with a batch size of 2, an initial learning rate of 10^{-3}, a momentum of 0.9, and a weight decay of 10^{-4}. We decrease our learning rate by a factor of 0.999 after every epoch. Our training took around 4.6 GPU days at around 40 minutes per epoch on an Nvidia GeForce GTX 1080Ti GPU.

8.4.2 Testing

At test time, we obtain weighted highlight scores $\sum_m \bar{h}_m$ following Equation 8.8 for each frame of the input video. We combine all contiguous frames above a threshold h_{thres} to generate highlight excerpts for the video. Based on our experiments, we have observed that values of $h_{\text{thres}} \geq 0.5$ leads to the detection of representative highlight excerpts in the benchmark datasets. The difference between h_{thres} and τ (Section 8.3.2) is that h_{thres} is used for trained highlight scores that capture domain- and user-preference-agnostic representativeness. In practice, we assign the individual highlight excerpts a score that is the mean of the weighted highlight scores for each of its constituent frames. We rank the excerpts based on these scores so that users can select their own thresholds to obtain the excerpts above those thresholds. The higher the threshold they choose, the fewer excerpts that survive the thresholding, thus reducing their manual effort of sifting through less representative excerpts.

Table 8.1: **HighlightMe: Mean Average Precision on the DSH dataset (Molino and Gygli 2018)**. Bold: **best**, underline: second-best. Our method performs second-best in the surfing domain, where not enough poses and faces were detected, and best in all the other domains.

Domain	RRAE (Yang et al. 2015)	Video2 GIF (Gygli, Song, and Cao 2016)	LSVM (Sun, Farhadi, and Seitz 2014)	Less is More (Xiong et al. 2019)	Ours
dog show	0.49	0.31	0.60	0.58	**0.63**
gymnastics	0.35	0.34	0.41	0.44	**0.73**
parkour	0.50	0.54	0.61	0.67	**0.72**
skating	0.25	0.55	0.62	0.58	**0.64**
skiing	0.22	0.33	0.36	0.49	**0.52**
surfing	0.49	0.54	0.61	**0.65**	0.62
Mean	0.38	0.46	0.54	0.57	**0.64**

8.5 Experiments

We evaluate the comparative performance of our method and current state-of-the-art methods on two large-scale public benchmark datasets: the Domain-Specific Highlights (DSH) dataset (Sun, Farhadi, and Seitz 2014) and the Personal Highlight Detection dataset (PHD^2) (Molino and Gygli 2018). We also evaluate on the smaller public datasets of TVSum (Song et al. 2015) and SumMe (Gygli et al. 2014). Unlike any of the current approaches, however, we do not train or fine-tune our method on any of these datasets. We also test the performance of ablated versions of our network by removing individual modalities from training and evaluation.

8.5.1 Datasets

The DSH dataset (Sun, Farhadi, and Seitz 2014) consists of YouTube videos across six domain-specific categories: dog show, gymnastics, parkour, skating, skiing, and surfing. There are roughly 100 videos in each domain, with a total duration of around 1,430 minutes. The PHD^2 dataset (Molino and Gygli 2018) consists of a total of around 10,000 YouTube videos in the test

Table 8.2: **HighlightMe: Mean Average Precision on PHD^2 (Molino and Gygli 2018).** Bold: **best**, underline: second-best.

Random	FCSN (Rochan, Ye, and Wang 2018)	Video2 GIF (Gygli, Song, and Cao 2016)	Ad-FCSN (Rochan et al. 2020)	Ours
0.12	0.15	0.15	0.16	**0.20**

set, totaling about 55,800 minutes. It consists of highlights annotated by 850 users based on their preferences. The TVSum dataset (Song et al. 2015) has 50 YouTube videos totaling about 210 minutes, collected across ten domains: beekeeping (BK), bike tricks (BT), dog show (DS), flash mob (FM), grooming animal (GA), making sandwich (MS), parade (PR), parkour (PK), vehicle tire (VT), and vehicle unstuck (VU). The SumMe dataset (Gygli et al. 2014) has 25 personal videos, totaling about 66 minutes.

8.5.2 Evaluation Metrics

We compute the commonly used mean average precision (mAP) of the detected highlights matching the annotated highlights (Sun, Farhadi, and Seitz 2014; Gygli, Song, and Cao 2016; Molino and Gygli 2018; Xiong et al. 2019; Rochan et al. 2020). For evaluating highlights, we consider the precision for each video individually rather than across videos, because the highlights detected from one video need not necessarily have higher highlight scores than the non-highlighted segments of another video (Sun, Farhadi, and Seitz 2014). We also report the mean F-score (harmonic mean of the precision and the recall, calculated per video, and then averaged over all videos) of our method on all the datasets and for the provided baselines on the SumMe dataset (Gygli et al. 2014).

Table 8.3: **HighlightMe: Mean Average Precision on the TVSum Dataset (Song et al. 2015).** Full domain names are in Section 8.5.1. Bold: **best**, underline: second-best. Our method performs second-best in the domains that are not fully human-centric (BK, DS, GA, MS), and best in all the other domains.

Domain	MBF (Chu, Song, and Jaimes 2015)	KVS (Potapov et al. 2014)	CVS (Panda and Roy-Chowdhury 2017)	Adv-LSTM (Mahasseni, Lam, and Todorovic 2017)	Less is More (Xiong et al. 2019)	Ours
BK	0.31	0.34	0.33	0.42	**0.66**	0.57
BT	0.37	0.42	0.40	0.48	0.69	**0.93**
DS	0.36	0.39	0.38	0.47	**0.63**	0.60
FM	0.37	0.40	0.37	0.46	0.43	**0.88**
GA	0.33	0.40	0.38	0.48	**0.61**	0.50
MS	0.41	0.42	0.40	0.49	**0.54**	0.50
PR	0.33	0.40	0.38	0.47	0.53	**0.84**
PK	0.32	0.38	0.35	0.46	0.60	**0.76**
VT	0.30	0.35	0.33	0.42	0.56	**0.65**
VU	0.36	0.44	0.41	0.47	0.50	**0.77**
Mean	0.35	0.40	0.37	0.46	0.58	**0.70**

8.5.3 Baselines

We compare with four baselines on the DSH dataset (Sun, Farhadi, and Seitz 2014), four on PHD^2 (Molino and Gygli 2018), five on the TVSum dataset (Song et al. 2015), and seven on the SumMe dataset (Gygli et al. 2014). We report the performances of the baselines as stated in the literature.

On the DSH dataset, we compare with the latent SVM-based highlight ranking (LSVM) method of Sun et al. (Sun, Farhadi, and Seitz 2014), Video2GIF (Gygli, Song, and Cao 2016), which uses C3D features with fully connected layers to learn highlight ranking, the unsupervised robust recurrent autoencoder method (RRAE) of Yang et al. (Yang et al. 2015), and the method of Xiong et al. (Less is More) (Xiong et al. 2019) that learns to rank highlights by using the duration of videos as weak supervision, with the insight that shorter videos are more likely to be edited and therefore more highlightable.

On PHD^2, we compare with Video2GIF (Gygli, Song, and Cao 2016) again, the fully convo-

Table 8.4: **HighlightMe: F-Scores on the SumMe Dataset (Gygli et al. 2014).** Bold: **best**, underline: second-best.

Method	F-Score
Int (Gygli et al. 2014)	0.39
Sub (Gygli, Grabner, and Van Gool 2015)	0.40
DPP-LSTM (Zhang et al. 2016b)	0.39
GAN-S (Mahasseni, Lam, and Todorovic 2017)	0.42
DRL-S (Zhou, Qiao, and Xiang 2018)	0.42
S^2N (Wei et al. 2018)	0.43
Ad-FCSN (Rochan et al. 2020)	<u>0.44</u>
Ours	**0.48**

lutional sequence network (FCSN) that uses GoogLeNet to learn image-based features for highlight detection (Rochan, Ye, and Wang 2018), and the adaptive FCSN method (Ad-FCSN) (Rochan et al. 2020), which additionally consists of a history encoder to adapt to a user's history of highlight preferences to detect personalized highlights. We also use a fully random highlight detector as the lowest baseline following (Rochan et al. 2020).

On the TVSum dataset, we compare again with the duration-based highlight detection method (Less is More) (Xiong et al. 2019), the visual correlation-based method of Chu et al. (Chu, Song, and Jaimes 2015) that uses maximal biclique finding (MBF) to obtain co-occurring shots that are also relevant to the original video, the kernel-based video summarization method (KVS) of Potapov et al. (Potapov et al. 2014) that trains an SVM on semantically consistent segments, the collaborative video summarization method (CVS) of Panda et al. (Panda and Roy-Chowdhury 2017) that uses a consensus regularizer to detect highlight segments satisfying sparsity, diversity, and representativeness, and the unsupervised video summarization method of Mahasseni et al. (Mahasseni, Lam, and Todorovic 2017) using LSTMs with adversarial loss (Adv-LSTM).

On the SumMe dataset, we compare again with adaptive FCSN (Ad-FCSN) (Rochan et al. 2020), the interestingness-based summarization method (Int.) of Gygli et al. (Gygli et al. 2014),

Table 8.5: **HighlightMe: Ablation Studies.** Comparison of mean mAP and mean F-score for different ablated versions of our method on benchmark datasets. Bold: **best**, underline: second-best.

Dataset	Using Modality					
	Face only		Pose only		Both	
	mAP	F	mAP	F	mAP	F
DSH (Sun, Farhadi, and Seitz 2014)	0.51	0.45	0.57	0.48	**0.64**	**0.56**
TVSum (Song et al. 2015)	0.57	0.46	0.64	0.56	**0.70**	**0.59**
PHD^2 (Molino and Gygli 2018)	0.16	0.20	0.15	0.18	**0.20**	**0.22**
SumMe (Gygli et al. 2014)	0.48	0.39	0.45	0.41	**0.52**	**0.48**

the submodularity-based summarization method (Sub.) of Gygli et al. (Gygli, Grabner, and Van Gool 2015), the LSTM network of Zhang et al. (Zhang et al. 2016b) employing a determinantal point process (DPP-LSTM), the GAN-based method of Lu and Grauman (Lu and Grauman 2013) with extra supervision (GAN-S), the deep reinforcement learning-based method of Zhou et al. (Zhou, Qiao, and Xiang 2018) with extra supervision (DRL-S), and the sequence to segments detection method (S^2N) (Wei et al. 2018) that uses an encoder-decoder architecture to detect segments with high relevance from sequence data.

8.5.4 Results

DSH (Sun, Farhadi, and Seitz 2014) and TVSum (Song et al. 2015). We report the mAP across all the domains in these datasets in Tables 8.1 and 8.3 respectively. We outperform the baselines on all but a few domains, which are either not fully human centric (beekeeping, dog show, grooming animals, and making sandwich in TVSum), or where sufficient poses and faces could not be detected (surfing in DSH). However, we come second-best on these domains, and on average, across the domains, we outperform the best baselines by an absolute 4% – 12%.

PHD^2 (Molino and Gygli 2018). We report the mAP across the dataset in Table 8.2.

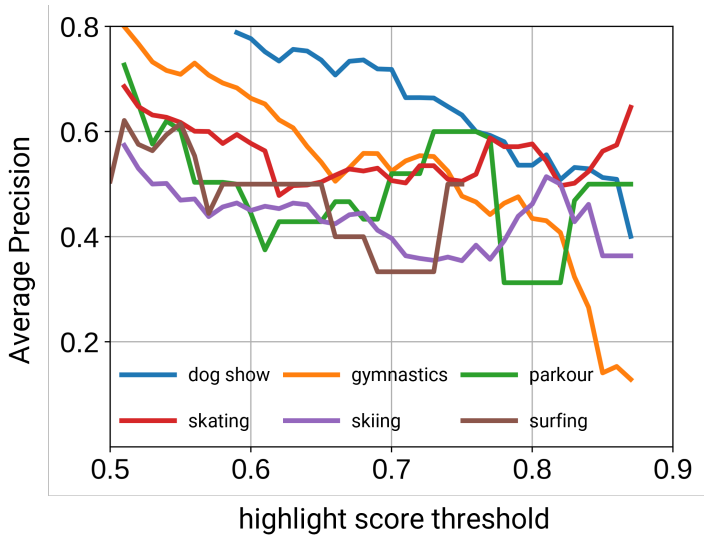


Figure 8.4: **HighlightMe: Average Precision by Highlight Score Threshold h_{thres}.** On the domains in the DSH dataset (Sun, Farhadi, and Seitz 2014).

Given the abundance of humans detected in the videos, we outperform the best baseline by an absolute 4%.

SumMe (Gygli et al. 2014). We report the mean F-scores across the dataset in Table 8.4. Following prior methods (Wei et al. 2018; Rochan et al. 2020), we randomly select 20% of the dataset for calculating the mean F-score, repeat this experiment five times, and report the mean performance. Based on these experiments, we outperform the best baseline by an absolute 4%.

These results demonstrate that our approach of using human-centric modalities to detect highlights leads to state-of-the-art performance on all these benchmark datasets.

8.5.5 Ablation Studies

In our experiments, we consider two modalities, poses and faces. We ablate each of these two modalities in turn and test the performance of our method by training it on the remaining modality. We report the mean mAP and the mean F-score of the ablated versions of our method on all four benchmark datasets in Table 8.5. Using only poses and no faces, we observe an absolute drop-off of 5% – 7% for the mean mAP and 3% – 8% for the mean F-score across the

datasets, compared to using both modalities. Using only faces and no poses, we observe more severe absolute drop-offs of 4% – 13% for the mean mAP and 2% – 13% for the mean F-score across the datasets. This happens because poses are generally more abundant and more easily detected compared to face landmarks. For example, poses can be detected even when a human is partially occluded, in the dark, or not in clear focus, whereas detection of face landmarks requires the face to be well-lit and in focus. Therefore, not detecting poses leads to missing a significant number of highlightable excerpts. This trend is reversed only in PHD^2, where faces were more commonly detected than poses.

We also show the qualitative performance of our method and all its ablated versions on one sample video from each of the four datasets, DSH, PHD^2, TVSum, and SumMe, in Figure 8.5. We see that when observing only poses and not faces, our method detects the representative highlight excerpts with pose-based expressions but fails to detect excerpts that primarily have facial expressions and emotions. Conversely, when observing only faces and not poses, our method can only detect the excerpts where the faces are prominent, and misses excerpts where the faces are too small, occluded, or in the dark. Using both modalities, our method can detect all the representative excerpts.

8.5.6 Effect of Highlight Score Threshold

We use a threshold h_{thres} on the highlight score predicted by our method to detect the highlightable excerpts (Section 8.4.2). To visualize the effect h_{thres} has on the average precision (AP), we show the plot of AP vs. h_{thres} on each domain in the DSH dataset (Sun, Farhadi, and Seitz 2014) in Figure 8.4. We observe a general trend of the AP decreasing as we increase the threshold, as our method returns fewer and fewer highlights. However, this is not true for some

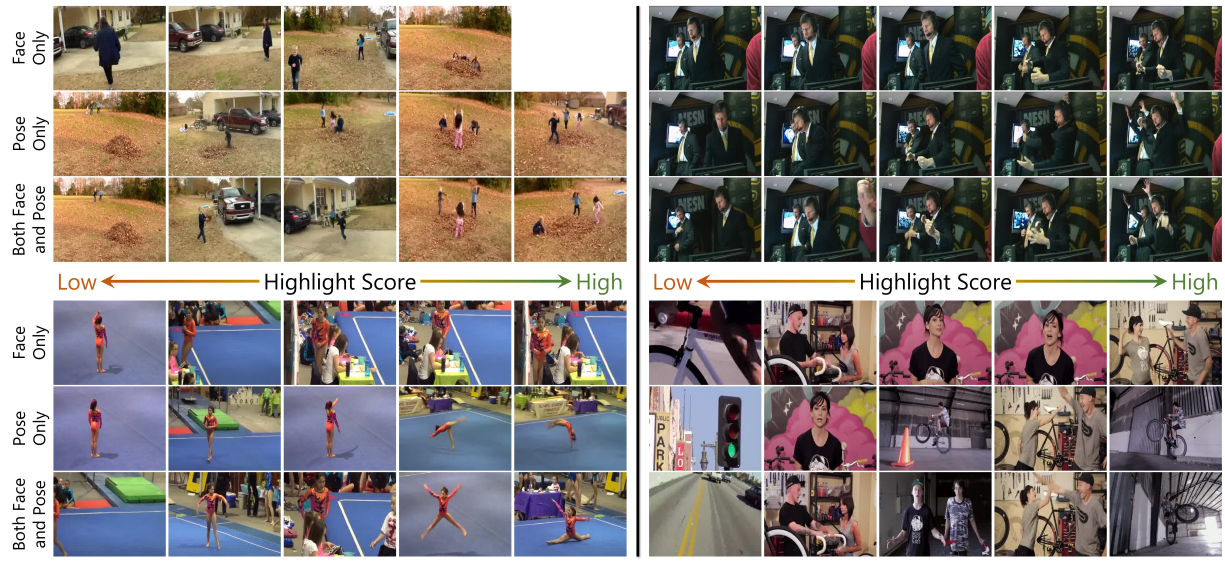


Figure 8.5: **HighlightMe: Qualitative Comparisons.** We show sample frames across the range of highlight scores as detected by different ablated versions of our method. We show one sample video from the datasets SumMe (Gygli et al. 2014), PHD^2 (Molino and Gygli 2018), DSH (Sun, Farhadi, and Seitz 2014), and TVSum (Song et al. 2015), in order from top to bottom. When using only faces or only poses, our method learns highlight scores based only on face- or pose-based representativeness. Combining both the modalities, our method learns highlight scores based on representativeness from both.

domains like surfing, where the highlight scores of the representative excerpts are already sufficiently high. In practice, we consider the choice of threshold to be user-configurable for each video.

8.6 Conclusion

We have presented a novel method for detecting highlights from human-centric videos, which leverages the observable human-centric modalities, such as faces and poses, and uses these modalities to automatically detect the most representative highlights from the video. Extensive experimental results on the domain-specific highlights (DSH) dataset (Sun, Farhadi, and Seitz 2014), the personal highlight detection dataset (PHD^2) (Molino and Gygli 2018), the TVSum dataset (Song et al. 2015), and the SumMe dataset (Gygli et al. 2014) demonstrate the benefits of

our proposed approach compared to several state-of-the-art baselines.

CHAPTER 9

Use Case 2: Detecting User-Specific Highlights from Videos

Abstract

We propose a method to detect individualized highlights for users on given target videos based on their preferred highlight clips marked on previous videos they have watched. Our method explicitly leverages the contents of both the preferred clips and the target videos using pre-trained features for the objects and the human activities. We design a multi-head attention mechanism to adaptively weigh the preferred clips based on their object- and human-activity-based contents, and fuse them using these weights into a single feature representation for each user. We compute similarities between these per-user feature representations and the per-frame features computed from the desired target videos to estimate the user-specific highlight clips from the target videos. We test our method on a large-scale highlight detection dataset containing the

annotated highlights of individual users. Compared to current baselines, we observe an absolute improvement of 2–4% in the mean average precision of the detected highlights. We also perform extensive ablation experiments on the number of preferred highlight clips associated with each user as well as on the object- and human-activity-based feature representations to validate that our method is indeed both content-based and user-specific.

9.1 Introduction

We expand our use-case to consider user-specific highlight detection in videos with both humans and non-human entities. Ongoing research efforts have led to the development of efficient content-based highlight detection methods (Sun, Farhadi, and Seitz 2014; Molino and Gygli 2018; Rochan, Ye, and Wang 2018; Bhattacharya et al. 2021c). More recently, highlight detection methods have also considered viewers' preferences to provide individualized or *user-specific* highlights (Rochan et al. 2020), making these methods more practically relevant. However, current user-specific methods either require the expensive computation of shot boundaries to divide videos into highlightable and non-highlightable segments (Molino and Gygli 2018) or uniformly pool the users' selected highlights to compute their highlight preferences (Rochan et al. 2020).

In practice, users' preferences may not be uniformly distributed across their preferred highlight clips but vary significantly based on the clip contents, especially in relation to the target videos. For example, in Figure 9.1, the cooking clips are more relevant indicators of user A's preferences compared to the workout clips, given the target video containing cooking and eating. Thus, combining the users' preferred clips to learn their relevant highlight preferences for each target video requires learning the relevant features describing the highlightable content

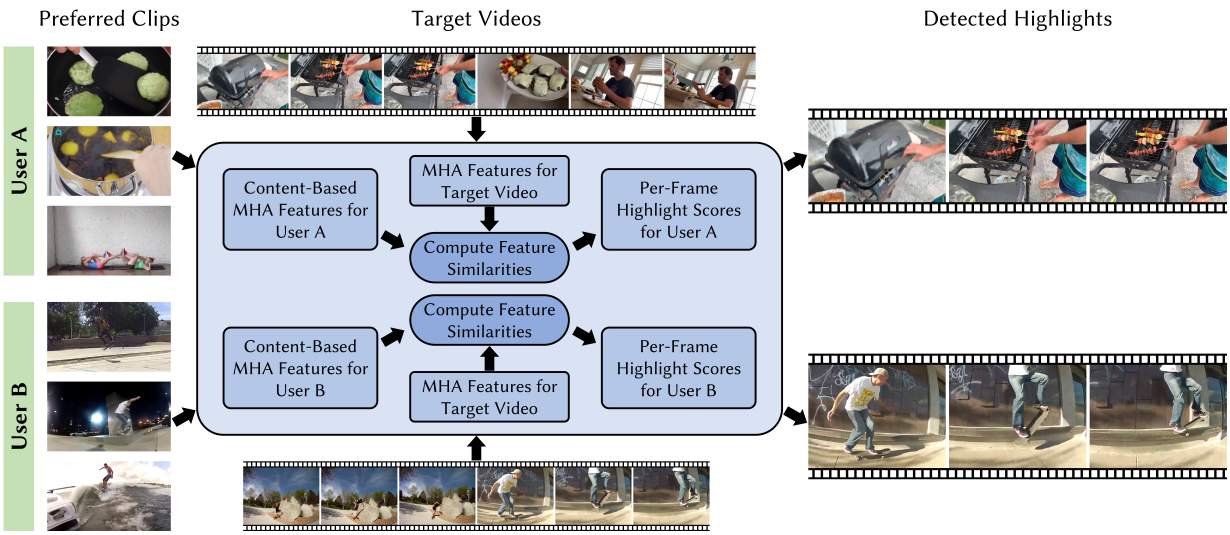


Figure 9.1: **User-Specific Highlights: Overview.** For each user, we consider a set of highlight clips denoting their individual preferences (left) and detect highlights for them (right) on different target videos (center top and bottom). Given the users' overall highlight preferences, our method employs a multi-head attention (MHA) mechanism to learn which segments of the target videos are relevant highlights based on feature similarities between them (center block). For example, our method learns that user A prefers watching cooking and workout videos. Therefore, given a target video containing cooking and eating, our method identifies that only the cooking segments are relevant between the preferred clips and the target videos, and detects those as highlights. Similarly, our method learns that user B prefers skating and surfing videos. Therefore, given a video containing parkour and skating activities, our method detects only the skating segments as highlights. Overall, our method significantly advances the state-of-the-art in user-specific highlight detection given a diverse, large-scale dataset of user preferences and target videos.

in each preferred clip. Subsequently, these relevant features need to be mapped to the contents of the target videos where the highlights are to be detected, requiring an automatic strategy to estimate the segments of the target videos that are similar to the relevant features from the preferred clips and, therefore, highlightable. Thus, for user A, a user-specific highlight detection approach needs to detect that the cooking segments in the target video are relevant to the user's preferred cooking clips, and that both the user's preferred workout clips and the eating segments in the target video are irrelevant in the current context.

Detecting highlights based on the similarities between the preferred clips and the target

videos can also be expressed as an attention-based retrieval problem. An attention mechanism adaptively weighs the keys of different key-value pairs based on their relative importance to a given query to predict the most suitable responses to the query (Vaswani et al. 2017). Depending on the data paradigm of the key, the value, and the query, attention mechanisms are used in a wide variety of tasks, including tasks in natural language understanding (Devlin et al. 2019), text-based image and video retrieval (Chen and Deng 2019), object and action recognition in images and videos (Wang, Cai, and Wang 2022; Song et al. 2018), and visual question answering (Yu et al. 2019). In the case of user-specific highlight detection, the key, value, and query need to be based on the video contents, *i.e.*, follow the paradigm of content-based highlight detection (Sun, Farhadi, and Seitz 2014; Rochan, Ye, and Wang 2018; Bhattacharya et al. 2021c) to perform meaningful retrieval of the highlightable clips per user.

Main Contributions. In this paper, we consider the visual content of the videos for highlight detection. Specifically, for both the preferred clips and the target videos, we consider the constituent non-human entities commonly clubbed as "objects" (Jocher et al. 2022), and the human activities expressed with their pose movements. We follow the current paradigm of leveraging the presence of and interactions between human activities and non-human entities, either directly or indirectly, for content-based highlight detection (Sun, Farhadi, and Seitz 2014; Molino and Gygli 2018; Bhattacharya et al. 2021c). We design an attention mechanism to leverage both the objects and the pose-based activities in the users' preferred highlight clips to detect user-specific highlights in different target videos. We only consider the preferred highlight clips and not the corresponding full videos, *i.e.*, we do not require information about clips the users did *not* select as highlights. Each highlight clip marked by the users contains partial information

about their highlight preferences, which we store in latent features learned from the objects and the poses detected in those clips. We pool these object- and pose-based features using learned weights to obtain combined features representing the users' overall highlight preferences. Given target videos for each user, we then query the object- and the pose-based features of the target videos against the combined features of the corresponding user to detect highlight clips for them in the target videos. To summarize, our main contributions are the following:

- **User-Specific Highlight Detection.** We leverage the video contents, in terms of the objects and the pose-based human activities, between the users' preferred highlight clips and the target videos to detect the user-specific highlights in the target videos.
- **Multi-Head Attention Mechanism.** We design an multi-head attention mechanism that learns content-based features per user from the users' preferred highlight clips. We also learn content-based features from the target videos and query those against the per-user features to detect the user-specific highlights.
- **State-of-the-art performance.** We perform extensive experiments on the benchmark personal highlight detection dataset (PHD^2) (Molino and Gygli 2018), containing more than 6,000 testing videos, to show that our method improves the mean average precision of highlight detection by an absolute 2–4% over the current baselines, as well as the F-score in the related problem of video summarization by an absolute 4–12% over the current baselines on the benchmark SumMe dataset (Gygli et al. 2014).

9.2 Related Work

We briefly review the prior work in highlight detection and the related problem of video summarization. For a more extensive discourse, including the development of multimedia-based

techniques for content-based clustering, scene understanding, and temporal variance optimization, we refer the readers to the works of (Yeung, Yeo, and Liu 1998; Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang 2003; Truong and Venkatesh 2007; Joho et al. 2011). Since our goal is to learn user-specific highlights from their past highlight preferences, our problem is also related to collaborative filtering, and we review prior work there as well.

Highlight Detection. Highlight detection aims to detect the most interesting moments in videos (Truong and Venkatesh 2007; Sun, Farhadi, and Seitz 2014). Since "interesting" is subjective for the viewer, highlight detection methods rely on the availability of annotated highlights to learn the interesting moments. Depending on the nature of the annotations, we can broadly classify these methods into fully supervised, weakly supervised, or unsupervised approaches. Fully supervised approaches assume that, for each video, highlight and non-highlight clips are available, and typically optimize variants of a pairwise ranking loss that ranks the highlight clips higher than the non-highlight clips (Sun, Farhadi, and Seitz 2014; Gygli, Song, and Cao 2016; Yao, Mei, and Rui 2016; Jiao et al. 2018; Yu et al. 2018; Jiao et al. 2019; Wei et al. 2018). Weakly supervised and unsupervised approaches do not require highlight annotations but instead utilize exemplars or video metadata. Exemplar-based methods use open-source images depicting highlightable moments, which guide the ranking of the individual shots or frames in the videos (Kim et al. 2018), *e.g.*, a surfing image used to detect highlights in a surfing video. Video metadata include information on the video domain (Yang et al. 2015), *e.g.*, collecting all edited videos with "surfing" in the title to learn surfing highlights, using the video duration itself as a weak signal with the argument that shorter videos are more likely to be edited and therefore have a higher percentage of highlightable moments (Xiong et al. 2019), and even focusing on certain subjects

in the videos, such as explicitly leveraging human activities to learn highlights in human-centric videos (Bhattacharya et al. 2021c). While weakly supervised and unsupervised methods alleviate over-reliance on annotated data, they are designed to be content-based and cannot adapt to the individual preferences of different users.

Preference-Based Highlight Detection. These methods solve the more challenging problem of learning individualized highlights for the users based on their annotated highlight preferences. Existing methods solve this problem by learning to score highlight frames higher than other frames conditioned on the users' preferences (Molino and Gygli 2018), learning parameters based on the users' preferred clips to guide a content-based highlight detection network (Rochan et al. 2020), or learning the users' overall preferences by averaging over features learned from their preferred clips (Chen et al. 2021). Other methods do not depend on the video contents but rank the clips in a video for each user based on the durations of their selected clips using a recommendation algorithm (Panagiotakis, Papadakis, and Fragopoulou 2020). Our work solves the same problem of predicting preference-based user-specific highlights and improves on the performances of these methods.

Video Summarization. Video summarization aims to condense videos into the summaries of their contents. Summaries are typically presented in the form of storyline graphs (Kim and Xing 2014; Xiong, Kim, and Sigal 2015), keyframe sequences (Lee, Ghosh, and Grauman 2012), clips (Gygli et al. 2014; Zhang, Grauman, and Sha 2018), as well as a mixture of formats (Gu and Swaminathan 2018). Video summarization can also be broadly classified into fully supervised, weakly supervised and unsupervised approaches. Fully supervised approaches leverage

annotated summaries, and learn relevant video subsets for summaries based on their relative importance in the videos (Gong et al. 2014; Lee, Ghosh, and Grauman 2012; Gygli et al. 2014; Gygli, Grabner, and Van Gool 2015; Xu et al. 2015; Zhang et al. 2016b; Zhang, Grauman, and Sha 2018; Zhao, Li, and Lu 2017; Ji et al. 2020). Weakly supervised approaches rely on learning the relevant contents for a summary based on exemplars such as images and edited videos (Kim and Xing 2014; Khosla et al. 2013; Cai et al. 2018b; Rochan and Wang 2019), or leveraging keywords in the video title to learn the relevant content-based features for summaries (Zhao and Xing 2014; Song et al. 2015; Potapov et al. 2014; Panda and Roy-Chowdhury 2017). Unsupervised approaches learn relevant information directly from the video contents, such as scenes that commonly co-occur in videos (Chu, Song, and Jaimes 2015), and temporal consistency across frames and shots to understand summary boundaries (Kim, Sigal, and Xing 2014; Mahasseni, Lam, and Todorovic 2017; Rochan, Ye, and Wang 2018; Zhang, Grauman, and Sha 2018). Other approaches for video summarization define and estimate specific parameters determining summaries, such as coherence (Lu and Grauman 2013), diversity, and representativeness (Panda and Roy-Chowdhury 2017; Zhou, Qiao, and Xiang 2018). Yet other approaches consider user preferences in the form of the text queries they use to search videos (Liu et al. 2015; Sharghi, Gong, and Shah 2016; Vasudevan et al. 2017; Yang et al. 2003), their feedback on individual summaries (Molino et al. 2017), and their preferences between pairs of summaries (Singla, Tschiatschek, and Krause 2016), to learn user-specific summaries. Highlight detection methods learn to leverage video contents using techniques similar to those employed for video summarization, albeit with different training objectives. Further, designing highlight detection methods using user-specific feedback requires the methods to be interactive at inference time and relies on the users' availability and ability to provide feedback. This is complementary to learning user-

specific highlights based only on the preferred highlight clips, which streamlines the inference, and the two approaches can be combined depending on the use cases.

Collaborative Filtering. Collaborative filtering aims to predict novel user preferences based on their preference history as well as the preferences of other users (Koren, Bell, and Volinsky 2009). It is widely used in tackling recommendation problems, such as the Netflix challenge (Bell and Koren 2007) and online video recommendations (Covington, Adams, and Sargin 2016). However, extending collaborative filtering to highlight detection requires the availability of multiple user responses to each clip within each video in a large training set of videos. Making such an approach practically feasible firstly requires the availability of multiple user responses for the each clip, which is not the case in current highlight detection datasets (Molino and Gygli 2018). Secondly, it requires us to define clip boundaries apriori; otherwise, each video can contain infinitely many clips. This, in turn, severely limits what the users can choose as their highlight preferences, as they can only detect highlight clips of fixed lengths. For these limitations, collaborative filtering is not amenable to user-specific highlight detection. By contrast, our highlight detection method works at the frame level and uses variable-length preferred clips marked by the users to detect their individualized highlights.

9.3 User-Specific Highlight Detection

Given preferred highlight clips marked by users and different target videos, our goal is to detect highlight clips per user from the target videos that are similar in content to the user's preferred highlight clips. For each user u, we consider $P^{(u)} = \left\{ i \mid i = 1, \ldots, N^{(u)} \right\}$ preferred highlight clips, and a target video $\tau^{(u)}$. For numerical consistency, we consider each highlight clip to be T

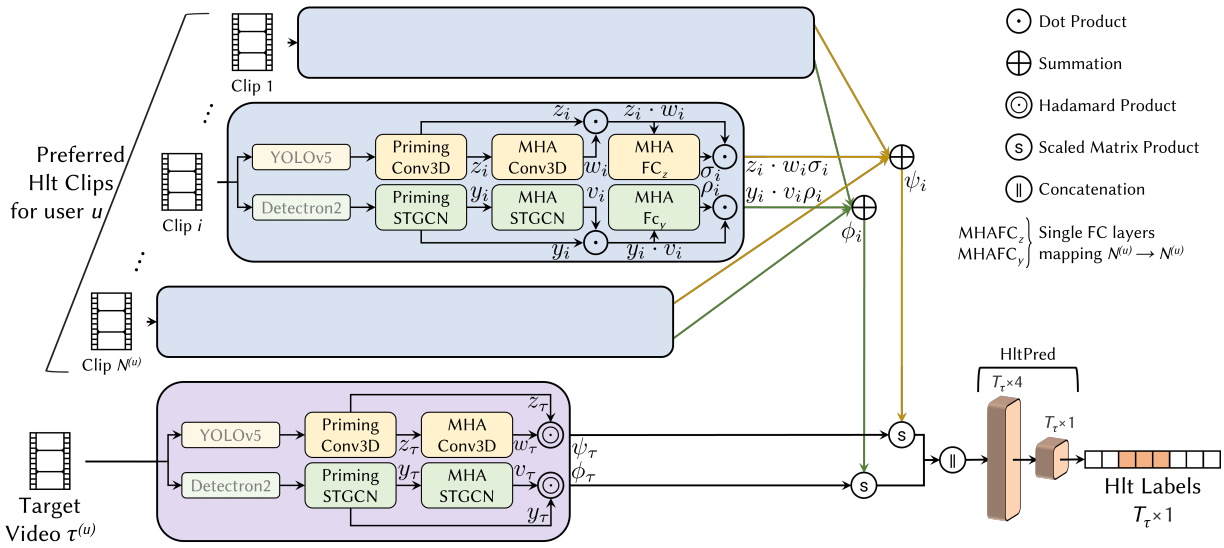


Figure 9.2: **User-Specific Highlights: Network Architecture to Learn Multi-User, Multimodal Highlights.** For each preferred clip i, we use the two priming blocks to map the object-based features and the pose-based features to respective features z_i and y_i. We use these features to learn the per-frame weights w_i and v_i using multi-head attention (MHA), perform per-frame attention pooling, learn the per-clip weights σ_i and ρ_i using MHA again, and fuse the per-clip features using weighted summation to get the fused features ψ_i and ϕ_i. For each target video τ, we train a separate set of attention priming and MHA layers to obtain fused features ψ_τ and ϕ_τ. We compute the similarities between the fused features of the preferred clips and the target video using scaled matrix products and concatenate and map the resultant features to per-frame highlight scores for the target video using a fully-connected prediction block.

frames long and zero-pad all videos shorter than T frames. In the subsequent text, we assume all the variables are for user u unless stated otherwise and drop the superscript (u) for brevity of notation.

For each user, our objective is to predict the highlight score $s_j \in [0, 1]$ for each frame $j = 1, \ldots, T_\tau$ for the target video, where T_τ is the number of frames in the target video (we zero-pad all videos shorter than T_τ frames). The highlight score s_j determines how highlightable a frame is, with 1 being the highest score. We require our highlight scores to be relative to the contents of both the target video and the user's preferred highlight clips. Thus, we aim to learn

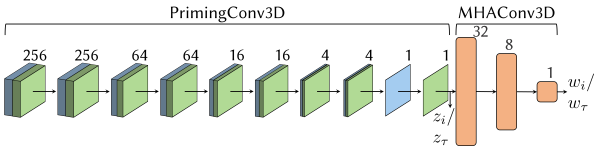


Figure 9.3: **Attention Priming and Multi-Head Attention for Objects.** Priming on the YOLOv5 (Jocher et al. 2022) features using 3D convolutions (blue blocks) and 3D batch norms (green blocks). We use the attention-primed features to learn the per-frame attention weights w_i and w_τ using fully-connected layers (orange blocks).

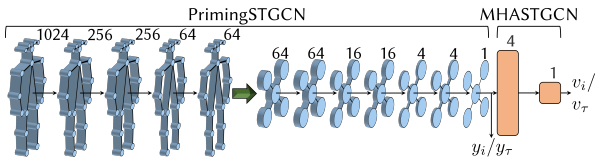


Figure 9.4: **Attention Priming and Multi-Head Attention for Poses.** Priming on the Detectron2 (Wu et al. 2019) features using spatial temporal graph convolutions with feature pooling (green arrow) on the five kinematic chains: trunk, two arms and two legs. We use the attention-primed features to learn the per-frame attention weights v_i and v_τ using fully-connected layers (orange blocks).

a scoring function $\mathcal{S}$ such that

$$\left\{s_1, \ldots, s_{T_\tau}\right\} = \mathcal{S}\left(\tau, P\right). \tag{9.1}$$

9.3.1 Attention Priming

We consider the contents to be both the non-human entities or "objects" (Jocher et al. 2022) and the pose-based human activities, and explicitly leverage their presence in all the clips. For each frame of each clip, we obtain pre-trained features x_{YOLOv5} from the penultimate layer of the YOLOv5 network (Jocher et al. 2022), which contains information on the object categories and their spatial positions. We also obtain pre-trained features $x_{\text{Detectron2}}$ from the penultimate layer of the Detectron2 network (Wu et al. 2019) containing information on the human poses. Our attention priming transforms these pre-trained object and pose features into lower-dimensional features for learning the per-frame attention weights for multi-head attention fusion. This provides the twin benefits of leveraging the localized information common to all the pre-trained features using the same set of convolutional layers, and reducing the parameter load required to learn the actual attention weights. It also ensures our network avoids memorization of the

attentions learned for the pre-trained features.

To perform attention priming, we pass the YOLOv5 x_{YOLOv5} and the Detectron2 features $_{\text{Detectron2}}$ through separate networks to get $z \in \mathbb{R}^{T \times d_z}$ and $y \in \mathbb{R}^{T \times d_y}$ respectively, which we use to learn attention weights. For priming the YOLOv5 features, we use a series of 3D convolutions, denoted PrimingConv3D (Figure 9.3), to leverage the spatial and temporal adjacency of the features x_{YOLOv5}, as

$$z = \text{PrimingConv3D}\left(x_{\text{YOLOv5}}; \theta_{\text{PrimingConv3D}}\right), \tag{9.2}$$

where $\theta_{\text{PrimingConv3D}}$ are trainable parameters. Similarly, for priming the Detectron2 features, we use a series of spatial temporal graph convolutions, denoted PrimingSTGCN (Figure 9.4), to leverage adjacencies between the body joints that comprise the features $x_{\text{Detectron2}}$, as

$$y = \text{PrimingSTGCN}\left(x_{\text{Detectron2}}; \theta_{\text{PrimingSTGCN}}\right), \tag{9.3}$$

where $\theta_{\text{PrimingSTGCN}}$ are trainable parameters.

9.3.2 Multi-Head Attention and Fusion

For each preferred highlight clip i, we denote the attention-primed features as $z_i \in \mathbb{R}^{T \times d_z}$ and $y_i \in \mathbb{R}^{T \times d_y}$. Each (z_i, y_i) contains information on the user's highlight preferences in clip i, which is different from the user's highlight preferences in all the other preferred clips. Thus, to obtain the user's overall highlight preferences, we consider a multi-head attention (MHA) mechanism that learns relative weights at both the frame and the clip levels.

At the frame level, the number of attention heads is the number of frames T for the preferred clips and T_τ for the target video. The parameters determining the attention weights are shared for all the preferred clips, implying that the order in which our network processes them is irrelevant. This is the desired behavior since the preferred clips need not have any content overlap or any specified ordering. However, we train a separate set of parameters to learn the attention weights from the target videos since the target video contains both highlightable and non-highlightable segments.

At the clip level, the number of attention heads is the number of preferred clips N that we consider for each user. We use the clip level attention weights to additively fuse the corresponding attention-primed features and then compute the similarity of the fused features to the attention-primed features of the target video. During training, the frame level and the clip level attention weights collectively determine the features describing the relevant contents of the preferred clips given the target videos per user. Conversely, during inference, our network uses these per-user relevant features to detect the highlightable frames from the target videos.

For our MHA mechanism, we train a pair of attention heads MHAConv3D (Figure 9.3) and MHASTGCN (Figure 9.4) to produce frame level attention weights $w_i \in \mathbb{R}^T$ and $v_i \in \mathbb{R}^T$ as

$$w_i = \text{MHAConv3D}\left(z_i; \theta_{\text{MHAConv3D}}\right), \tag{9.4}$$

$$v_i = \text{MHASTGCN}\left(y_i; \theta_{\text{MHASTGCN}}\right), \tag{9.5}$$

where $\theta_{\text{MHAConv3D}}$ and θ_{MHASTGCN} are trainable parameters. We train another pair of attention heads MHAFC_z and MHAFC_y (Figure 9.2, rightmost layers in central block) to get clip level

attention weights $\sigma_i \in \mathbb{R}$ and $\rho_i \in \mathbb{R}$ from the weighted frame level features as

$$\sigma_i = \text{MHAFC}_z \left(z_i^\top w_i; \theta_{\text{MHAFC}_z} \right), \tag{9.6}$$

$$\rho_i = \text{MHAFC}_y \left(y_i^\top v_i; \theta_{\text{MHAFC}_y} \right), \tag{9.7}$$

where θ_{MHAFC_z} and θ_{MHAFC_y} are trainable parameters. We obtain the fused, attention-weighted features $\psi \in \mathbb{R}^{d_z}$ and $\phi \in \mathbb{R}^{d_y}$ as

$$\psi = \sum_{i=1}^{N} z_i^\top w_i \sigma_i, \tag{9.8}$$

$$\phi = \sum_{i=1}^{N} y_i^\top v_i \rho_i. \tag{9.9}$$

Parallelly, we compute the per-frame attention-weighted features of the target video as $\psi_\tau = z_\tau \bigodot w_\tau \in \mathbb{R}^{T_\tau \times d_z}$ and $\phi_\tau = y_\tau \bigodot v_\tau \in \mathbb{R}^{T_\tau \times d_y}$. Given ψ_τ, ϕ_τ, ψ, and ϕ, we use scaled matrix products to compute features $h_z \in \mathbb{R}^{T_\tau}$ and $h_y \in \mathbb{R}^{T_\tau}$ representing their pairwise similarities, as

$$h_z = \text{softmax} \left(\psi d_z^{-1/2} \right) \cdot \psi_\tau, \tag{9.10}$$

$$h_y = \text{softmax} \left(\phi d_y^{-1/2} \right) \cdot \phi_\tau. \tag{9.11}$$

We then use a predictor block HltPred (Figure 9.2, bottom-right block) that fuses the features h_z and h_y using concatenation, and reduces the fused features to per-frame highlight scores $s \in [0, 1]^{T_\tau}$, as

$$s = \text{HltPred} \left(h_z \parallel h_y; \theta_{\text{HltPred}} \right), \tag{9.12}$$

where θ_{HltPred} are trainable parameters.

9.4 Training and Inference

We discuss the loss function we use to train our network, the implementation details, and how we use our network to detect user-specific highlights on test videos.

9.4.1 Loss Function

Our network predicts a highlight score between 0 and 1 for each frame in the target video. For training, we consider a frame to be a highlight if and only if its score is above a predetermined threshold ζ. We train our network using a combination of a label loss, a margin loss, and a sparsity loss, assuming a threshold $\zeta = 0.5$.

- **Label loss** $L_{\tau j}$ for each frame j in target video τ measures the weighted cross entropy loss between the ground-truth highlight labels $y_{\tau j}$ (1 for highlight, 0 otherwise) and the predicted highlight scores $s_{\tau j}$, as

$$L_{\tau j} = -w y_{\tau j} \log\left(s_{\tau j}\right) - \left(1 - y_{\tau j}\right) \log\left(1 - s_{\tau j}\right), \tag{9.13}$$

  where w denotes the relative weight assigned to the highlight class. The label loss is the baseline loss that guides the training process as a binary classification problem.
- **Margin loss** $M_{\tau j}$ for each frame j in target video τ measures the one-sided distance between the highlight score of that frame and the threshold ζ as

$$M_{\tau j} = \max\left(0, \widetilde{y}_{\tau j}\left(\zeta - s_{\tau j}\right)\right), \tag{9.14}$$

where $\widetilde{y}_{\tau j} = 1$ if j is a highlight frame, and -1 otherwise. The margin loss provides additional constraints that the scores for the highlight and the non-highlight frames should be on opposite sides of the threshold ζ.

- **Sparsity loss** S_τ for each target video τ enforces that the total number of highlight frames should be low, as

$$S_\tau = \sum_{j=1}^{T} \max\left(0, \text{sign}\left(s_{\tau j} - \zeta\right)\right). \tag{9.15}$$

The sparsity loss incentivizes our network to detect as few highlight frames as possible, following the intuition that highlight frames make up a small fraction of total video (Sun, Farhadi, and Seitz 2014; Xiong et al. 2019; Bhattacharya et al. 2021c). Moreover, the sparsity loss improves the precision of highlight detection by its design of minimizing the number of frames detected as highlights.

Combining the individual loss terms, we write the overall loss function $\mathcal{L}_\tau$ for the target video τ as

$$\mathcal{L}_\tau = \frac{1}{T} \sum_{j=1}^{T} \left(L_{\tau j} + M_{\tau j}\right) + S_\tau. \tag{9.16}$$

Based on our experiments, we did not observe significant performance differences when using relative weights for the margin and the sparsity losses up to five times that of the baseline label loss. Consequently, we propose using the same weights for all the loss terms for simplicity. We also note that none of our losses enforce the temporal continuity of labels, *i.e.*, adjacent frames should have the same labels (highlight or non-highlight) except at the highlight clip boundaries. There are two reasons for this. First, the durations of highlight clips are not fixed. In our experiments, the longest highlight clip can be about seven times the length of the shortest highlight clip for the same video. Thus, defining clip boundaries for enforcing temporal

continuity is non-trivial and would, in fact, limit the variety of highlight clips that we can detect during inference. Second, our network explicitly leverages information along both the spatial and the temporal dimensions using 3D convolutions and spatial temporal graph convolutions. Consequently, temporal continuity is implicitly enforced in the learned features and highlight scores, and in practice, we observe the highlight scores varying smoothly across the frames, with only minor variance due to noise.

9.4.2 Implementation

We first extract the YOLOv5 (Jocher et al. 2022) and the Detectron2 (Wu et al. 2019) features for each frame in both the preferred highlight clips and the target videos in the training dataset. This takes about 30 seconds per frame on an NVIDIA Tesla A100 GPU. We then train our network using the Adam optimizer (Kingma and Ba 2014) with a batch size of 36, an initial learning rate of 1E−4 that we decay at a rate of 0.999 per epoch, and a weight decay of 3E−4. We train for 500 epochs on 8 NVIDIA Tesla V100 GPUs at a speed of about 1,480 seconds per epoch. Once the network is fully trained, inference takes about 9 seconds for each user using the same GPU configuration.

Hyperparameter Tuning We have experimented with batch sizes between 2 (minimum possible) and 36 (maximum allowable given our memory constraints), initial learning rates between 1E−5 (very slow convergence) and 1E−3 (optimization diverges), learning rate decays between 0.9 (very slow convergence) and 1 (optimization oscillates), weight decays between 1E−4 (optimization oscillates) and 5E−4 (very slow convergence), and training epochs between 100 and 1000 (validation loss saturates around 500 epochs).

9.4.3 Inference

Similar to the training set up, we consider a set of preferred highlight clips and target videos for each user during inference. We use the preferred highlight clips to learn the features h_z and h_y (Equations 9.10 and 9.11), and subsequently the per-frame highlight scores s_j (Equation 9.1). We also consider the scenario where the users' preferred highlight clips are not available, where our network falls back to content-based highlight detection (Bhattacharya et al. 2021c). Instead of the softmax operations in Eqs. 9.10 and 9.11, we perform uniform weighting along the feature dimension, *i.e.*, we have $h_z = \sum_{k=1}^{d_z} \psi_{\tau k}/d_z$ and $h_y = \sum_{k=1}^{d_y} \phi_{\tau k}/d_y$.

9.5 Experiments and Results

We provide details on the dataset we use for training and testing our method, the baselines we compare with, and the performance of our method on quantitative metrics. We also show the improvement in performance as we add more preferred highlight clips to our input, the benefits of the two components of object detection and pose detection used in our approach, and the contribution of each of the loss functions we use in training.

9.5.1 Dataset

We evaluate on the personal highlight detection dataset (PHD^2) introduced by del Molino and Gygli (Molino and Gygli 2018). This dataset consists of URLs of YouTube videos, IDs of annotators or "users", and the segments they selected as highlight clips from those videos as per their preferences. The last video that a user annotated is designated as the target video for that user. Since PHD^2 only provides the YouTube URLs, we scraped the videos from YouTube for our ex-

Table 9.1: **Mean Average Precision (mAP) and normalized Meaningful Summary Duration (nMSD) for Highlight Detection.** We report the numbers of all methods on PHD^2 (Molino and Gygli 2018). **Bold** indicates best.

	Method	mAP	nMSD
User-Agnostic	Random	0.112	0.536
	FCSN (Rochan, Ye, and Wang 2018)	0.152	-
	HighlightMe (Bhattacharya et al. 2021c)	0.200	-
User-Specific	Personalized Summ. (Panagiotakis, Papadakis, and Fragopoulou 2020)	0.216*	0.288*
	Video2GIF (Gygli, Song, and Cao 2016)	0.158	0.420
	PHD-GIF (Molino and Gygli 2018)	0.166	0.402
	Adaptive-FCSN (Rochan et al. 2020)	0.168	-
	PR-Net (Chen et al. 2021)	0.187	-
	MHA+Fusion (Ours)	**0.228**	**0.271**

* Numbers reported for a subset of the test set consisting of at least 5 selected highlights per user. For a similar test subset, our method has mAP of 0.262 and nMSD of 0.223.

periments, subject to video availability and IP restrictions. Both our training and testing sets consist of up to 15 highlight clips and a target video per user. The highlight clips are between 1 and 672 seconds long and have a mean length of 5.19 seconds. The target videos are between 1 and 37,434 seconds long and have a mean length of 443.29 seconds. The training set contains a total of 6,596 users and 26,390 videos. The testing set contains 727 users and 6,004 videos that do not overlap with any user or video in the training set.

9.5.2 Baselines

We compare with the methods Video2GIF by Gygli et al. (Gygli, Song, and Cao 2016), PHD-GIF by del Molino and Gygli (Molino and Gygli 2018), Personalized Summarization by Panagiotakis et al. (Panagiotakis, Papadakis, and Fragopoulou 2020), Adaptive-FCSN by Rochan et al. (Rochan et al. 2020), and PR-Net by Chen et al. (Chen et al. 2021), which, like our method, consider user-specific history when detecting highlights for target videos. Video2GIF computes shot boundaries in the videos and uses C3D features, used for action recognition, to map the shots

Table 9.2: **User-Specific Highlights: F-Scores for Video Summarization.** We report the F-scores of all methods on the SumMe dataset (Gygli et al. 2014). **Bold** indicates best.

Method	F-Score
SumMe baseline (Gygli et al. 2014)	0.394
Submodular mixtures (Gygli, Grabner, and Van Gool 2015)	0.397
DPP-LSTM (Zhang et al. 2016b)	0.386
Unsup. Adversarial LSTM (Mahasseni, Lam, and Todorovic 2017)	0.417
Sup. Deep RL (Zhou, Qiao, and Xiang 2018)	0.421
S^2N (Wei et al. 2018)	0.433
Adaptive-FCSN (Rochan et al. 2020)	0.444
HighlightMe (Bhattacharya et al. 2021c)	0.480
MHA+Fusion (Ours)	**0.526**

to highlight labels and train using a ranking loss. Adaptive-FCSN trains an encoder network to learn affine parameters based on the users' preferred highlight clips and uses those parameters to guide the detection of per-frame highlight labels in the target videos. We also compare with state-of-the-art user-agnostic approaches FCSN by Rochan et al. (Rochan, Ye, and Wang 2018), and HighlightMe by Bhattacharya et al. (Bhattacharya et al. 2021c), which detect highlights based only on the video contents. FCSN utilizes the GoogLeNet backbone to learn image-based features for per-frame highlight detection. HighlightMe leverages pose-based activities and facial expressions in human-centric videos and trains an autoencoder-based architecture to detect per-frame highlights.

9.5.3 Quantitative Comparison

We evaluate the performance of each method using the metrics of mean average precision (mAP) and normalized meaningful summary duration (nMSD). We compute mAP as the mean of the average precision of matching the highlight labels in each target video following (Rochan, Ye, and Wang 2018; Panagiotakis, Papadakis, and Fragopoulou 2020; Rochan et al. 2020; Bhattacharya

et al. 2021c; Chen et al. 2021), and nMSD at a recall rate of 0.5 following (Gygli, Song, and Cao 2016; Molino and Gygli 2018; Panagiotakis, Papadakis, and Fragopoulou 2020). As empirical lower bounds on mAP and nMSD, we also report the performance of randomly choosing the highlight frames in the target videos. Table 9.1 shows the mAP and the nMSD of each method on PHD^2 (Molino and Gygli 2018). Our method achieves the best mAP of 0.219 and nMSD of 0.271, outperforming the current best user-agnostic method HighlightMe (Bhattacharya et al. 2021c) by an absolute 2% and the current best user-specific methods of PR-Net (Chen et al. 2021) and Personalized Summarization (Panagiotakis, Papadakis, and Fragopoulou 2020) by absolute 4% and 6% respectively.

Beyond highlight detection, we also evaluate our method on the related problem of video summarization, following the approach of Rochan et al. (Rochan et al. 2020). Video summarization aims to condense videos into summaries of their contents, which may or may not correspond to the most interesting moments that users annotate as highlights. We test the performance of our trained network on the SumMe (Gygli et al. 2014) dataset, which consists of 25 videos totaling about 66 minutes, for video summarization. We compute the F-score of matching the summary frames and compare with the summarization methods listed by Rochan et al. (Rochan et al. 2020), their own highlight detection method, and the highlight detection method of HighlightMe (Bhattacharya et al. 2021c). We show the results in Table 9.2, where we observe our method improves the F-score by an absolute 4% over the next-best method of HighlightMe and 12% over the SumMe baseline. Our results further corroborate Rochan et al.'s argument that methods for video summarization can benefit from using networks with parameters pre-trained using loss functions for highlight detection.

Table 9.3: **User-Specific Highlights: Ablation Studies.** We report the mAPs of all ablated versions on PHD^2 (Molino and Gygli 2018). **Bold** indicates best. As we increase the number of preferred highlights, we observe that the mAP increases at a decreasing rate, indicating diminishing marginal benefits.

(a) Ablation 1: Changing the Number of Preferred Highlights.

# Pref. Clips	mAP
0	0.151
5	0.197
10	0.211
15 (max)	**0.228**

(b) Ablation 2: Using Only One Pre-Trained Backbone for Highlight Detection.

Backbone	mAP
YOLOv5 (Jocher et al. 2022)	0.164
Detectron2 (Wu et al. 2019)	0.193
Both	**0.228**

(c) Ablation 3: Ablating the Training Loss Functions.

W/o Loss	mAP
Label	0.155
Margin	0.192
Sparsity	0.163
None	**0.228**

9.5.4 Ablation Studies

We ablate our method in terms of (i) the number of preferred highlight clips used, (ii) the two backbones leveraging objects and poses, and (iii) the loss functions used for training our network.

To evaluate the benefit of using the preferred highlight clips, we train four versions of our network using at most 0, 5, 10, and 15 preferred highlight clips per user. We report the mAPs of the corresponding trained networks on PHD^2 in Table 9.3a. We observe a monotonic increase in performance as we increase the number of preferred highlight clips. This shows that our network learns more accurate representations of user-specific highlight preferences as we feed it more data on the preferred highlight clips. At the same time, the rate of increase drops as we increase the number of preferred highlight clips, indicating a diminishing marginal benefit of providing additional clips. We also note that the performance of our method without a substantial number of preferred highlight clips is worse than the user-agnostic method of HighlightMe (Bhattacharya et al. 2021c). This is because the highlights in PHD^2 are largely human-

centric, which HighlightMe is fine-tuned to detect. However, the performance of HighlightMe remains fixed, while ours grows and outperforms it as we increase the number of preferred highlight clips. We expect that combining our user-specific approach with the human-centric approach of HighlightMe will lead to the best performance on datasets such as PHD^2.

We also ablate each of our two network components, the one with the YOLOv5 backbone and the one with the Detectron2 backbone, and train the network using only the remaining component. For these experiments, there is no concatenation of the two components. Instead, we adjust the number of parameters of the first fully-connected layer of the prediction block HltPred (Figure 9.2 bottom-right block) accordingly. We show the mAP of each of these ablated networks on PHD^2 in Table 9.3b. We observe a more significant drop in mAP when ablating the component with the Detectron2 backbone, which results in the network not explicitly learning from the human activities in the videos. This is again a consequence of the annotated highlights in PHD^2 being largely human-centric.

Lastly, we ablate each of the three loss functions we describe in Section 9.4.1 and train our network using the remaining loss functions. We report the mAP of these ablated versions in Table 9.3c. Without the label loss, the label assignment becomes random, being only regularized by the margin and the sparsity losses. This results in a sharp drop in the mAP. Without the margin loss, all the predicted highlight scores are closer to the threshold ζ, resulting in more noise in the highlight scores around the threshold ζ and consequently more confusion between the highlight and the non-highlight labels. Without the sparsity loss, more frames are incorrectly labeled as highlights, again resulting in a sharp drop in the mAP.

Table 9.4: User-Specific Highlights: Mean average precision on the DSH dataset (Sun, Farhadi, and Seitz 2014). **Bold** indicates best, underline indicates second-best.

Domain	RRAE (Yang et al. 2015)	Video2 GIF (Gygli, Song, and Cao 2016)	LSVM (Sun, Farhadi, and Seitz 2014)	Less is More (Xiong et al. 2019)	HighlightMe (Bhattacharya et al. 2021c)	Ours
dog show	0.49	0.31	0.60	0.58	**0.63**	0.60
gymnastics	0.35	0.34	0.41	0.44	**0.73**	**0.73**
parkour	0.50	0.54	0.61	0.67	**0.72**	0.71
skating	0.25	0.55	0.62	0.58	**0.64**	0.62
skiing	0.22	0.33	0.36	0.49	0.52	**0.61**
surfing	0.49	0.54	0.61	**0.65**	0.62	0.58
Mean	0.38	0.46	0.54	0.57	**0.64**	0.62

Table 9.5: User-Specific Highlights: Mean average precision on the TVSum dataset (Song et al. 2015). **Bold** indicates best, underline indicates second-best.

Domain	MBF (Chu, Song, and Jaimes 2015)	KVS (Potapov et al. 2014)	CVS (Panda and Roy-Chowdhury 2017)	Adv-LSTM (Mahasseni, Lam, and Todorovic 2017)	Less is More (Xiong et al. 2019)	HighlightMe (Bhattacharya et al. 2021c)	Ours
BK	0.31	0.34	0.33	0.42	**0.66**	0.57	0.60
BT	0.37	0.42	0.40	0.48	0.69	**0.93**	0.85
DS	0.36	0.39	0.38	0.47	**0.63**	0.60	0.62
FM	0.37	0.40	0.37	0.46	0.43	**0.88**	0.77
GA	0.33	0.40	0.38	0.48	0.61	0.50	**0.63**
MS	0.41	0.42	0.40	0.49	0.54	0.50	**0.55**
PR	0.33	0.40	0.38	0.47	0.53	**0.84**	0.70
PK	0.32	0.38	0.35	0.46	0.60	**0.76**	0.68
VT	0.30	0.35	0.33	0.42	0.56	**0.65**	0.61
VU	0.36	0.44	0.41	0.47	0.50	**0.77**	0.75
Mean	0.35	0.40	0.37	0.46	0.58	**0.70**	0.68

9.5.5 Performance in Domain-Specific Highlight Detection

For the sake of completeness, we further test the performance of our method on domain-specific highlight detection datasets that do not contain any annotated user-specific preferred clips. We test on the domain-specific highlight dataset (DSH) (Sun, Farhadi, and Seitz 2014) and the title-based summarization dataset (TVSum) (Song et al. 2015). DSH consists of six domains, namely, dog show, gymnastics, parkour, skating, skiing, and surfing. Each domain contains about 100 videos, and the total duration over all the six domains is around 1,430 minutes. TVSum consists of 50 videos, totaling around 210 minutes, from ten domains, namely, beekeeping (BK), bike

tricks (BT), dog show (DS), flash mob (FM), grooming animal (GA), making sandwich (MS), parade (PR), parkour (PK), vehicle tire (VT), and vehicle unstuck (VU). However, neither of these datasets provide any user preferences. Therefore, to test our method on these datasets, we use the testing setup with no preferred highlight clips for any user (Section 9.4.3). We show the results on the DSH dataset in Table 9.4, and on the TVSum dataset in Table 9.5. The results of all the methods we compare with are taken from Bhattacharya et al. (Bhattacharya et al. 2021c). We observe that our method has the second-best mean average precision of all the methods over all the domains. Our method is also at par with the human-centric approach of HighlightMe (Bhattacharya et al. 2021c) for most of the domains, being only significantly outperformed in domains that have an abundance of both face and pose modalities and no other detected objects in the highlighted segments, *e.g.*, flash mob (FM) and parade (PR) in TVSum. The precision of our method also suffers in the surfing domain in DSH and the bike tricks (BT) domain in TVSum where our method assigns highlight labels to many objects and human interactions not related to the ground-truth highlights (such as prepping surfboards, people interviewing, cars on the street). As part of our future work, for such scenarios, we can incorporate more human-centric modalities into our pipeline or provide domain-specific videos as the preferred clips to improve performance.

9.5.6 Qualitative Results

We show sample visual results in Figure 9.5. We show sample frames from the users' preferred highlights, the ground-truth highlights in the users' target videos, and the highlights detected by HighlightMe (Bhattacharya et al. 2021c) and our method. For our method, we observe content similarities between the users' preferred highlights and the target videos, which also largely

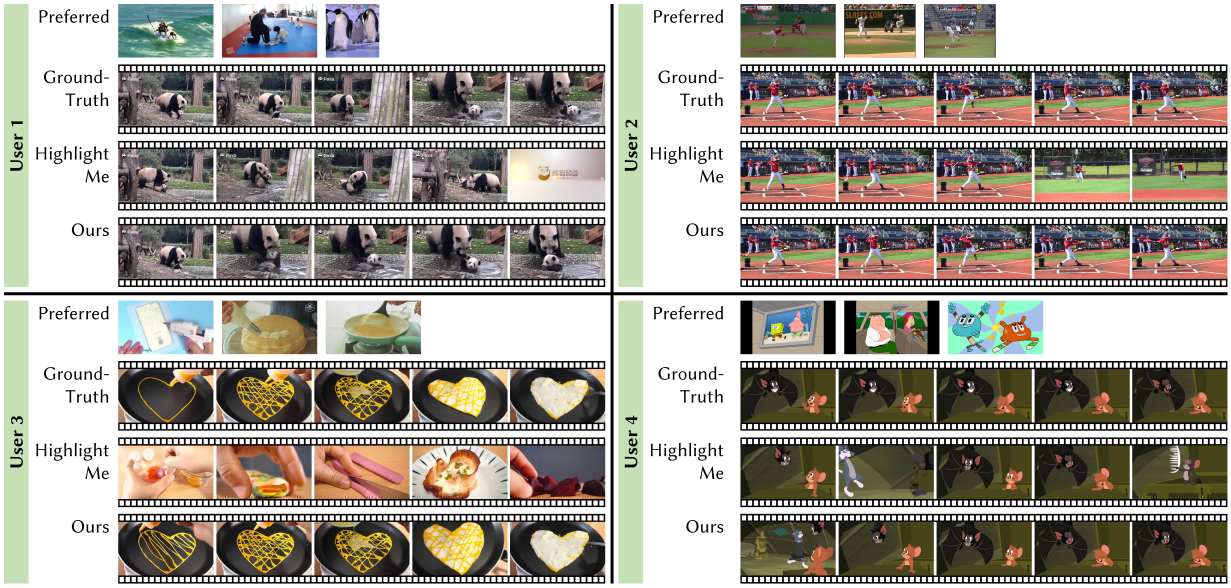


Figure 9.5: **User-Specific Highlights: Qualitative Comparisons.** We show qualitative results of our method and the current best baseline of HighlightMe (Bhattacharya et al. 2021c) for four users in the testing set of PHD^2(Molino and Gygli 2018). For each user, we show sample frames of (i) their preferred clips, (ii) ground-truth highlights selected by them from their target videos, (iii) highlights detected by HighlightMe (Bhattacharya et al. 2021c), and (iv) highlights detected by our method. For each user, we observe that our method matches the ground-truth more closely than HighlightMe (Bhattacharya et al. 2021c). For user 1, our method looks at the preferences containing surfing, children, and animals, and detects clips of the pandas playing in the pool. For user 2, our method identifies that the user only selects clips of baseball hits and detects similar clips. User 3 prefers videos of creating designs and cooking, and detects clips of designer fried eggs. User 4 prefers looking at the interactions between two characters, and detects clips consisting of two or more interacting characters.

match the ground-truth. By contrast, HighlightMe (Bhattacharya et al. 2021c) only looks at the target videos and does not adapt to the users' preferences. Our method also fails when the users break their preference patterns in the target videos, or the target videos did not contain any features similar to the preferred highlights. Please refer to our video results for details.

9.6 Conclusion

We have presented a method for user-specific highlight detection based on multi-head attention fusion of features from the users' preferred highlight clips. Our method only requires the

highlight clips and not the clips users did not select as highlights, thus significantly reducing the data overhead for training. Our multi-head attention mechanism leverages both objects and human activities in videos to learn highlights based on their spatial and temporal variations and mutual interactions. We also perform experiments to show that our method advances the state-of-the-art in both highlight detection in a large-scale dataset with annotated user preferences and the related problem of video summarization. We also show that, unlike user-agnostic highlight detection methods, the performance of our method increases as we increase the maximum number of preferred highlight clips considered for each user.

CHAPTER 10

Conclusion and Future Directions

We now take a step back to understand how all our proposed methodologies connect to our bigger picture. Our main contribution has been the development and implementation of automated techniques for affect detection and synthesis using body expressions, and their application in video understanding tasks. We have also incorporated the subjectivity of affect understanding and individualistic styles of expressing affects throughout our work, by learning from both variable affect representations and integrating the variable agent styles into our learning methods. Our methods can also learn efficiently from large-scale data collected using commercially available devices such as video cameras, thus making it possible to deploy it on affordable hardware and enabling democratized usage.

10.1 Summary of Our Work

We began by developing a combination of a supervised classifier network to detect one-hot affect labels from gaits and a generative network to synthesize gaits given one-hot affect labels to provide additional training data to our supervised detection method. While this method advances the state-of-the-art in affect detection, the saturation of the network performance indicates that our generative network may have overfitted to the small real-world dataset at our disposal. Second, our classifier network was fully supervised and therefore could not make use of other real-world gait datasets which do not have affect labels. We also assumed our affects to be one-hot discrete, which is a strong assumption that does not allow for the subjectivity of the individual observers' affect perception.

To overcome some of these limitations, we then designed a semi-supervised learning method that can leverage widely available unlabeled gait-based datasets for affect detection in addition to small-scale labeled datasets. This method also incorporates multi-hot affect labels to allow for some subjectivity in affect perception. However, it still considers discrete affect labels, which do not span all possible affects. Also, like our previous method, it considers gaits in isolation, without any other modalities or context. This makes its utility limited in real-world scenarios that contain multiple simultaneous modalities of affect expressions (such as speech, facial expressions, and body expressions) and additional context of the human-human interactions. Nevertheless, this method makes good improvements over our previous method and sets a new baseline for affect detection.

Moving further towards our goal of achieving real-world affect understanding, we next considered the more real-world centric problem of affect synthesis. Affect synthesis requires

us to not only learn various affect patterns from data, but also learn how to use them to create novel data that we can use in real-world applications. Our first method for affect synthesis generates affective gaits for digital human characters or virtual agents, as they walk on predefined trajectories. To better incorporate subjectivity in affect perception in this method, we have also used probabilistic labels distributed over the four affect classes instead of multi-hot labels. While probabilistic labels allow for smooth control over the affects and affect transitions during gait synthesis, it still does not span the full VAD space. This method also uses only pose-based affective and movement features to generate affective gaits, and our subsequent objective was to incorporate information corresponding to other modalities such as speech and facial expressions as well as the context of human-human interactions to emulate more real-world scenarios.

To this end, our next method tackled the problem of synthesizing affective body gestures synced with the virtual agents' speech, or affective *co-speech* body gestures. This method takes as input the intended affect as a point in the VAD space, some of the agents' characteristics such as gender and handedness as binary labels, and the text transcript of the agents' speech to synthesize their body gestures. To the best of our knowledge, this was the first method on synthesizing affective co-speech body gestures, and it had room for improvement. First, this method maps complete text sentences to gestures, which we could extend to phrase-level mapping to synthesize affects with finer details. Second, the synthesized virtual agents returned to the rest pose after every sentence. One way to improve this is to learn the transitions of the gestures between sentences. Lastly, we assumed binary labels for stylistic features such as gender and handedness. A possible extension is to relax these assumptions and design a continuous distribution space of speakers based on their individual speaking styles.

We have addressed these limitations with our follow-up method, which synthesizes affective co-speech body gestures for multiple sentences, and assume a continuous space of speaker styles instead of discrete speaker attributes. This method also does not assume any availability of intended affect information, but learns it from the speech audio and the gesture data using separate encoders. Crucially, this method consists of a generative adversarial framework that we have specifically designed to learn to match the affective information in the synthesized body gestures with the affective information in original speaker gestures in the dataset. As a result, this method significantly advances the baseline for affective co-speech gesture synthesis. Nevertheless, we identify scopes for further improvement. First, we have not built a mechanism to control the affective expressions, but compute them automatically from the input modalities. Further investigation on the interface between affective expressions from the speech and the gestures, especially when expressing contradictory cues such as sarcasm and irony, can lead to more controllable synthesis. Second, using a finer representation of poses can also improve the synthesis quality since affective expressions in the gestures are often associated with even more subtle movements not captured by our current representation. Lastly, our method synthesizes only body gestures synced with speech. It is possible to incorporate additional real-world context by considering additional modalities such as co-speech facial expressions that synchronize with the co-speech body gestures.

Our next and latest method on affective human motion synthesis looks into this synchronous synthesis of affective co-speech facial expressions and body gestures, which we collectively term *unified expressions*. This method extends our previously introduced generative adversarial framework to consider the multimodal embedding space of speech, facial expressions, and body gestures, and encourages the synthesis of synchronous unified expressions via

dedicated face and body pose decoders coupled with appropriate reconstruction and adversarial losses. As scopes for future extensions, we note that this method uses sparse face landmarks and pose joints to synthesize co-speech face and pose expressions. We do not account for more fine-grained expressions, particularly on the cheeks and other parts of the lower face, the hands, and the fingers. To synthesize these expressions, we one can use more detailed face meshes and additional pose joints that can be extracted from videos. Further, given the sparsity of our face and pose representations and the noise associated with extracting them from videos, the quality of our synthesized motions do not match those synthesized from high-end facial scans and motion-capture data. We aim to bridge this gap by building techniques to develop more robust face and pose representations from videos. We also plan to combine our work with lower-body actions such as sitting, standing, and walking to synthesize 3D animated digital humans in a wider variety of scenarios. In terms of its running-time cost, our method uses high-end GPUs to obtain real-time performance. We plan to explore knowledge distillation techniques to reduce our running-time cost and implement our method in real-time on commodity devices such as digital personal assistants.

Having developed methods for affective human motion detection and synthesis, we have also explored their applications in the area of video understanding, specifically, in video highlight detection. To this end, we have developed two highlight detection methods.

Our first method detects highlights from human-centric videos, by learning from human actions and affects in videos without any supervision in the form of highlight labels. While this method leads to state-of-the-art performance in highlight detection on average, its current implementation considers only faces and poses as the human modalities. However, many videos (*e.g.*, videos on grooming animals, making sandwich in the TVSum dataset Song et al.

2015) exhibit other modalities such as hands and fingers. Nonetheless, our theoretical network design can accommodate any number of modalities, and we can extend its implementation to incorporate more human-centric modalities as needed. Also, our method, being human-centric, does not offer much improvements in videos from domains that are not fully human-centric, *e.g.*, videos focusing on other living beings, inanimate objects, and natural scenes. To address this limitation, our follow-up method considers the presence of both humans and non-human entities in videos to detect highlights.

The key contribution of our follow-up method is to detect user-specific highlights from videos. Its goal is to learn individual user preferences based on the highlight clips they have previously watched, and predict highlights for them accordingly from novel videos. This method leverages the multi-head attention mechanism to learn content-based similarities between the prior videos users had watched and the novel videos where we are required to detect highlights. We design our network to consider both the human actions and affects and the non-human entities in the videos to learn the feature representations for the video contents. This method sets a new baseline for video highlight detection, with scopes for further improvement. First, we have considered users' highlight preferences only at training time and do not consider user feedback at inference. To this end, we can plan to combine our method with complementary methods that rely on user feedback to gradually learn their preferences. Second, we can extend our attention mechanism to incorporate a broader definition of content, including the associated audio, semantic segmentation of the video frames, and other human-centric modalities such as faces. Lastly, we do not fine-tune our method to video categories based on domains (*e.g.*, surfing videos) or the constituent subjects (*e.g.*, human-centric videos). Rather, we design an approach to combine diverse highlight clips assuming the presence of only generic elements, *i.e.*, objects

and humans. Nevertheless, our method is suitable for fine-tuning as necessary to improve user-specific highlight detection in various video categories.

10.2 Applications and Future Research

Looking at the broader scope of our work, we note its impact on modern industry applications and impact on future research.

In terms of industry applications, our work is integral to the design and implementation of social VR. We identify three different ways of developing virtual agents for such social VR applications. One is the virtual assistant that can work as embodied counterparts of the digital assistants of today. Another is in virtual chat rooms, where people can interact with each other through expressive virtual avatars, thus ensuring both privacy and engagement. And the third is developing fully immersive large-scale virtual worlds, where people can interact socially in semi-structured or unstructured environments, using highly plausible virtual avatars and in-environment virtual agents at high render resolutions.

Our work also has some limitations that can be addressed as future research directions. We note that our affect body expression synthesis is automatic given either affect labels or other modalities such as speech. While these inputs allow indirect control over the body expression affects, we do not consider a mechanism to directly control how the body expressions should look. This is particularly useful when adapting to specific speaker styles not captured in our datasets, especially from a wider variety of social and cultural backgrounds. There has been some recent work in this direction where the gestures can be directly modified using controllable features and parameters (Habibie et al. 2021), and we can consider incorporating social, cultural and affective expressions to such methods.

Also, while we perform affect detection and synthesis on gaits and gestures, our methods do not consider all possible modalities of body expressions at once. We have not worked with some of smaller scale and higher frequency modalities such as eye gaze and hand movements, particularly because it is challenging to collect data that uniformly represent modalities appearing at such different scales and frequencies. Nevertheless, it is possible to extend our work to include multi-scale modalities at very different frequencies.

In terms of plausible motion synthesis, our work only considers optimizing data-driven loss terms such as Euclidean distances between the synthesized and the original human motions, and differences between affective features either computed or learned from the data. To improve the plausibility of the synthesized motions, we can also include kinematic constraints of the human body (An 1984), such as optimizing the energy expended by the body muscles when performing different motions.

To further enable democratized usage of our proposed methodologies, we can continue to optimize the design and training of our proposed neural networks. Particularly, we can focus on reducing the parameter load via knowledge transfer and distillation mechanisms (Ahn et al. 2019) to eventually make it suitable for edge devices with low compute power such as smartphones and head-mounted displays.

In the longer term, our work can also extend into broader interpretations of affects, including reactive affects in people working in groups, and individual behavior understanding, both in isolation and during social interactions (Sinha, Bai, and Cassell 2022).

Bibliography

Ahn, Sungsoo, Shell Xu Hu, Andreas Damianou, Neil D. Lawrence, and Zhenwen Dai (June 2019). "Variational Information Distillation for Knowledge Transfer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ahsan, Unaiza, Chen Sun, and Irfan Essa (2018). "Discrimnet: Semi-supervised action recognition from videos using generative adversarial networks". In: *arXiv preprint arXiv:1801.07230*.

Ahuja, C. and L. Morency (2019). "Language2Pose: Natural Language Grounded Pose Forecasting". In: *2019 International Conference on 3D Vision (3DV)*, pp. 719–728. DOI: 10.1109/3DV.2019.00084.

Ahuja, Chaitanya, Dong Won Lee, Yukiko I. Nakano, and Louis-Philippe Morency (2020). "Style Transfer for Co-Speech Gesture Animation: A Multi-Speaker Conditional-Mixture Approach". In: *European Conference on Computer Vision*.

Albanie, Samuel, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman (2018). "Emotion Recognition in Speech Using Cross-Modal Transfer in the Wild". In: *Proceedings of the 26th ACM International Conference on Multimedia*. MM '18. Seoul, Republic of Korea: Association for Computing Machinery, 292–301. ISBN: 9781450356657. DOI: 10.1145/3240508.3240578.

Alexanderson, Simon, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow (2020). "Style-Controllable Speech-Driven Gesture Synthesis Using Normalising Flows". In: *Computer Graphics Forum* 39.2, pp. 487–496. DOI: 10.1111/cgf.13946. eprint: 10.1111/cgf.13946.

Allen, Chris T, Karen A Machleit, and Susan Schultz Kleine (1992). "A comparison of attitudes and emotions as predictors of behavior at diverse levels of behavioral experience". In: *Journal of consumer research* 18.4, pp. 493–504.

Altarriba, Jeanette, Dana M Basnight, and Tina M Canary (2003). "Emotion representation and perception across cultures". In: *Online readings in psychology and culture* 4.1, pp. 1–17.

Ambady, Nalini and Robert Rosenthal (1992). "Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis." In: *Psychological bulletin* 111.2, p. 256.

An, Kai-Nan (1984). "Kinematic analysis of human movement". In: *Annals of biomedical engineering* 12.6, pp. 585–597.

Argyle, Michael (2013). *Bodily communication*. Routledge.

Aristidou, Andreas and Joan Lasenby (2011). "FABRIK: A fast, iterative solver for the Inverse Kinematics problem". In: *Graphical Models* 73.5, pp. 243–260. ISSN: 1524-0703. DOI: https://doi.org/10.1016/j.gmod.2011.05.003.

Atcheson, Mia, Vidhyasaharan Sethu, and Julien Epps (2017). "Gaussian Process Regression for Continuous Emotion Recognition with Global Temporal Invariance". In: *IJCAI-W*, pp. 34–44.

Aviezer, Hillel, Yaacov Trope, and Alexander Todorov (2012). "Body cues, not facial expressions, discriminate between intense positive and negative emotions". In: *Science* 338.6111, pp. 1225–1229.

Babu, Ashwin Ramesh, Akilesh Rajavenkatanarayanan, James Robert Brady, and Fillia Makedon (2018). "Multimodal approach for cognitive task performance prediction from body postures, facial expressions and EEG signal". In: *Proceedings of the Workshop on Modeling Cognitive Processes from Multimodal Data*. ACM, p. 2.

Badler, Norman I, Cary B Phillips, and Bonnie Lynn Webber (1993). *Simulating humans: computer graphics animation and control*. Oxford University Press.

Bagher Zadeh, AmirAli, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency (July 2018). "Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 2236–2246. DOI: 10.18653/v1/P18-1208.

Banerjee, Abhishek, Uttaran Bhattacharya, and Aniket Bera (June 2022). "Learning Unseen Emotions from Gestures via Semantically-Conditioned Zero-Shot Perception with Adversarial Autoencoders". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.1, pp. 3–10. DOI: 10.1609/aaai.v36i1.19873.

Barrett, Lisa Feldman (2017). *How emotions are made: The secret life of the brain*. Houghton Mifflin Harcourt.

Barrett, Lisa Feldman, Batja Mesquita, and Maria Gendron (2011). "Context in emotion perception". In: *Current Directions in Psychological Science* 20.5, pp. 286–290. DOI: 10.1177\%2F0963721411422522.

Bauer, Andrea, Klaas Klasing, Georgios Lidoris, Quirin Mühlbauer, Florian Rohrmüller, Stefan Sosnowski, Tingting Xu, Kolja Kühnlenz, Dirk Wollherr, and Martin Buss (2009). "The autonomous city explorer: Towards natural human-robot interaction in urban environments". In: *IJSR* 1.2, pp. 127–140.

Baur, Tobias, Ionut Damian, Patrick Gebhard, Kaska Porayska-Pomsta, and Elisabeth André (2013). "A Job Interview Simulation: Social Cue-Based Interaction with a Virtual Character". In: *2013 International Conference on Social Computing*, pp. 220–227. DOI: 10.1109/SocialCom.2013.39.

Bell, Robert M. and Yehuda Koren (Dec. 2007). "Lessons from the Netflix Prize Challenge". In: *SIGKDD Explor. Newsl.* 9.2, 75–79. ISSN: 1931-0145. DOI: 10.1145/1345448.1345465.

Bengio, Samy, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer (2015). "Scheduled sampling for sequence prediction with recurrent neural networks". In: *Advances in Neural Information Processing Systems*, pp. 1171–1179.

Bhattacharya, Uttaran, Elizabeth Childs, Nicholas Rewkowski, and Dinesh Manocha (2021a). "Speech2-AffectiveGestures: Synthesizing Co-Speech Gestures with Generative Adversarial Affective Expression Learning". In: *Proceedings of the 29th ACM International Conference on*

Multimedia. New York, NY, USA: Association for Computing Machinery, 2027–2036. ISBN: 9781450386517.

Bhattacharya, Uttaran, Trisha Mittal, Rohan Chandra, Tanmay Randhavane, Aniket Bera, and Dinesh Manocha (2020). "STEP: Spatial Temporal Graph Convolutional Networks for Emotion Perception from Gaits". In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI'20. AAAI Press, 1342–1350.

Bhattacharya, Uttaran, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha (2021b). "Text2Gestures: A Transformer-Based Network for Generating Emotive Body Gestures for Virtual Agents". In: *2021 IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR)*. IEEE.

Bhattacharya, Uttaran, Nicholas Rewkowski, Pooja Guhan, Niall L. Williams, Trisha Mittal, Aniket Bera, and Dinesh Manocha (2020). "Generating Emotive Gaits for Virtual Agents Using Affect-Based Autoregression". In: *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 24–35. DOI: 10.1109/ISMAR50242.2020.00020.

Bhattacharya, Uttaran, Christian Roncal, Trisha Mittal, Rohan Chandra, Kyra Kapsaskis, Kurt Gray, Aniket Bera, and Dinesh Manocha (2020). "Take an Emotion Walk: Perceiving Emotions from Gaits Using Hierarchical Attention Pooling and Affective Mapping". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, pp. 145–163. ISBN: 978-3-030-58607-2.

Bhattacharya, Uttaran, Gang Wu, Stefano Petrangeli, Viswanathan Swaminathan, and Dinesh Manocha (Oct. 2021c). "HighlightMe: Detecting Highlights From Human-Centric Videos". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8157–8167.

— (2022). "Show Me What I Like: Detecting User-Specific Video Highlights Using Content-Based Multi-Head Attention". In: *Proceedings of the 30th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery. ISBN: 978145039203.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. ISSN: 2307-387X.

Bruna, Joan, Wojciech Zaremba, Arthur Szlam, and Yann LeCun (2013). "Spectral networks and locally connected networks on graphs". In: *arXiv:1312.6203*.

Busso, Carlos, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan (Nov. 2008). "IEMOCAP: interactive emotional dyadic motion capture database". In: *Language Resources and Evaluation* 42.4, p. 335. DOI: 10.1007/s10579-008-9076-6.

Cai, Haoye, Chunyan Bai, Yu-Wing Tai, and Chi-Keung Tang (2018a). "Deep video generation, prediction and completion of human action sequences". In: *ECCV*, pp. 366–382.

Cai, Sijia, Wangmeng Zuo, Larry S. Davis, and Lei Zhang (Sept. 2018b). "Weakly-supervised Video Summarization using Variational Encoder-Decoder and Web Prior". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Carreira, Joao and Andrew Zisserman (2017). "Quo vadis, action recognition? a new model and the kinetics dataset". In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308.

Castillo, Gabriel and Michael Neff (2019). "What Do We Express without Knowing? Emotion in Gesture". In: *Proceedings of the 18th International Conference on Autonomous Agents and*

MultiAgent Systems. AAMAS '19. Montreal QC, Canada: International Foundation for Autonomous Agents and Multiagent Systems, 702–710. ISBN: 9781450363099.

Chanel, Guillaume, Julien Kronegg, Didier Grandjean, and Thierry Pun (2006). "Emotion assessment: Arousal evaluation using EEG's and peripheral physiological signals". In: *IWMCRCS*. Springer, pp. 530–537.

Chen, Binghui and Weihong Deng (June 2019). "Hybrid-Attention Based Decoupled Metric Learning for Zero-Shot Image Retrieval". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, C., R. Jafari, and N. Kehtarnavaz (2015). "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor". In: *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 168–172. DOI: 10.1109/ICIP.2015.7350781.

Chen, Runnan, Penghao Zhou, Wenzhe Wang, Nenglun Chen, Pai Peng, Xing Sun, and Wenping Wang (Oct. 2021). "PR-Net: Preference Reasoning for Personalized Video Highlight Detection". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7980–7989.

Chen, Ying, Wenjun Hou, Xiyao Cheng, and Shoushan Li (2018). "Joint Learning for Emotion Classification and Emotion Cause Detection". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 646–651.

Chiu, Chung-Cheng, Louis-Philippe Morency, and Stacy Marsella (2015). "Predicting Co-verbal Gestures: A Deep and Temporal Modeling Approach". In: *Intelligent Virtual Agents*. Cham: Springer International Publishing, pp. 152–166. ISBN: 978-3-319-21996-7.

Chiu, Mangtik, Jiayu Shu, and Pan Hui (2018). "Emotion Recognition through Gait on Mobile Devices". In: *PerCom Workshops*. IEEE, pp. 800–805.

Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang (2003). "Automatic video summarization by graph modeling". In: *Proceedings Ninth IEEE International Conference on Computer Vision*, 104–109 vol.1. DOI: 10.1109/ICCV.2003.1238320.

Choutas, Vasileios, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid (2018). "Potion: Pose motion representation for action recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7024–7033.

Chowanda, Andry, Peter Blanchfield, Martin Flintham, and Michel Valstar (2016). "Computational Models of Emotion, Personality, and Social Relationships for Interactions in Games: (Extended Abstract)". In: *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '16. Singapore, Singapore: International Foundation for Autonomous Agents and Multiagent Systems, 1343–1344. ISBN: 9781450342391.

Chu, Wen-Sheng, Yale Song, and Alejandro Jaimes (June 2015). "Video Co-Summarization: Video Summarization by Visual Co-Occurrence". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chuah, Joon Hao, Brent Rossen, and Benjamin Lok (2009). "Automated Generation of Emotive Virtual Humans". In: *Intelligent Virtual Agents*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 490–491. ISBN: 978-3-642-04380-2.

Cisco (2020). "Annual Internet Report (2018–2023)". In: *CISCO White paper*.

Clavel, Céline, Justine Plessier, Jean-Claude Martin, Laurent Ach, and Benoit Morel (2009). "Combining Facial and Postural Expressions of Emotions in a Virtual Character". In: *Intelli-*

gent Virtual Agents. Ed. by Zsófia Ruttkay, Michael Kipp, Anton Nijholt, and Hannes Högni Vilhjálmsson. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 287–300.

Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter (2015). "Fast and accurate deep network learning by exponential linear units (elus)". In: *arXiv preprint arXiv:1511.07289*.

CMU-MOCAP (2018). "CMU Graphics Lab Motion Capture Database". In: *http://mocap.cs.cmu.edu/*.

Covington, Paul, Jay Adams, and Emre Sargin (2016). "Deep Neural Networks for YouTube Recommendations". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys '16. Boston, Massachusetts, USA: Association for Computing Machinery, 191–198. ISBN: 9781450340359. DOI: 10.1145/2959100.2959190.

Crenn, Arthur, Rizwan Ahmed Khan, Alexandre Meyer, and Saida Bouakaz (2016). "Body expression recognition from animated 3D skeleton". In: *IC3D*. IEEE, pp. 1–7.

Cudeiro, Daniel, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J. Black (June 2019). "Capture, Learning, and Synthesis of 3D Speaking Styles". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cui, Qiongjie, Huaijiang Sun, and Fei Yang (June 2020). "Learning Dynamic Relationships for 3D Human Motion Prediction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dabral, Rishabh, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain (2018). *Learning 3D Human Pose from Structure and Motion*. Lecture Notes in Computer Science. Springer International Publishing.

Dael, Nele, Marcello Mortillaro, and Klaus R Scherer (2012). "Emotion expression in body action and posture". In: *Emotion* 12.5, p. 1085. DOI: 10.1037/a0025737.

Daoudi, Mohamed, Stefano Berretti, Pietro Pala, Yvonne Delevoye, and Alberto Del Bimbo (2017). "Emotion recognition by body movement representation on the manifold of symmetric positive definite matrices". In: *ICIAP*. Springer, pp. 550–560.

De Gelder, Beatrice (2006). "Towards the neurobiology of emotional body language". In: *Nature Reviews Neuroscience* 7.3, pp. 242–249.

Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst (2016). "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., pp. 3844–3852.

Deng, Jun, Xinzhou Xu, Zixing Zhang, Sascha Frühholz, and Björn Schuller (2018). "Semisupervised autoencoders for speech emotion recognition". In: *IEEE/ACM ASLP* 26.1, pp. 31–43.

Denham, Susanne A, Elizabeth Workman, Pamela M Cole, Carol Weissbrod, Kimberly T Kendziora, and Carolyn Zahn-Waxler (2000). "Prediction of externalizing behavior problems from early to middle childhood". In: *Development and Psychopathology* 12.1, pp. 23–45.

DeVault, David, Ron Artstein, Grace Benn, Teresa Dey, Ed Fast, Alesia Gainer, Kallirroi Georgila, Jon Gratch, Arno Hartholt, Margaux Lhommet, et al. (2014). "SimSensei Kiosk: A virtual human interviewer for healthcare decision support". In: *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1061–1068.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis,

Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

Du, Han, Erik Herrmann, Janis Sprenger, Noshaba Cheema, Somayeh Hosseini, Klaus Fischer, and Philipp Slusallek (2019). "Stylistic Locomotion Modeling with Conditional Variational Autoencoder." In: *Eurographics (Short Papers)*, pp. 9–12.

Ekman, P and W V Friesen (1967a). "Head and body cues in the judgment of emotion: A reformulation". In: *Perceptual and motor skills*.

— (1967b). "Head and body cues in the judgment of emotion: A reformulation". In: *Perceptual and motor skills*.

Ekman, Paul (1993). "Facial expression and emotion". In: *American psychologist* 48.4, p. 384.

Ekman, Paul and Wallace V Friesen (1969). "The repertoire of nonverbal behavior: Categories, origins, usage, and coding". In: *semiotica* 1.1, pp. 49–98.

— (1978). "Facial action coding system". In: *Environmental Psychology & Nonverbal Behavior*.

Ekman, Paul and Dacher Keltner (1970). "Universal facial expressions of emotion". In: *California mental health research digest* 8.4, pp. 151–158.

Fabian Benitez-Quiroz, C., Ramprakash Srinivasan, and Aleix M. Martinez (June 2016). "EmotioNet: An Accurate, Real-Time Algorithm for the Automatic Annotation of a Million Facial Expressions in the Wild". In: *CVPR*.

Fan, Yin, Xiangju Lu, Dian Li, and Yuanliu Liu (2016). "Video-based emotion recognition using CNN-RNN and C3D hybrid networks". In: *ICMI*. ACM, pp. 445–450.

Feichtenhofer, Christoph, Axel Pinz, and Richard Wildes (2016). "Spatiotemporal residual networks for video action recognition". In: *NeurIPS*, pp. 3468–3476.

Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman (2016). "Convolutional two-stream network fusion for video action recognition". In: *CVPR*, pp. 1933–1941.

Fernández-Dols, José-Miguel and Maria-Angeles Ruiz-Belda (1995). "Expression of emotion versus expressions of emotions". In: *Everyday conceptions of emotion*. Springer, pp. 505–522.

Ferstl, Ylva and Rachel McDonnell (2018a). "A Perceptual Study on the Manipulation of Facial Features for Trait Portrayal in Virtual Agents". In: *Proceedings of the 18th International Conference on Intelligent Virtual Agents*. IVA '18. Sydney, NSW, Australia: Association for Computing Machinery, 281–288. ISBN: 9781450360135. DOI: 10.1145/3267851.3267891.

— (Nov. 2018b). "IVA: Investigating the use of recurrent motion modelling for speech gesture generation". In: *IVA '18 Proceedings of the 18th International Conference on Intelligent Virtual Agents*.

Ferstl, Ylva, Michael Neff, and Rachel McDonnell (2019). "Multi-Objective Adversarial Gesture Generation". In: *Motion, Interaction and Games*. MIG '19. Newcastle upon Tyne, United Kingdom: Association for Computing Machinery. ISBN: 9781450369947. DOI: 10.1145/3359566.3360053.

Fleiss, Joseph L (1971). "Measuring nominal scale agreement among many raters." In: *Psychological bulletin* 76.5, p. 378.

Franco, Annalisa, Antonio Magnani, and Dario Maio (2020). "A multimodal approach for human activity recognition based on skeleton and RGB data". In: *Pattern Recognition Letters* 131, pp. 293 –299. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2020.01.010.

Frick, Robert W (1985). "Communicating emotion: The role of prosodic features". In: *Psychological Bulletin* 97.3, p. 412. DOI: 10.1037/0033-2909.97.3.412.

Geitgey, Adam (2020). *Face Recognition*.

Gendron, Maria, Debi Roberson, Jacoba Marietta van der Vyver, and Lisa Feldman Barrett (2014). "Perceptions of emotion from facial expressions are not culturally universal: evidence from a remote culture." In: *Emotion* 14.2, p. 251. DOI: 10.1037/a0036052.

Giannopoulos, Panagiotis, Isidoros Perikos, and Ioannis Hatzilygeroudis (2018). "Deep Learning Approaches for Facial Emotion Recognition: A Case Study on FER-2013". In: *Advances in Hybridization of Intelligent Methods: Models, Systems and Applications*. Ed. by Ioannis Hatzilygeroudis and Vasile Palade. Cham: Springer International Publishing, pp. 1–16. ISBN: 978-3-319-66790-4. DOI: 10.1007/978-3-319-66790-4_1.

Ginosar, Shiry, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik (June 2019). "Learning Individual Styles of Conversational Gesture". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Girdhar, Rohit, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran (June 2018). "Detect-and-Track: Efficient Pose Estimation in Videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gong, Boqing, Wei-Lun Chao, Kristen Grauman, and Fei Sha (2014). "Diverse Sequential Subset Selection for Supervised Video Summarization". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger. Vol. 27. Curran Associates, Inc., pp. 2069–2077.

Gonzalez-Franco, Mar, Markus Wojcik, Eyal Ofek, Anthony Steed, and Dave Garagan (2020). *Microsoft Rocketbox: https://github.com/microsoft/Microsoft-Rocketbox*.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger. Vol. 27. Curran Associates, Inc.

Grassia, F Sebastian (1998). "Practical parameterization of rotations using the exponential map". In: *Journal of graphics tools* 3.3, pp. 29–48.

Greenwood, David, Stephen Laycock, and Iain Matthews (2017). "Predicting head pose from speech with a conditional variational autoencoder". In: ISCA.

Gross, M Melissa, Elizabeth A Crane, and Barbara L Fredrickson (2012). "Effort-shape and kinematic assessment of bodily expression of emotion during gait". In: *Human movement science* 31.1, pp. 202–221.

Gu, Hongxiang and Viswanathan Swaminathan (2018). "From thumbnails to summaries-a single deep neural network to rule them all". In: *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, pp. 1–6.

Gu, Junxia, Xiaoqing Ding, Shengjin Wang, and Youshou Wu (2010). "Action and gait recognition from recovered 3-D human joints". In: *Cybernetics* 40.4, pp. 1021–1033.

Gunes, Hatice and Massimo Piccardi (2007). "Bi-modal emotion recognition from expressive face and body gestures". In: *Journal of Network and Computer Applications* 30.4. Special issue on Information technology, pp. 1334–1345. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2006.09.007.

Gygli, Michael, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool (2014). "Creating Summaries from User Videos". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, pp. 505–520. ISBN: 978-3-319-10584-0.

Gygli, Michael, Helmut Grabner, and Luc Van Gool (June 2015). "Video Summarization by Learning Submodular Mixtures of Objectives". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gygli, Michael, Yale Song, and Liangliang Cao (June 2016). "Video2GIF: Automatic Generation of Animated GIFs From Video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Habibie, Ikhansul, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura (2017). "A Recurrent Variational Autoencoder for Human Motion Synthesis". In: *BMVC*.

Habibie, Ikhsanul, Weipeng Xu, Dushyant Mehta, Lingjie Liu, Hans-Peter Seidel, Gerard Pons-Moll, Mohamed Elgharib, and Christian Theobalt (2021). "Learning Speech-Driven 3D Conversational Gestures from Video". In: *Proceedings of the 21st ACM International Conference on Intelligent Virtual Agents*. IVA '21. Virtual Event, Japan: Association for Computing Machinery, 101–108. ISBN: 9781450386197. DOI: 10.1145/3472306.3478335.

Harvey, Félix G, Julien Roy, David Kanaa, and Christopher Pal (2018). "Recurrent semi-supervised classification and constrained adversarial generation with motion capture data". In: *Image and Vision Computing* 78, pp. 42–52.

Hasegawa, Dai, Naoshi Kaneko, Shinichi Shirakawa, Hiroshi Sakuta, and Kazuhiko Sumi (2018). "Evaluation of Speech-to-Gesture Generation Using Bi-Directional LSTM Network". In: *Proceedings of the 18th International Conference on Intelligent Virtual Agents*. IVA '18. Sydney, NSW, Australia: Association for Computing Machinery, 79–86. ISBN: 9781450360135. DOI: 10.1145/3267851.3267878.

Heidicker, Paul, Eike Langbehn, and Frank Steinicke (2017). "Influence of avatar appearance on presence in social VR". In: *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 233–234. DOI: 10.1109/3DUI.2017.7893357.

Henter, Gustav Eje, Simon Alexanderson, and Jonas Beskow (Nov. 2020). "MoGlow: Probabilistic and Controllable Motion Synthesis Using Normalising Flows". In: *ACM Transactions on Graphics* 39.6. ISSN: 0730-0301. DOI: 10.1145/3414685.3417836.

Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter (2017). "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *NIPS*, pp. 6626–6637.

Hoffmann, Holger, Andreas Scheck, Timo Schuster, Steffen Walter, Kerstin Limbrecht, Harald C Traue, and Henrik Kessler (2012). "Mapping discrete emotions into the dimensional space: An empirical approach". In: *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, pp. 3316–3320.

Holden, Daniel, Taku Komura, and Jun Saito (2017). "Phase-functioned neural networks for character control". In: *ACM Transactions on Graphics (TOG)* 36.4, p. 42.

Holden, Daniel, Jun Saito, and Taku Komura (July 2016). "A Deep Learning Framework for Character Motion Synthesis and Editing". In: *ACM Transactions on Graphics* 35.4. ISSN: 0730-0301. DOI: 10.1145/2897824.2925975.

Hu, Ping, Dongqi Cai, Shandong Wang, Anbang Yao, and Yurong Chen (2017). "Learning supervised scoring ensemble for emotion recognition in the wild". In: *ICMI*. ACM, pp. 553–560.

Huber, Peter J (1965). "A robust version of the probability ratio test". In: *The Annals of Mathematical Statistics*, pp. 1753–1758.

Ioffe, Sergey and Christian Szegedy (July 2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 448–456.

Ionescu, Catalin, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu (2014). "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7, pp. 1325–1339. DOI: 10.1109/TPAMI.2013.248.

Jacob, Agnes and P Mythili (2015). "Prosodic feature based speech emotion recognition at segmental and supra segmental levels". In: *SPICES*. IEEE, pp. 1–5.

Jaques, Natasha, Daniel J. McDuff, Yoo Lim Kim, and Rosalind W. Picard (2016). "Understanding and Predicting Bonding in Conversations Using Thin Slices of Facial Expressions and Body Language". In: *Intelligent Virtual Agents - 16th International Conference, IVA 2016, Los Angeles, CA, USA, September 20-23, 2016, Proceedings*. Vol. 10011. Lecture Notes in Computer Science, pp. 64–74. DOI: 10.1007/978-3-319-47665-0.

Ji, Shuiwang, Wei Xu, Ming Yang, and Kai Yu (2013). "3D convolutional neural networks for human action recognition". In: *PAMI* 35.1, pp. 221–231.

Ji, Zhong, Kailin Xiong, Yanwei Pang, and Xuelong Li (2020). "Video Summarization With Attention-Based Encoder–Decoder Networks". In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.6, pp. 1709–1717. DOI: 10.1109/TCSVT.2019.2904996.

Jiao, Yifan, Zhetao Li, Shucheng Huang, Xiaoshan Yang, Bin Liu, and Tianzhu Zhang (2018). "Three-dimensional attention-based deep ranking model for video highlight detection". In: *IEEE Transactions on Multimedia* 20.10, pp. 2693–2705.

Jiao, Yifan, Tianzhu Zhang, Shucheng Huang, Bin Liu, and Changsheng Xu (2019). "Video Highlight Detection via Region-Based Deep Ranking Model". In: *International Journal of Pattern Recognition and Artificial Intelligence* 33.07, p. 1940001.

Jocher, Glenn, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, tkianai, yxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900, Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh (Feb. 2022). *ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference*. Version v6.1. DOI: 10.5281/zenodo.6222936.

Joho, Hideo, Jacopo Staiano, Nicu Sebe, and Joemon M Jose (2011). "Looking at the viewer: analysing facial activity to detect personal highlights of multimedia contents". In: *Multimedia Tools and Applications* 51.2, pp. 505–523.

Joo, Hanbyul, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh (2017). "Panoptic Studio: A Massively Multiview System for Social Interaction Capture". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Kanazawa, Angjoo, Jason Y Zhang, Panna Felsen, and Jitendra Malik (2019). "Learning 3d human dynamics from video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5614–5623.

Karg, M., A. Samadani, R. Gorbet, K. Kühnlenz, J. Hoey, and D. Kulić (2013). "Body Movements for Affective Expression: A Survey of Automatic Recognition and Generation". In: *IEEE Transactions on Affective Computing* 4.4, pp. 341–359.

Karg, Michelle, Kolja Kuhnlenz, and Martin Buss (2010). "Recognition of affect based on gait patterns". In: *Cybernetics* 40.4, pp. 1050–1061.

Karras, Tero, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen (July 2017). "Audio-Driven Facial Animation by Joint End-to-End Learning of Pose and Emotion". In: *ACM Transactions on Graphics* 36.4. ISSN: 0730-0301. DOI: 10.1145/3072959.3073658.

Keltner, Dacher and Jonathan Haidt (2001). "Social functions of emotions". In.

Kfir, Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen (July 2020). "Unpaired Motion Style Transfer from Video to Animation". In: *ACM Transactions on Graphics* 39.4. DOI: 10.1145/3386569.3392469.

Khodabandeh, Mehran, Hamid Reza Vaezi Joze, Ilya Zharkov, and Vivek Pradeep (2018). "DIY Human Action Dataset Generation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1448–1458.

Khosla, Aditya, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan (June 2013). "Large-Scale Video Summarization Using Web-Image Priors". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kim, Gunhee, Leonid Sigal, and Eric P. Xing (June 2014). "Joint Summarization of Large-scale Collections of Web Images and Videos for Storyline Reconstruction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kim, Gunhee and Eric P. Xing (June 2014). "Reconstructing Storyline Graphs for Image Recommendation from Web Community Photos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kim, Hoseong, Tao Mei, Hyeran Byun, and Ting Yao (2018). "Exploiting web images for video highlight detection with triplet deep ranking". In: *IEEE Transactions on Multimedia* 20.9, pp. 2415–2426.

Kim, Sujeong, Stephen J Guy, Wenxi Liu, David Wilkie, Rynson WH Lau, Ming C Lin, and Dinesh Manocha (2015). "Brvo: Predicting pedestrian trajectories using velocity-space reasoning". In: *The International Journal of Robotics Research* 34.2, pp. 201–217.

Kim, Yelin, Honglak Lee, and Emily Mower Provost (2013). "Deep learning for robust feature generation in audiovisual emotion recognition". In: *ICASSP*, pp. 3687–3691.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kingma, Diederik P. and Max Welling (2019). "An Introduction to Variational Autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4, pp. 307–392. ISSN: 1935-8237. DOI: 10.1561/2200000056.

Kipf, Thomas N and Max Welling (2016). "Semi-supervised classification with graph convolutional networks". In: *arXiv:1609.02907*.

Kleinsmith, A. and N. Bianchi-Berthouze (2013). "Affective Body Expression Perception and Recognition: A Survey". In: *IEEE Transactions on Affective Computing* 4.1, pp. 15–33.

Kleinsmith, Andrea and Nadia Bianchi-Berthouze (2013). "Affective body expression perception and recognition: A survey". In: *IEEE Transactions on Affective Computing* 4.1, pp. 15–33.

Knapp, Mark L, Judith A Hall, and Terrence G Horgan (2013). *Nonverbal communication in human interaction*. Cengage Learning.

Kocabas, Muhammed (2019). *Simple Multi Person Tracker*.

Koren, Yehuda, Robert Bell, and Chris Volinsky (2009). "Matrix Factorization Techniques for Recommender Systems". In: *Computer* 42.8, pp. 30–37. DOI: 10.1109/MC.2009.263.

Kosti, Ronak, Jose Alvarez, Adria Recasens, and Agata Lapedriza (2019). "Context Based Emotion Recognition using EMOTIC Dataset". In: *IEEE transactions on pattern analysis and machine intelligence*.

Kovar, Lucas, Michael Gleicher, and Frédéric Pighin (2008). "Motion Graphs". In: *ACM SIGGRAPH 2008 Classes*. SIGGRAPH '08. Los Angeles, California: Association for Computing Machinery. ISBN: 9781450378451. DOI: 10.1145/1401132.1401202.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *NeurIPS*, pp. 1097–1105.

Kucherenko, Taras, Dai Hasegawa, Gustav Eje Henter, Naoshi Kaneko, and Hedvig Kjellström (2019). "Analyzing Input and Output Representations for Speech-Driven Gesture Generation". In: *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*. IVA '19. Paris, France: Association for Computing Machinery, 97–104. ISBN: 9781450366724. DOI: 10.1145/3308532.3329472.

Kucherenko, Taras, Patrik Jonell, Sanne van Waveren, Gustav Eje Henter, Simon Alexandersson, Iolanda Leite, and Hedvig Kjellström (2020). "Gesticulator: A Framework for Semantically-Aware Speech-Driven Gesture Generation". In: ICMI '20. Virtual Event, Netherlands: Association for Computing Machinery, 242–250. ISBN: 9781450375818. DOI: 10.1145/3382507.3418815.

Kucherenko, Taras, Patrik Jonell, Youngwoo Yoon, Pieter Wolfert, and Gustav Eje Henter (2021). "A Large, Crowdsourced Evaluation of Gesture Generation Systems on Common Data: The GENEA Challenge 2020". In: *26th International Conference on Intelligent User Interfaces*. IUI '21. College Station, TX, USA: Association for Computing Machinery, pp. 11–21. ISBN: 9781450380171. DOI: 10.1145/3397481.3450692.

Laban, Rudolf and Lisa Ullmann (1971). "The mastery of movement". In.

Lahiri, Avisek, Vivek Kwatra, Christian Frueh, John Lewis, and Chris Bregler (June 2021). "LipSync3D: Data-Efficient Learning of Personalized 3D Talking Faces From Video Using Pose and Lighting Normalization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2755–2764.

Latoschik, M. E., F. Kern, J. Stauffert, A. Bartl, M. Botsch, and J. Lugrin (2019). "Not Alone Here?! Scalability and User Experience of Embodied Ambient Crowds in Distributed Social Virtual Reality". In: *IEEE Transactions on Visualization and Computer Graphics* 25.5, pp. 2134–2144.

Latoschik, Marc Erich, Daniel Roth, Dominik Gall, Jascha Achenbach, Thomas Waltemate, and Mario Botsch (2017). "The Effect of Avatar Realism in Immersive Social Virtual Realities". In: *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. VRST '17. Gothenburg, Sweden: Association for Computing Machinery. ISBN: 9781450355483. DOI: 10.1145/3139131.3139156.

Lea, Colin, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager (2017). "Temporal convolutional networks for action segmentation and detection". In: *CVPR*, pp. 156–165.

Lee, Kang Hoon, Myung Geol Choi, and Jehee Lee (2006). "Motion patches: building blocks for virtual environments annotated with motion data". In: *ACM Transactions on Graphics* 25.3, pp. 898–906.

Lee, Seunghwan, Moonseok Park, Kyoungmin Lee, and Jehee Lee (2019). "Scalable muscle-actuated human simulation and control". In: *ACM Transactions on Graphics (TOG)* 38.4, p. 73.

Lee, Y. J., J. Ghosh, and K. Grauman (2012). "Discovering important people and objects for egocentric video summarization". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1346–1353. DOI: 10.1109/CVPR.2012.6247820.

Levine, Sergey, Philipp Krähenbühl, Sebastian Thrun, and Vladlen Koltun (2010). "Gesture Controllers". In: *ACM SIGGRAPH 2010 Papers*. SIGGRAPH '10. Los Angeles, California: Association for Computing Machinery. ISBN: 9781450302104. DOI: 10.1145/1833349.1778861.

Li, Ang, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman (2020). "The AVA-Kinetics Localized Human Actions Video Dataset". In: *arXiv preprint arXiv:2005.00214*.

Li, Chen, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee (June 2018). "Convolutional Sequence to Sequence Model for Human Dynamics". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, G., Y. Zhao, M. Ji, X. Yuan, and L. Fang (2020). "Zoom in to the Details of Human-Centric Videos". In: *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 3089–3093. DOI: 10.1109/ICIP40778.2020.9190977.

Li, Jamy, René Kizilcec, Jeremy Bailenson, and Wendy Ju (2016). "Social robots and virtual agents as lecturers for video instruction". In: *Computers in Human Behavior* 55, pp. 1222 –1230. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2015.04.005.

Li, Jianan, Xuemei Xie, Qingzhe Pan, Yuhan Cao, Zhifu Zhao, and Guangming Shi (2020). "SGM-Net: Skeleton-guided multimodal network for action recognition". In: *Pattern Recognition* 104, p. 107356. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107356.

Li, S. and W. Deng (2020). "Deep Facial Expression Recognition: A Survey". In: *IEEE Transactions on Affective Computing*, pp. 1–1. DOI: 10.1109/TAFFC.2020.2981446.

Li, Yitong, Martin Renqiang Min, Dinghan Shen, David E Carlson, and Lawrence Carin (n.d.). "Video Generation From Text". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI'18. AAAI Press, pp. 7065–7072.

Liao, M., C. Sung, H. Wang, and W. Lin (2019). "Virtual Classmates: Embodying Historical Learners' Messages as Learning Companions in a VR Classroom through Comment Mapping". In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 163–171.

Liebold, Benny and Peter Ohler (2013). "Multimodal Emotion Expressions of Virtual Agents, Mimic and Vocal Emotion Expressions and Their Effects on Emotion Recognition". In: *Proceedings of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. ACII '13. USA: IEEE Computer Society, 405–410. ISBN: 9780769550480. DOI: 10.1109/ACII.2013.73.

Lipton, Zachary C, John Berkowitz, and Charles Elkan (2015). "A critical review of recurrent neural networks for sequence learning". In: *arXiv preprint arXiv:1506.00019*.

Liu, Wu, Tao Mei, Yongdong Zhang, Cherry Che, and Jiebo Luo (June 2015). "Multi-Task Deep Visual-Semantic Embedding for Video Thumbnail Selection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Ziyu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang (June 2020). "Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lu, Yifei, Wei-Long Zheng, Binbin Li, and Bao-Liang Lu (2015). "Combining eye movements and EEG to enhance emotion recognition". In: *IJCAI*.

Lu, Zheng and Kristen Grauman (June 2013). "Story-Driven Summarization for Egocentric Video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ma, Yingliang, Helena M Paterson, and Frank E Pollick (2006). "A motion capture library for the study of identity, gender, and emotion perception from biological motion". In: *Behavior Research Methods* 38.1, pp. 134–141.

Mahasseni, Behrooz, Michael Lam, and Sinisa Todorovic (2017). "Unsupervised video summarization with adversarial lstm networks". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 202–211.

Majumder, Anima, Laxmidhar Behera, and Venkatesh K Subramanian (2014). "Emotion recognition from geometric facial features using self-organizing map". In: *Pattern Recognition* 47.3, pp. 1282–1293.

Mascarenhas, Samuel, Manuel Guimarães, Rui Prada, João Dias, Pedro A. Santos, Kam Star, Ben Hirsh, Ellis Spice, and Rob Kommeren (2018). "A Virtual Agent Toolkit for Serious Games Developers". In: *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–7. DOI: 10.1109/CIG.2018.8490399.

Mason, I., S. Starke, H. Zhang, H. Bilen, and T. Komura (2018). "Few-shot Learning of Homogeneous Human Locomotion Styles". In: *Computer Graphics Forum* 37.7, pp. 143–153. DOI: 10.1111/cgf.13555. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13555.

Matsumoto, David, Mark G Frank, and Hyi Sung Hwang (2012). *Nonverbal communication: Science and applications*. Sage Publications.

McHugh, Joanna Edel, Rachel McDonnell, Carol O'Sullivan, and Fiona N Newell (2010). "Perceiving emotion in crowds: the role of dynamic body postures on the perception of emotion in crowded scenes". In: *Experimental brain research* 204.3, pp. 361–372.

McNeill, David (1992). *Hand and mind: What gestures reveal about thought*. University of Chicago press.

Meeren, Hanneke K. M., Corné C. R. J. van Heijnsbergen, and Beatrice de Gelder (2005a). "Rapid perceptual integration of facial expression and emotional body language". In: *Proceedings of the National Academy of Sciences* 102.45, pp. 16518–16523. ISSN: 0027-8424. DOI: 10.1073/pnas.0507650102. eprint: https://www.pnas.org/content/102/45/16518.full.pdf.

Meeren, Hanneke KM, Corné CRJ van Heijnsbergen, and Beatrice de Gelder (2005b). "Rapid perceptual integration of facial expression and emotional body language". In: *Proceedings of NAS* 102.45, pp. 16518–16523.

Mehrabian, Albert (1996). "Analysis of the big-five personality factors in terms of the PAD temperament model". In: *Australian journal of Psychology* 48.2, pp. 86–92.

Mehrabian, Albert and James A Russell (1974). *An approach to environmental psychology*. the MIT Press.

Mehta, D., O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt (2018). "Single-Shot Multi-person 3D Pose Estimation from Monocular RGB". In: *2018 International Conference on 3D Vision (3DV)*, pp. 120–130.

Mesquita, Batja and Michael Boiger (2014). "Emotions in Context: A Sociodynamic Model of Emotions". In: *Emotion Review* 6.4, pp. 298–302. DOI: 10.1177/1754073914534480. eprint: https://doi.org/10.1177/1754073914534480.

Michalak, Johannes, Nikolaus F Troje, Julia Fischer, Patrick Vollmar, Thomas Heidenreich, and Dietmar Schulte (2009). "Embodiment of sadness and depression—gait patterns associated with dysphoric mood". In: *Psychosomatic Medicine* 71.5, pp. 580–587.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Vol. 26. Curran Associates, Inc., pp. 3111–3119.

Mittal, Trisha, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha (2020a). "M3ER: Multiplicative Multimodal Emotion Recognition using Facial, Textual, and Speech Cues". In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI'20. AAAI Press, pp. 1359–1367.

Mittal, Trisha, Pooja Guhan, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha (2020b). "EmotiCon: Context-Aware Multimodal Emotion Recognition Using Frege's Principle". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14234–14243.

Mohammad, Saif (July 2018). "Obtaining Reliable Human Ratings of Valence, Arousal, and Dominance for 20,000 English Words". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 174–184. DOI: 10.18653/v1/P18-1017.

Molino, Ana Garcia del, Xavier Boix, Joo-Hwee Lim, and Ah-Hwee Tan (Feb. 2017). "Active Video Summarization: Customized Summaries via On-line Interaction with the User". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1.

Molino, Ana Garcia del and Michael Gygli (2018). "PHD-GIFs: Personalized Highlight Detection for Automatic GIF Creation". In: *Proceedings of the 26th ACM International Conference on Multimedia*. MM '18. Seoul, Republic of Korea: Association for Computing Machinery, 600–608. ISBN: 9781450356657. DOI: 10.1145/3240508.3240599.

Montepare, Joann M, Sabra B Goldstein, and Annmarie Clausen (1987). "The identification of emotions from gait information". In: *Journal of Nonverbal Behavior* 11.1, pp. 33–42.

Moustafa, Fares and Anthony Steed (2018). "A Longitudinal Study of Small Group Interaction in Social Virtual Reality". In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. VRST '18. Tokyo, Japan: Association for Computing Machinery. ISBN: 9781450360869. DOI: 10.1145/3281505.3281527.

Murty, K. S. R. and B. Yegnanarayana (2006). "Combining evidence from residual phase and MFCC features for speaker recognition". In: *IEEE Signal Processing Letters* 13.1, pp. 52–55. DOI: 10.1109/LSP.2005.860538.

Narang, Sahil, Andrew Best, Andrew Feng, Sin-hwa Kang, Dinesh Manocha, and Ari Shapiro (2017). "Motion recognition of self and others on realistic 3D avatars". In: *Computer Animation and Virtual Worlds* 28.3-4, e1762.

Narayanan, Venkatraman, Bala Murali Manoghar, Vishnu Sashank Dorbala, Dinesh Manocha, and Aniket Bera (2020). "ProxEmo: Gait-based Emotion Learning and Multi-view Proxemic Fusion for Socially-Aware Robot Navigation". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020*. IEEE.

Neff, Michael, Michael Kipp, Irene Albrecht, and Hans-Peter Seidel (Mar. 2008). "Gesture Modeling and Animation Based on a Probabilistic Re-Creation of Speaker Style". In: *ACM Transactions on Graphics* 27.1. ISSN: 0730-0301. DOI: 10.1145/1330511.1330516.

Neiberg, Daniel, Kjell Elenius, and Kornel Laskowski (2006). "Emotion recognition in spontaneous speech using GMMs". In: *Ninth international conference on spoken language processing*.

NEON: https://www.neon.life/ (2020).

Nestler, Eric J, Michel Barrot, Ralph J DiLeone, Amelia J Eisch, Stephen J Gold, and Lisa M Monteggia (2002). "Neurobiology of depression". In: *Neuron* 34.1, pp. 13–25. DOI: 10.1016/S0896-6273(02)00653-0.

Nisbett, Richard E and Timothy D Wilson (1977). "Telling more than we can know: Verbal reports on mental processes." In: *Psychological review* 84.3, p. 231.

NVIDIA (n.d.). *NVIDIA Omniverse*.

Oculus (2019). *Facebook Horizon, https://www.oculus.com/facebook-horizon/*.

Osking, Hunter and John A. Doucette (2019). "Enhancing Emotional Effectiveness of Virtual-Reality Experiences with Voice Control Interfaces". In: *Immersive Learning Research Network*. Ed. by Beck et al. Cham: Springer International Publishing, pp. 199–209. ISBN: 978-3-030-23089-0.

Panagiotakis, Costas, Harris Papadakis, and Paraskevi Fragopoulou (2020). "Personalized Video Summarization Based Exclusively on User Preferences". In: *Advances in Information Retrieval*. Ed. by Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins. Cham: Springer International Publishing, pp. 305–311. ISBN: 978-3-030-45442-5.

Panda, Rameswar and Amit K. Roy-Chowdhury (July 2017). "Collaborative Summarization of Topic-Related Videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Park, Soohwan, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee (2019). "Learning Predict-and-Simulate Policies From Unorganized Human Motion Data". In: *ACM Transactions on Graphics* 38.6.

Parkinson, Brian, Agneta H Fischer, and Antony SR Manstead (2005). *Emotion in social relations: Cultural, group, and interpersonal processes*. Psychology press.

Pavllo, Dario, Christoph Feichtenhofer, David Grangier, and Michael Auli (2019). "3D human pose estimation in video with temporal convolutions and semi-supervised training". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7753–7762.

Pavllo, Dario, David Grangier, and Michael Auli (2018). "QuaterNet: A Quaternion-based Recurrent Model for Human Motion". In: *British Machine Vision Conference 2018, BMVC 2018*, p. 299.

Pelczer, Ildikó, Francisco Cabiedes Contreras, and Fernando Gamboa Rodríguez (2007). "Expressions of Emotions in Virtual Agents: Empirical Evaluation". In: *2007 IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, pp. 31–35.

Peng, Xue Bin, Pieter Abbeel, Sergey Levine, and Michiel van de Panne (July 2018). "DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills". In: *ACM Transactions on Graphics* 37.4. ISSN: 0730-0301. DOI: 10.1145/3197517.3201311.

Peng, Xue Bin, Glen Berseth, Kangkang Yin, and Michiel Van De Panne (July 2017). "DeepLoco: Dynamic Locomotion Skills Using Hierarchical Deep Reinforcement Learning". In: *ACM Transactions on Graphics* 36.4. ISSN: 0730-0301. DOI: 10.1145/3072959.3073602.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

Phillips, Mary L, Wayne C Drevets, Scott L Rauch, and Richard Lane (2003). "Neurobiology of emotion perception I: The neural basis of normal emotion perception". In: *Biological psychiatry* 54.5, pp. 504–514.

Potapov, Danila, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid (2014). "Category-Specific Video Summarization". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, pp. 540–555. ISBN: 978-3-319-10599-4.

Quigley, Karen S, Kristen A Lindquist, and Lisa Feldman Barrett (2014). *Inducing and measuring emotion and affect: Tips, tricks, and secrets.* Cambridge University Press.

Randhavane, Tanmay, Aniket Bera, Kyra Kapsaskis, Uttaran Bhattacharya, Kurt Gray, and Dinesh Manocha (2019a). "Identifying Emotions from Walking using Affective and Deep Features". In: *arXiv preprint arXiv:1906.11884*.

Randhavane, Tanmay, Aniket Bera, Kyra Kapsaskis, Kurt Gray, and Dinesh Manocha (2019b). "FVA: Modeling Perceived Friendliness of Virtual Agents Using Movement Characteristics". In: *IEEE transactions on visualization and computer graphics* 25.11, pp. 3135–3145.

Randhavane, Tanmay, Aniket Bera, Kyra Kapsaskis, Rahul Sheth, Kurt Gray, and Dinesh Manocha (2019c). "EVA: Generating Emotional Behavior of Virtual Agents using Expressive Features of Gait and Gaze". In: *ACM Symposium on Applied Perception 2019*. ACM, p. 6.

Randhavane, Tanmay, Aniket Bera, Emily Kubin, Kurt Gray, and Dinesh Manocha (2021). "Modeling Data-Driven Dominance Traits for Virtual Characters Using Gait Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 27.6, pp. 2967–2979. DOI: 10.1109/TVCG.2019.2953063.

Rao, K. Sreenivasa, Shashidhar G. Koolagudi, and Ramu Reddy Vempada (2013). "Emotion recognition from speech using global and local prosodic features". In: *International Journal of Speech Technology*.

Richard, Alexander, Michael Zollhöfer, Yandong Wen, Fernando de la Torre, and Yaser Sheikh (Oct. 2021). "MeshTalk: 3D Face Animation From Speech Using Cross-Modality Disentanglement". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1173–1182.

Riek, Laurel D., Tal-Chen Rabinowitch, Bhismadev Chakrabarti, and Peter Robinson (2009). "How Anthropomorphism Affects Empathy toward Robots". In: *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*. HRI '09. La Jolla, California, USA: Association for Computing Machinery, 245–246. ISBN: 9781605584041. DOI: 10.1145/1514095.1514158.

Riggio, Heidi R. (2017). "Emotional Expressiveness". In: *Encyclopedia of Personality and Individual Differences*.

Rivas, Jesús J, Felipe Orihuela-Espina, L Enrique Sucar, Lorena Palafox, Jorge Hernández-Franco, and Nadia Bianchi-Berthouze (2015). "Detecting affective states in virtual rehabilitation". In: *Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, pp. 287–292.

Rochan, Mrigank, Mahesh Kumar Krishna Reddy, Linwei Ye, and Yang Wang (Aug. 2020). "Adaptive Video Highlight Detection by Learning from User History". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Rochan, Mrigank and Yang Wang (June 2019). "Video Summarization by Learning From Unpaired Data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rochan, Mrigank, Linwei Ye, and Yang Wang (Sept. 2018). "Video Summarization Using Fully Convolutional Sequence Networks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Roether, Claire L, Lars Omlor, Andrea Christensen, and Martin A Giese (2009). "Critical features for the perception of emotion from gait". In: *Journal of vision* 9.6, pp. 15–15.

Rokbani, Nizar, Boudour Ammar Cherif, and Adel M Alimi (2009). "Toward intelligent biped-humanoids gaits generation". In: *Humanoid Robots*, pp. 259–271.

Rosenberg, Hannah, Skye McDonald, Jacob Rosenberg, and Reginald Frederick Westbrook (2019). "Measuring emotion perception following traumatic brain injury: The Complex Audio Visual Emotion Assessment Task (CAVEAT)". In: *Neuropsychological Rehabilitation* 29.2. PMID: 28030989, pp. 232–250. DOI: 10.1080/09602011.2016.1273118. eprint: https://doi.org/10.1080/09602011.2016.1273118.

Roth, Daniel, Jean-Luc Lugrin, Dmitri Galakhov, Arvid Hofmann, Gary Bente, Marc Erich Latoschik, and Arnulph Fuhrmann (2016). "Avatar realism and social interaction quality in virtual reality". In: *2016 IEEE Virtual Reality (VR)*, pp. 277–278. DOI: 10.1109/VR.2016.7504761.

Sadoughi, N. and C. Busso (2018). "Novel Realizations of Speech-Driven Head Movements with Generative Adversarial Networks". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6169–6173.

Sadoughi, Najmeh and Carlos Busso (2019). "Speech-driven animation with meaningful behaviors". In: *Speech Communication* 110, pp. 90 –100. ISSN: 0167-6393. DOI: https://doi.org/10.1016/j.specom.2019.04.005.

Santos, Igor, Nadia Nedjah, and Luiza de Macedo Mourelle (2017). "Sentiment analysis using convolutional neural network with fastText embeddings". In: *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pp. 1–5. DOI: 10.1109/LA-CCI.2017.8285683.

Scholz, F. W. and M. A. Stephens (1987). "K-Sample Anderson–Darling Tests". In: *Journal of the American Statistical Association* 82.399, pp. 918–924. DOI: 10.1080/01621459.1987.10478517. eprint: https://doi.org/10.1080/01621459.1987.10478517.

Schurgin, MW, J Nelson, S Iida, H Ohira, JY Chiao, and SL Franconeri (2014). "Eye movements during emotion recognition in faces". In: *Journal of vision* 14.13, pp. 14–14.

Shahroudy, A., T. Ng, Y. Gong, and G. Wang (2018). "Deep Multimodal Feature Analysis for Action Recognition in RGB+D Videos". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.5, pp. 1045–1058. DOI: 10.1109/TPAMI.2017.2691321.

Shahroudy, Amir, Jun Liu, Tian-Tsong Ng, and Gang Wang (2016). "Ntu rgb+ d: A large scale dataset for 3d human activity analysis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1010–1019.

Sharghi, Aidean, Boqing Gong, and Mubarak Shah (2016). "Query-Focused Extractive Video Summarization". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, pp. 3–19. ISBN: 978-3-319-46484-8.

Shi, Lei, Yifan Zhang, Jian Cheng, and Hanqing Lu (2019a). "Skeleton-Based Action Recognition with Directed Graph Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7912–7921.

Shi, Lei, Yifan Zhang, Jian Cheng, and Hanqing Lu (2019b). "Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12026–12035.

Si, Chenyang, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan (2019). "An Attention Enhanced Graph Convolutional LSTM Network for Skeleton-Based Action Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1227–1236.

Simeone, Adalberto L, Marco Speicher, Andreea Molnar, Adriana Wilde, and Florian Daiber (2019). "LIVE: The Human Role in Learning in Immersive Virtual Environments". In: *Symposium on Spatial User Interaction*. SUI '19. New Orleans, LA, USA: Association for Computing Machinery. ISBN: 9781450369756. DOI: 10.1145/3357251.3357590.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv:1409.1556*.

Singla, Adish, Sebastian Tschiatschek, and Andreas Krause (Mar. 2016). "Noisy Submodular Maximization via Adaptive Sampling with Applications to Crowdsourced Image Collection Summarization". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 30.1.

Sinha, Tanmay, Zhen Bai, and Justine Cassell (2022). "A Novel Multimodal Approach for Studying the Dynamics of Curiosity in Small Group Learning". In: *arXiv preprint arXiv:2204.00545*.

Sohn, Kihyuk, Honglak Lee, and Xinchen Yan (2015). "Learning structured output representation using deep conditional generative models". In: *NeurIPS*, pp. 3483–3491.

Sohn, Samuel S., Xun Zhang, Fernando Geraci, and Mubbasir Kapadia (2018). "An Emotionally Aware Embodied Conversational Agent". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '18. Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, 2250–2252.

Soleymani, Mohammad, Maja Pantic, and Thierry Pun (2012). "Multimodal Emotion Recognition in Response to Videos". In: *IEEE Transactions on Affective Computing* 3.2, pp. 211–223. DOI: 10.1109/T-AFFC.2011.37.

Song, Sijie, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu (2018). "Spatio-Temporal Attention-Based LSTM Networks for 3D Action Recognition and Detection". In: *IEEE Transactions on Image Processing* 27.7, pp. 3459–3471. DOI: 10.1109/TIP.2018.2818328.

Song, Yale, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes (June 2015). "TVSum: Summarizing Web Videos Using Titles". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Springenberg, Jost Tobias, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller (2014). "Striving for simplicity: The all convolutional net". In: *arXiv:1412.6806*.

Stangl, Brigitte, Dandison C. Ukpabi, and Sangwon Park (2020). "Augmented Reality Applications: The Impact of Usability and Emotional Perceptions on Tourists' App Experiences". In: *Information and Communication Technologies in Tourism 2020*. Ed. by Julia Neidhardt and Wolfgang Wörndl. Cham: Springer International Publishing, pp. 181–191. ISBN: 978-3-030-36737-4.

Starke, Sebastian, He Zhang, Taku Komura, and Jun Saito (2019). "Neural state machine for character-scene interactions". In: *ACM Transactions on Graphics (TOG)* 38.6, p. 209.

Stock, Jan Van den, Ruthger Righart, and Beatrice De Gelder (2007). "Body expressions influence recognition of emotions in the face and voice." In: *Emotion* 7.3, p. 487.

Strapparava, Carlo and Rada Mihalcea (2008). "Learning to identify emotions in text". In: *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, pp. 1556–1560.

Sun, Min, Ali Farhadi, and Steve Seitz (2014). "Ranking domain-specific highlights by analyzing edited videos". In: *European conference on computer vision*. Springer, pp. 787–802.

Suwajanakorn, Supasorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman (July 2017). "Synthesizing Obama: Learning Lip Sync from Audio". In: *ACM Transactions on Graphics* 36.4. ISSN: 0730-0301. DOI: 10.1145/3072959.3073640.

Tang, Zongheng, Yue Liao, Si Liu, Guanbin Li, Xiaojie Jin, Hongxu Jiang, Qian Yu, and Dong Xu (2021). "Human-centric Spatio-Temporal Video Grounding With Visual Transformers". In: *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1. DOI: 10.1109/TCSVT.2021.3085907.

Truong, Ba Tu and Svetha Venkatesh (Feb. 2007). "Video Abstraction: A Systematic Review and Classification". In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 3.1, 3–es. ISSN: 1551-6857. DOI: 10.1145/1198302.1198305.

Tulyakov, Sergey, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz (2018). "Mocogan: Decomposing motion and content for video generation". In: *CVPR*, pp. 1526–1535.

Umeyama, Shinji (1991). "Least-squares estimation of transformation parameters between two point patterns". In: *TPAMI*, pp. 376–380.

Vasudevan, Arun Balajee, Michael Gygli, Anna Volokitin, and Luc Van Gool (2017). "Query-Adaptive Video Summarization via Quality-Aware Relevance Estimation". In: *Proceedings of the 25th ACM International Conference on Multimedia*. MM '17. Mountain View, California, USA: Association for Computing Machinery, 582–590. ISBN: 9781450349062. DOI: 10.1145/3123266.3123297.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pp. 5998–6008.

Venture, Gentiane, Hideki Kadone, Tianxiang Zhang, Julie Grèzes, Alain Berthoz, and Halim Hicheur (2014). "Recognizing emotions conveyed by human gait". In: *IJSR* 6.4, pp. 621–632.

Vergin, R., D. O'Shaughnessy, and A. Farhat (1999). "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition". In: *IEEE Transactions on Speech and Audio Processing* 7.5, pp. 525–532. DOI: 10.1109/89.784104.

Vicol, Paul, Makarand Tapaswi, Lluís Castrejón, and Sanja Fidler (June 2018). "MovieGraphs: Towards Understanding Human-Centric Situations From Videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Volkova, Ekaterina, Stephan De La Rosa, Heinrich H Bülthoff, and Betty Mohler (2014). "The MPI Emotional Body Expressions Database for Narrative Scenarios". In: *PloS one* 9.12, e113647.

Wagner, Petra, Zofia Malisz, and Stefan Kopp (2014). "Gesture and speech in interaction: An overview". In: *Speech Communication* 57, pp. 209 –232. ISSN: 0167-6393. DOI: https://doi.org/10.1016/j.specom.2013.09.008.

Wang, Jack M, David J Fleet, and Aaron Hertzmann (2007). "Gaussian process dynamical models for human motion". In: *IEEE transactions on pattern analysis and machine intelligence* 30.2, pp. 283–298.

Wang, Lei, Du Q. Huynh, and Piotr Koniusz (2020). "A Comparative Review of Recent Kinect-Based Action Recognition Algorithms". In: *IEEE Transactions on Image Processing* 29, pp. 15–28. DOI: 10.1109/TIP.2019.2925285.

Wang, Lijuan and Frank K Soong (2015). "HMM trajectory-guided sample selection for photo-realistic talking head". In: *Multimedia Tools and Applications* 74.22, pp. 9849–9869.

Wang, Ting-Chun, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro (2018). "Video-to-Video Synthesis". In: *NeurIPS*.

Wang, Weiyi, Valentin Enescu, and Hichem Sahli (2016). "Adaptive real-time emotion recognition from body movements". In: *TiiS* 5.4, p. 18.

Wang, Wenju, Yu Cai, and Tao Wang (2022). "Multi-view dual attention network for 3D object recognition". In: *Neural Computing and Applications* 34.4, pp. 3201–3212.

Wang, Xueyang, Xiya Zhang, Yinheng Zhu, Yuchen Guo, Xiaoyun Yuan, Liuyu Xiang, Zerun Wang, Guiguang Ding, David Brady, Qionghai Dai, and Lu Fang (June 2020). "PANDA: A Gigapixel-Level Human-Centric Video Dataset". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Watson, Katie, Samuel S. Sohn, Sasha Schriber, Markus Gross, Carlos Manuel Muniz, and Mubbasir Kapadia (2019). "StoryPrint: An Interactive Visualization of Stories". In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*. IUI '19. Marina del Ray, California: Association for Computing Machinery, 303–311. ISBN: 9781450362726. DOI: 10.1145/3301275.3302302.

Wei, Zijun, Boyu Wang, Minh Hoai Nguyen, Jianming Zhang, Zhe Lin, Xiaohui Shen, Radomir Mech, and Dimitris Samaras (2018). "Sequence-to-Segment Networks for Segment Detection". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., pp. 3507–3516.

Wu, Yuxin, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick (2019). *Detectron2*. https://github.com/facebookresearch/detectron2.

Wu, Zuxuan, Yanwei Fu, Yu-Gang Jiang, and Leonid Sigal (2016). "Harnessing object and scene semantics for large-scale video understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3112–3121.

Xia, Shihong, Congyi Wang, Jinxiang Chai, and Jessica Hodgins (July 2015). "Realtime Style Transfer for Unlabeled Heterogeneous Human Motion". In: *ACM Transactions on Graphics* 34.4. ISSN: 0730-0301. DOI: 10.1145/2766999.

Xiong, Bo, Yannis Kalantidis, Deepti Ghadiyaram, and Kristen Grauman (2019). "Less is more: Learning highlight detection from video duration". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1258–1267.

Xiong, Bo, Gunhee Kim, and Leonid Sigal (Dec. 2015). "Storyline Representation of Egocentric Videos With an Applications to Story-Based Search". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Xu, Jia, Lopamudra Mukherjee, Yin Li, Jamieson Warner, James M. Rehg, and Vikas Singh (June 2015). "Gaze-Enabled Egocentric Video Summarization via Constrained Submodular Maximization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yan, An, Yali Wang, Zhifeng Li, and Yu Qiao (2019a). "PA3D: Pose-Action 3D Machine for Video Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7922–7931.

Yan, Sijie, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin (2019b). "Convolutional Sequence Generation for Skeleton-Based Action Synthesis". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4394–4402.

Yan, Sijie, Yuanjun Xiong, and Dahua Lin (2018). "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, pp. 7444–7452.

Yang, Ceyuan, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin (2018a). "Pose guided human video generation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 201–216.

— (2018b). "Pose guided human video generation". In: *ECCV*, pp. 201–216.

Yang, Huan, Baoyuan Wang, Stephen Lin, David Wipf, Minyi Guo, and Baining Guo (Dec. 2015). "Unsupervised Extraction of Video Highlights Via Robust Recurrent Auto-Encoders". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Yang, Hui, Lekha Chaisorn, Yunlong Zhao, Shi-Yong Neo, and Tat-Seng Chua (2003). "VideoQA: Question Answering on News Video". In: *Proceedings of the Eleventh ACM International Conference on Multimedia*. MULTIMEDIA '03. Berkeley, CA, USA: Association for Computing Machinery, 632–641. ISBN: 1581137222. DOI: 10.1145/957013.957146.

Yang, Huiyuan, Umur Ciftci, and Lijun Yin (2018). "Facial expression recognition by de-expression residue learning". In: *CVPR*, pp. 2168–2177.

Yao, Ting, Tao Mei, and Yong Rui (2016). "Highlight detection with pairwise deep ranking for first-person video summarization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 982–990.

Yates, Heath, Brent Chamberlain, Greg Norman, and William H Hsu (2017). "Arousal detection for biometric data in built environments using machine learning". In: *IJCAI-W*, pp. 58–72.

Yeung, Minerva, Boon-Lock Yeo, and Bede Liu (1998). "Segmentation of Video by Clustering and Graph Analysis". In: *Computer Vision and Image Understanding* 71.1, pp. 94 –109. ISSN: 1077-3142. DOI: https://doi.org/10.1006/cviu.1997.0628.

Yoon, Youngwoo, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee (2020). "Speech Gesture Generation from the Trimodal Context of Text, Audio, and Speaker Identity". In: *ACM Transactions on Graphics* 39.6.

Yoon, Youngwoo, Woo-Ri Ko, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee (2019). "Robots Learn Social Skills: End-to-End Learning of Co-Speech Gesture Generation for Humanoid Robots". In: *Proc. of The International Conference in Robotics and Automation (ICRA)*.

Yu, Youngjae, Sangho Lee, Joonil Na, Jaeyun Kang, and Gunhee Kim (2018). "A Deep Ranking Model for Spatio-Temporal Highlight Detection from a 360 Video". In: pp. 7525–7533.

Yu, Zhou, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian (June 2019). "Deep Modular Co-Attention Networks for Visual Question Answering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yumer, M. Ersin and Niloy J. Mitra (July 2016). "Spectral Style Transfer for Human Motion between Independent Actions". In: *ACM Transactions on Graphics* 35.4. ISSN: 0730-0301. DOI: 10.1145/2897824.2925955.

Zeng, Wenjun (2020). "Toward human-centric deep video understanding". In: *APSIPA Transactions on Signal and Information Processing* 9, e1. DOI: 10.1017/atsip.2019.26.

Zhang, Feifei, Tianzhu Zhang, Qirong Mao, and Changsheng Xu (2018a). "Joint pose and expression modeling for facial expression recognition". In: *CVPR*, pp. 3359–3368.

Zhang, Jason Y, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik (2019). "Predicting 3d human dynamics from video". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7114–7123.

Zhang, K., Z. Zhang, Z. Li, and Y. Qiao (Oct. 2016a). "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks". In: *IEEE Signal Processing Letters* 23.10, pp. 1499–1503. ISSN: 1070-9908. DOI: 10.1109/LSP.2016.2603342.

Zhang, Ke, Wei-Lun Chao, Fei Sha, and Kristen Grauman (2016b). "Video Summarization with Long Short-Term Memory". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, pp. 766–782.

Zhang, Ke, Kristen Grauman, and Fei Sha (Sept. 2018). "Retrospective Encoders for Video Summarization". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zhang, Songyang, Yang Yang, Jun Xiao, Xiaoming Liu, Yi Yang, Di Xie, and Yueting Zhuang (2018b). "Fusing geometric features for skeleton-based action recognition using multilayer LSTM networks". In: *IEEE Transactions on Multimedia* 20.9, pp. 2330–2343.

Zhang, Zonghua and Nikolaus F Troje (2005). "View-independent person identification from human gait". In: *Neurocomputing* 69.1-3, pp. 250–256.

Zhao, Bin, Xuelong Li, and Xiaoqiang Lu (2017). "Hierarchical Recurrent Neural Network for Video Summarization". In: *Proceedings of the 25th ACM International Conference on Multimedia*. MM '17. Mountain View, California, USA: Association for Computing Machinery, 863–871. ISBN: 9781450349062. DOI: 10.1145/3123266.3123328.

Zhao, Bin and Eric P. Xing (June 2014). "Quasi Real-Time Summarization for Consumer Videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, Mingmin, Fadel Adib, and Dina Katabi (2016). "Emotion recognition using wireless signals". In: *ICMCN*. ACM, pp. 95–108.

Zhou, Kaiyang, Yu Qiao, and Tao Xiang (2018). "Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity-Representativeness Reward". In: pp. 7582–7589.

Zhou, Xingran, Siyu Huang, Bin Li, Yingming Li, Jiachen Li, and Zhongfei Zhang (June 2019). "Text Guided Person Image Synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhou, Yang, Zhan Xu, Chris Landreth, Evangelos Kalogerakis, Subhransu Maji, and Karan Singh (July 2018). "Visemenet: Audio-Driven Animator-Centric Speech Animation". In: *ACM Transactions on Graphics* 37.4. ISSN: 0730-0301. DOI: 10.1145/3197517.3201292.