ABSTRACT

Title of Dissertation: Interpreting Visual Representations and Mitigating their Failures

Neha Kalibhat, 2024

Dissertation Directed by: Professor Soheil Feizi
Department of Computer Science
University of Maryland, College Park

Deep learning has become the cornerstone of artificial intelligence (AI), particularly in language and computer vision domains. The progression in this field is reflected in numerous applications accessible to the general public, such as information retrieval via virtual assistants, content generation, autonomous vehicles, drug discovery, and medical imaging. This unprecedented rate of AI adoption raises the critical need for research on the fundamental underpinnings of deep neural networks to understand what leads to their decisions and why they fail.

This thesis concentrates on self-supervised representation learning, a prevalent unsupervised method employed by foundational models to extract patterns from extensive visual data. Specifically, our focus lies in examining the low-dimensional representations generated by these models and dissecting their failure modes. In our initial investigation, we discover that self-supervised representations lack robustness to domain shifts, as they are not explicitly trained to distinguish image content from its domain. We remedy this issue by proposing a module that can be plugged into existing self-supervised baselines to disentangle their representation spaces and promote domain invariance and generalization.

Our subsequent analysis delves into the patterns within representations that influence downstream classification. We scrutinize the discriminative capacity of

individual features and their activations. We then propose an unsupervised quality metric that can preemptively determine whether a given representation will be correctly or incorrectly classified, with high precision. In the next segment of this thesis, we leverage our findings to further demystify the representation space, by uncovering interpretable subspaces which have unique concepts associated with them. We design a novel explainability framework that uses a vision-language model (such as CLIP) to provide natural language explanations for neural features (or groups) of a given pre-trained model.

We next investigate the role of augmentations and format transformations in learning generalizable visual representations. Drawing inspiration from advancements in audio and speech modalities, we examine how presenting visual data in multiple formats affects learning, separating this from the impact of augmentations. In the final segment, we reveal compositionality as a notable failure mode in current state-of-the-art representation methods. We critique the use of fixed-size patches in vision transformers and demonstrate the benefits of employing semantically meaningful patches based on visual priors. This design adjustment leads to significant improvements in image-text retrieval tasks and, more importantly, enhances performance on compositionality benchmarks.

Interpreting Visual Representations and Mitigating their Failures

by

Neha Kalibhat

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
2024

Advisory Committee:
Professor Soheil Feizi, Advisor
Professor Furong Huang
Professor Abhinav Shrivastava
Professor David Jacobs
Professor Doug Oard, Dean's Representative

To my partner, family and friends,

whose unwavering support made this journey possible.

Acknowledgements

This thesis represents the culmination of several years of hard work and determination, fueled by the guidance, collaboration, and support of many individuals to whom I owe a deep debt of gratitude.

My advisor, Prof. Soheil Feizi, has been a guiding beacon of mentorship throughout this five-year journey. Despite my limited background in machine learning and computer vision when I first approached him, he recognized my potential and allowed me to take ownership of my very first publication. His guidance and trust have empowered me to produce impactful research. I am sincerely grateful for the time and effort he dedicated to shaping my research career and preparing me for the greater intellectual challenges that lie ahead.

I would like to extend my gratitude to my collaborators at the University of Maryland, who helped bring my research ideas to fruition. This includes Yogesh Balaji, who mentored me during my initial years and guided me in building machine learning systems, as well as my outstanding project mates, Shweta Bhardwaj, Priyatham Kattakinda, and Arman Zarei, for their significant contributions. I have also had the privilege of collaborating with industry researchers who supported and directed many of my projects. I thank Bayan Bruss, Samuel Sharpe, Senthil Kumar, Jeremy Goodsitt, and Nikita Seleznev at Capital One Research. A special thanks goes to Maziar Sanjabi, my internship mentor and long-term collaborator at Meta AI, for

his invaluable guidance and counsel. I also appreciate the continued assistance of Kanika Narang and Hamed Firooz from Meta AI, even after my internship engagement. During the final two years of my graduate studies, I had the opportunity to work closely with several researchers at Google DeepMind through two internships. I am profoundly thankful to Warren Morningstar and Philip Mansfield who actively coached me to produce research at industry standards. Additionally, I thank Alex Bijamov, Karan Singhal, and Luyang Liu at Google DeepMind for their valuable insights.

I would like to acknowledge and thank all the essential sources of support and resources that made it possible for me to conduct my research projects with ease. These include a grant from CapitalOne, an NSF CAREER AWARD 1942230, ONR YIP award N00014-22-1-2271, ARO's Early Career Program Award 310902- 00001, Meta grant 23010098, HR001119S0026 (DARPA/GARD), HR00112090132 (DARPA/RED), Army Grant No. W911NF2120076, NIST 60NANB20D134, the NSF award CCF2212458, NSF Award No. 2229885 (NSF Institute for Trustworthy AI in Law and Society, TRAILS) and an Amazon Research Award.

I extend my heartfelt thanks to my thesis committee members including, Prof. Abhinav Shrivastava, Prof. Furong Huang, Prof. David Jacobs and Prof. Douglas Oard for their insightful discussions and feedback. I am incredibly grateful to Prof. Vanessa Frias-Martinez for admitting me into this PhD program and guiding me during my first semester. Her decision to offer me this opportunity has truly shaped my career trajectory toward a brighter future. I am also deeply appreciative of Prof. Abhinav Shrivastava for his timely advice, support, and encouragement during pivotal moments of my journey. I am constantly inspired by his empathy, friendship, and

cheerfulness, qualities I hope to carry forward in both my professional and personal life.

I feel overwhelmingly fortunate to have formed some remarkable friendships, each of which has enriched this journey with warmth, compassion, and guidance. I want to thank my friends - Pulkit, Sneha, Amanpreet, Sharath, Saksham, Nirat, Samyadeep, Gowthami, Sanchita, Ahana, Roni, Noor, Anubhav, Lillian, Mara, Kamal, Archana, Namitha, Vatsal, Soumik, Shishira, Matt, Shaan, Vedant, Divya, Geonsun, Saketh, my lab mates - Aya, Mazda and Vinu and many others who I may have indavertently missed.

I would also like to thank the administrative staff at UMD, including Tom Hurst, Migo Gui, Jodie Gray, Vivian Lu, and all the members of UMIACS and ISSS, for their timely assistance throughout various stages of this journey. Additionally, I sincerely appreciate CS GradCo for organizing social activities that fostered our community and for their proactive efforts in advocating for the rights, well-being, and financial aid of international students and graduate assistants.

Finally, I extend my deepest gratitude to my family, my greatest source of strength and motivation. My partner, Pavan Gurudath, was the first to instigate the idea in me to pursue a doctoral degree, a decision that has transformed my life in ways I could never have imagined. I am endlessly grateful for his unconditional love, patience, and understanding, and for standing by me every step of the way. My mother, Anitha Kalibhat, is my main pillar of inspiration and encouragement. She raised me to be a resilient and independent woman, which laid the foundation for my professional growth. I am forever indebted to her for her constant affection and the countless

sacrifices she made to ensure my well-being. My sister, Nikhita Kalibhat, shared in the responsibility for my welfare, and I am deeply appreciative of her companionship and selflessness. I am profoundly thankful to my grandparents, Usha and Ranganath Tankasali, my uncles, Sharad and Anand Tankasali, and my brother-in-law, Anand Madanapalle, each of whom has generously motivated me and provided essential advice during defining moments. I also feel extremely privileged that my family has expanded over the last few years. I am immensely grateful to Nikita Gurudath and Ravi Chollangi, whose warmth, kindness, and support have been a true gift, as well as to Sharada and Gurudath KP for always believing in my abilities.

I am indeed exceptionally blessed to have embarked on this journey with each and every one of you by my side. From the bottom of my heart, thank you.

Table of contents

Chapter 1

Introduction

The success of deep learning in computer vision is grounded in its ability to extract relevant information from large amounts of data. One way to achieve this is through *representation learning* where, deep neural networks are optimized to produce low-dimensional representations (also called embeddings) of the high-dimensional data (like images) presented to it. These representations encode sufficient information from the data and are then used for downstream tasks. A large portion of foundational models, especially in computer vision, rely on self-supervised learning to learn powerful representations. Unlike supervised learning, this approach does not require any labelled data. It instead relies on pretext tasks, driven by augmentations, to extract semantically relevant information from images in the form of representations. Using this approach, large vision models can be pre-trained on millions of images after which frozen representations can be used for various downstream tasks like classification, transfer learning, object detection, semantic segmentation, image retrieval etc. State-of-the-art self-supervised approaches [Che+20a; Car+21; Car+20b; Gri+20] have shown on par classification performance compared to supervised methods and significantly beat supervised methods in their generalization capability. Self-supervised learning has also been deployed to align data from multiple modalities like language and vision

[Rad+21] and shown impressive results in zero-shot classification.

While large pre-trained vision models can contain billions of parameters, only low-dimensional frozen representations are exposed when they are deployed. In this thesis, we study how representations encode information, their non-trivial properties and what leads to their success or failure. In Chapter 2, we observe that current state-of-the-art self-supervised approaches, are effective when trained on individual domains but show limited generalization on unseen domains. We observe that these models poorly generalize even when trained on a mixture of domains, making them unsuitable to be deployed under diverse real-world setups. Upon investigation, we observe that the core issue lies in representations being unable to distinguish domain information from content information. We therefore propose a general-purpose, lightweight Domain Disentanglement Module (DDM) that can be plugged into any self-supervised encoder to effectively perform representation learning on multiple, diverse domains with or without shared classes. During pre-training according to a self-supervised loss, DDM enforces a disentanglement in the representation space by splitting it into a domain-variant and a domain-invariant portion. When domain labels are not available, DDM uses a robust clustering approach to discover pseudo-domains. We show that pre-training self-supervised encoders with DDM can improve downstream performance on classification and transfer learning on various domain generalization benchmarks.

We next study how individual features in representations and their activation can affect downstream classification in Chapter 3. Without the use of class label information, we discover discriminative features that correspond to unique physical attributes in images, present mostly in correctly-classified representations. Using these features, we can compress the representation space by up to 40% without significantly affecting linear classification performance. We then propose Self-Supervised Representation Quality Score (or Q-Score), an unsupervised score that can reliably predict if a given sample is likely to be mis-classified during linear evaluation. Q-Score can also be used

as a regularization term on pre-trained encoders to remedy low-quality representations. Fine-tuning with Q-Score regularization can boost the linear probing accuracy of self-supervised models compared to their baselines. Finally, using gradient heatmaps and Salient ImageNet masks, we define a metric to quantify the interpretability of each representation. We show that discriminative features are strongly correlated to core attributes and, enhancing these features through Q-score regularization makes representations more interpretable.

Owing to our finding that representation coordinates correspond to unique concepts, we next attempt to explain these neural features using natural language in Chapter 4. We propose Automatic Feature Explanation using Contrasting Concepts (FALCON), an interpretability framework to explain features of image representations. For a target feature, FALCON captions its highly activating cropped images using a large captioning dataset (like LAION-400m) and a pre-trained vision-language model like CLIP. Each word among the captions is scored and ranked leading to a small number of shared, human-understandable concepts that closely describe the target feature. FALCON also applies *contrastive interpretation* using lowly activating (counterfactual) images, to eliminate spurious concepts. Although many existing approaches interpret features independently, we observe in state-of-the-art self-supervised and supervised models, that less than 20% of the representation space can be explained by individual features. We show that features in larger spaces become more interpretable when studied in groups and can be explained with high-order scoring concepts through FALCON. We discuss how extracted concepts can be used to explain and debug failures in downstream tasks. Finally, we present a technique to transfer concepts from one (explainable) representation space to another unseen representation space by learning a simple linear transformation.

We next shift the focus to the use of data augmentations/perturbations, one of the pillars of generalizable vision representation learning. For audio and other

temporal signals, augmentations are commonly used alongside format transforms such as Fourier transforms or wavelet transforms. Unlike augmentations, format transforms do not change the information contained in the data; rather, they express the same information in different coordinates. In Chapter 5, we study the effects of format transforms and augmentations both separately and together on vision SSL. We define augmentations in the frequency space called Fourier Domain Augmentations (FDA) and show that training SSL models on a combination of these and image augmentations can improve the downstream classification accuracy by up to 1.3% on ImageNet-1K. We also show improvements against SSL baselines in few-shot and transfer learning setups using FDA. Surprisingly, we also observe that format transforms can improve the quality of learned representations even without augmentations; however, the combination of the two techniques yields better quality.

Our studies identify several aspects of internal representations learned by models that are prone to failures including; out-of-distribution generalization, interpretability and data augmentations. These failures point to a key design of image understanding architectures which rely on learning fixed-length representations from billions of examples. While such learned representations are currently the state-of-the-art in almost every domain where AI is used, they are prone to suffer significant bottlenecks when it comes to continual/lifelong learning, multi-modal alignment and reasoning. In the final Chapter 6, we instigate a new direction of research i.e., utilizing better suited data structures and priors to learn more generalizable visual representations. Vision transformers have established a precedent of patchifying images into uniformly-sized chunks before processing. We hypothesize that this design choice may limit models in learning comprehensive and compositional representations from visual data. This paper explores the notion of providing semantically-meaningful visual tokens to transformer encoders within a vision-language pre-training framework. Leveraging off-the-shelf segmentation and scene-graph models, we extract representations of

instance segmentation masks (referred to as tangible tokens) and relationships and actions (referred to as intangible tokens). Subsequently, we pre-train a vision-side transformer by incorporating these newly extracted tokens and aligning the resultant embeddings with caption embeddings from a text-side encoder. To capture the structural and semantic relationships among visual tokens, we introduce additive attention weights, which are used to compute self-attention scores. Our experiments on COCO demonstrate notable improvements over ViTs in learned representation quality across text-to-image (+47%) and image-to-text retrieval (+44%) tasks. Furthermore, we showcase the advantages on compositionality benchmarks such as ARO (+18%) and Winoground (+10%).

Chapter 2

Adapting Self-Supervised Representations to Multi-Domain Setups[1]

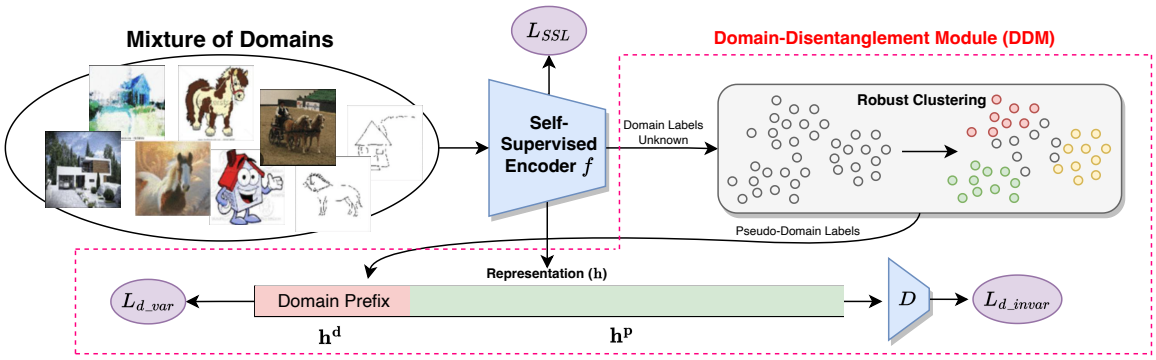


Figure 2.1: **Framework of our proposed Domain Disentanglement Module:** In our proposed DDM framework, the representation space ($\mathbf{h}$) of any given self-supervised encoder is split into two portions, a domain prefix ($\mathbf{h}^d$) and a domain-invariant ($\mathbf{h}^p$) portion. Along with the self-supervised loss (L_{ssl}), $\mathbf{h}^d$ is trained to be distinguishable across domains (L_{d_var}) and $\mathbf{h}^p$ is trained to be invariant to any domain information (L_{d_invar}). DDM also supports scenarios when domain labels are not available using *robust clustering*, an iterative process that reduces outlier noise.

[1]Full paper available at https://arxiv.org/abs/2309.03999. Contributing authors include Sam Sharpe, Jeremy Goodsitt, Bayan Bruss and Soheil Feizi.

2.1 Introduction

Self-supervised learning [Che+20a; Car+20b; He+20; Gri+20; CH21; Car+18a; Kho+20; Car+21] has become a popular paradigm for unsupervised representation learning as it shows impressive results on downstream tasks. However, we find that current self-supervised models when trained on a single-domain show very poor generalizability to domain shifts. This can hinder their deployment in large scale real-world settings where data almost always comes from multiple diverse domains. We illustrate this issue in Figure 2.2, where we show that popular self-supervised models, SimCLR [Che+20a], MoCo [He+20] and BYOL [Gri+20], trained on individual domains of PACS [Li+17] significantly under-perform on unseen domains. This means that a different self-supervised model needs to be trained for every new domain, which can add significant computational overheads given that training these models often require large batch sizes and a large number of training epochs [Che+20a; He+20; Wu+18].

One potential solution for self-supervised learning on multi-domain datasets is to train the models on the *union* of all input domains. We illustrate this in Figure 2.2 were we plot the multi-domain training results for each baseline on a mixture of PACS Photo, Sketch and Cartoon. We observe that this solution may show improved performance on the training domains, however they do not match the single-domain baselines in all cases. Moreover, they show poor generalization to unseen domains (PACS Painting). In Section 2.3, we study the representation space closely under multi-domain regimes to find that they can under perform compared to single-domain regimes because domain-related and content-related information overlap in the representation space, affecting their quality for instance classification.

To tackle these issues, we propose a **Domain-Disentanglement Module (DDM)**, that can be plugged in to any self-supervised model during multi-domain training. With DDM, we enforce a disentanglement in the representation space where a domain prefix is trained to be distinguishable across domains and the remaining portion is

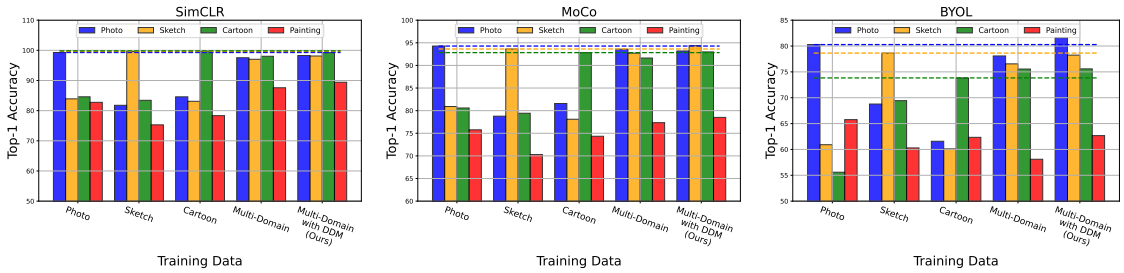


Figure 2.2: **Self-supervised baselines under single and multi-domain setups:** We plot 3 SOTA self-supervised baselines, SimCLR, BYOL and MoCo, trained individually on PACS Photo, Sketch and Cartoon and on their mixture. We observe that both single-domain and mult-domain training generalizes poorly to unseen domains on all baselines. These baselines when pre-trained with DDM (our method), outperforms even single-domain baselines and shows significantly improved generalization to the unseen domains.

trained to be *domain-invariant* to produce better structured representations. This is achieved by minimizing the Wasserstein Distance [ACB17] between the known and predicted domain label distributions. We also extend DDM to more realistic, entirely unsupervised multi-domain setups where domain labels are unknown. In such scenarios, we present a *robust clustering* approach that iteratively reduces outlier noise and detects pseudo-domain-labels that are used in DDM.

By pre-training with DDM, we show that we can improve the generalization capability of various state-of-the-art self-supervised baselines including SimCLR [Che+20a], MoCo [He+20], BYOL [Gri+20], DINO [Car+21], SimSiam [CH21] and Barlow Twins [Zbo+21a]. We perform extensive experiments on generalization benchmarks including PACS [Li+17], DomainNet [Pen+19] and WILDS [Koh+21]. Upon linear probing on unseen domains, we observe an improvement of 6.1% on PACS, 7.4% on DomainNet and 5.9% on WILDS using DDM. In summary, we propose a lightweight module called DDM which can be simply attached to any self-supervised encoder to enable training over multiple diverse domains to produce well-structured, generalizable representations (See Figure 2.1).

2.2 Related Work

Building on the success of unsupervised learning techniques [BJ17; Dos+14; YCA20; Bau+16; Car+18b; Car+19; Hua+19], self-supervised models have shown unprecedented capabilities when used in a range of downstream tasks. Among a number of self-supervised baselines, we focus on SimCLR [Che+20a], MoCo [He+20], BYOL [Gri+20], DINO [Car+21], SimSiam [CH21] and Barlow Twins [Zbo+21a]. These are joint-embedding self-supervised learning methods, which involve taking two augmented views of the same input and ensuring their representations are close using the same encoder or two encoders sharing the same weights.

Extending these self-supervised methods to multiple diverse domains, other than ImageNet [Rus+15a], is a relatively less explored topic [WH20]. Existing approaches [Kim+21; Li+21; Kim+20] use pre-trained encoders and assume few source labels for unsupervised domain adaption and domain generalization. [SL22] uses available class information and novelty discovery to learn new samples in the wild. These works do not consider fully unsupervised multi-domain setups, where even domain label information is unavailable. [FXT19] assumes domain labels and uses mutual information to encode common invariant information and domain-specific information for each image. [Yan+22a] uses multiple domain-specific decoders to reconstruct images according to their domains such that the encoder is domain-invariant. This method may not be scalable and is contingent upon the number of available domains. [Zha+22b] proposes a contrastive method that selects negatives across domains to train invariant representations. Our method reports better numbers on the PACS dataset compared to these baselines. Our method also does not assume domain labels and can be flexibly applied on any self-supervised setup.

In our paper, we focus on a general multi-domain setup with diverse related or unrelated domains, with and without shared classes, and evaluate on individual domain-specific tasks. We make it possible to efficiently pre-train a single encoder on any

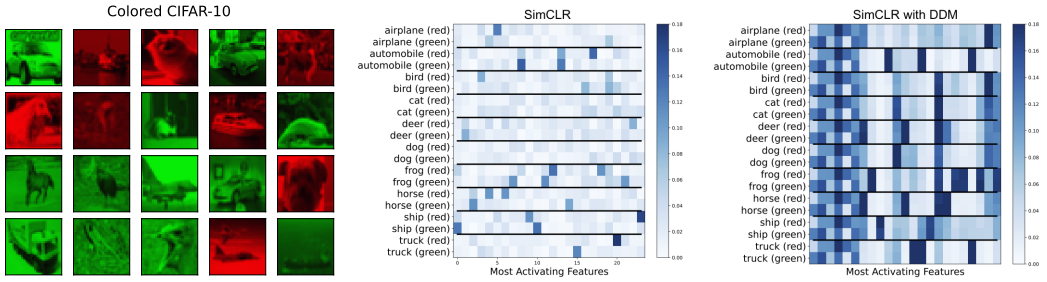


Figure 2.3: **Visualizing Colored-CIFAR representations:** We prepare Colored-CIFAR, multi-domain version of CIFAR-10 [KNHa] where the images are randomly colored red or green. We visualize the top activating features of the class-averaged representations of both domains. In the SimCLR baseline, we observe a clear difference in feature distribution within the same classes, across domains. DDM enables representations to have a shared domain-specific prefix while the remaining portion is domain-invariant and almost identical across classes. This structure significantly improves linear evaluation performance (See Figure 2.4).

existing state-of-the-art self-supervised setup, over multiple domains, to significantly improve their generalizability.

2.3 Self-Supervised Models under Multi-domain Setups

We observed in Figure 2.2, that state-of-the-art self-supervised learning methods like SimCLR [Che+20a], MoCo [Che+20b] and BYOL [Gri+20] show low transfer performance on unseen domains on both single-domain and multi-domain regimes. In this section, we take a closer look at the learned representation space under these regimes to explain this behavior.

We first define some notations. Let us consider a self-supervised model with a base encoder $f(.)$. We apply data transformations and pass the input samples, $\mathbf{x}_i \in \mathbb{R}^n$, through the base encoder to get self-supervised representations denoted by $f(\mathbf{x}_i) = \mathbf{h}_i \in \mathbb{R}^r$ where r is the size of the representation space.

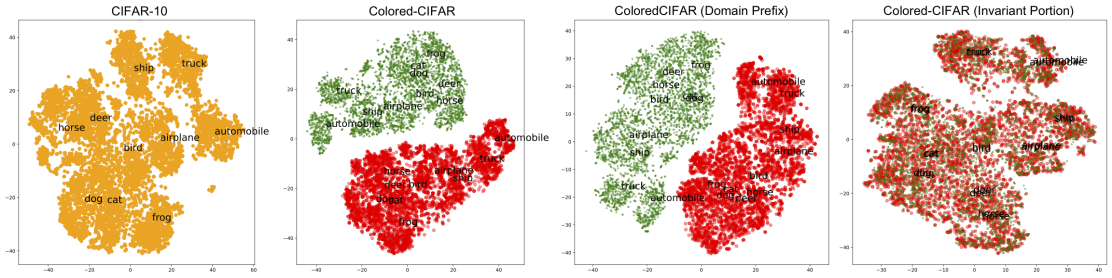
Let us take the example of SimCLR [Che+20a] trained on CIFAR-10 [KNHa]

dataset. In the first t-SNE [MH08] plot in Figure 2.4(a), we observe that the representations are naturally clustered based on their classes, which allows us to achieve a top-1 accuracy of 90.18 after linear probing. Let us now define a multi-domain version of CIFAR-10 called *Colored-CIFAR* where, each sample is randomly colored either red or green as shown in the first panel of Figure 2.3. In this dataset, the domains refer to the colors of the image, while the labels are of the objects. When SimCLR is trained on Colored-CIFAR, there is a significant drop in top-1 accuracy (78.52). We observe that the representation space is divided into two large clusters, corresponding to the domains (red or green) as shown in 2.4(b). We attribute the loss in accuracy to this significant change in representation structure.

We now study the SimCLR representation space of Colored-CIFAR to further understand and explain multi-domain behavior. In Figure 2.3, in the second panel, we show a heatmap of the domain-wise averaged representations of each class in CIFAR-10. Each column corresponds to specific feature indices of the class-averaged representations. The darker the column, the higher the magnitude of the feature. For fair comparison, we L2 normalize every feature. For ease of visualization, we display only the subset of feature indices (called *most activating features*) that are strongly deviated from the mean in at least one row. The remaining features show low activation across the board and are omitted from visualization [Jin+21; Kal+22]. Top activating features correspond to important physical attributes discovered from the training data [Kal+22; SF21]. Two images of a car, one in each domain, would share all physical attributes except for the color. An ideal self-supervised encoder is expected to encode all physical attributes independent of any domain shift.

However, in multi-domain SimCLR, we observe that there is almost no overlap between the most activating features of each class between the red and green domains. This suggests that the domain information (color) and instance information (actual content of the image) are somewhat interleaved in these representations, causing

different sets of features to be strongly activated for the same class based on the domain. In single-domain SimCLR on CIFAR-10 (no colors), the representations only encode content information, which results in linearly separable representations by class. In multi-domain SimCLR on Colored-CIFAR, a combination of both domain and content information is encoded in every representation which directly affects linear classification performance. Therefore, to achieve comparable performance to single-domain setups, we propose to **disentangle** domain information from representations by plugging in a general-purpose a Domain-Disentanglement Module (DDM) for Self-Supervised Models which is discussed in the next section.



(a) Accuracy: 90.18 (b) Accuracy: 78.52 (c) Accuracy: 87.06 (DDM) (d) Accuracy: 87.06 (DDM)

Figure 2.4: **SimCLR Representation t-SNE before and after DDM:** CIFAR-10 representations are naturally clustered by class, however, Colored-CIFAR representations are clustered by domain which leads to a significant reduction in classification performance. When SimCLR is trained with DDM, the prefix alone has domain-distinguishable representations, while the remaining portion of the representation is domain-invariant, clustered by class. This structure notably improves the classification performance.

2.4 Domain-Disentanglement Module for Self-Supervised Representations

As described in the previous section, self-supervised models in their current state, are not trained to learn content and domain information independently. We hypothesize that disentangling domain information from the learned representations can improve the performance of existing state-of-the-art SSL models in multi-domain setups. We

therefore propose a general-purpose Domain-Disentanglement Module (DDM) that can be simply attached at any SSL encoder during its pre-training. In this work focus on joint-embedding (involving two transformed views) self-supervised encoders [Che+20a; Car+20b; Che+20b; Gri+20; Car+21; BPL21; Zbo+21a; CH21] and not masked image models [He+22].

Recall that for a given sample $\mathbf{x}_i$, its representation is denoted by $f(\mathbf{x}_i) = \mathbf{h}_i \in \mathbb{R}^r$. Let y_i denote the domain of the i^{th} representation. We allocate the first k features of the representation as the domain prefix, $\mathbf{h}_{i,0..k}$, denoted by $\mathbf{h}_i^d$ for ease of notation. The remaining portion of the representation $\mathbf{h}_{i,k..r}$ is denoted by $\mathbf{h}_i^p$. We call $\mathbf{h}_i^d$ as the *domain-variant* portion and $\mathbf{h}_i^p$ as the *domain-invariant* portion. We train the domain prefix of the i^{th} sample according to the following contrastive optimization, $L_{i_{d\_var}} = \log \frac{\sum_{j=1}^{2N} \mathbb{1}_{j \neq i} \mathbb{1}_{y_i = y_j} sim(\mathbf{h}_i^d, \mathbf{h}_j^d)}{\sum_{j=1}^{2N} \mathbb{1}_{y_i \neq y_j} sim(\mathbf{h}_i^d, \mathbf{h}_j^d)}$.

where $sim(a, b) = \exp\left(\frac{1}{\tau} \frac{a^T b}{\|a\| \|b\|}\right)$. This loss maximizes the similarity of the domain prefixes within each domain and minimizes the similarity of domain prefixes across domains. $\mathbf{h}_i^p$ is learned according to any self-supervised loss like SimCLR, MoCo, DINO etc., denoted by $L_{i_{ssl}}$. Splitting the representation in this manner helps us control each portion independently. L_{ssl} ensures that all content information is encoded in a self-supervised manner such that representations can be utilized for downstream tasks. L_{d_var} ensures that the domain prefixes across samples of different domains are distinguishable.

We next ensure that $\mathbf{h}_i^p$ does not contain any domain-related information (domain-invariance constraint). In other words, it should not be possible to predict the the domain label y_i from the representation $\mathbf{h}_i^p$. To achieve this, we pass each $\mathbf{h}_i^p$ through a domain discriminator $D(.)$ and minimize the Wasserstein distance (using the dual form as proposed in [ACB17]), $L_{i_{d\_invar}} = D(\mathbf{h}_i^p, y_i) - D(\mathbf{h}_i^p, y_{rand})$, where $y_{rand} \sim \mathbb{P}(y)$, i.e., randomly drawn from the distribution of domain labels. The final optimization

for the encoder ($f(.)$) and the discriminator ($D(.)$) is,

$$\max_{f} \sum_{i=1}^{2N} \left[\lambda L_{i_{ssl}} + L_{i_{d\_var}} + L_{i_{d\_invar}} \right] \tag{2.1}$$

where λ is a tunable hyperparameter. We optimize both the encoder $f(.)$ and the discriminator D using alternating gradient descent ascent. We train $D(.)$ using gradient penalty to improve its stability as proposed in [Gul+17]. This formulation is similar to [She+17; KLF22], except that we use Wasserstein Distance to disentangle domain information from the remaining portion of the representation space. In summary, our module DDM consists of splitting the representation space into two parts and applying two additional loss terms, L_{d_var} and L_{d_invar}. Note that, DDM can be plugged in while training any existing state-of-the-art self-supervised model.

In Figure 2.3, in the last panel, we show the representation space of SimCLR trained on Colored-CIFAR using DDM. We observe that among the most activating features, the first few features (which are part of the domain prefix) are equivalent for all classes within a domain and clearly distinguishable between both domains. The remaining portion of the representation is completely invariant to any domain information as each class shows very similar feature distribution in both red and green domains. In the t-SNE plots (Figure 2.4(c) and (d)), we observe that the domain prefix is separable by domain whereas the domain-invariant portion shows natural class clusters with overlapping red and green images. This update in structure leads to a significant improvement in top-1 accuracy from 78.52 to 87.06.

2.4.1 Experimental Setup

We use ViT-S [Dos+21] as the base encoder ($f(.)$) for all of our experiments. Our domain discriminator ($D(.)$) is an MLP with LeakyReLU activations. The representations are 384-dimensional with a 24-dimensional domain prefix. We train the

encoder according to various self-supervised baselines including SimCLR [Che+20a], MoCo [He+20], BYOL [Gri+20], DINO [Car+21], SimSiam [CH21] and Barlow Twins [Zbo+21a]. We use the same optimization and scheduling for the encoder as the respective papers. While training with DDM, we use the Adam optimizer for the domain discriminator with a learning rate of 0.005 and cosine-annealing scheduling and $\lambda = 0.5$. We experiment with PACS [Li+17], DomainNet [Pen+19] and the WILDS [Koh+21] multi-domain benchmarks. We use Nvidia GeForce RTX A4000 GPUs for pre-training. We evaluate representations using the linear evaluation protocol [KZB19; BHB19; OLV19] where we train a linear classifier on top of frozen representations and compute the top-1 accuracy over the training and unseen domains.

2.4.2 Self-Supervised Baselines Trained with DDM

In Figure 2.2, we observed that self-supervised baselines (SimCLR, BYOL and MoCo), when trained on a single domain or multiple domains, generalize poorly to unseen domains. These baselines, when pre-trained with DDM, show improved performance on the training domains (PACS Photo, Sketch and Cartoon) as well as significantly improved generalization to the unseen domain (PACS Art Painting). Pre-training on multiple domains with DDM outperforms every self-supervised baseline as shown in Table 2.1 with a maximum of 2.6% improvement on average top-1 accuracy on SimSiam. We also tabulate our results on DomainNet using Painting, Real and Sketch as training domains and Clipart, Infograph and Quickdraw as the unseen domains in Table 2.2. We observe that pre-training with DDM improves upon each self-supervised baseline with a maximum of 3.5% improvement on average top-1 accuracy on BYOL. DDM generalizes significantly better than its baselines showing a 6.1% (SimSiam) increase in PACS (Painting) and a 7.4% (DINO) in DomainNet (Clipart).

To further evaluate the generalization of self-supervised baselines with DDM, we utilize the WILDS benchmark [Koh+21]. In this benchmark, pre-train on iWildCam

(200K samples, 182 classes, 323 domains), Camelyon17 (456K samples, 2 classes, 5 domains), FMoW (141K samples, 62 classes, 80 domains) and RxRx1 (125K samples, 1139 classes, 51 domains). We summarize our results in Table 2.3. On each benchmark, we observe that DDM outperforms the baselines on the unseen validation set. The accuracy in rxrx1 is low since it is a very hard classification task as it contains 1139 classes and 51 domains. We observe a 5.9% increase linear classification accuracy on iWildCam on SimCLR

Table 2.1: SSL baselines trained on PACS (Photo, Sketch and Cartoon) with DDM

Model	**Top-1 Accuracy (Baseline / with DDM)**				
	Photo	**Sketch**	**Cartoon**	**Painting (Unseen)**	**Average**
SimCLR	97.54 / **98.28**	**98.12** / 97.04	98.03 / **99.24**	87.59 / **89.42**	95.32 / **96.00**
MoCo	**93.59** / 93.19	92.71 / **94.36**	91.63 / **92.98**	77.34 / **78.51**	88.81 / **89.76**
BYOL	78.08 / **81.61**	76.55 / **78.24**	75.55 / **75.58**	58.10 / **62.67**	72.07 / **74.53**
DINO	93.67 / **95.25**	94.33 / **96.42**	79.44 / **81.77**	72.12 / **74.43**	85.89 / **86.97**
SimSiam	83.68 / **84.71**	80.97 / **85.44**	**93.75** / 92.59	57.98 / **64.09**	79.09 / **81.71**
Barlow Twins	**85.09** / 83.94	85.44 / **88.07**	92.0 / **92.83**	59.01 / **62.67**	80.39 / **81.89**

Table 2.2: SSL baselines trained on DomainNet (Painting, Real and Sketch) with DDM

Model	**Top-1 Accuracy (Baseline / with DDM)**						
	Painting	**Real**	**Sketch**	**Clipart (Unseen)**	**Infograph (Unseen)**	**Quickdraw (Unseen)**	**Average**
SimCLR	74.49 / **75.99**	79.31 / **82.02**	85.86 / **86.26**	68.60 / **70.48**	34.75 / **39.25**	22.98 / **24.38**	60.99 / **63.06**
MoCo	70.20 / **73.08**	**89.79** / 86.37	86.66 / **88.15**	65.10 / **68.91**	34.56 / **34.75**	19.89 / **22.12**	61.03 / **62.23**
BYOL	56.87 / **59.82**	77.60 / **79.67**	71.43 / **75.21**	50.67 / **55.86**	27.4 / **30.68**	19.33 / **22.85**	50.55 / **54.02**
DINO	**79.53** / 79.11	86.46 / **86.88**	75.8 / **76.50**	66.32 / **73.76**	30.83 / **32.12**	27.71 / **29.08**	61.11 / **62.90**
SimSiam	77.55 / **78.78**	82.02 / **85.88**	86.52 / **88.38**	67.43 / **71.53**	27.03 / **30.56**	22.29 / **25.67**	60.47 / **63.47**
Barlow Twins	56.78 / **61.18**	79.06 / **80.16**	71.56 / **73.90**	60.40 / **64.33**	26.11 / **28.82**	18.67 / **21.70**	52.09 / **55.01**

Table 2.3: SSL baselines trained on WILDS with DDM

Model	**Top-1 Accuracy (Baseline / with DDM)**			
	iWildCam	**Camelyon17**	**FMoW**	**RxRx1**
SimCLR	66.01 / **71.87**	95.19 / **95.68**	38.94 / **41.23**	8.43 / **11.20**
MoCo	67.05 / **69.12**	91.45 / **93.47**	40.04 / **40.23**	5.67 / **5.93**
BYOL	71.69 / **74.88**	95.15 / **96.38**	38.74 / **39.78**	4.39 / **6.20**
DINO	64.55 / **68.07**	94.38 / **95.38**	33.57 / **34.52**	7.32 / **7.66**
SimSiam	60.45 / **61.16**	88.37 / **89.16**	39.27 / **40.05**	6.39 / **7.26**
Barlow Twins	63.17 / **63.84**	96.38 / **97.62**	44.40 / **47.46**	5.79 / **6.65**

Figure 2.5: **DDM with clustering:** When domain labels are not available, we perform DDM with clustering to identify pseudo-domain-labels. In the above plots we show the t-SNE of the SimCLR representations trained on "Cartoon" and "Sketch" domains in PACS. We observe that DDM with robust clustering produces a better separation between domains.

2.5 DDM without Domain Labels

Most real-world multi-domain datasets are unlabelled (i.e., domain label information is not available). In this section, we develop an extension of DDM for such setups by identifying pseudo domain labels via a clustering approach in the representation space. As it is common in clustering, we assume the number of domains (denoted by M) is known. Depending on the multi-domain setup, we can also approximate the number of domains by studying any available meta-data like data sources, geo-location, quality, etc. We can also estimate the number of domains empirically through clustering and visualization.

Domain labels are required in both DDM losses (L_{d_var}, L_{d_invar}) as described in the previous section. Let us consider a fully unlabelled setup, with no domain labels while the number of domains M is known. We first warm up our self-supervised encoder $f(.)$ treating it as a single-domain setup for a few iterations to get somewhat distinguishable representations by domain. We next cluster the representations into M clusters using K-Means clustering [HW79]. Using the cluster assignments as pseudo-domain-labels (y), we continue training the encoder $f(.)$ along with a discriminator

using the DDM optimization, to learn domain-disentangled representations.

In practice, clustering does not discover 100% accurate domain labels, especially for datasets that are distributionally similar. We therefore use a **robust clustering** approach coupled with DDM to prevent outlier clustering noise from affecting the pseudo-domain-labels. Suppose we discover M clusters with centroids $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M$, before assigning pseudo-domain-labels to each sample, we first determine if they are outliers or not. If so, we ignore these samples in the next stages of training to prevent assigning a noisy label to them. We say a representation is *not* an outlier if it is significantly closer to one of the clustering centroids compared to another. Concretely, $\mathbf{h}_i$ is not an outlier if

$$\max \left\{ \frac{\|\mathbf{h}_i - \mathbf{c}_m\|^2}{\|\mathbf{h}_i - \mathbf{c}_n\|^2} \ : 1 \leq m \leq M, 1 \leq n \leq M \right\} > 1 + \epsilon \tag{2.2}$$

where $\epsilon \geq 0$ is defined as the *outlier threshold*. When ϵ is high, it means that the given sample is very close to its respective centroid. When ϵ approaches 0, it indicates that the sample is almost equidistant from at least two centroids and therefore, may not be reliably assigned one pseudo-label. We ignore such samples going forward in training. When we perform clustering for the first time, we start with $\epsilon = 1$. We repeat the clustering at regular intervals of training on the representations $\mathbf{h}$ to get improved cluster centroids. Each time we repeat clustering, we decay the value of ϵ exponentially such that it approaches 0. By the end of training, all samples will contribute to the training of the self-supervised encoder with DDM. In Figure 2.5, we illustrate the difference between regular clustering and robust clustering with MDSSL trained on the PACS dataset [Li+17] ("Cartoon" and "Sketch" domains). We observe that robust clustering helps in identifying more accurate and distinguishable clusters.

To evaluate DDM with robust clustering, we combine CIFAR-10 [KNHa], CIFAR-

100 [KNHb] and STL-10 [CLN] to form a multi-domain dataset. The constituent datasets are distributionally similar with several shared classes (CIFAR-10 and STL-10 share 9 out of 10 classes). With this setup, we try to simulate a real-world scenario where data arises from various domains however the actual domains are undefined. We therefore apply DDM with robust-clustering to identify pseudo-domain-labels. We then evaluate the pre-trained representations by linear probing the validation portion of each constituent dataset. We include Tiny-ImageNet [LY15] as an unseen domain to test generalization.

In Table 2.4, we tabulate the results on this prepared multi-domain dataset on various self-supervised baselines with and without DDM and robust clustering. We observe an improvement in the average top-1 accuracy across all baselines with 1.7% improvement in MoCo. DDM shows improved generalization on Tiny-ImageNet with a 2.9% increase in DINO.

Table 2.4: SSL baselines trained on a mixture of CIFAR-10, STL-10 and CIFAR-100 using DDM and robust clustering

Model	**Top-1 Accuracy (Baseline / with DDM and robust clustering)**				
	CIFAR-10	**STL-10**	**CIFAR-100**	**Tiny-ImageNet (Unseen)**	**Average**
SimCLR	89.43 / **90.03**	79.77 / **81.01**	63.33 / **64.90**	49.58 / **51.22**	70.53 / **71.79**
MoCo	**90.80** / 90.69	80.02 / **81.60**	61.57 / **64.28**	37.16 / **39.55**	67.38 / **69.03**
BYOL	88.31 / **89.68**	75.07 / **75.72**	64.82 / **65.56**	50.04 / **51.10**	69.56 / **70.52**
DINO	90.61 / **92.96**	**84.7** / 82.35	62.63 / **63.57**	49.52 / **52.46**	71.87 / **72.84**
SimSiam	87.02 / **87.38**	72.15 / **73.78**	**62.08** / 61.90	33.11 / **34.78**	63.59 / **64.46**
Barlow Twins	88.31 / **89.01**	75.59 / **76.11**	65.03 / **66.89**	40.27 / **41.31**	67.30 / **68.33**

2.6 Conclusion

We proposed a Domain Disentanglement Module (DDM) for self-supervised encoders that provide better structured representations, domain-invariant representations that can be used for diverse multi-domain tasks. DDM also supports training over setups where domain labels are not available by using a robust clustering technique that reduces outlier noise. With DDM, we prevent the need for having to train multiple

single-domain encoders and instead leverage a single encoder to perform comparably on multiple domains. The benefit of invariant representations is better generalization which we show on various benchmarks including PACS, DomainNet and WILDS.

Chapter 3

Measuring Self-Supervised Representation Quality for Downstream Classification using Discriminative Features[1]

3.1 Introduction

Self-supervised models [Che+20a; Car+20b; Che+20b; Gri+20; CH21; Car+18a; Kho+20; Car+21; BPL21; Zbo+21b] learn to extract useful representations from data without relying on human supervision, and perform comparably to supervised models in downstream classification tasks. Pre-training these models can be highly resource-intensive and time-consuming. It is therefore crucial that the learned representations are of high quality such that they are explainable and generalizable. However, in practice, these representations are often quite noisy and un-interpretable, causing difficulties in understanding and debugging their failure modes [Jin+22; HYZ21;

[1]Full paper available at https://arxiv.org/abs/2203.01881. Contributing authors include Kanika Narang, Hamed Firooz, Maziar Sanjabi and Soheil Feizi.

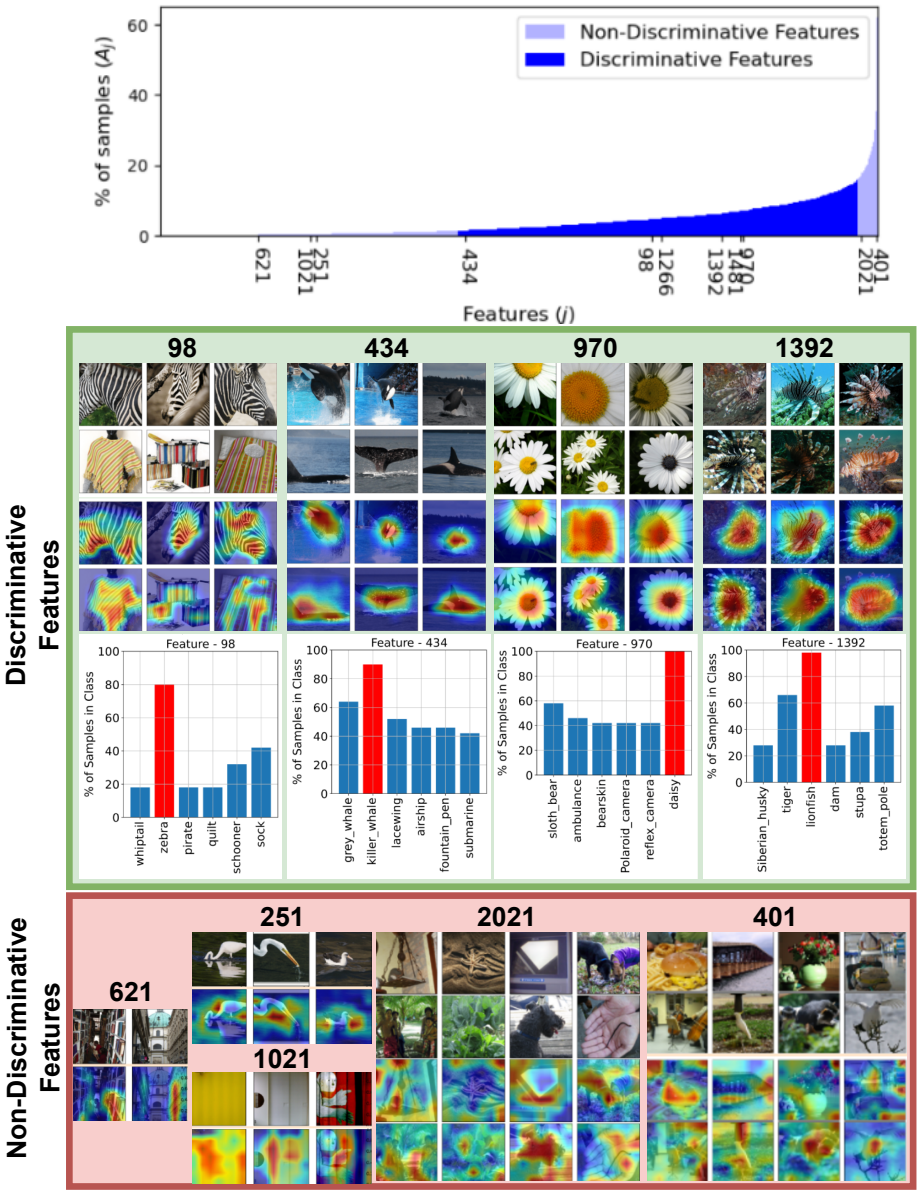


Figure 3.1: **Discriminative Features in Self-Supervised (SSL) Models:** We plot the percentage of highly activating samples for each feature in the SimCLR (ResNet-50) representation space. The features that show very low or very high percentage activations are *non-discriminative* as they likely correspond to very uncommon (lower tail) or very general attributes (upper tail). The features that activate a moderate number of samples (middle portion) are called *discriminative features*. As shown in the gradient heatmaps, these features encode important physical attributes shared among specific classes. These features play a key role in assessing the quality of SSL representations for downstream linear classification tasks.

EGH21a].

In this paper, our goal is to study the representation space of pre-trained self-supervised encoders (SSL) such as SimCLR [Che+20a], SwaV [Car+20b], MoCo [Che+20b], BYOL [Gri+20], SimSiam [CH21], DINO [Car+21], VICReg [BPL21] and Barlow Twins [Zbo+21b] and discover their informative features in an unsupervised manner. We observe that representations are mostly sparse, containing a small number of *highly activating features*. These features can strongly activate a small, moderate or large number of samples in the population. We refer to the moderate category of features as *discriminative features*.

We observe some intriguing properties of discriminative features: (i) Although discovered without any class label information, they can be strongly correlated to a particular class or group of classes (See Figure 3.1); (ii) They highlight informative concepts in the activating samples which are often related to the ground truth of those samples; (iii) They activate strongly in correctly classified representations rather than mis-classified representations (as shown in Figure 3.3) and finally, iv) Representations can be compressed by up to 40% using discriminative features without significantly affecting linear evaluation performance.

Building on these observations, we propose an unsupervised, sample-wise **Self-Supervised Representation Quality Score (Q-Score)**. A high Q-Score for a sample implies that its representation contains highly activating discriminative coordinates which is a favorable representation property. We empirically observe that Q-Score can be used as a zero-shot predictor in distinguishing between correct and incorrect classifications for any SSL model achieving AUPRC of 91.45 on ImageNet-100 and 78.78 AUPRC on ImageNet-1K.

We next apply Q-Score as a regularizer and further-train pre-trained SSL models at a low rate to improve low-quality representations. This improves the linear probing performance across all our baselines, highest on BYOL (5.8% on ImageNet-100 and

3.7% on ImageNet-1K). The representations, after regularization, show increased activation for discriminative features (Figure 3.3) due to which several previously mis-classified samples get correctly classified with higher confidence.

Finally, we define a metric for quantifying representation interpretability using Salient ImageNet [SF21] masks as ground truth. Discriminative features are strongly correlated to *core* features of Salient ImageNet. We can potentially explain these features by correlating their meanings with the feature annotations provided for core features in Salient ImageNet. We also observe that discriminative features in mis-classified representations are less correlated with core features compared to correct classifications. Q-score regularization improves this correlation for both correct and mis-classified representations, thereby making representations more explainable.

3.2 Related Work

Unsupervised methods for classification has been a long-standing area of research, traditionally involving the use of clustering techniques [BJ17; Dos+14; YCA20; Bau+16; Car+18b; Car+19; Hua+19]. Self-supervised learning, is a powerful approach that enables learning by preparing own labels for every sample [BJ17; Dos+14; Wu+18; Dos+16] usually with the help of a contrastive loss [Aro+19; TKH21; BHB19]. Positive views in SSL losses are multiple transformations [Tia+20b] of a given sample using stochastic data augmentation. Through this approach, several state-of-the-art SSL techniques [Che+20a; Car+20b; CH21; Gri+20; Che+20b; Kho+20] have produced representations that show competitive linear classification accuracy to that of supervised approaches.

Understanding these learned representations is relatively less explored. Several feature interpretability techniques exist [Bau+17a; Kal+23; Her+22a], that aim to explain individual neurons with natural language. However, our goal is to study

representations through the lens of failure modes and generalization. [Jin+22], observes that self-supervised representations collapse to a lower dimensional space instead of the entire embedding space. Other methods [Küg+21; Xia+21], propose to separate the representation space into variant and invariant information so that augmentations are not task-specific. [Gri+21] observes representations across layers of the encoder and compare it to supervised setups. Clustering-based or prototypical-based methods have also been proposed where the representation space is collapsed into a low-rank space [Dwi+21; KTP21]. [BBV21] uses an RCDM model to understand representation invariance to augmentations. [Gar+22; LEP22] propose a score based on the rank of all post-projector embeddings that can be used to judge and compare various self-supervised models.

In this work, we focus more on studying the properties of representations across correct and incorrect classifications in downstream linear probing (without using any labels). We investigate the connection between these unsupervised properties in the representation space and mis-classifications. Unlike [Gar+22; LEP22] which requires computing rank over the entire dataset, our analysis leads to the development of an unsupervised *sample-wise* quality score which can be used as a regularizer and effectively improve downstream classification performance.

3.3 Understanding Representations and their Failure Modes

Let us consider a pre-trained self-supervised model with a ResNet [He+16] backbone encoder $f(.)$. Given an input sample, $\mathbf{x}_i \in \mathbb{R}^n$ its representation is denoted by, $f(\mathbf{x}_i) = \mathbf{h}_i \in \mathbb{R}^r$, where r is the size of the representation space.

Upon visual analysis (See Appendix for more details), we observe that each representation is *nearly* sparse, i.e., most feature activations are close to zero [Jin+22].

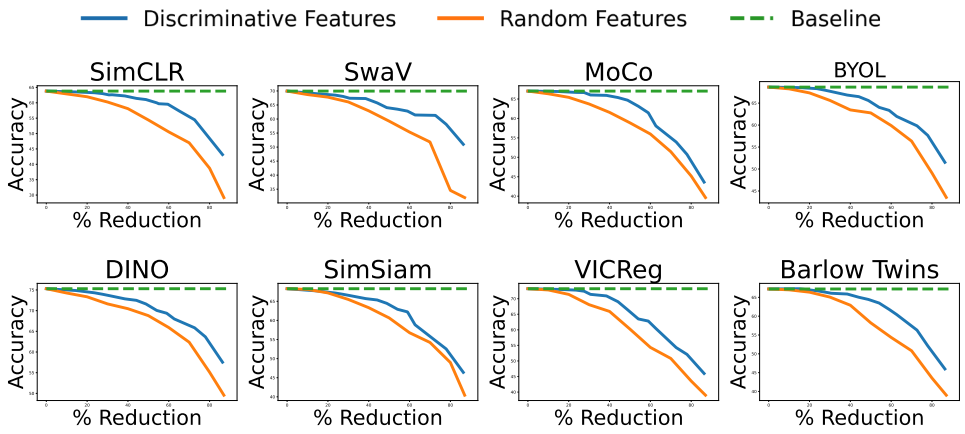


Figure 3.2: **Linear probing discriminative features:** We train linear classifiers after selecting subsets of discriminative features of various sizes (middle portion of Figure 3.1) and plot their top-1 accuracy for various SSL baselines. Classifiers trained using discriminative features consistently outperform those of randomly selected features (averaged over 4 random seeds). We can achieve up to 40% reduction in representations size using discriminative features without significantly affecting the top-1 accuracy.

There exists a select few features that are strongly deviated from the remaining features in that representation. For any given representation $\mathbf{h}_i \in \mathbb{R}^r$, we formally define the **set of highly activating features** (L_i) as $L_i := \{j : h_{ij} > \mu_i + \epsilon\sigma_i\}$, where μ_i and σ_i denote the mean and standard deviation of $\mathbf{h}_i$ respectively and ϵ is a hyperparameter that is empirically selected. We use $\epsilon = 4$ in our experiments. In all our analysis, we perform L2 normalization over every $\mathbf{h}$ to ensure fair comparison of features. For every feature j, the percentage of highly activating samples is denoted by, $A_j = \frac{100}{N}\sum_{i=1}^{N} \mathbb{1}_{j \in L_i}$ where N is the size of the population. In the top panel of Figure 3.1, we plot A_j for all features j in the SimCLR representation space of the ImageNet-1K train set. The x-axis is ordered in ascending order of A_j. In the next section, we dissect this plot to group features based on A_j.

Our observations do not directly extend to ViT-based SSL encoders since, unlike ResNet encoders, their representations can also contain negatively activated features. They need not be sparse in nature (no ReLU before representation layer), rather, features can be both positively (highly activating) or negatively (lowly activating)

correlated to important class-specific concepts. In our work, we observe several unique properties of highly activating features in ResNet representations which are beneficial to detect failures. We see ViT encoders as an important direction for future work and focus on ResNet-based encoders for our study.

3.3.1 Discriminative Features

Based on Figure 3.1, we can define three broad categories of highly activating features: (i) Features that are highly activating across a very small fraction of the population, corresponding to the lower tail features in Figure 3.1. We take the example of features 621, 251 and 1021 and visualize their highly activating samples and gradient heatmaps (using GradCAM [Sel+19]). Since these features activate very few samples, they likely correspond to image-specific or uncommon concepts. Such features would also not be useful in classification tasks as these are not shared, class-relevant attributes. (ii) Features that highly activate a very large number of samples in the population i.e, the upper tail features in Figure 3.1. Like feature 2021 and 401, such features are likely to encode very broad and general characteristics (like texture, color etc.) common to most samples (spanning various classes) and therefore, are not class-discriminative. The third category includes, (iii) Features that highly activate a moderate number of samples in the population (i.e. the middle part in Figure 3.1). These features are most likely to encode unique physical attributes associated with particular classes. For example, feature 98 corresponds to the "stripe" pattern which is an important property of the zebra class. Similarly, feature 970 corresponds to the style of the daisy class, and feature 1392 corresponds to lionfish in different scenes. We refer to this subset of highly activating features as *discriminative features*. Note that we did not use any label information for this analysis. We can identify discriminative and non-discriminative features in a fully unsupervised manner by simply observing their percentage activations (A). The bar plots in Figure 3.1, show that these features

activate more than 80% of particular classes which confirms that these features are strongly class-correlated.

Discriminative features can be regarded as a summarization of the top concepts related to each class of the dataset the encoder is trained on. We justify the described method of selection in Figure 3.2, where we plot the top-1 accuracy of a linear classifier trained on ImageNet-1K using subsets of discriminative features of varying sizes as chosen from Figure 3.1 (middle portion). We compute the percentile for each point in the distribution A and gradually increase the lower limits (from 0^{th} percentile), and decreasing the upper limits (from 100^{th} percentile) to get multiple sets of discriminative features of varying sizes. We also plot the top-1 accuracy when random subsets of features are selected. We observe that discriminative features perform significantly better compared to randomly selected features. We also observe that we can reduce the representation size up to 40% using the discriminative features, with minimal reduction in performance. In practice, we select the discriminative features between the 50^{th} and the 95^{th} percentile of A (as shown in Figure 3.1). This range can be discovered empirically and can be further tuned for each model-dataset pair. In the Appendix, we also show that selecting features from either the lower or upper tail of A also under-perform compared to discriminative features from the middle portion.

While we analyze features independently in our work, it has been shown [Kal+23; Elh+22] that not all neural features are axis-aligned. Meaningful class-related concepts can also be encoded by multiple features (See Appendix for examples). In such cases, the whole *group* of features can be considered as discriminative. These groups can be highly activating for specific classes and lie in the middle portion of A. We also perform a PCA analysis on the representation space (see Appendix) to partially validate our selection method.

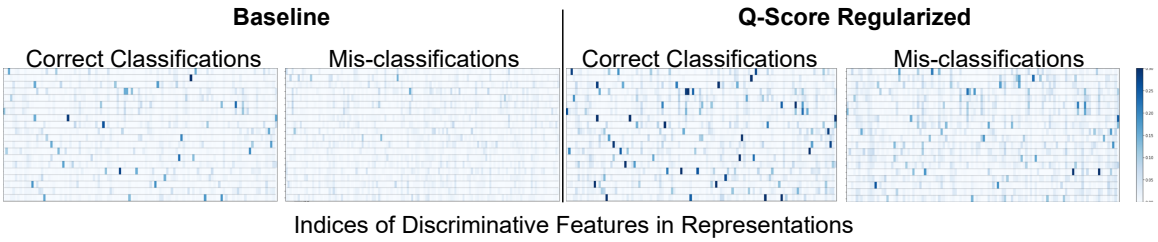


Figure 3.3: **Comparing correct and mis-classified representations:** In these heatmaps, we visualize the discriminative features of average SimCLR representations of several ImageNet-1K classes - correct (left) and incorrect (right) classifications. In the baseline, we observe that discriminative features are strongly activated only in correctly classified representations. Q-Score regularization improves discriminative features' activations, even in mis-classified representations.

3.3.2 Mis-classified Representations

We now study how discriminative features play a key role in detecting potential mis-classifications in a fully unsupervised manner. In Figure 3.3, we take SimCLR ImageNet-1K representations and visualize the discriminative features. On the left, we show the average representations of correctly classified samples (after linear probing) in a subset of classes, while on the right, we show the same for the mis-classified samples in those classes. The subset of features we display is the same for correct and incorrect classifications.

As we can see, in Figure 3.3, in the first panel, there is a clear difference between representations of correctly and incorrectly classified examples. Both correct and mis-classified representations are *nearly* sparse, however, the discriminative features are significantly more activated in correct classifications. This is especially interesting because we can visually distinguish between correct and incorrect classifications, just by observing the discriminative features, without using any label information. Note that this observation does not depend on the actual ground truths or predicted labels of the linear classifier rather, just a binary outcome or whether or not a sample was correctly classified.

The correlation of discriminative features to unique physical attributes as studied in

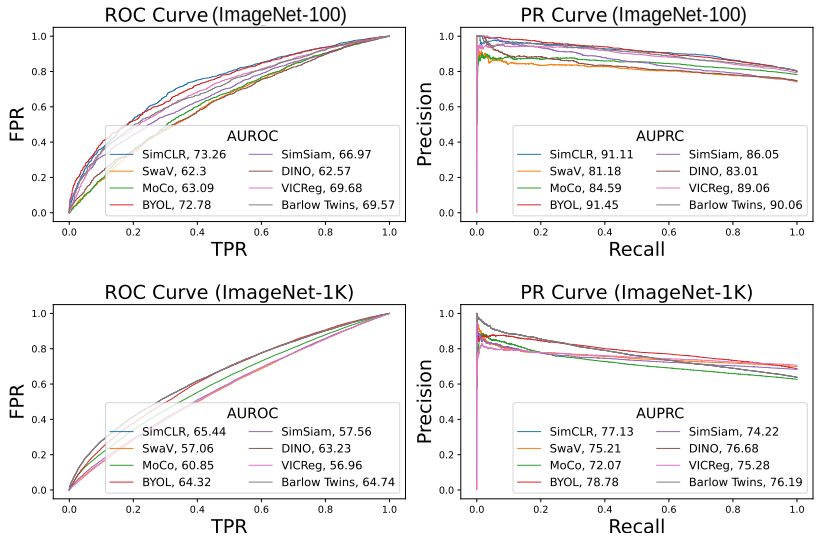


Figure 3.4: **Precision-Recall and ROC curves of Q-Score:** We measure the effectiveness of Q-Score when used as a predictor in distinguishing between correct and mis-classified representations on ImageNet-100 and ImageNet-1K on each SSL model. Q-Score shows an AUPRC of up to 91.45 on ImageNet-100, 78.78 on ImageNet-1K and AUROC of 73.26 on ImageNet-100, 65.44 on ImageNet-1K.

the previous section, suggests that their presence may be useful in correctly classifying representations. In Figure 3.3, our claim is confirmed as we observe that mis-classified representations do not show high activations on these features. Therefore, for any given sample, we can consider discriminative features as strong signals indicating classification outcome without requiring to train a linear classification head. We would like to emphasize that our results only indicate an *association* between these structural properties and classification accuracy and we do not claim any causal relationship between the two.

3.4 Self-Supervised Representation Q-Score

Our study of learned representation patterns helps us discover discriminative features in an unsupervised manner. These features encode class-specific attributes and help us visually distinguish between correct and incorrect classifications. We combine these observations to design a sample-wise quality score for SSL representations. Let us

define D, such that $|D| < r$, as the set of discriminative features for a given SSL model trained on a given dataset. For the i^{th} sample, we have $\mathbf{h}_i$ (representation), μ_i (mean of $\mathbf{h}_i$), σ_i (standard deviation of $\mathbf{h}_i$) and the set of highly activating features $L_i = \{j : h_{ij} > \mu_i + \epsilon\sigma_i\}, |L_i| < r$. We define our Self-Supervised Quality Score for sample i as,

$$Q_i := \frac{1}{|L_i \cap D|} \sum_{j \in L_i \cap D} (h_{ij} - \mu_i) \tag{3.1}$$

where, $L_i \cap D$ is the set of discriminative features specific to the i^{th} sample. Intuitively, higher Q_i implies that the representation contains highly activated discriminative features which are strongly deviated from the mean. Our objective with this metric is to compute a sample-specific score in an unsupervised manner indicating the quality of its representations. Ideally, we would like to argue that samples with higher Q-score have improved representations and thus are more likely to be classified correctly in the downstream task. This is a general score that can be applied to any ResNet-based SSL model trained on any dataset. See Appendix for a discussion on Q-Score in supervised models.

Next, we measure how effective our score is in differentiating between correctly and incorrectly classified representations in an unsupervised manner. In Figure 3.4, we plot the Precision-Recall (PR) curve and the Receiver Operating Characteristic (ROC) curve of Q-Score when used as a predictor of classification outcome (correct or incorrect). We show this for SimCLR, SwaV, MoCo, BYOL, DINO and SimSiam for ImageNet-100 (top panel) and ImageNet-1K (bottom panel). We also compute the AUROC (area under receiver operating characteristic curve) and AUPRC (area under precision-recall curve) of these curves. We observe AUPRC up to 91.45 on ImageNet-100 and 78.78 on ImageNet-1K on BYOL. On SimCLR, we observe AUROC

up to 73.26 on ImageNet-100 and 65.44 on ImageNet-1K. Based on these results we can conclude that, Q-Score is a reliable metric in assessing the quality of representations, meaning that representations with lower Q-Score (quality), are more likely to be mis-classified.

We now check if promoting Q-Score on pre-trained representations is helpful. To do so, we take state-of-the-art pre-trained SSL models and further train them for a small number of iterations with Q-Score as a regularizer. For example, we can apply this regularizer to the SimCLR optimization as follows,

$$\max_{\theta} \frac{1}{2N} \sum_{i=1}^{2N} \Big[\log \frac{sim(\mathbf{z}_i, \tilde{\mathbf{z}}_i)}{\sum_{j=1}^{2N} \mathbb{1}_{j \neq i} sim(\mathbf{z}_i, \mathbf{z}_j)} + \lambda_1 \mathbb{1}_{Q_i < \alpha}(Q_i) \Big] \tag{3.2}$$

where $\mathbf{z}$ is the latent vector computed by passing $\mathbf{h}$ through a projector network and $sim(.)$ denotes the exponentiated cosine similarity of the normalized latent vector. α is a threshold with which we select the low-score samples whose Q-Scores should be maximized and λ_1 is the regularization coefficient. In other words the goal of this regularization is to improve low-quality representations, similar to the ones shown in Figure 3.3, by maximizing their discriminative features for downstream classification.

In practice, directly applying this regularization could lead to a trivial solution where a small set of features get activated for all samples. This is not a favorable situation because these representations become harder to classify accurately and more importantly, the discriminative features are no longer *informative* because they are activated for all samples (similar to the upper tail in Figure 3.1). Such features have significantly large L1 norms *across* samples compared to the remaining features. Therefore, in our revised optimization, we penalize features that have large L1 norms across samples. Let us denote the representation matrix of a given batch by $\mathbf{H} \in \mathbb{R}^{2N \times r}$ and $\|\mathbf{H}_{*,k}\|_1$ represents the L1 norm of the k^{th} column (corresponding to

Table 3.1: **Boosting linear classification performance with Q-Score regularization:** We tabulate the top-1 accuracy of linear evaluation on SimCLR, SwaV, MoCo, BYOL, DINO, SimSiam, VICReg and Barlow Twins with and without Q-Score regularized fine-tuning and a simple lasso regularization. We observe that Q-Score regularization consistently improves each SSL state-of-the-art baseline achieving up to 5.8% relative improvement on ImageNet-100 and 3.7% on ImageNet-1K.

Model	ImageNet-100			ImageNet-1K		
	Baseline	Lasso	Q-Score	Baseline	Lasso	Q-Score
SimCLR	78.64	75.63	**80.79** (+2.2%)	63.80	61.48	**66.18** (+2.3%)
SwaV	74.36	74.56	**78.90** (+4.5%)	69.95	67.35	**71.05** (+1.1%)
MoCo	79.62	78.81	**85.16** (+5.5%)	67.03	65.12	**69.31** (+2.2%)
BYOL	80.88	78.73	**86.72** (+5.8%)	69.14	68.47	**72.81** (+3.7%)
DINO	75.41	75.18	**76.39** (+1.0%)	75.52	72.89	**75.78** (+0.3%)
SimSiam	78.80	78.42	**81.41** (+2.6%)	68.62	68.63	**70.47** (+1.9%)
VICReg	79.77	76.95	**81.56** (+1.8%)	73.63	72.86	**74.72** (+1.1%)
Barlow	80.63	80.32	**81.03** (+0.4%)	67.85	66.47	**69.58** (+1.7%)

the k^{th} feature). Our regularized objective would then be,

$$\max_{\theta} \frac{1}{2N} \sum_{i=1}^{2N} \Big[\log \frac{sim(\mathbf{z}_i, \tilde{\mathbf{z}}_i)}{\sum_{j=1}^{2N} \mathbb{1}_{j \neq i} sim(\mathbf{z}_i, \mathbf{z}_j)} + \lambda_1 \mathbb{1}_{Q_i < \alpha}(Q_i) \Big] - \lambda_2 \sum_{k=1}^{r} \mathbb{1}_{\|\mathbf{H}_{*,k}\|_1 > \beta}(\|\mathbf{H}_{*,k}\|_1) \tag{3.3}$$

where the threshold β helps us select the uninformative features whose L1 norms should be minimized. In practice, we choose α and β for each batch as the mean values of Q_i and $\|\mathbf{H}_{*,k}\|_1$ respectively.

3.4.1 Experimental Setup

Our setup consists of state-of-the-art self-supervised ResNet encoders ($f(.)$) - SimCLR, SwaV, MoCo, BYOL, DINO (ResNet-based), SimSiam, VICReg and Barlow Twins that are pre-trained on datasets - ImageNet-1K, ImageNet-100 [Rus+15b]. We use a ResNet-50 encoder for our ImageNet-1K experiments and ResNet-18 encoder for all other datasets. We discover discriminative features for each pre-trained model using the train set of each dataset. For Q-Score regularization, maintaining the same

encoder architecture as the respective papers, we use the LARS [YGG17a] optimizer with warmup-anneal scheduling. We further-train each pre-trained model with and without Q-Score regularization (controlled by λ_1 and λ_2) using a low learning rate of 10^{-5} for 50 epochs. We find that $\lambda_1 = \lambda_2 = 10^{-4}$ generally works well for fine tuning. We use a maximum of 4 NVIDIA RTX A4000 GPUs (16GB memory) for all our experiments. Using the implementations from solo-learn [Cos+22], we have tried to match our baseline numbers as much as possible within the error bars reported in the papers using the available resources. We follow the standard evaluation by training a linear classifier on frozen pre-trained representations for 100 epochs. For all our gradient heatmap visualizations, we utilize GradCAM [Sel+19].

3.4.2 Q-Score Regularization

We tabulate our linear evaluation results of various SSL baselines before and after Q-Score regularization in Table 3.1. We also include results on lasso (L1) regularization [Tib96] on pre-trained models. Lasso promotes sparsity by minimizing the L1 norm of representations. Q-Score regularization improves the linear probing top-1 accuracy on all of the SSL state-of-the-art models. We observe the most improvement on BYOL showing 5.8% increase in accuracy on ImageNet-100 and 3.7% on ImageNet-1K. Lasso regularization shows degraded performance across most models since naively sparsifying representations can lead to loss of information. In contrast, Q-Score regularization promotes highly activating discriminative coordinates which we have shown to be essential for downstream classification. We include more results on CIFAR-10 [KNHa], STL-10 [CLN] and CIFAR-100 [KNHb] in the Appendix. We also include the results on the transfer performance of discriminative features and Q-Score regularized ImageNet-1K models on unseen datasets in the Appendix. Q-Score is therefore a powerful regularizer that can boost the performance of state-of-the-art SSL baselines.

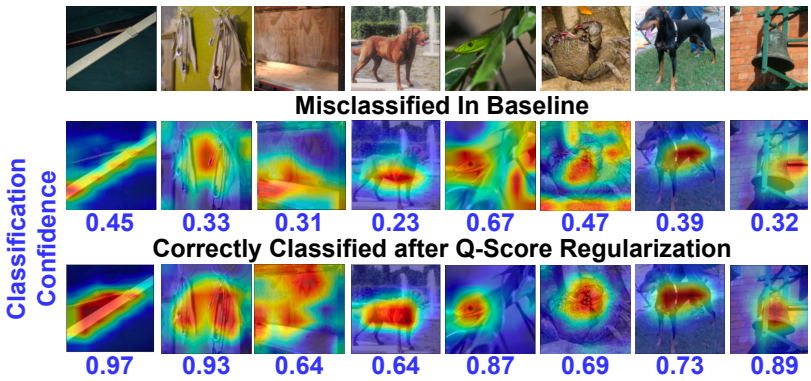


Figure 3.5: **Discriminative features in mis-classified samples:** The discriminative features' heatmaps on the SimCLR (baseline) activate portions that may not be relevant to the image ground truth, leading to incorrect predictions. After Q-Score regularization on these representations, the heatmaps become more localized and less noisy, whilst improving predictions and confidence.

In addition to top-1 accuracy, Q-Score also shows significant improvement in representation quality. In Figure 3.3, we compare the discriminative features of representations before and after Q-Score regularization. We observe that the magnitude of discriminative features and consequently the Q-Score, increases for both correct and mis-classified representations after regularization, making it harder to differentiate between them. For example on SimCLR ImageNet-1K, the AUROC reduces to 59.81 and AUPRC to 71.28. We also observe improved classification confidence as representations become more disentangled (see Appendix for a discussion on this). Our regularization produces better quality representations with clear discriminative features making them more distinguishable across classes and therefore, easier to classify. Due to this, we can attribute the improvement in performance to improved representation quality. Although Q-Score improves accuracy, it does not entirely prevent mis-classifications as mis-classifications may occur due to a variety of reasons such as, training augmentations, hardness of samples, encoder complexity, dataset imbalance etc.

Our motivation for using discriminative features as discussed in Section 3.3 is because - a) they are at clear contrast between correct and incorrect classifications,

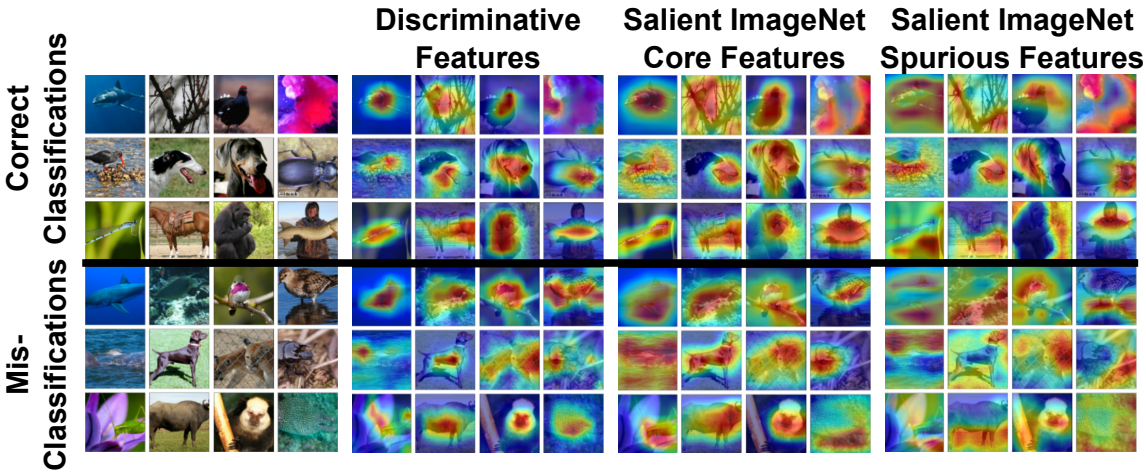


Figure 3.6: **Comparing discriminative features with Salient ImageNet *core* and *spurious* features:** We compare the gradient heatmaps of discriminative features correct and incorrect classifications of SimCLR on ImageNet-1K with the core and spurious masks of the same images in Salient ImageNet. We observe that discriminative features generally overlap more with core features in Salient ImageNet.

and b) they show strong correlation to ground truth. We observed in Figure 3.3 in the baseline, that the discriminative features in correctly classified samples are not strongly activated in mis-classified samples. We now study some mis-classified samples and observe how their features may improve with Q-Score regularization. In Figure 3.5, we visualize the gradient heatmaps of the discriminative features of some mis-classified examples in SimCLR. In the baseline, we observe that discriminative features do highlight portions of the image relevant to the ground truth, however, they may also activate other portions that are not necessarily important (see rock crab and green mamba). These heatmaps reflect low quality representations where the discriminative features are not strongly deviated from the mean. After Q-Score regularization, the maximization of discriminative features also leads to better gradient heatmaps that are more localized and cover almost all important portions of the image relevant to the ground truth. Therefore, these samples get classified correctly with higher confidence after regularization.

3.5 Quantifying Representation Interpretability with Salient ImageNet

We have observed that discriminative features in representations correspond to meaningful physical attributes through gradient heatmaps and they play a key role in deciding the downstream classification outcome. In this section, we quantify the interpretability of these features between correct and incorrect classifications. We utilize Salient ImageNet [SF21] as the ground truth baseline to compare our gradient heatmaps with. The Salient ImageNet dataset contains annotated masks for both *core* and *spurious* features extracted from a supervised robust ResNet-50 model for 6858 images spanning 327 ImageNet classes. It also contains some natural language keywords, provided by workers to explain each feature. Core features are those that are highly correlated with the ground truth of the image, whereas, spurious features are those that activate portions irrelevant to the ground truth. In Figure 3.6, we study some correct and mis-classified samples in the SimCLR baseline. We plot the gradient heatmaps of the discriminative features (combining each individual feature heatmap) of SimCLR for each respective image. We also plot the core and spurious masks of the same images from the Salient ImageNet dataset. We observe that discriminative SimCLR features mostly capture relevant and defining characteristics of the images, therefore are highly correlated with the ground-truth. Moreover, for every correctly classified image, these heatmaps overlap more with core features than spurious features in Salient ImageNet. Discriminative features in mis-classified images also overlap with core features in most cases. Since discriminative features are very closely related (in terms of overlap) to *core* features, we can potentially explain these features better with the help of worker annotations in Salient ImageNet. Therefore, these features can be considered as *interpretable*.

We quantitatively measure the interpretability of a given representation of a

given model by computing the Intersection over Union (mIoU) between the heatmap of discriminative features and the core or spurious mask of that image in Salient ImageNet. We can extend this to measure the overall interpretability of a given model by computing the mean Intersection over Union (mIoU) over the population. For the i^{th} image , we define Ar_i as the area of the heatmap of its discriminative features. Let Ar_i^{core} and Ar_i^{sp} be the area of the core and spurious masks respectively. The mIoU scores are defined as follows,

$$\text{mIoU}^{core} = \frac{1}{N} \sum_i \frac{s(Ar_i \cap Ar_i^{core})}{s(Ar_i \cup Ar_i^{core})}$$
$$\text{mIoU}^{sp} = \frac{1}{N} \sum_i \frac{s(Ar_i \cap Ar_i^{sp})}{s(Ar_i \cup Ar_i^{sp})}$$

where $s(.)$ calculates the sum of the pixel values of the discriminative features' heatmap in the given area. Higher $\text{mIoU}^{core}\%$ indicates that, on an average higher percentage of the feature heatmap overlaps with the annotated core region, meaning that the model features are more interpretable.

In Figure 3.7, we show that for all SSL baselines, $\text{mIoU}^{core} > \text{mIoU}^{sp}$ for both correct and incorrect classifications which confirms that discriminative features generally encode important and core attributes over the whole population. Among correct and mis-classified samples in the baselines, we observe that the mIoU^{core} of correct classifications is higher than mis-classifications. This aligns with our observations in Figure 3.5, which shows that discriminative features in mis-classified samples may not be strongly deviated from the mean and therefore, may correspond to less important portions of the image. After Q-Score regularization, we observe an increase in mIoU^{core} for both correct and mis-classified samples compared to the baseline. This shows that our regularization which enhances discriminative features produces better gradient heatmaps which are more overlapped with core portions of images

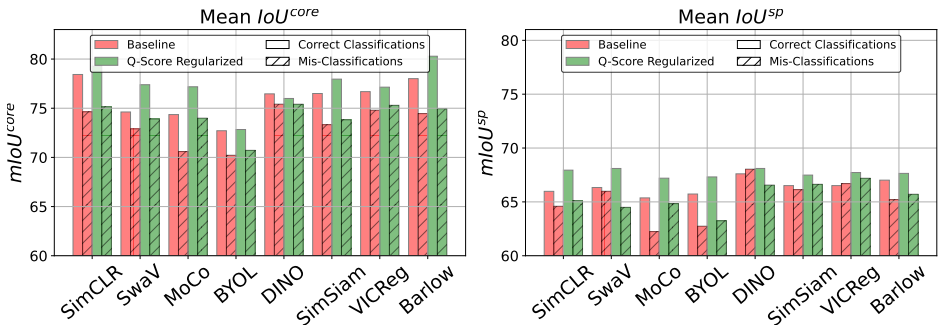


Figure 3.7: **mIoU scores with Salient ImageNet features:** We compute the mean mIoU^{core} and mIoU^{sp} scores of SSL baselines (using their discriminative features) before and after Q-Score regularization. We observe that discriminative features for all models generally show higher % IoU with core features than spurious features. Mis-classified representations show relatively lower % IoU with core features. After Q-score regularization, we observe that mIoU^{core} generally improves for both correct and mis-classified representations.

and therefore, improves the overall model interpretability. Note that, spurious feature heatmaps (Figure 3.6) cover almost all the image content. As shown in Figure 3.5, discriminative feature heatmaps, after regularization, become larger or smaller to capture all core characteristics of the image. This can cause mIoU^{sp} to be higher or lower after regularization. We therefore only use mIoU^{sp} to compare with mIoU^{core} and not to analyze the regularization effect.

3.6 Conclusion

We studied the representation space of SSL models to identify discriminative features in a fully unsupervised manner. Using discriminative features, we compress the representation space by up to 40% without largely affecting the downstream performance. We defined an unsupervised sample-wise score, Q-Score, that uses discriminative features and is effective in determining how likely samples are to be correctly or incorrectly classified. We regularized with Q-Score and remedied low-quality samples, thereby, improving the overall accuracy of state-of-the-art SSL models on ImageNet-1K by up to 3.7%, also producing more explainable representations. Our work poses several

questions and directions for future work: (i) What are other causes for failures and poor generalization in SSL models apart from representation quality?; (ii) Studying properties for ViT-based representations in a similar fashion is a crucial next step to our work as ViT is a widely used SSL encoder; (iii) How can representations be better structured for non-classification-based downstream tasks such as object detection and semantic segmentation, where discriminative features should correlate to the object/segment categories (which can be several per-image).

Chapter 4

Identifying Interpretable Subspaces in Image Representations[1]

4.1 Introduction

Learning generalizable representations has a growing requirement given the considerable cost of pre-training and inference. More importantly, understanding what is encoded in representations is a necessity for deployment, particularly in medical and safety-critical applications [Sal+21]. Large pre-trained self-supervised models [Car+21; Che+20a; Che+20b; CH21] have shown successful generalization capability with frozen representations, however, their representation spaces are still not fully understood. Prior works attempt to understand neural features through detailed visualization of concepts [Ola+20; OMS17; Sel+19; Zha+21; Gho+19]. Visualization (via saliency) helps discover various attributes that neurons react to, but can be noisy and greatly ambiguous requiring manual inspection to achieve any useful explanation. Natural language explanations can complement saliency heatmaps by providing a small number of conceptual keywords that accurately describe the salient component.

[1]Full paper available at https://arxiv.org/abs/2307.10504. Contributing authors include Shweta Bhardwaj, Bayan Bruss, Hamed Firooz, Maziar Sanjabi and Soheil Feizi.

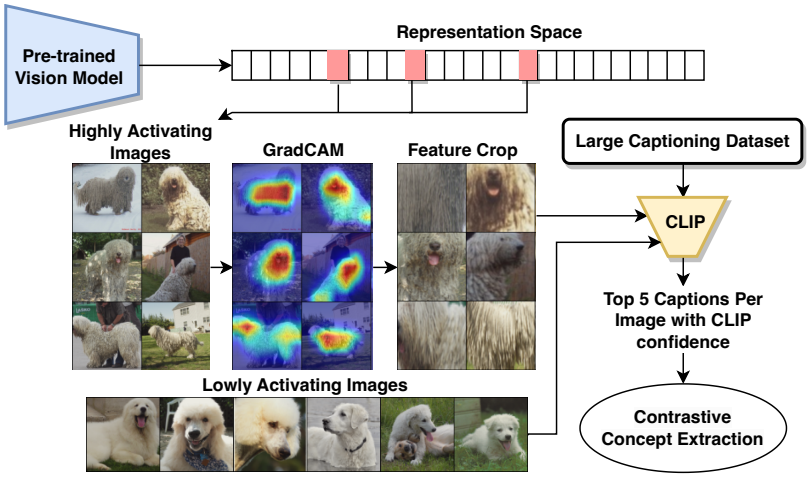


Figure 4.1: **Framework of FALCON:** We outline the process of interpreting any given feature(s) in the representation space of a pre-trained model using a probe dataset $\mathcal{D}$ and a captioning dataset $\mathcal{S}$. Taking the set of highly activating images (from $\mathcal{D}$) for the target features we compute their gradient heatmap [Sel+19] crops, keeping only the highly activating regions. We compute CLIP [OW22] image representations of the cropped images and text representations of a large captioning dataset (in our case, LAION-400m [Sch+21]). For *contrastive interpretation*, we also caption lowly activating (counterfactual) images. Using cosine similarity, we select the top 5 captions per image and pass them through our concept extraction module (Described in Figure 4.3).

Figure 4.2: **Concepts extracted by FALCON for various features in the SimCLR representation space:** We explain various features of the final layer representations of SimCLR [Che+20a] pre-trained on ImageNet [Rus+15b] with a ResNet-18 [He+16] backbone (512 features). For each feature, we show the top activating images as well as the lowly activating images. We crop the top activating images to highlight only the activated regions and extract concepts using the approach outlined in Section 4.2. The lowly activating images are used to filter spurious concepts using our approach called *contrastive interpretation* (See Equation 4.2).

Text-based explanations of model features can also enable scalable analysis of model interpretability. We can automatically identify concept frequency and sensitivity, their contribution in downstream tasks and debug failures modes. We note that such analysis is not easily possible using traditional interpretation methods involving saliency (gradient heatmaps). One way to achieve automatic text-explanation is by using supervision datasets [Bau+17b; Her+22b] with fine-grained conceptual labels for each sample. Such approaches can prove to be expensive, requiring expert annotations. They also may not be generalizable as explanations can be dataset-specific.

In the first part of this paper, we propose **Automatic Feature Explanation using Contrasting Concepts (FALCON)**, a framework to explain neural features, with no densely-labelled dataset or human intervention. We mainly study final-layer self-supervised representations as they contain no label-bias, however, our approach is model-agnostic and can be extended to any deep neural feature. We are also

particularly interested in understanding final-layer representations since they alone are accessible to downstream tasks, and their richness and quality is shown to be essential for better generalization [BBV21; Kal+22; Gar+22]. Nevertheless, our framework is general and can be extended to explain any layer neurons.

FALCON is described in Figures 4.1 and 4.3. For a target feature, we first compute crops of highly activating images from a given dataset (like, ImageNet [Rus+15b]) based on gradient activation. We then caption each cropped image by matching their CLIP [Rad+21] image embeddings to the closest CLIP text embeddings from a large captioning dataset (like, LAION-400m [Sch+21]). We collect illustrative captions for each image with high CLIP cosine similarity, without having to train additional captioning models [Her+22b; Wan+20; Yu+22; WAG22]. The next step in FALCON is described in Figure 4.3, where we show how a compact set of shared, human-understandable *concepts* are extracted from image captions using *Word Score*. We define concepts as the words which closely relate to the attributes that are likely to be encoded by the target feature, based on the set of cropped highly activating images. Unlike prior methods ([OW22]), FALCON is not restricted to output a single concept since features can encode complex physical information which can compose of multiple facets [MA20]. We recognize, however, that top-ranking concepts can relate to spurious attributes which may not be true descriptors for the target feature, although the attributes exist in most of the highly activating images. Current interpretability techniques [OW22; Her+22b; Bau+17b], tend to produce misguided explanations as they do not account for spuriosity and simply report the highest scoring concept. FALCON eliminates spurious concepts by applying a *contrastive interpretation* technique, where we use lowly activating (counterfactual) images for the target feature whose concepts can be discarded. We therefore produce the minimum sufficient set of concepts that best explain the target. We show the results of successfully annotated features of SimCLR [Che+20a] in Figure 4.2.

In the second part of our paper, we study which features in the representation space can be explained. We observe that individual features that are very strongly activating for an adequate number of samples can correspond to easily detectable concepts. However, such features constitute a very small portion of the whole representation space. We observe that most features activate a diverse set of images where the hidden concept is not apparent (See Figure 4.4). We discover that pairs (or groups) of such features are surprisingly more interpretable than individual features. The highly activating images of feature groups are strongly correlated allowing FALCON to produce high scoring concepts. We can therefore explain a much larger portion of the representation space with descriptive and robust concepts.

We evaluate FALCON through human evaluation on Amazon Mechanical Turk (AMT). We show participants images and their FALCON concepts to collect ground truths (relevant or not relevant) for each concept of each annotated feature. The results from our study show a precision of 0.86 and recall of 0.84 for the top-5 concepts, indicating that FALCON concepts are agreeably explanatory (See Section 4.4).

Since the extracted concepts are unique physical attributes for only the portions that a given feature encodes, we can decompose the content of any given image into a set of concepts corresponding to different elements (See Figure 4.5). This helps us understand which physical components of the image have been encoded in its representation. This is also not possible with approaches that conceptualize entire images (like [OW22]). We further utilize concepts to explain failures, like mis-classifications in downstream tasks (See Figure 4.6). By discovering the most contributing concepts in classification, we can detect what the model pays attention to while making its prediction and communicate these in terms of human-understandable concepts. This can help practitioners find and debug issues like hard examples, multi-object scenarios and mis-labelled examples.

Finally, we propose an approach to transfer concepts from an explained represen-

tation space to a new representation space by learning a simple, linear transformation. We train a linear head that maps representations from a target (unseen) model to the source (interpretable) model. This function lets us map any interpretable feature (or group of features) in the source model to the corresponding feature (or group of features) in the target model, and transfer the extracted concepts. We show that the top activating images of the features in the new representation space, exactly match the transferred concepts from the source representation space (See Figure 4.7).

We summarize our contributions below:

- We propose Automatic Feature Explanation using Contrasting Concepts (FALCON), an interpretability framework that automatically detects concepts encoded by any feature of image representations, without any labelled datasets or human intervention.
- We show that representation spaces can be largely explained by interpretable feature groups rather than independent features.
- We show that concepts can be used to explain failures in downstream tasks and can be transferred across representation spaces with a simple linear transformation.

4.2 Automatic Feature Explanation using Contrasting Concepts (FALCON)

4.2.1 Image Captioning Using CLIP

We discuss the general workflow of FALCON to explain features of vision model representations. Let us consider a pre-trained backbone denoted by $\mathbf{f}_\theta(\cdot)$. For a given input image $\mathbf{x}$, this model outputs a representation vector of size r, i.e, $\mathbf{f}_\theta(\mathbf{x}) = \mathbf{h} \in \mathbb{R}^r$.

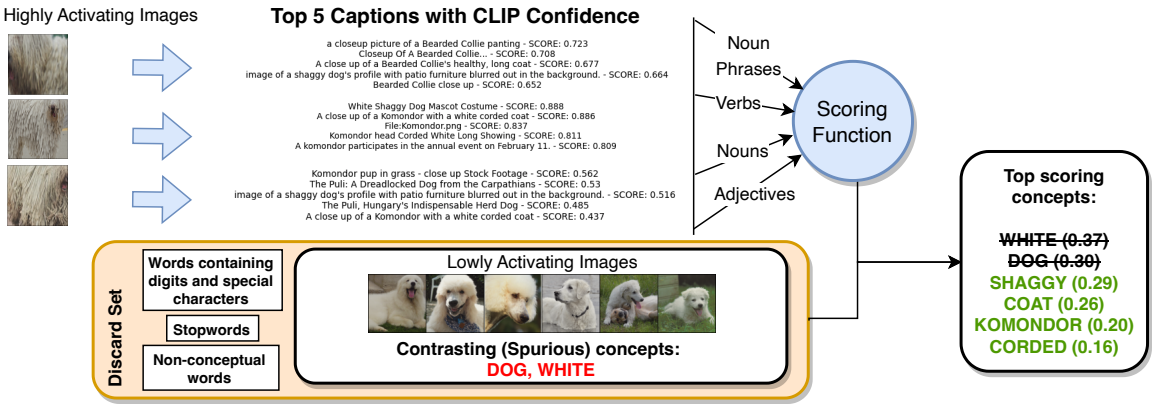


Figure 4.3: **Concept extraction in FALCON using contrasting concepts:** We extract a bag of words (nouns, verbs, adjectives) from the top 5 captions (from LAION-400M [Sch+21]) of every image in the set of highly activating images of a given feature. We use a scoring function (Equation 4.1) to extract top scoring words and phrases which we refer to as *concepts*. We also apply *contrastive interpretation* where we discard any concept that is extracted from the lowly activating images (mined through Equation 4.2). In this case, "dog" and "white" are spurious concepts that exist in both highly and lowly activating images, implying that they are not discriminative explanations. Therefore, final set of discriminative concepts include "shaggy", "coat", "komondor" and "corded" which are all closely related to the given image set.

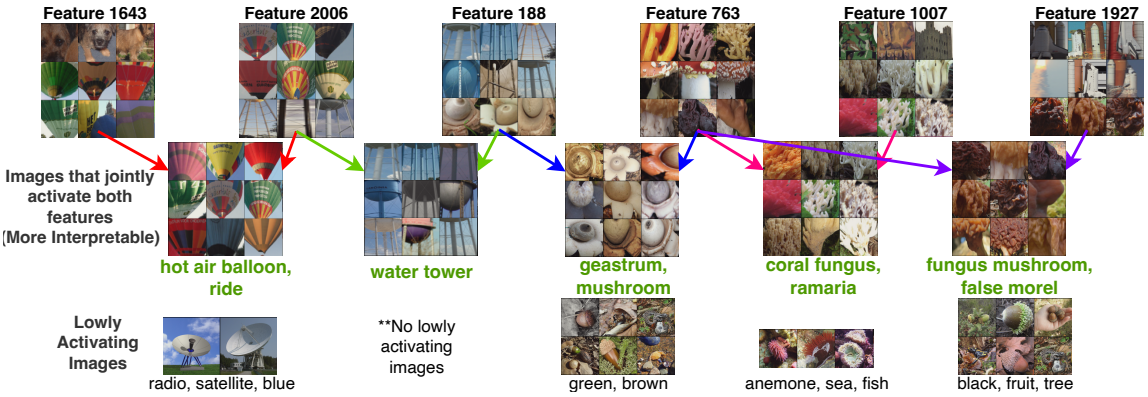


Figure 4.4: **Groups of features can be more interpretable than individual features:** In the first panel, we show the highly activating images of some features of DINO [Car+21] representations trained on ImageNet [Rus+15b] with a ResNet-50 [He+16] backbone. We observe that the images are highly diverse with seemingly no shared concept, like "mushrooms" and "water towers" in feature 188. In the second panel, we observe that images that highly activate pairs of features are significantly more connected. The concepts that our framework extracts are strongly correlated to each group of images. For each feature group, we use the lowly activating images (mined from Equation 4.2) to filter out spurious concepts.

Any downstream task only utilizes these representation vectors, therefore, our objective is to provide human-understandable explanations for these features.

In order to explain features (different indices in $\mathbf{h}$), we utilize two datasets ; 1) A probing dataset consisting of a diverse set of images ($\mathcal{D}$), and 2) A large text dataset to extract concepts ($\mathcal{S}$). In our experiments, we use ImageNet-1K [Rus+15b] validation set for $\mathcal{D}$ and LAION-400m [Sch+21] for $\mathcal{S}$, however, the framework of FALCON is general and can be used with other datasets as well.

Let us consider the task of explaining the $i^{th}(0 \leq i \leq r)$ feature in the representation space of a pre-trained vision model $f_\theta(\cdot)$. From the probing dataset, $\mathcal{D}$ of size N, we first extract the set of highly activating images for feature i defined by, $\mathcal{T}_i = \{j : h_{ji} > \alpha, 1 \leq j \leq N\}$, where α is a threshold we empirically select (more discussed in Section 4.3). As shown in Figure 4.1, for SimCLR [Che+20a] with a ResNet-18 [He+16] backbone, the set of highly activating images for feature 10 are images of a certain breed of dogs. We next compute the gradient of feature i with respect to these images using GradCAM [Sel+19] as shown. We crop the images keeping only the maximally activating portions by thresholding the GradCAM mask. This set of cropped images as well as a large scale text dataset ($\mathcal{S}$) like LAION-400m, serve as the input to a pre-trained vision-language model, i.e., CLIP (ViT-B/32) [Rad+21]. LAION-400m is a large, diverse image captioning dataset which has been used to pre-train vision-language models like CLIP.

We define the CLIP text encoder as $g_{tx}(\cdot)$ and image encoder as $g_{im}(\cdot)$. Given our captioning dataset ($\mathcal{S}$) of size M, we extract the text embedding matrix denoted by $A \in \mathbb{R}^{M \times k}$ where k is the size of the CLIP text embedding space. Since our captioning dataset is fixed for interpreting any feature, we only need to compute its embeddings once. In fact, LAION also provides pre-computed text embeddings on CLIP which saves compute time significantly. We next compute the image embeddings of the cropped highly activating images of feature i denoted by $B \in \mathbb{R}^{|\mathcal{T}_i| \times k}$. Using A and

B, we compute the CLIP confidence matrix, which is essentially the cosine similarity matrix, denoted by $C = BA^T \in \mathbb{R}^{|\mathcal{T}_i| \times M}$. Note that both text and image embeddings are L2-normalized before computing C. Using C, we extract the top 5 captions for each image in $\mathcal{T}_i$.

4.2.2 Contrastive Concept Extraction

The second component of FALCON involves extracting concepts out of the captioned batch of highly activating images for the given feature. In Figure 4.3, we show the top-5 concepts for three highly activating images along with the CLIP confidence. From each caption, we extract the noun phrases, nouns, verbs and adjectives to form a bag of words. Verbs and adjectives are extracted to qualify complex concepts which cannot be described with nouns alone. We remove all stop words and words containing digits or special characters from the bag. We also prepare a discard word set including general, non-conceptual words like “photo", “picture", “background" etc. Given a word w, the word confidence for the p^{th} caption in the q^{th} image is given by, $C^w_{q,p}$ if the word exists in the caption, otherwise 0. We get the maximum value of $C^w_{q,p}$ for each image (q). The *Word Score* is defined as,

$$\textit{Word Score}^w = \frac{1}{|\mathcal{T}_i|} \sum_{q=1}^{|\mathcal{T}_i|} \max_p C^w_{q,p} \tag{4.1}$$

Word Score gives a normalized score for every word among the captions we extract. The best shared concepts describing a given feature i are the highest ranking words, by applying a threshold (in practice, 0.08).

Contrastive Interpretation: In practice, the above method of concept extraction provides a number of high-scoring keywords, shared between the highly activating images. However, in many cases, these keywords can be too general or related to high-level spurious attributes which may be common to all the activating images but

not necessarily relevant to the feature we want to interpret. Many existing techniques [OW22; Bau+17b; Her+22b; MA20], do not account for such cases and they only report a single best scoring concept. In FALCON, we overcome this issue by discovering images in $\mathcal{D}$ that share all other concepts with the highly activating images of feature i, except the actual concepts that feature i encodes. We refer to these images as *lowly activating counterfactual images*. The concepts extracted out of lowly activating images can be regarded as spurious concepts for feature i and added to the discard set.

Let us define the set of feature indices without the index i as $\mathcal{V}_i = \{j : 0 \leq j \leq r, j \neq i\}$. The mean representation of the highly activating images ignoring the i^{th} feature can be written as $\mathbf{h}^{\mu} = mean_{\mathcal{T}_i}(\mathbf{h}_{\mathcal{T}_i,\mathcal{V}_i}) \in \mathbb{R}^{|\mathcal{V}_i|}$. The set of lowly activating images for the target feature i is given by,

$$\mathcal{L}_i = \{j : h_{ji} < \epsilon, \mathbf{h}_{j,\mathcal{V}_i} \cdot \mathbf{h}^{\mu} \geq \beta, 0 \leq j \leq N\} \tag{4.2}$$

where β and ϵ are limits we select empirically. In our experiments, ϵ is the mean value of that feature across the population of normalized representations. Since β is used to threshold the dot product of representations (excluding the target feature), a larger value for β would give us true counterfactuals. We therefore select β to be 0.7. This method of conditional selection gives us lowly activating images that contain all the concepts in the highly activating image set, except the concept represented by the i^{th} feature. We apply FALCON (without feature cropping) to extract concepts out of the lowly activating image set. As shown in Figure 4.3, concepts like “dog" and “white" are in lowly activating images. These keywords can be relevant to the highly activating image batch as well, however, they are not discriminative explanations for that feature. Therefore, we include the concepts of lowly activating images in the discard set and arrive at the final minimum sufficient set of concepts “shaggy", “coat",

“komondor" and “corded".

In Figure 4.2, we show the extracted concepts from FALCON for 8 different features of SimCLR on a ResNet-18 backbone. In cases like Feature 337, the lowly activating images match almost all the object properties i.e, vehicle or van. However, after extracting concepts, it becomes clear that the feature concept is the side view of an emergency vehicle which is explained by - “ambulance" and “side". Contrastive interpretation therefore lets us ignore generic and spurious concepts to derive a compact set of discriminative explanations. We also compare FALCON with MILAN [Her+22b], a recent approach that trains a generative model on a human-annotated fine-grained image region-caption dataset, and uses this model to generate natural language explanations. We observe that FALCON produces more feature-specific concepts compared to the generic high-level explanations of MILAN. We show more annotated features (including supervised and previous-layer features) comparing with MILAN in the Appendix (See Figures B.7, B.8). We also discuss the generalizability of concepts to an unseen dataset like STL-10 [CLN] (See Figure B.7).

4.3 Which Features are Explainable?

So far we discussed our method to explain individual features, given the representation space of a pre-trained model. In this section, we understand which features in the representation space can be considered as *explainable*. Let us go back to the set of highly activating images for a given feature i, defined by $\mathcal{T}_i = \{j : h_{ji} > \alpha, 1 \leq j \leq N\}$. Note that the representations are all L2-normalized. In order to extract meaningful and generalizable shared concepts with high Word Scores, we require a sufficient number of highly activating images. In our experiments, we select the features where $|\mathcal{T}_i| > 10$. If α is large enough, we may expect the set of highly activating images to be more connected where the feature concept is clearly detectable (See features in

Figure 4.2).

We choose features with a strong value for α according to the distribution of the representation space of the selected model. The features where $|\mathcal{T}_i| > 10$, only comprises of roughly 20% of the representation space. See Table B.3 for this percentage for various pre-trained models. Upon empirical inspection of the activated images, we find that thresholding α alone, may not guarantee explainability. Some of the features can correspond to human recognizable concepts (activating correlated images), like the examples shown in Figure 4.2. While other features, although strongly activated for a sufficient number of samples, correspond to very high level, abstract concepts that are not apparent to humans. We show examples of such features in the top panel of Figure 4.4, on DINO [Car+21] with a ResNet-50 backbone. Although these features are activating with high α, the images are quite diverse, making it almost impossible to decipher any shared properties. One possible way to understand such features could be by explaining previous layer neurons in the network which may perhaps encode higher level properties [OW22; MA20; Her+22b; Bau+17b]. This is however computationally inefficient as previous layer features may still activate dissimilar image sets or may correspond to entirely different concepts.

In the second panel of Figure 4.4, we make a key observation; images that jointly activate a given pair of features are significantly more related and explainable than those of individual features. For example, visually, we cannot identify any shared property between rockets and morel mushrooms in feature 1927 and similarly, fly argaric mushrooms and underwater coral plants in feature 763. However, when both feature 763 and 1927 are highly activated, the shared concepts become more apparent, showing only morel mushroom textures. When the same feature 763 is jointly activating with another feature like 1007, it corresponds to a totally different concept of coral reef patterns. A similar observation has been made in [Elh+22; FV18]. Note that the threshold for α is the same for both individual and groups of features

(for fair comparison in Figure 4.4), however, less rigorous α can still be used for groups of features. By observing highly activating images for a combination of features, we can explain a larger portion of the representation space (even by relaxing α) compared to independent features.

Automatically discovering all interpretable feature groups: Given a model $f_\theta(.)$ and a probe dataset $\mathcal{D}$ of N samples, we compute the top activating set of features (group) for every sample (using α as the threshold). We save each feature group and the indices of the samples that highly activate that group. We use the average CLIP cosine similarity of the samples within each group to decide if a group is interpretable or not (using a threshold, γ). A higher value for average similarity implies that the top activating samples are *similar* with interpretable shared concepts. Other metrics LPIPS [Zha+18] can also be used. In Algorithm 1, we provide PyTorch-like code highlighting the steps required for identifying all possible interpretable feature groups in the representation space of a given model.

FALCON can be used to extract concepts out of groups of features in the same manner as individual features, with some modifications. First, the feature crop is calculated by taking the intersection of the gradient heat map of each feature individually as shown in Figure 4.4. Second, the lowly activating images are mined such that all the features in the group show low activation and the remaining features are close to that of the highly activating representations. That is, Equation 4.2 is updated to compute $\mathcal{L}_\mathcal{I}$ where $\mathcal{I}$ represents the feature group. As shown in Figure 4.4, FALCON uses the lowly activating images to help in finding discriminative concepts for groups of features that best explain the highly activating images.

In Appendix Section B.1, we analyze the extracted concepts across various models (supervised and self-supervised) and discuss some key insights.

Algorithm 1: Pytorch-like pseudocode for discovering interpretable feature groups in a given representation space

Input: **H** is the set of representations (of the given model $f_\theta(.)$) of N samples in the probing dataset $\mathcal{D}$,
α is a threshold for feature activation,
γ is a threshold for interpretable feature groups.

```
# Identify all feature groups
groups = {}
for j in range(N):
    group = torch.where(h[j] > alpha)
    groups[group].append(j) # groups[group] is a list
# Filter out interpretable groups
int_groups = {}
for group in groups:
    if len(groups[group]) > 10:
        # top activating samples for group
        top_act_idx = groups[group]
        clip_feat = get_clip_feat(top_act_idx)
        avg_cos = torch.matmul(clip_feat, clip_feat.T).mean()
        if avg_cos > gamma:
            int_groups[group] = groups[group]
return int_groups
```

4.4 Evaluating Extracted Concepts

FALCON produces a simple, compact set of concepts to describe any explainable feature in an automatic fashion without any human intervention, or densely-labelled datasets. We performed a human study on Amazon Mechanical Turk (AMT) to evaluate the concepts generated by FALCON and provide some quantitative metrics. In each task, we showed the AMT participant the set of highly activating cropped image set (Group A) and the lowly activating image set (Group B) for a target feature and, the top 6 concepts ranked by FALCON. We asked the participant to - identify the concepts that are related to Group A and not Group B. This lets us assign binary ground-truth labels to each concept as 0 (not related) if it has been chosen by at least 65% of the participants and 1 (related) otherwise. We can partition the six FALCON concepts for each feature based on their rank such that the first K concepts where $1 \leq K < 6$ can be predicted as 1 (related), otherwise 0 (not related). In Table 4.1, we plot the Precision and Recall for each K. Precision in our case measures how many of the “related" concepts predicted by FALCON are actually related according to our human study. Recall measures how many “related" concepts

was FALCON able to predict among the total number of related concepts (from our human study). We observe that the Recall improves from the 4^{th} caption, meaning that, the participants agree that the first 4-5 concepts are related to the given set of images. 84.23% (precision at top-6) of all FALCON concepts are considered relevant by our participants. This study confirms that the top ranking concepts generated by FALCON are considered relevant and explainable among humans. We collect ground truths for 600 concepts each from 3 participants. We measure the agreement between participants for each feature by averaging the % overlap of the concepts selected by each participant. The average agreement among the participants is 79%. More details about our human study can be found in Appendix Section B.4.

Existing methods (MILAN [Her+22b], NetDissect [Bau+17b]) use human-annotated datasets for natural language descriptions. Through FALCON, we automatically extract a minimal sufficient set of noise-free concepts with no human intervention. We performed another user-study on MTurk to provide a quantitative comparison of FALCON with MILAN and Net-Dissect. We display the highly (Group A) and lowly (Group B) activating images for each target feature and requested the participants to select the concept set which best describes the images in Group A but not Group B. We tabulate the percentage of times the concept set of each framework was selected as the best explanation in Table 4.2. FALCON performs significantly better than the baselines in our study of 115 features.

4.5 Explaining Failure Modes in Vision Models

An interpretable representation space of a given model, allows us to decompose and label different groups of concepts in any given image. In the previous sections, we found that each interpretable feature (or group of features) encodes only a portion of images that correspond to a unique concept set. Therefore, images with multiple

Table 4.1: **Precision and Recall for human evaluation of top K concepts:** Using Amazon Mechanical Turk (AMT), we ask human participants to choose the un-related captions, among 6 top-ranking captions for each feature. We use the annotations as ground truth labels (relevant or not relevant) and compare them to the predictions of FALCON at different levels of K (number of predicted concepts labelled as relevant).

Top K Concepts	Precision (%)	Recall (%)
1	94.62	18.72
2	92.47	36.60
3	88.89	52.77
4	86.82	68.72
5	85.60	84.68
6	84.23	100.00

Table 4.2: **Comparing explanations generated by FALCON with existing baselines:** We request participants to select the best explanation generated by the following 3 frameworks for a given set of highly and lowly activating images. FALCON beats other baselines by a significant amount.

Framework	% of times selected as best explanation
FALCON	**86.40**
MILAN [Her+22b]	13.47
Net-Dissect [Bau+17b]	0.12

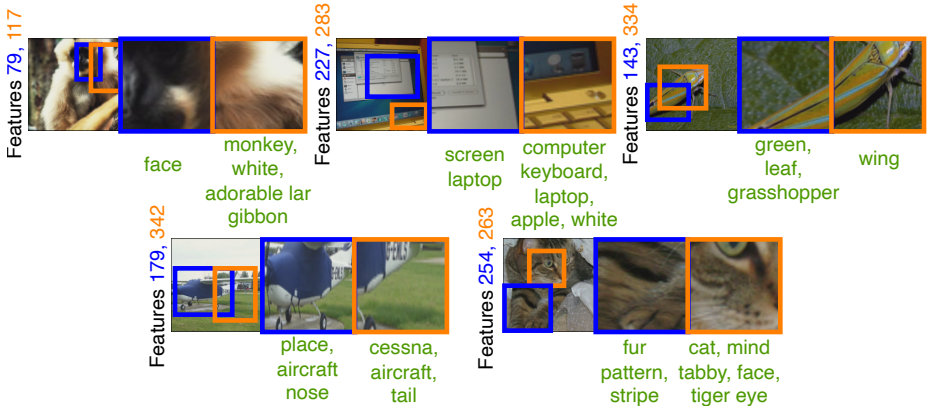


Figure 4.5: **Decomposing images into various concepts:** We show some images which highly activate multiple interpretable features. FALCON extracts concepts from feature crops rather than entire images, therefore, each image can be broken down into components, each describing a different physical attribute.

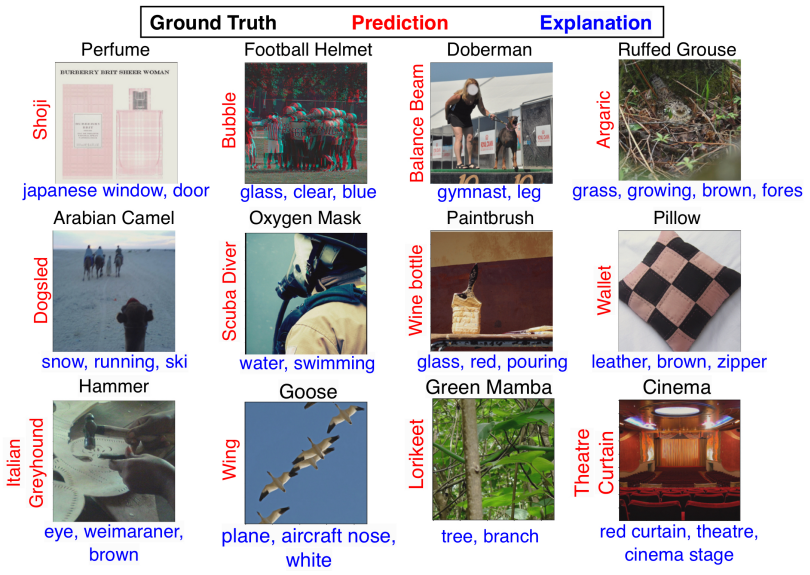


Figure 4.6: **Explaining failures in downstream tasks using concepts:** Given SimCLR [Che+20a] pre-trained on ImageNet [Rus+15b], we show some mis-classified examples along with the most contributing concepts for their prediction. This allows us to detect and explain concepts which contributed to a model's decision and help us debug model failures.

highly activating features [Kal+22], can be decomposed into multiple components, each representing a unique concept. We illustrate this in Figure 4.5, where we show the feature crop of highly activating features (of SimCLR with ResNet-18) in each image, and their corresponding physical concepts that our framework has extracted. This is only possible because FALCON uses feature crops to discover concepts rather than whole images (unlike CLIP-Dissect [OW22]).

Another advantage of feature specific concepts is the ability to explain failures in downstream tasks. It is often not obvious what led to a model's prediction without some qualifying explanations. Images in the real-world could contain several spurious attributes interfering with the main content of the image. In such cases, it can be difficult for even experts to localize the exact reason for mis-classification. Moreover, it is often too tedious to have humans make guesses as to what could be the reason for failures as each human can interpret images in a unique manner. With our automatic explanation framework, FALCON, we eliminate this need for human-in-the-loop and

can inspect grounded explanations directly.

We consider the task of classification using a linear head defined by the weight matrix $\mathbf{U} \in \mathbb{R}^{o \times r}$, where o is the number of classes. The most contributing features (and corresponding concepts) for a sample $\mathbf{x}_j$ with prediction y_j, can be given by, $\arg\max(\mathbf{h}_j \odot \mathbf{U}_{y_j})$. In Figure 4.6, we show some mis-classified examples of SimCLR trained on ImageNet and the most contributing concepts for each prediction. The concepts we find add novel insight into model behavior apart from the readily available information i.e., the image, label and prediction. They help describe the attributes to which model paid attention while making its prediction, potentially helping us automatically debug models at inference time.

For example, the eighth "Goose" image, looked more like an aircraft to the model, leading to the prediction "Wing". This is an example where the model may be spuriously associating shape (like an aircraft) and background information (like the sky) in making its prediction, failing to identify the subtle features of geese. The texture in the "Perfume" and Football Helmet" images is also an example of spurious attributes. The sixth "Green Mamba" image can be regarded as a *hard example*, where the core object is largely hidden, causing the model to focus more on concepts like tree and branch. Explanations can also help uncover images which may have multiple ground truths like the eleventh example of "Cinema" and "Theatre Curtain" (similar to the images in Figure 4.5). The "Pillow" and "Hammer" images indicate that the training paradigm of the model ignored global object information and made decisions based on local attributes. One possible approach to improve such models relying on spurious correlations is by fine-tuning on synthetic images generated using the relevant FALCON concepts via methods like Stable Diffusion [Rom+21]. Explanations can also help define optimal training augmentations that could prevent spurious dependencies.

4.6 Transferring Concepts to New Representation Spaces

So far, we have discussed the process of feature captioning and concept extraction for a given vision model. We hypothesize that the representations learned by different models can be mapped from one to another. This would allow us to map the features of an explainable representation space to any unseen representation space, without having to re-run our explanation framework. Let us consider the representations of a model that we have extracted concepts for, denoted by $\mathbf{H}_{source} \in \mathbb{R}^{N \times r}$. The representation space of an unseen model can be denoted by $\mathbf{H}_{target} \in \mathbb{R}^{N \times r}$. Using a linear head, our goal is to learn a transformation matrix $\mathbf{Z} \in \mathbb{R}^{r \times r}$, that transforms $\mathbf{H}_{target}$ to $\mathbf{H}_{source}$, by solving the optimization,

$$\min_{\mathbf{Z}} \|\mathbf{Z}^T \mathbf{H}_{target} - \mathbf{H}_{source}\|_2 \tag{4.3}$$

We solve optimization by training a linear head for only 10 epochs with a learning rate of 1, using an SGD optimizer. Once the mapping is learned, we can take any explainable feature i in $\mathbf{H}_{source}$, and find the features in $\mathbf{H}_{target}$ with have the highest weights in $\mathbf{Z}$. Hence, the concepts described by feature i in $\mathbf{H}_{source}$ can be mapped to features in $\mathbf{H}_{target}$ efficiently.

We confirm that this transformation works by matching the concepts of $\mathbf{H}_{target}$ to the highly activating images of $\mathbf{H}_{target}$. As shown in Figure 4.7, we successfully map individual interpretable features in SimCLR to features in MoCo [Che+20b] which is an unseen representation space. The highly activating images in MoCo interestingly contain all of the concepts of the source feature. Note that, features across representation spaces need not have a 1:1 relationship. Similarly, concepts can also correspond to compositional features (as described in Section 4.3). We do not

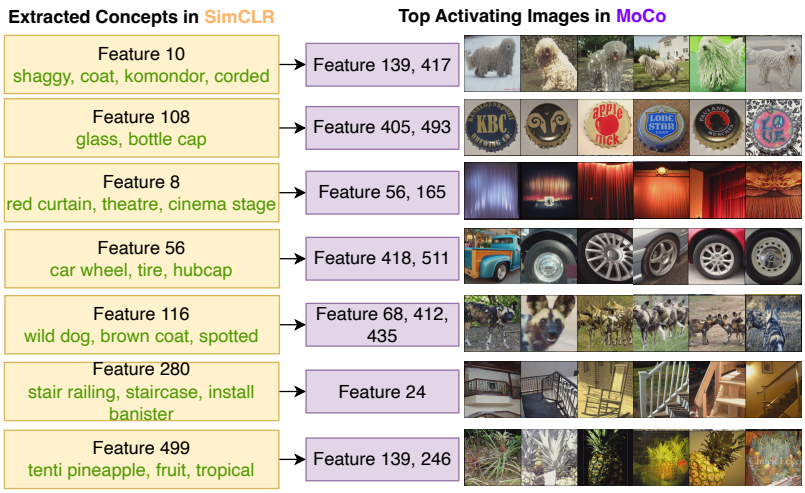


Figure 4.7: **Transferring concepts from an explained representation space to an unseen representation space:** We show that representations of self-supervised models can be mapped from one to another by learning a transformation $\mathbf{Z}$ (See Equation 4.3). We transfer extracted concepts from SimCLR (source) [Che+20a], to an unseen model, MoCo (target) [Che+20b] by mapping the source features to the target features with the highest weights in $\mathbf{Z}$. We observe that the top activating images of the mapped features in the MoCo very closely match the concepts extracted in SimCLR.

constrain the sparsity of $\mathbf{Z}$. In practice, $\mathbf{Z}$ is not sparse however, it can be considered as *nearly sparse* where most weights are close to zero. When we discover feature maps, we only extract the weights in $\mathbf{Z}$ if they are large enough ($> \text{mean} + 4 \times \text{std}$ based on the weight distribution). If we do so, Z becomes quite sparse, indicating that some directions in the target model can be mapped to a dedicated set of features in the source model.

This observation of transferrability can potentially be extended to any pair of pre-trained models (supervised or unsupervised), preventing the need to interpret the representations of each specific model. It also gives us an important insight that various vision models, regardless of their pre-training regime, learn mostly similar concepts. To the best of our knowledge, ours is the first approach in the direction of transferring explanations across model spaces.

4.7 Conclusion

We proposed FALCON, an automatic framework to explain individual neurons in vision models. These explanations can be utilized for classification tasks (as shown in Figure 4.6) as well as non-classification tasks like object detection and segmentation. We show that features become more interpretable when regarded in groups and propose a simple algorithm to discover all possible interpretable groups in a given representation space.

FALCON utilizes three components: 1) A probe image dataset, 2) A large text vocabulary and 3) An off-the-shelf pre-trained vision-language encoder. With FALCON we propose a general-purpose framework, where the above components can be flexibly customized depending on the target model we wish to investigate. The concepts learned by the target model is governed by the data it was trained on. In order to explain these concepts via FALCON, we choose the probe image dataset and text vocabulary such that it is representative of the target model's training domain and encapsulates all the concepts learned by the target model. In our experiments, we use FALCON with ImageNet, LAION-400M and CLIP to explain deep models pre-trained on ImageNet-1K. These components could potentially generalize to a range of domains, since CLIP is already pre-trained on a very large scale and LAION-400M is diverse and expressive. FALCON can be updated to use even larger zero-shot vision-language encoders and vocabulary, when developed in future. To deploy FALCON on target models trained on medical images like chest x-rays, we can utilize vision-language encoders like ConVIRT [Zha+20] or MedCLIP [Wan+22], combined with expressive vocabulary from radiology reports (ex. Mimic-cxr).

Limitations and directions for future work: Understanding how FALCON can be applied to explain vision-language models can be an extension of our work. Since vision-language models are trained to align representations in the vision and language space, we could potentially learn a great deal about the model's under-

standing by applying FALCON directly on the vision encoder. It would however be interesting to understand what information is represented uniquely by the text encoder. Understanding the equivalent of localized gradient heatmaps in the language space is still unclear and requires further research.

Supporting FALCON for non-image domains remains a topic for further research. Another limitation of FALCON is the requirement of a pre-trained vision-language model for the task of matching images to captions. While CLIP is trained on very diverse data and domain-specific versions of CLIP exist, there may be target models which are trained for uncommon tasks and data, that is unknown to CLIP. In our transferrability example, we show that concepts extracted from one model may be transferred to another by learning a simple linear transformation. Another important direction for future work is to test the limits of transferrability on multi-domain setups. For example, how do concepts learned by a model trained on painting images, transfer to a model trained on sketch images.

Acknowledgments: We would like to thank Mazda Moayeri, Hemant Kumar, Samyadeep Basu, Sharath Gokarn, Saksham Suri, Pavan Gurudath, Nirat Saini, Pulkit Kumar and Archana Swaminathan for their help with testing our human studies.

Chapter 5

Disentangling the Effects of Data Augmentation and Format Transform in Self-Supervised Learning of Image Representations[1]

5.1 Introduction

In the fast-evolving landscape of deep learning and computer vision, self-supervised learning has emerged as a powerful paradigm for foundation models [Rad+21; Oqu+23]. Its success is rooted in its ability to learn robust and generalizable representations from unlabelled data with no supervision. Existing SSL approaches have been categorized into two main types: generative [He+21] and invariance-based [Che+20a; CH21; Car+20a; Oqu+23; Gri+20; Zbo+21b]. The latter involves joint-embedding pre-training with two or more *views* of the same input data sample. To prevent joint-embedding representations from collapsing (converge to identical representations)

[1]Full paper available at https://arxiv.org/abs/2312.02205. Contributing authors include Warren Morningstar, Alex Bijamov, Luyang Liu, Karan Singhal and Philip Mansfield.

during pre-training, it is crucial to employ stochastic augmentations like random crop, color jitter, Gaussian blur, solarization etc. These augmentations are often hand-crafted for specific downstream tasks and may not transfer well to other tasks [Tia+20a; EGH21b]. We show evidence (Figure 5.2) that progressively adding more hand-crafted augmentations improves downstream linear probing performance and conversely, removing any given augmentation always hurts performance among 3 self-supervised baselines. We therefore hypothesize that increasing augmentation diversity during pre-training allows representations to become invariant to more nuisance concepts and could improve downstream linear probing performance.

Meanwhile, in the audio and speech domain, recent works [WO21; Sae+21; Zha+22a] have successfully performed self-supervised learning by maximizing the mutual information between time and frequency formats in the latent space using the Fourier Transform and a small number of format-specific augmentations. This mode of transformation allows us to represent the same data under different coordinates. This is unlike hand-crafted augmentations, since the data remains unperturbed. Prior works [Xu+21; Cai+21; YS20; Kot+22] have utilized the Fourier space to unify multi-domain latent spaces to benefit tasks like domain generalization and image-to-image translation. We use the term *format transform* and *Fourier transform* interchangeably in the context of images.

In this paper, we integrate both notions presented above. We study the effect of incorporating augmentations in the Fourier domain of images with the goal of increasing overall augmentation diversity. To this end, we propose a pipeline of augmentations called **Fourier Domain Augmentations (FDA)** that can be applied in the complex Fourier domain. When data after these FDAs are inverted back to the image space, we observe that they produce unique textures and patterns, which cannot be easily reproduced by directly perturbing the image space.

We study the combined effect of applying FDA along with standard image aug-

mentations on pre-training state-of-the-art self-supervised baselines including SimCLR [Che+20a], BYOL [Gri+20], MoCov2 [Che+20b] and SimSiam [CH21] on ImageNet-1K [Rus+15b]. We show an average improvement of 1% in the top-1 accuracy during downstream linear probing. We also evaluate other downstream tasks including few-shot learning and transfer learning and show qualitative improvements on image retrieval with the use of FDAs.

Our results confirm our initial hypothesis of the need for augmentation diversity. We perform ablations where we study the independent effects of augmentations in the image space and the frequency space in a single-encoder contrastive learning setup (SimCLR). We explore the results of maximizing agreement between two augmented views where the augmentation can be any one of (i) standard image augmentations (ii) Fourier-mode augmentations and (iii) the combination of both.

Finally, we examine the individual effect of using the format transform itself disentangled from any augmentations. This experiment is to understand if self-supervised learning can benefit from encoding images presented in multiple formats without any augmentations i.e., the raw image and Fourier transform. To achieve this we design a dual-encoder setup with contrastive learning where each encoder is exposed to one modality, either raw image or Fourier image. We observe that providing the Fourier transform as one of the views during pre-training improves linear probing performance by 16% compared to raw image pre-training in lieu of any augmentations. We further explore the benefit of augmentations (both image and frequency) in this dual-encoder setup. Across all ablations, we observe that combining image and FDA while pre-training in the image domain results in the best downstream performance.

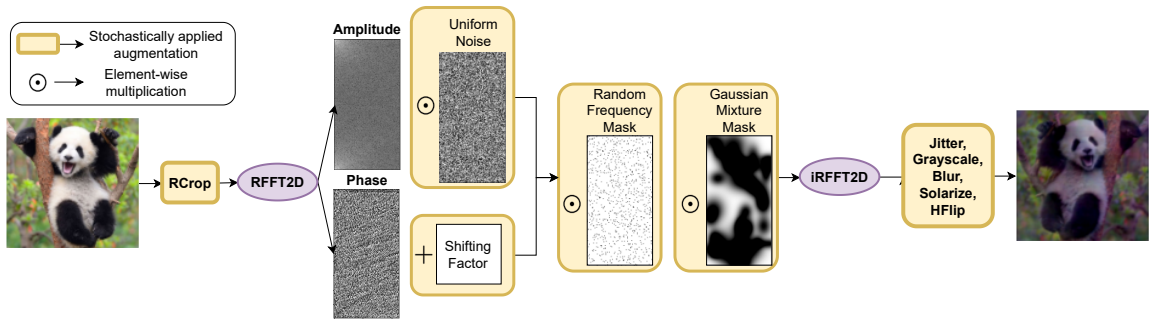


Figure 5.1: **Diversifying image augmentations with Fourier Domain Augmentations (FDA):** We show the pipeline of applying Fourier Mode Augmentations integrated with standard image augmentations like random cropping, color jitter, grayscale etc. We use RFFT2D (available in PyTorch and TensorFlow) to transform a random resized crop image into the Fourier space. Here, we stochastically apply *amplitude rescale*, *phase-shift*, *random frequency mask* and *Gaussian mixture mask* which together constitute Fourier Domain Augmentations (FDA). The remaining image augmentations are applied after inverting the augmented Fourier spectrum back to the image space using iRFFT2D.

5.2 Background

Self-Supervised Learning is a powerful approach of learning representations from large amounts of data without the use of labels. Learned representations can later be used for downstream tasks [Bal+23] directly or with inexpensive fine-tuning. Representations are learned by solving *pretext tasks* which can involve predicting simple transformations on a given image like rotations [GSK18], jigsaw [NF16] or color [ZIE16]. However, more successful self-supervised approaches involve joint-embedding methods which force latent space similarity between multiple augmented views of the same image sample. This can be achieved via contrastive or InfoNCE loss [OLV18; Che+20a; Che+20b; CH21], self-distillation [Oqu+23; Gri+20] or by redundancy reduction in the latent space [Zbo+21b; Car+20a]. Regardless of the training paradigm, all joint-embedding methods rely on powerful data augmentations to control the degree of invariance beneficial for downstream tasks.

Augmentations in Self-Supervised Learning engender invariances which in turn introduce good inductive biases for downstream tasks [EGH21b; PG20]. However,

for any given downstream task, specific augmentations may be better suited over others [EGH21b; Tia+20a]. This property tends to restrict the generalization capability of many self-supervised models as using an inappropriate set of augmentations can significantly hurt downstream performance. Therefore, a standard protocol followed by most self-supervised approaches is to identify optimal augmentations for best downstream linear probing performance on ImageNet-1K.

Fourier-based Methods in Audio: Self-Supervised learning has shown success in the audio/speech domain [Sch+19; Bae+20] in predicting embeddings of future audio samples from a sequence of prior embeddings, by comparing with a context embedding derived from the sequence. Recently, Wang et. al [WO21] have extended these results by directly comparing two augmented versions of a given audio sample rather than utilizing a context embedding. In their work one version of the audio sample is in the time-domain format, with augmentations directly applied to the waveform, while the other has been Fourier-transformed into the frequency-domain, with augmentations applied to the spectrogram. Encoders for the two formats are simultaneously trained so that their output embedding vectors align when they arise from the same data source. Specifically, time-domain augmentations involve masking (removing) some time intervals and adding noise. Frequency-domain augmentations involve masking (removing) some frequency intervals and shifting all frequencies by an integer constant. [Sae+21] also did contrastive learning on representations of two different signal formats; namely a waveform (not necessarily audio), and a scaleogram arising from a wavelet transform. However, no data augmentations were explored in that work. [Zha+22a] train a joint time-frequency representation, where self-supervision is implemented by penalizing the distance between a signal's time and frequency representations, each pretrained contrastively.

The contrasting of multiple formats (raw and frequency) of the same input is especially interesting even in the image space, as it potentially allows generating rich

embeddings that encode both modalities. To the best of our knowledge, no analysis has been done of the separate and combined effects of Fourier space augmentations and image augmentations. Moreover, neither augmentations in the Fourier space nor the direct use of Fourier space in self-supervision have been properly explored on image data for vision models.

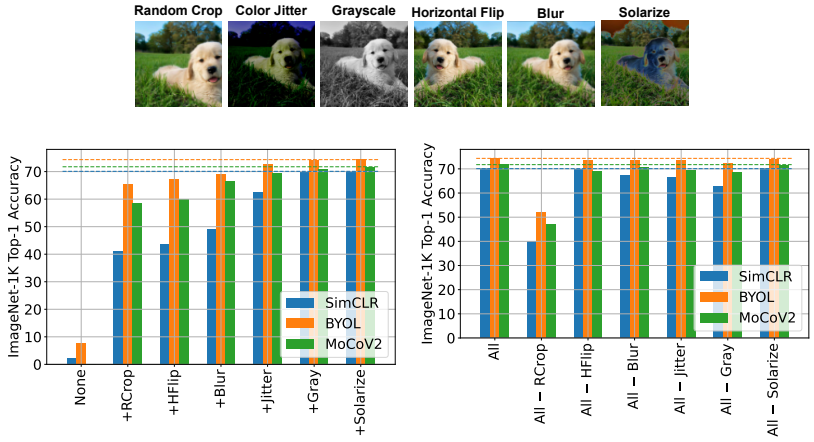


Figure 5.2: **Augmentation Diversity:** We display commonly used hand-crafted augmentations for self-supervised learning in the top panel. We demonstrate the effect of increasing diversity in pre-training augmentations (bottom, left plot) and removing individual augmentations (bottom, right plot). The best performance is retained when all given augmentations are used in 3 baselines.

5.3 Importance of Diversity in Pre-Training Augmentations

In this section, we illustrate the strong dependence that joint-embedding self-supervised models have on pre-training augmentations. We hypothesize that each augmentation tackles a specific type of *invariance*. Depending on the downstream task, a model's generalization power can be improved by enforcing invariance to physical properties irrelevant to the ground truth [EGH21b; PG20; Tia+20a]. The standard set of augmentations used by self-supervised models are - random cropping and resizing,

horizontal flip, color jittering, grayscale, Gaussian blurring and solarization. We display an example of these augmentations in Figure 5.2 (top panel). These augmentations have been hand-crafted to show competitive performance in downstream classification, particularly on ImageNet-1K.

In Figure 5.2 (bottom left plot), we show the effect of progressively adding individual augmentations while pre-training SimCLR, BYOL and MoCov2 on ImageNet-1K and measuring the linear probing accuracy. Each baseline demonstrates the best performance when all of the above augmentations are used. This result supports our claim of diversity playing an important role in producing easily classifiable representations.

While the diversity of augmentations is necessary, it is also important that each augmentation attacks specific invariances. In Figure 5.2 (bottom, right plot), we show the impact of removing individual augmentations while maintaining the rest. Each model shows a drop in performance when any of the augmentations are removed. Among these, removing random cropping shows the strongest reduction in performance (followed by grayscale) compared to the baselines which retain all augmentations.

It is important to note that regardless of the pre-training paradigm, self-supervised models only demonstrate state-of-the-art performance when all of the above augmentations are used. As more augmentations are incorporated during pre-training, the downstream performance steadily improves. This begs the question - can we additionally incorporate new augmentations to further improve linear classification performance? While most of the proposed augmentation strategies [Car+20a; NF16; Prz+23; Lee+21] perturb the image directly, we shift the focus to leverage the format transform of images to incorporate new information and invariances. We first explore these benefits by augmenting the Fourier spectrum and returning to the image space via an inverse transform. We then explore utilizing the Fourier spectrum directly in joint-embedding pre-training to study its independent effect.

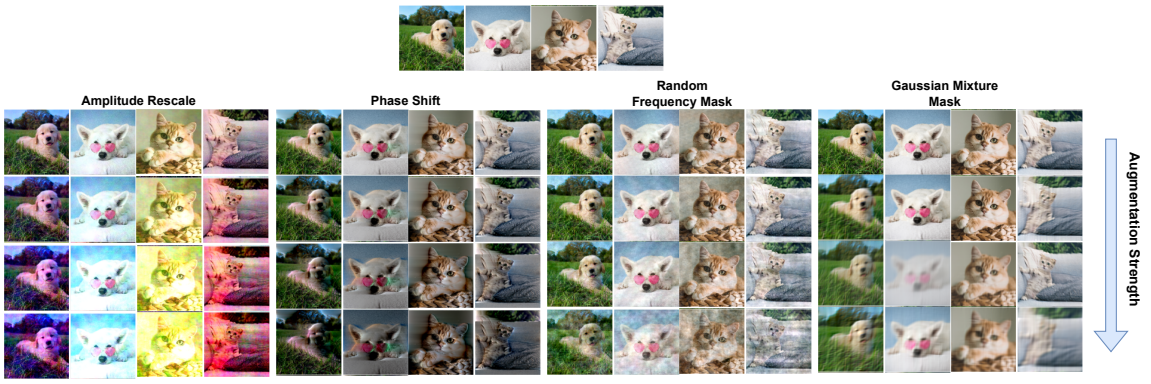


Figure 5.3: **Fourier Domain Augmentations (FDA):** We illustrate the result of applying each augmentation in FDA when inverted back into the image domain - amplitude rescale, phase shift, random frequency mask, Gaussian mixture mask. We vary the strength using augmentation-specific hyper-parameters m, n, p, q, k, o (see Section 5.4). We tune these hyper-parameters (no training required) such that images are perturbed sufficiently without hiding the core ground-truth attributes.

5.4 Fourier Domain Augmentations (FDA)

The Discrete Fourier Transform of a single-channel 2-dimensional image $\mathbf{x} \in \mathbb{R}^{H \times W}$ is given by,

$$\mathcal{F}(\mathbf{x})_{u,v} = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} e^{-2\pi i \left(\frac{h}{H}u + \frac{w}{W}v\right)} x_{h,w} \tag{5.1}$$

where, $u = \{0...H-1\}$ and $v = \{0...W-1\}$. The Fourier transform can be applied over every image channel (RGB). Both $\mathcal{F}$ and $\mathcal{F}^{-1}$ can be computed efficiently using the Fast Fourier Transform algorithm [BM67]. Since the FFT of a real signal is Hermitian-symmetric, we use the RFFT2D operation (provided by PyTorch and TensorFlow), which provides only the positive frequency terms to avoid redundancy.

Let $\mathbf{f}$ denote the complex-valued Fourier spectrum of the image $\mathbf{x}$ (Equation 5.1). The real and imaginary components of $\mathbf{f}$ are denoted by $\mathcal{R}(\mathbf{f}) = \mathcal{A}(\mathbf{f}) \cos \mathcal{P}(\mathbf{f})$ and $\mathcal{I}(\mathbf{f}) = \mathcal{A}(\mathbf{f}) \sin \mathcal{P}(\mathbf{f})$ respectively where, $\mathcal{A}(\mathbf{f})$ is the amplitude and $\mathcal{P}(\mathbf{f})$ is the phase of the spectrum. Conversely, $\mathcal{A}(\mathbf{f}) = \sqrt{\mathcal{R}^2(\mathbf{f}) + \mathcal{I}^2(\mathbf{f})}$, and $\mathcal{P}(\mathbf{f}) = atan2\left(\mathcal{I}(\mathbf{f}), \mathcal{R}(\mathbf{f})\right)$.

The Fourier spectrum provides a number of unique insights into the image signal.

A well-known and often exploited property [HH07; OL81; Opp+79; PC82] is that the amplitude represents low-level statistics and superficial patterns in the image while the phase preserves structural and semantic information. Traditional image processing techniques [HJB84] involved using a circular kernel mask on the Fourier spectrum to turn off high-frequency modes (low-pass filter) to create a *blurring* effect, after inverting back to the image space ($\mathcal{F}^{-1}$). On the other hand, turning off low-frequency modes (high-pass filter) creates a *sharpening* effect. Inverting the Fourier spectrum back to the image space lets us apply our method as new augmentations in addition to standard image augmentations and does not require us to re-define the self-supervised training pipeline. In Section 5.6, we study disentangle the effect of format transform and augmentations with the use of a designated image encoder and frequency encoder where we directly encode Fourier input ($\mathbf{f}$) into representations.

We propose the following general-purpose format transformations that perturb different properties in the Fourier spectrum, producing unique augmentations when inverted back to the image space.

- **Amplitude Re-scale:** We prepare a uniform noise vector $\mathbf{p} \in \mathbb{R}^{H \times W}$ within a range $[m, n)$ where, $m, n > 0$ (selected empirically). We multiply this noise with the amplitude of the spectrum,

$$\mathcal{A}(\mathbf{f}) = \mathcal{A}(\mathbf{f}) \odot \mathbf{p}$$

  A randomly sampled noise is applied to each channel of the FFT of the 3-channel image. When this augmentation is inverted to the image domain ($\mathcal{F}^{-1}$), it results in non-uniform perturbations to the image color scope.

- **Phase Shift:** We randomly sample a constant *shifting factor* $\theta \in \mathbb{R}$ within the range $[p, q)$ where, $p, q > 0$ (selected empirically). The phase is shifted as

follows,

$$\mathcal{P}(\mathbf{f}) = \mathcal{P}(\mathbf{f}) \pm \theta$$

This transform brings about a *movement effect* in the image wherein certain high-frequency attributes are brightened.

- **Random Frequency Mask:** We define a binary mask $\mathbf{h}$, commonly across all channels where $k\%$ of frequencies are set to 0. We also ensure that the zero frequency mode ($h_{0,0}$) is always enabled so that semantic information is largely retained.

$$\mathbf{f} = \mathbf{f} \odot \mathbf{h}$$

This transform randomly turns off both high and low frequency modes across all channels. This preserves the color scope but results in a unique *cloudy* texture non-uniformly applied across the image.

- **Gaussian Mixture Mask:** Unlike, low-pass and high-pass filters which apply a single circular kernel at the center of the spectrum, we propose a more general form of frequency-band masking. We prepare a Gaussian Mixture Mask with a randomly sampled set of origins, $\mathbf{c} \in \mathbb{R}^{o \times 2}$ and standard deviations, $\sigma \in \mathbb{R}^{o \times 2}$. We draw a 2D Gaussian kernel around each origin given by,

$$\mathcal{G}(u, v, \mathbf{c}, \sigma) = \exp - \left(\frac{(u - o_0)^2}{2\sigma_0^2} + \frac{(v - o_1)^2}{2\sigma_1^2} \right)$$

An illustration of the resulting mask is shown in Figure 5.1. This method flexibly masks low and high frequencies and the resulting images show unique textures containing both blurred and sharpened artifacts.

Figure 5.3 illustrates each proposed augmentation on a common set of images. We vary the strength of each augmentation via their respective hyperparameters (including m, n, p, q, k, o). Each augmentation's strength can be tuned such that it introduces

Table 5.1: **ImageNet-1K Pre-Training with FDA:** We report the linear probing top-1 accuracy of 4 self-supervised baselines pre-trained on ImageNet-1K. When FDA is applied in addition to standard image augmentations, we observe $\sim 1\%$ improvement in performance across all models. We report the mean and standard deviation across 3 random seeds.

	Top-1 Accuracy - ImageNet-1K			
	SimCLR	**BYOL**	**MoCo v2**	**SimSiam**
Baseline	69.2 (0.3)	74.3 (0.5)	71.7 (0.7)	73.7 (0.2)
+ FDA (Ours)	**70.5** (0.1)	**74.7** (0.6)	**73.0** (0.4)	**74.3** (0.5)

sufficient invariance but does not obfuscate the main content of the image relevant to the ground truth (in the downstream task). More importantly, we confirm this effect when each augmentation is used together with other FDA or image augmentations. Note that this is a subjective process involving visual examination of images. Due to resource constraints, we apply the same set of augmentation parameters for all our experiments (detailed in the Appendix) however, these can be further tuned for each specific baseline. In the next section, we perform pre-training experiments on a combination of both FDA and image augmentations following the pipeline illustrated in Figure 5.1.

5.5 Experimental Results

5.5.1 Experimental Setup

We examine 4 self-supervised baselines including SimCLR [Che+20a], MoCov2 [Che+20b], BYOL [Gri+20] and SimSiam [CH21]. Our TensorFlow [Mar+15] implementation replicates the training paradigms of each model including their encoder architecture (projector, predictor, momentum encoder etc.), loss, learning rate scheduling (cosine anneal) and optimizer (LARS [YGG17b]). More details about training detailed in the Appendix. We use the ResNet-50 [He+15] backbone for all our experiments. To be consistent, we apply the following image augmentations across all baselines - random

resized crop, color jitter, horizontal flip, Gaussian blur, grayscale and solarize. Within this augmentation pipeline, we incorporate our Fourier Domain Augmentations (FDA) as shown in Figure 5.1. All other training details and hyper-parameters are mentioned in the Appendix. SimCLR follows a single-encoder setup while MoCo, BYOL and SimSiam use a dual-encoder setup where one of the encoders is used for downstream tasks. We find that applying FDA to only the left view (left encoder is used for downstream tasks) in addition to existing image augmentations provides the best results as opposed to applying on both views. We perform standard linear probing for evaluation where we train a linear classifier on frozen pre-trained representations.

5.5.2 ImageNet Pre-Training

We pre-train SimCLR, BYOL, MoCov2 and SimSiam on ImageNet-1K [Rus+15b] by further diversifying the left view image augmentations with FDA. In Table 5.1, we summarize the linear probing top-1 accuracy for each model compared to their baselines which do not use FDA. We observe that FDA shows as average improvement of $\sim$ 1% with the highest improvement in MoCo v2 of 1.3%. Recall Figure 5.2 where we demonstrated the steady improvement in downstream performance as more augmentations are added. Our improvements with FDA solidifies our initial claims about the importance of diversity.

5.5.3 Transfer and Few-Shot Learning

We perform few-shot and transfer learning on the above frozen ImageNet pre-trained self-supervised baselines. In the few-shot setup, we apply 5-shot and 10-shot learning regimes where the training set contains 5 or 10 images per label respectively. We test for transfer learning on iNaturalist (5089 classes) [Hor+18], DomainNet Painting (345 classes) [Pen+19], Food101 (101 classes) [BGV14a] and Places365 (400 classes) [Zho+17]. We observe that pre-training with FDA largely benefits both few-shot

Table 5.2: **Transfer and few-shot learning with FDA pre-trained encoders:** We evaluate the few-shot (5-shot, 10-shot) and transfer learning performance on the frozen encoder from Table 5.1. We observe that baselines pre-trained with FDA improve the top-1 accuracy over most setups.

	ImageNet (5-shot)	ImageNet (10-shot)	iNaturalist	DomainNet (Painting)	Food101	Places365	Average
SimCLR	38.2	45.6	45.9	61.8	73.7	50.4	52.6
+ FDA	**39.3**	**46.6**	**47.7**	**63.6**	**74.4**	**51.0**	**53.8**
BYOL	47.5	54.2	52.4	67.1	77.0	50.6	58.1
+ FDA	**47.6**	53.7	**52.5**	**68.0**	76.7	**51.1**	**58.3**
MoCo v2	43.5	51.4	46.7	63.9	75.4	51.1	55.3
+ FDA	**43.8**	**52.8**	**48.6**	**64.0**	**76.8**	**51.9**	**56.2**
SimSiam	46.2	53.3	51.5	66.7	76.5	**50.3**	57.4
+ FDA	46.2	**53.5**	**52.7**	**67.3**	**76.7**	50.1	**57.8**

and transfer learning tasks across all baselines. We observe the highest average improvement in MoCo.

5.5.4 Image Retrieval on Transfer Datasets

We also employ image retrieval as a qualitative evaluator of the learned representations. Given a query image, we retrieve the top-4 nearest neighbours in the representation space using cosine similarity as the distance metric. Specifically, given a sample $\mathbf{x}$ we retrieve $\arg\max 4_{\mathbf{y}} \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$ from the dataset. In Figure 5.4, we display these results on 5 query images each from the test set of iNaturalist and DomainNet and compare the retrieved neighbours between MoCov2 baseline and MoCov2 trained with FDA on ImageNet-1K. The objective of this experiment is that the nearest neighbours should closely match the semantics of the retrieved images. This property is upheld in some FDA trained MoCo examples like the ice cream, teapot and woman with suitcase in DomainNet.

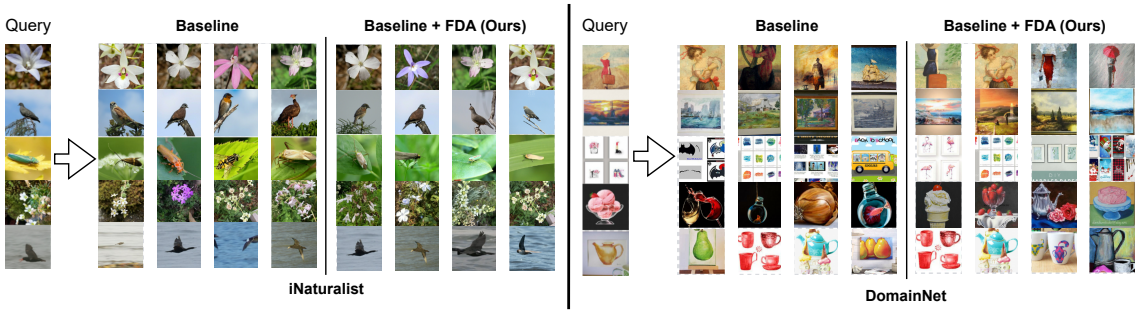


Figure 5.4: **Image retrieval:** We test the image retrieval quality of vanilla MoCov2 and MoCov2 pre-trained with FDA on ImageNet on transfer datasets, iNaturalist and DomainNet. We observe that the top retrieved images in MoCo FMA visibly match the semantics of the query image better.

Table 5.3: **Disentangling the effect of FDA and image augmentations:** In a single-encoder contrastive learning setup, we ablate between the pair of augmentations used going from $\mathbf{x}$ (no augmentations) to $A_{im}(\mathcal{F}^{-1}(A_{freq}(\mathbf{f})))$ (FDA + image augmentations). Here $A_{im}(.)$ denotes the image augmentations (random crop, color jitter, blur etc.) and $A_{freq}(.)$ denotes FDA transforms we propose (amplitude rescale, phase shift etc.). We observe the best performance when using FDA + image augmentations one view and image augmentations alone in the second view. All setups are pre-trained on ImageNet-1K and we report the linear probing top-1 accuracy.

Augmentation		**Left View**			
		$\mathbf{x}$	$A_{im}(\mathbf{x})$	$\mathcal{F}^{-1}(A_{freq}(\mathbf{f}))$	$A_{im}(\mathcal{F}^{-1}(A_{freq}(\mathbf{f})))$
Right View	$\mathbf{x}$	1.5	68.6	34.7	69.6
	$A_{im}(\mathbf{x})$		69.2 *(SimCLR baseline)*	68.8	**70.5**
	$\mathcal{F}^{-1}(A_{freq}(\mathbf{f}))$			38.9	67.8
	$A_{im}(\mathcal{F}^{-1}(A_{freq}(\mathbf{f})))$				70.4

5.6 Disentangling the Effects of Augmentation and Format Transform

We showed that pre-training state-of-the-art self-supervised baselines with FDA and standard image augmentations improves the linear classification performance of ImageNet-1K, its few-shot variants and various transfer learning datasets. This also confirms our initial hypothesis that more diverse augmentations ultimately benefit downstream tasks. However, a key aspect of our method is the utilization of the Fourier domain to introduce further diversity. Recall, our method involves multiple stages of transformations over a given image i.e., (i) The format transform (via FFT operation $\mathcal{F}$) (ii) Fourier Domain Augmentations (FDA) (iii) Inverse FFT operation to return to the image space ($\mathcal{F}^{-1}$) (iv) Standard image augmentations like color jittering, blur, grayscale etc. Therefore, it is essential to study the the effect of each operation independently to properly attribute the improvement in downstream performance.

We represent the raw input image as $\mathbf{x}$ and its Fourier transform $\mathcal{F}(\mathbf{x})$ as $\mathbf{f}$. We define the standard image augmentations, such as random crop, jitter, blur, as a function $A_{im}(.)$ and the FDAs as $A_{freq}(.)$. We train SimCLR in a single-encoder setup with a contrastive loss and various combinations of augmented views on ImageNet-1K. SimCLR uses the InfoNCE [OLV18] objective to learn image representations. For every query sample, we maximize its similarity in the latent space with one positive view of the same sample and minimize the similarity with the remaining samples in the batch. The objective is as follows,

$$\max \log \frac{\exp\left(\text{sim}(A(\mathbf{x}_i), A(\mathbf{x}_i))/\tau\right)}{\sum_{j=0}^{2N} \mathbb{1}_{i \neq j} \exp\left(\text{sim}(A(\mathbf{x}_i), A(\mathbf{x}_j))/\tau\right)} \tag{5.2}$$

where $sim(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$ and $A(.)$ is the stochastically applied set of augmentations. In Table 5.3, we test different pairs of augmentations between the positive

Table 5.4: **Sequence of augmentations:** We follow the sequence of augmentations illustrated in Figure 5.1 where we apply FDA before any of the image augmentations (except random crop which is applied first). In this table, we test to see if applying FDA after image augmentations is beneficial. We observe comparable performance in both setups (on SimCLR pre-trained on ImageNet-1K).

Augmentation		**Left View** $A_{im}(\mathcal{F}^{-1}(A_{freq}(\mathbf{f})))$	**Left View** $\mathcal{F}^{-1}(A_{freq}(\mathcal{F}(A_{im}(\mathbf{x}))))$
Right View	$A_{im}(\mathbf{x})$	**70.5**	70.4

views including, (i) $\mathbf{x}$: un-augmented and center-cropped image, (ii) $A_{im}(\mathbf{x})$, (iii) $\mathcal{F}^{-1}(A_{freq}(\mathbf{f}))$: FDA applied in the Fourier space and inverted back to the image space, (iv) $A_{im}(\mathcal{F}^{-1}(A_{freq}(\mathbf{f})))$: standard image augmentations applied on top of inverted FDA image. Due to the single-encoder contrastive learning setup, we present the results as an upper triangular matrix as swapping the views does not alter the overall objective.

We follow the SimCLR ImageNet-1K setup including the architecture, learning rate, scheduling, loss and optimizer. We define a naive baseline as the setup that uses a pair of raw un-augmented views ($\mathbf{x}$). The use of large batch sizes allows the model to contrast with a sufficient number of negative views, preventing collapse i.e., when all representations are identical. Nevertheless, this model achieves a low performance of 1.48% as lack of augmentations inhibits the learning of informative representations. Keeping the right view un-augmented, we next experiment with different View 1 augmentations (first row in Table 5.3). We observe significant improvements with both FDA (34.7%) and standard augmentations (68.6%) applied individually, but the performance gains are highest when they are used together (69.6%). Applying both augmentations to a single view also outperforms all methods which apply individual augmentations to both views. A similar trend is seen in the second row when we apply standard image augmentations to the right view. While we find that standard augmentations outperform FDA when applied individually, we attribute this mainly to the use of random cropping in standard augmentations, which significantly improves

their performance (from 40% to 69%).

As applying FDA in conjunction with image augmentations gives the best result, we next ablate how the order of FDA and image augmentations sequence affects the accuracy. In all previous experiments, we apply FDA before any other image augmentations (except random crop) following the sequence in Figure 5.1. In Table 5.4 we reverse this order and apply FDA after traditional image augmentations. Formally, this can be defined as $\mathcal{F}^{-1}(A_{freq}(\mathcal{F}(A_{im}(\mathbf{x}))))$. We observe a comparable performance in SimCLR ImageNet-1K with this image-augmentation-first strategy.

5.6.1 The Effect of Format Transform

We next disentangle the effect of using the Fourier transform ($\mathbf{f}$) directly as input to the self-supervised encoder. This experiment explores if we can produce better representations from input expressed in multiple formats (image and frequency) similar to the approach discussed in [WO21]. Since the Fourier spectrum of an image is complex-valued, it cannot be directly supplied to an image encoder. We therefore convert it to a real-valued 3 channel by re-scaling the spectrum to bring the values between $[0, 1]$ (same as image input). Since the RFFT2D output is of half the width ($\in \mathbb{R}^{H \times W/2 \times 3}$) as the image, we interleave the real and imaginary components such that the resulting frequency image is the same shape as that of the image ($\in \mathbb{R}^{H \times W \times 3}$). This procedure is available in our code.

Format transforms represent the information in frequency coordinates, which are incompatible with the image coordinate system. We therefore deploy a two-encoder setup where the first encoder $g_{im}(.)$ (left) only takes image input and the second $g_{freq}(.)$ (right) only takes frequency input. The two encoders do not share any weights and are trained independently. The representations of $g_{im}(.)$ are used for downstream tasks. We maximize agreement in the latent space using the standard InfoNCE loss described in Equation 5.2. Figure 5.5 illustrates this two-encoder setup.

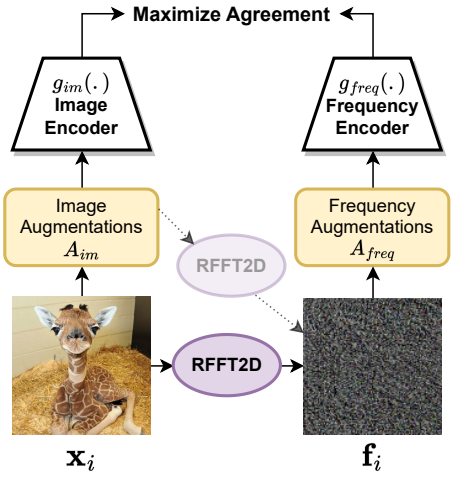


Figure 5.5: **Dual-encoder setup for multi-format contrastive learning:** To disentangle the effect of the format transform, we design a two-encoder setup where the left encoder $g_{im}(.)$ encodes the image view and the right encoder $g_{freq}(.)$ encodes the Fourier transform of the same view. Format-specific augmentations are applied to both views. Both encoders are trained independently (no shared weights) and are aligned in latent space via contrastive loss.

Note that our goal is to disentangle format transforms from augmentations. We take the naive baseline that uses two raw un-augmented views ($(\mathbf{x}, \mathbf{x})$ single encoder setup) and substitute the right view with the frequency image and train under the dual encoder regime (Figure 5.5). In Table 5.5, we present an interesting finding where contrasting raw image and frequency $(\mathbf{x}, \mathbf{f})$ results in 17.5% top-1 accuracy on ImageNet pre-training which is a 16% improvement over the raw baseline of 1.5%. Keeping the left view un-augmented, we augment the right view (i) in the frequency space ($A_{freq}(\mathbf{f})$) which improves the performance to 20.6% and (ii) in the image space before applying Fourier transform ($\mathcal{F}(A_{im}(\mathbf{x}))$) which improves the performance to 48.8%.

Next, we augment the left view ($A_{im}(\mathbf{x})$) and contrast against the set of frequency space right views. We do not observe improved performance with format transforms in this scenario. In fact, the performance degrades further when the frequency view is augmented. We hypothesize that this behavior may be caused by our choice of

Table 5.5: **Disentangling the effect of format transform:** We examine the effect of contrasting image and frequency views using the dual-encoder setup outlined in Figure 5.5 (cells highlighted in blue). We compare this against the single-encoder setup which uses both image views (first row). When the left image is not augmented, we observe noticeable improvements with format transform (and augmentations) in the right view. We do not observe similar improvements when the left image is augmented.

Augmentation		**Left View**	
		$\mathbf{x}$	$A_{im}(\mathbf{x})$
Right View	$\mathbf{x}$	1.5	68.6
	$\mathbf{f}$	17.5	63.3
	$A_{freq}(\mathbf{f})$	20.6	62.4
	$\mathcal{F}(A_{im}(\mathbf{x}))$	48.8	59.0

architecture for the frequency encoder i.e., ResNet (ConvNets). The *translation equivariance* property of convolutional neural networks that applies to real images, need not directly transfer to frequency images. The improvements we observe from the format transform in lieu of image augmentations in the left view are still non-trivial, opening a new direction for further research.

5.7 Discussion

We examined the need for diverse augmentations in self-supervised pre-training and proposed Frequency-Domain Augmentations (FDA) to introduce further diversity by tapping into the format transform of the image. FDA, when used in conjunction with image augmentations, showed improved performance on ImageNet-1K top-1 accuracy on 4 baselines - SimCLR, BYOL, MoCov2 and SimSiam. We also showed improvements in transfer learning, few-shot learning and image retrieval. We studied the disentangled effect of format transform using a dual-encoder setup with a dedicated frequency encoder. When no augmentations are used, we observed a 16% improvement in performance with the use of format transform in one view as compared to images in both views. Pre-training with the format transform improves over raw images, however, the best performance is still seen in the image space through diverse Fourier

(FDA) and image augmentations. Our findings open several questions for further research – (i) What are better methods to utilize and encode the format transform and FDA without requiring to invert back into the image space?, (ii) How can complex Fourier input be better structured to feed through real valued encoders?, (iii) How does FDA behave in specialized domains that are not real images (e.g., medical scans).

Chapter 6

Understanding the Effect of using Semantically Meaningful Tokens for Visual Representation Learning[1]

6.1 Introduction

Vision transformers (ViTs) [Dos+21; Liu+21; Tou+20] have emerged as a groundbreaking innovation in the field of computer vision, leveraging the power of transformer architectures [Vas+23] to process and interpret visual data. They also have shown unprecendented performance in multi-modal setups [Rad+21; Sin+22; ZZL22; Li+19; Li+22] where vision and language data are aligned for downstream retrieval, question-answering, captioning and other tasks. Tokenization is a key feature of transformers where the input is split into small chunks which are converted into vectors and then processed by the model. Text sentences (in the English language) is generally tokenized into words. Image data for ViTs is generally split into a grid of equally sized patches

[1]Full paper available at https://arxiv.org/abs/2405.16401. Contributing authors include Priyatham Kattakinda, Arman Zarei, Nikita Seleznev, Samuel Sharpe, Senthil Kumar and Soheil Feizi.

and then flattened into a sequence. Although this practice is widely accepted, we propose a different approach to patchifying which attempts to capture more high-level semantic information in patches.

Using off-the-shelf segmentation and scene-graph generation techniques, we extract panoptic segmentation mask embeddings of all objects a given image and CLIP [Rad+21] text embeddings of the actions or relationships between them. We call the object embeddings as *tangible tokens* since these are visible in the image and the relationship embeddings as the *intangible tokens* since these are not visible but are still observable. Both sets of tokens have independent semantic meaning, similar to words in a sentence but unlike equal-sized patches in ViTs. We also extract other metadata including image features, [subject, object, predicate] triplets and K-nearest neighbors in the image for each object. This metadata helps us capture both directional relationship and relative position information for complete visual comprehension.

We demonstrate a proof-of-concept that applies such semantically-meaningful tokens in visual representation learning and study its potential. We train a transformer model (called Visual Token Encoder) on the set extracted tokens of the COCO [Lin+15] dataset. We apply an additive attention mechanism using the relational and structural information from the metadata, ranked by importance. The learned image embeddings are contrastively aligned with the COCO caption embeddings from the CLIP text encoder which is fine-tuned alongside our model. We compare our method with 2 other vision-language pre-training setups which follow the same training regime as ours but the image-side encoder is replaced with (i) A ViT (randomly-initialized) or (ii) The CLIP image encoder (fine-tuned) and trained directly on COCO images.

Our experiments show that our tokenization process significantly improves representation quality, resulting in a 47% improvement in text-to-image retrieval over a ViT and 9% over CLIP (fine-tuned) on the COCO validation split. Moreover, we show improved compositional reasoning capabilities of the learned image representa-

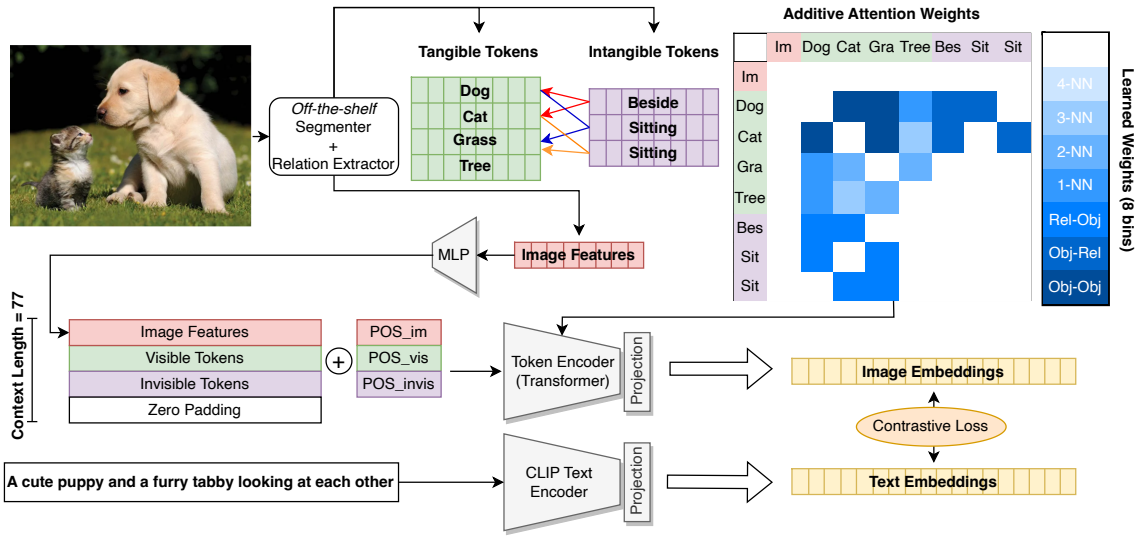


Figure 6.1: **Training with meaningful visual tokens:** We present a framework that uses off-the-shelf segmentation and relation extraction models to prepare a set of *tangible tokens* ($\mathcal{V}$) and *intangible tokens* ($\mathcal{U}$) for any arbitrary image, along with directional semantic relationships between them. These tokens and image features ($\mathbf{l}$) are then passed as input to our visual token encoder ($f(.)$). We utilize the semantic relation ($\mathcal{E}$) and relative location ($\mathcal{N}$) information of all tokens to compute additive attention weights, ranked by importance. The learned image embeddings ($\mathbf{s}$) are contrastively aligned with the text embeddings ($\mathbf{t}$) of the CLIP text encoder ($g(.)$), which is simultaneously fine-tuned with our model.

tions by exploring the ARO [Yuk+23] and Winoground [Thr+22] benchmarks. Our Visual Token Encoder outperforms the ViT by 18% on ARO and 9% on Winoground. These results indicate a promising angle for upcoming research - to re-think better encoder architectures that encapsulate high-level, semantic entities for improved visual understanding.

6.2 Related Work

Transformers and ViTs. Transformers [Vas+23] have revolutionized the field of natural language processing and have been increasingly applied to computer vision tasks. The self-attention mechanism in transformers enables the model to capture long-range dependencies, making them highly effective for various applications. Vision

Transformers (ViTs) [Dos+21] introduced the concept of patching, where an image is divided into fixed-size patches, and these patches are treated as tokens similar to words in NLP tasks. This patching approach allows transformers to process images efficiently, leveraging their powerful attention mechanisms. However, ViTs often struggle with computational efficiency and local feature extraction. Subsequent ViT variants, such as Swin Transformers [Liu+21], have introduced hierarchical structures that enhance the model's ability to capture multi-scale features. These hierarchical transformers divide the image into non-overlapping windows and compute self-attention within each window, allowing for better handling of larger images and more detailed features.

Large-Scale Image Segmentation. In the realm of large-scale image segmentation, several prominent models have made significant strides in addressing complex visual tasks. Among these, the Segment Anything Model (SAM) [Kir+23] and Segment Everything Everywhere All At Once (SEEM) [Zou+23] stand out due to their innovative approaches. These models excel at dividing an image into different components and parts, enabling detailed analysis and interpretation of complex scenes by segmenting and categorizing each visual element distinctly. Building on their innovative frameworks, they are adept at performing a variety of segmentation tasks, including semantic segmentation [LSD15], panoptic segmentation [Kir+19], and instance segmentation [He+18].

Architectural Improvements. To enhance understanding of complex images and processing them more efficiently, various methods have explored novel techniques. [Han+22] introduced Vision GNN (ViG), which models images as graphs by treating patches as nodes and their relationships as edges, effectively capturing complex structures and spatial relationships within images, thus outperforming traditional CNNs and transformers on some benchmarks. In another approach, [Ma+24] presented Groma, a Multimodal Large Language Model (MLLM) utilizing localized visual tokenization

to handle region-level tasks effectively, demonstrating superior performance on COCO [Lin+15] and Visual Genome benchmarks by efficiently grounding textual outputs to specific image regions. [Xia+23] proposed the Deformable Multi-Head Attention (DMHA) module in the Deformable Attention Transformer (DAT), which dynamically allocates key and value pairs to relevant regions, enhancing representation power while reducing computational overhead, achieving state-of-the-art results on benchmarks like ImageNet [Rus+15b], MS-COCO [Lin+15], and ADE20K [Zho+18]. Additionally, [Che+21] introduced the Deformable Patch-based Transformer (DPT), featuring a Deformable Patch (DePatch) module that dynamically adjusts patch positions and scales to preserve local structures and semantic integrity, thereby significantly improving performance in image classification and object detection tasks.

6.3 Re-thinking Tokenization in Vision Transformers

Text: Given a text corpus, tokenizers are typically constructed to concisely represent the text by capturing the sentences with as few tokens as possible while maintaining meaningful information. A simple yet effective practice to tokenize is, by splitting a given sentence into constituent words using delimiting characters like spaces, periods, commas, etc. This results in words that have semantic (sometimes physical) meaning when considered independently as well as in the sentence context. In Figure 6.2, we show an example of the sentence *A scenic view of the sea with a dog in front* and tokenize it into words such as *a, scenic, view, of, the, sea,* etc. In majority of written languages (including English) humans are conditioned to read a sentence word by word (right to left or left to right depending on the language). As we read each word, we associate that word with its meaning and simultaneously deduce the meaning of the sentence. The design of transformers and tokenization therefore makes intuitive sense as it closely mimics how humans process sentences.

Image: Visual tokenizers (in ViTs), on the other hand, treat images as a grid of patches, which are then flattened into a sequence and processed with several multi-head attention layers. Specifically, an image of size 224×224 is divided into small, equally-sized patches (say 16×16), resulting in 196 patches or *tokens* which are processed as if they were a sequence. This process of tokenization was adapted from transformers for text data. But unlike text data, each visual token (or patch) does not always have independent semantic meaning. This is illustrated in Figure 6.2, where tokens 3, 4, 13, and 16, when examined independently, have ambiguous semantic and physical meaning. For example, token 3 and 4 can be associated with a blue marble stone, and 16 can be associated with brushed metal. Each token needs to be studied in the context of the surrounding tokens or the entire image to be associated with a physical meaning. Therefore, there is a fundamental difference in the significance of text tokens and visual tokens although they are processed by transformer architectures in almost identical manners.

An alternative approach is to divide up the image into larger entities that each have independent physical meaning. Each of these entities possesses several observable constituent attributes. For example, in Figure 6.2, we have *sky, mountain, grass, sea, dog*, etc., where the grass is green with small yellow flowers and the mountain is rocky with alpine trees. Apart from these physical entities, we also draw several conclusions that are not necessarily associated with tangible (or visible) concepts. For example, the sea is in front of the mountain, the dog is sitting on the rock, and the rock lies on the grass. Both sets of entities, tangible and intangible, play a vital role in fully comprehending the image.

Our observation leads us to propose a set of modifications to the transformer architecture so that images are tokenized and processed in a more semantically meaningful way. In the next section, we define tangible and intangible tokens and describe how these can be extracted for any given image using off-the-shelf models.

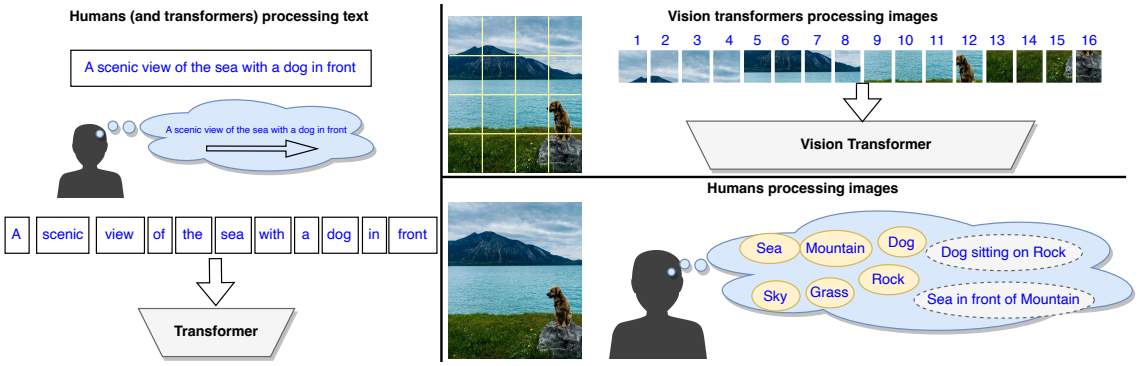


Figure 6.2: **Processing of text vs image data:** In this simple illustration, we demonstrate the notable difference in how text and visual data are processed by humans and transformers. Humans are capable of deciphering larger concepts from images (both tangible and intangible), where each concept has independent semantic meaning.

We hypothesize that this tokenization strategy facilitates the transformer's ability to process and reason about the various objects and their interrelationships, as each high-level visual entity is represented as an individual token readily accessible to the transformer.

6.4 Approach

6.4.1 Using Off-the-shelf Models to Extract Visual Tokens

Let's consider a real-world image $\mathbf{x}$ which may contain several entities, both in the foreground and background. We define the set of tangible entities (tokens) as $\mathcal{V}$. $\mathcal{V}$ may include several items of varying sizes, ranging from small details like the coffee cup in Figure 6.3 to much larger entities like the trees in the background. As discussed in Section 6.3, there are several observable components in the image that cannot be localized. These entities usually correspond to actions or relationships among objects in the scene. We denote this set of *intangible entities (tokens)* as $\mathcal{U}$.

We utilize an off-the-shelf instance segmentation model titled Segment Everything Everywhere All At Once (SEEM) [Zou+23] to extract *mask embeddings* of all the

tangible tokens $\mathcal{V} = \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n$. These mask embeddings are outputs from an X-Decoder head [Zou+22] which encode both localization and object-related information. We set a threshold on the instance segmentation scores to 0.9 to select only high-scoring detections. We hypothesize that these mask embeddings capture visual information of the corresponding objects within the context of the given image more effectively than extracting separate embeddings for each bounding box using off-the-shelf image encoders like DINO [Oqu+24] or CLIP [Rad+21].

In addition to $\mathcal{V}$, we also extract global image features ($\mathbf{l}$) computed by the segmenter. SEEM, like most segmentation models, follows a U-Net [RFB15] design which contracts and expands the image input to result in localized bounding boxes and masks. We therefore collect 2-D average pooled image features at each layer and finally concatenate the resulting vectors.

Next, we extract the set of intangible tokens $\mathcal{U}$ using the Relate-Anything Model (RAM). RAM is built on the Segment-Anything Model (SAM) [Kir+23] and is trained on the Panoptic Scene Graph Generation dataset (PSG) [Yan+22c] to reason about relationships between any two arbitrary object masks provided with an input image. After extracting the tangible token vectors (mask embeddings) $\mathcal{V}$, we pass pairs of 2D masks corresponding to $(\mathbf{v}_a, \mathbf{v}_b), 1 \leq a, b \leq |\mathcal{V}|$ to RAM to obtain a prediction for the relationship between the corresponding objects. We set a threshold on the classification score at 0.05 (as specified by the RAM model) to select only high-scoring relationships. We then embed the relationship class using the CLIP text encoder. This process results in a set of intangible tokens $\mathcal{U} = \mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_m$.

Finally, we also extract the directional *(subject, object, predicate)* triplet indices, denoted by $\mathcal{E} = \{(a, b, c) : 1 \leq a, b \leq |\mathcal{V}|, 1 \leq c \leq |\mathcal{U}|, \forall c\}$. This means that the subject a is performing the action c, received by the object b, and the term *object* in this context refers to the part-of-sentence tag in language terminology (a, b, c are indices of corresponding tokens or visual entities). These triplets capture the semantic

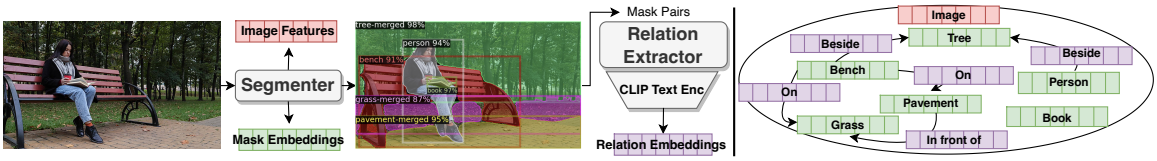


Figure 6.3: **Using off-the-shelf models to extract tokens:** We extract image features ($\mathbf{l}$) and mask embeddings ($\mathcal{V}$) from a panoptic segmentation model. Next, we pass pairs of object masks to a relation extractor and collect the highly probable relationships ($\mathcal{E}$). We compute CLIP text embeddings of all relationships ($\mathcal{U}$). This information is distilled into a scene graph representing the image as shown.

correlation between tangible and intangible tokens, resulting in a scene graph of vector nodes and vector edges as illustrated in Figure 6.3. Beyond scene graphs, we also obtain structural information on how objects are co-located in the image by computing the 4-nearest neighbors of each tangible token. Specifically, for every instance we discover from the segmenter, we rank the 4-nearest neighbor instances by computing the Euclidean distance between the centers of the bounding boxes. We formally define this set as $\mathcal{N} = \{(n_k^{(a)}), 1 \leq i \leq |\mathcal{V}|, \forall a, k \in [1, 2, 3, 4]\}$.

We note that our method of extracting tokens can be substituted with alternative segmentation models or scene graph generation methods [Yan+22c]. Scene graph datasets like Visual Genome [Kri+16] provide pre-defined image graphs with object and relationship sets. Our goal of extracting semantic tokens is to capture larger, concrete visual entities as token embeddings rather than tiny patches that are flattened. Therefore, alternate approaches to extract object masks and embeddings can be used in our framework as long as the stated goal is met.

In summary, our token extraction process uses an off-the-shelf segmenter and relation extractor to obtain (i) Image features, denoted by $\mathbf{l}$, (ii) Set of tangible tokens, denoted by $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_n\}$, (iii) Set of intangible tokens, denoted by $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, ...\mathbf{u}_m\}$, (iv) Set of (subject, object, predicate) triplets, denoted by $\mathcal{E} = \{(a, b, c) : 1 \leq a, b \leq |\mathcal{V}|, 1 \leq c \leq |\mathcal{U}|, \forall c\}$, (v) Set of 4-nearest neighbors, denoted by $\mathcal{N} = \{(n_k^{(a)}), 1 \leq a \leq |\mathcal{V}|, \forall a, k \in [1, 2, 3, 4]\}$. We outline this process in Figure 6.3.

6.4.2 Training with Additive Attention

In this section, we explain the process of training our model using all the elements we extracted in Section 6.4.1. Rather than training a vision transformer on image data, we train a transformer model $f(\cdot)$ on the extracted token vectors. To this end, we prepare each sample i from the training data as a concatenated set of $\mathbf{l}_i$ (image features), $\mathcal{V}_i$ (tangible tokens), and $\mathcal{U}_i$ (intangible tokens). We formally define this set as $\mathcal{T}_i = \{\mathbf{l}_i\} \cup \mathcal{V}_i \cup \mathcal{U}_i$. Similar to text tokens in a transformer, where each token has independent meaning, we now have a set of *visual tokens* where each token has some semantic association. Since each component of $\mathcal{T}_i$ is extracted in a unique manner, involving different deep networks, we add separate positional embeddings to each token based on its type. Specifically, we initialize 3 learnable positional embeddings $\mathbf{p}_v$, $\mathbf{p}_u$, and $\mathbf{p}_l$, and add them to each token: $\mathbf{v} = \mathbf{v} + \mathbf{p}_v \, \forall \mathbf{v} \in \mathcal{V}_i$, $\mathbf{u} = \mathbf{u} + \mathbf{p}_u \, \forall \mathbf{u} \in \mathcal{U}_i$, $\mathbf{l}_i = \mathbf{l}_i + \mathbf{p}_l$.

The tokens and positional encodings account for most of the visual information available in the given image, including intangible information about actions and relationships. However, the structural connectivity between tokens as illustrated in Figure 6.3 is still lacking. Since our data is of graph structure, it makes intuitive sense to use Graph Neural Networks and variants [Sca+09; KW17] as primary encoders. However, the scaling and computational complexity of these models make us explore a simpler idea. Transformers, on the other hand, are a proven recipe to train large-scale models efficiently and learn generalizable representations. The attention mechanism [Vas+23] already encapsulates varying levels of importance between tokens and their neighbors.

In the context of text data, attention allows us to identify strong correlations of each word with surrounding words simultaneously as models make sense out of a given sentence. In our setup, we have several tangible tokens, correlated with each other in two manners defined by i) Semantic relations that we extract as intangible

tokens $\mathcal{U}$ and (subject, object, predicate) triplets $\mathcal{E}$ and ii) Relative positions in the image defined by the set of nearest neighbors $\mathcal{N}$. We therefore simulate both of these correlations by applying a ranked additive weight to the computed attention scores between each pair of tokens.

We prepare a weight matrix $A_i \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{T}|}$ for any i^{th} data sample. We populate A_i with 7 types of relationships between the tokens in $\mathcal{T}_i$, ranked by their importance in image comprehension.

$$A_i^{(a,b)} = \begin{cases} 7, & \text{if } (a,b,.) \in \mathcal{E}_i \\ 6, & \text{if } (a,.,b) \in \mathcal{E}_i \text{ or } (.,a,b) \in \mathcal{E}_i \\ 5, & \text{if } (b,.,a) \in \mathcal{E}_i \text{ or } (.,b,a) \in \mathcal{E}_i \\ 5-k, & \text{if } b = \mathcal{N}_i^{(k)}, k = \{1,2,3,4\} \end{cases} \tag{6.1}$$

Here, a and b represent the indices of any pair of tokens in the given set $\mathcal{T}_i$ and for simplicity of notation, we directly denote (subject, object, predicate) sets using a and b. With reference to the scene graph shown in Figure 6.4.1, the ranks are applied as (i) 7 for a directional node to node connection, (ii) 6 for every node to edge connection (*person* to *beside*, *tree* to *beside*), (iii) 5 for edge to node connections (*beside* to *person* and *beside* to *tree*), (iv) 4, 3, 2, 1 for the first to the fourth nearest neighbor (in the image) of the given node respectively.

We define an attention weight encoding $\mathbf{a} \in \mathbb{R}^8$ that is learned based on the rankings in the attention weight matrix A_i for all samples. The first element in $\mathbf{a}$ is set to 0 and each subsequent element is learned such that the difference between that element and the previous element is equal to a rank $\{1, 2, ..7\}$. This is done simply by computing the cumulative sum, $cumsum(\exp(\mathbf{a}))$ and substituting each rank in A_i with the learned rank weight in $cumsum(\exp(\mathbf{a}))$. Finally, we add this updated

learned weight matrix A_i to the computed self-attention score across all attention heads in our model.

We deploy our *visual token encoder* that applies all of the strategies described above in a vision-language pre-training setup. As shown in Figure 6.1, we preprocess every image in the training data using our visual tokenization method, which extracts the set of tokens $\mathcal{T}$ and prepares attention rank matrices A for those tokens. The token embeddings are padded to a fixed context length and added to positional encodings $\mathbf{p}_v$, $\mathbf{p}_u$, and $\mathbf{p}_l$. The tokens, along with the attention weight matrix (computed using $\mathbf{a}$ and A_i), are fed into our visual token encoder, $f(\cdot)$, which follows a standard transformer architecture to extract fixed-length image embeddings $\mathbf{s}$. We simultaneously also train a text encoder, $g(\cdot)$ (also a transformer), on the image captions to extract fixed-length text embeddings $\mathbf{t}$. We follow the CLIP [Rad+21] optimization, which applies a simple contrastive loss between all s_i's and t_i's in large batches.

Practically, training this model is more efficient compared to ViTs and CLIP since we process relatively low-dimensional data compared to high-dimensional large images. We use a context length of 77 during token extraction and our token embedding width is 512 (arising from the segmentation model). Therefore, each sample is of 77×512 dimensions along with A, a $|\mathcal{T}| \times |\mathcal{T}|$ dimensional weight matrix, where $|\mathcal{T}| < 77$. While the training speed and compute cost is significantly lower, we cannot ignore the added overhead of token extraction itself. The compute cost and memory overhead comes from the segmentation and relation extraction process where each image needs to be processed individually and the extracted token and metadata need to be saved on the disk for training.

6.5 Results

6.5.1 Experimental Setup

Our experimental premise is to demonstrate a proof-of-concept of our hypothesis stating - using semantically meaningful tokens can be beneficial in learning comprehensive, compositional representations. For token extraction, as discussed in Section 6.4.1, we use the segmenter, SEEM [Kir+23] (Focal-L [Yan+22b] backbone) and relation extractor RAM [Yan+22c]. SEEM is trained on COCO [Lin+15] while RAM is trained on the Panoptic Scene Graph Generation (PSG) dataset which contains 49K images arising from COCO and Visual Genome [Kri+16]. We extract and save all sets of tokens and metadata as listed in Section 6.4.1 for the COCO train (118M samples) and validation (5K samples) splits. We set our context length to 77 tokens and add zero-padding, if needed.

Next, we train our Visual Token Encoder from scratch on the synthesized COCO token dataset to confirm our hypothesis. Our tangible and intangible tokens are of 512 dimensions and we use a 3-layer MLP with ReLU [Aga18] activation to project the concatenated intermediate image features from the segmentation model backbone into 512 dimensions. We use the PyTorch implementation of the Transformer model using 8 layers and 8 attention heads with a linear projection head that dimension of 512. We simultaneously fine-tune the CLIP (ViT-B/32) text encoder, which outputs 512-dimensional text embeddings, on the COCO caption data by loading 1 randomly sampled caption out of 5 per image in the dataset.

We sweep over 4 learning rates $\{1e^{-6}, 5e^{-6}, 1e^{-5}, 5e^{-5}\}$ and choose the best performing model. We use the AdamW optimizer [LH19] and train for 100 epochs, using a batch size of 256, with a warmup of 10 epochs and cosine annealing learning rate schedule. We perform experiments with and without additive attention as an ablation.

In order to understand the benefits of using our proposed tokenization approach,

we compare with 3 baseline setups which are directly trained on image data using standard tokenization techniques i.e., image patchification. The first setup replaces our tokenizer and transformer with a standard ViT [Dos+21] trained directly on COCO images. We use the PyTorch implementation of the VisionTransformer model and train a ViT-s/16 variant which has 8 layers and 8 attention heads, closely matching the architecture of our Visual Token Encoder. We align the learned image embeddings with the CLIP text embeddings in the same manner as described above. Our second setup, is the pre-trained CLIP (ViT-B/32) model which is trained on very large-scale data [Rad+21] of roughly 400M samples. Finally, in our last setup, we fine-tune the pre-trained CLIP (ViT-B/32) model on COCO images and captions. This setup is similar to that of the ViT-s/16 except CLIP is already pre-trained on a large amount of data while the ViT is trained from scratch. We use the same training pipeline described for our Visual Token Encoder for all experiments - including number of epochs, learning rate sweeps, optimizer, schedulers, etc. and all results are averaged over 2 random seeds. Our Visual Token Encoder can be trained efficiently using 2 A5000's, however, the ViT-s/16 and CLIP models need to be trained on 4 A6000's.

6.5.2 Learned Representations

We measure several metrics through the course of training our model and baselines to understand the quality of the learned visual representations. As mentioned, in our training pipeline, we fine-tune the CLIP text encoder with a vision component which can be any one of (i) Visual Token Encoder (Ours), (ii) Visual Token Encoder (Ours), without Additive Attention, (iii) ViT-s/16 (iv) CLIP (fine-tuned). In Figure 6.4, we show that the training loss converges across all setups. The CLIP model which is already pre-trained maintains a low loss through the course of training.

We evaluate the alignment between visual and text representations by calculating the image-to-text and text-to-image retrieval scores on the COCO validation split

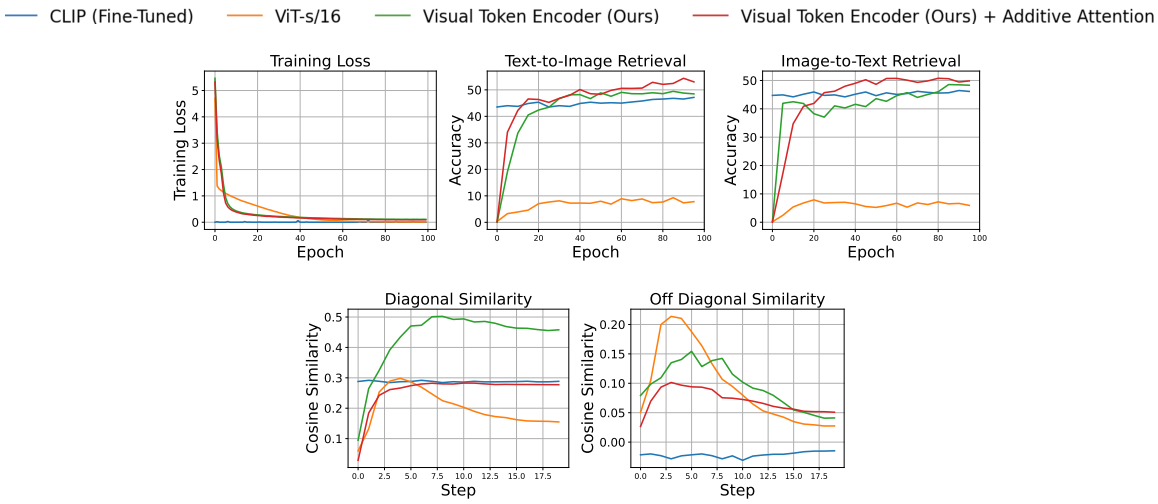


Figure 6.4: **Learned Representations:** In text-to-image and image-to-text retrieval accuracy, we observe that our visual token encoders perform best beating both CLIP (fine-tuned) and ViT-s/16 baselines. We also show the average diagonal and off-diagonal similarity of the learned representations across training iterations. From these plots, we observe that the contrast is strongest for our visual token encoder when additive attention is not used.

across iterations. We randomly sample 1 caption out of 5 per COCO sample and compute all text and visual embeddings. We then measure the zero-shot retrieval accuracy of matching the closest image to a given text (text-to-image) and closest text to a given image (image-to-text). These metrics are plotted in the second and third subfigures in Figure 6.4. We observe that the retrieval scores of our Visual Token Encoder are the highest amongst all experiments. The usage of additive attention results in 54.35 text-to-image retrieval accuracy which is a 9% improvement from the fine-tuned CLIP model and 47% improvement over ViT-s/16. This model also performs best for image-to-text retrieval achieving 49.76 accuracy which is 4% improvement over the fine-tuned CLIP model and a 44% improvement over ViT-s/16. These results are especially noteworthy because both our Visual Token Encoder and the ViT-s/16 are trained from scratch to convergence on the same data (COCO) but our method learns significantly more powerful representations. Our model also beats CLIP which is pre-trained on 1000× more data and fine-tuned on COCO for the same number of iterations.

Table 6.1: **ARO Benchmark:** We evaluate our model and baselines on 4 components of the ARO [Yuk+23] benchmark and measure the accuracy for each. We outperform both CLIP and ViT across VG-Relation, VG-Attribution and COCO-Order and beats the ViT on Flickr-Order. * indicates models that are trained from scratch on COCO. FT indicates pretrained models that are fine-tuned on COCO.

Image Encoder	Text Encoder	VG-Relation	VG-Attribution	COCO-Order	Flickr-Order
CLIP	CLIP	59.9	63.1	47.4	**58.0**
CLIP^{FT}	CLIP^{FT}	65.8	65.9	56.4	32.7
ViT-s/16*	CLIP^{FT}	53.5	53.9	38.7	28.9
Visual Token Encoder* (Ours) (without additive attn.)	CLIP^{FT}	67.8	64.1	17.2	34.8
Visual Token Encoder* (Ours)	CLIP^{FT}	**68.9**	**66.2**	**56.8**	41.4

In the last two plots in Figure 6.4, we compare the average diagonal and off-diagonal similarities in the COCO validation text-image embedding cosine similarity matrix. Since the CLIP model has already converged, we do not observe major changes in embedding similarity as training progresses. Other models which are trained from scratch show an increasing trajectory in the off-diagonal similarity, followed by a decrease, finally leading to a similarity score lower than that of the diagonal values. We observe the strongest contrast between diagonal and off-diagonal scores in our visual token encoder when additive attention is not used and the weakest contrast in the ViT-s/16 model.

6.5.3 Compositionality Benchmarks

As vision-language models gained popularity, several follow up works have challenged their compositional reasoning capabilities. Compositionality benchmarks like ARO [Yuk+23] and Winoground [Thr+22] propose a set of evaluation datasets which can be used to understand the depth of vision-language model reasoning. Both benchmarks have highlighted the significant lack of compositional understanding in state-of-the-art vision language models like CLIP. In this section, we study the behavior of the Visual Token Encoder we proposed on these benchmarks, compared to our baselines.

The ARO benchmark consists of 4 datasets - Visual Genome-Relation (VG-

Relation), Visual Genome-Attribution (VG-Attribution), COCO-Order and Flickr-Order. A sample arising from VG-Relation and VG-Attribution consists of an image with 2 caption options, a correct caption and an incorrect caption where either the relations (between objects) or the attributions (object properties) are interchanged across objects. Samples from COCO-Order and Flickr-Order consist of an image and 5 caption options, where only one is correct and the others have shuffled words to test order sensitivity. The accuracy for each dataset measures the percentage of images matched with the correct caption by the given model using its corresponding similarity metric (cosine similarity). We evaluate our model and baselines on each of these datasets and present our results in Table 6.1.

In VG-Relation, VG-Attribution and COCO-Order benchmarks, our model with additive attention performs best, showing a 10% improvement over CLIP (off-the-shelf) and 18% over ViT-s/16. We consider the ViT-s/16 as a fair competitor with our model since it has seen the same data (COCO train) and Visual Genome and Flickr are both out-of-distribution datasets. In COCO-Order, we observe a degraded accuracy in our model when additive attention is not used. Without additive attention, the visual tokens of COCO are simply stacked and presented to the transformer with no information of the nature of their relations. We suspect that this prevents the model from choosing the correct permutation of words in COCO-Order. Our model outperforms the ViT in Flickr-Order but does not beat the CLIP (off-the-shelf) baseline. This may be because Flickr may be closer to the training distribution of $400M$ samples that CLIP has seen.

Winoground, like ARO, also tests for relation and attribution reasoning. Each sample in Winoground consists of 2 images and 2 captions and the accuracy is measured by a given model's capability of associating the correct image to the correct caption and vice versa. The resulting metrics are Text Correct (assigning text to correct image), Image Correct (assigning image to correct text) and Group Correct (a combination of

Table 6.2: **Winoground Benchmark:** We evaluate our model and baselines on Winoground [Thr+22] and measure the 3 metrics given by the dataset. Our model outperforms CLIP and ViT in the image correct and group correct metrics. * indicates models that are trained from scratch on COCO. FT indicates pretrained models that are fine-tuned on COCO.

Image Encoder	Text Encoder	Text Correct	Image Correct	Group Correct
CLIP	CLIP	30.75	10.50	8.00
CLIP^{FT}	CLIP^{FT}	28.25	12.00	7.25
ViT-s/16*	CLIP^{FT}	18.00	13.00	7.00
Visual Token Encoder* (Ours) (without additive attn.)	CLIP^{FT}	**28.25**	15.25	9.25
Visual Token Encoder* (Ours)	CLIP^{FT}	27.00	**16.00**	**9.75**

the previous two). In Table 6.2, we summarize these metrics across our experiments. Compared to ARO, Winoground is a harder benchmark where even a large-scale model like CLIP only reaches a 10.50 image correct score. Our model outperforms all others in image correct and group correct scores showing 3% improvement over ViT and 4% over CLIP (fine-tuned). We beat the ViT by 10% on text correct scores with and without additive attention. Our tokenization process is beneficial to the image correct scores rather than text correct, because we attempt to learn compositional image embeddings such that they are better associated with correct captions.

6.6 Discussion

We challenge the premise of equal-sized patching in vision transformers and propose to use variable-sized, semantically meaningful tokens for visual understanding. We use off-the-shelf segmentation models and scene graph generation models which can detect high-level patches, such as objects in real images, which possess independent physical meaning. Additionally, we show that visual comprehension can be enhanced with intangible tokens like actions and relations that have semantic significance but are not physically localized in the image. We train a transformer model, referred to as the Visual Token Encoder, on the extracted set of tokens on the COCO dataset to learn

image representations and align them with caption representations from a fine-tuned CLIP text encoder. We incorporate other metadata, such as directional relationships and relative positions of tokens, by applying additive attention weights ranked by importance. These updates result in a 47% improvement in text-to-image retrieval compared to using a Vision Transformer (ViT) and a 9% improvement over the fine-tuned CLIP model. Additionally, we show an 18% improvement over ViT in the ARO benchmark and a 10% improvement in the Winoground benchmark, indicating that our Visual Token Encoder produces higher-quality compositional representations. Our contribution presents a proof-of-concept for rethinking tokenization in vision models and the associated potential benefits. Our findings open new avenues for empirical and theoretical research, specifically: (i) How does this tokenization approach perform in large-scale setups? (ii) Can we develop a unified model for both scene-graph generation and representation learning? (iii) How can our tokenization method be made more compute and memory efficient?

Chapter 7

Conclusion

In this thesis, we studied various failure modes of deep learning methods through the lens of visual representation learning. We discovered vulnerabilities in representation patterns that lead to these failures and identified methods to enable more structured, generalizable learning.

The first failure we investigated, is domain generalization. In Chapter 2, we showed that naive self-supervised representations are not trained to distinguish domain and content information in images. Our Domain Disentanglement Module (DDM) plugged into existing SSL models, pushed them to encode content information invariant to domain information which boosted downstream linear classification accuracy. In Chapter 3, we further dissected the representation space to define a per-sample, model-agnostic, quality metric that measures the chances of a representation being correctly classified. Using this metric as a regularizer while fine-tuning, we showed significant improvements over linear evaluation, transfer learning and interpretability. We next proposed an interpretability method called FALCON in Chapter 4, which can be used to explain subspaces in representations in a human-understandable format and help debug failures.

The next failure in representation learning we identified is the role of diverse aug-

mentations in ensure required invariances for generalization. In Chapter 5, we showed that the Fourier space of images can be utilized to further diversify augmentations and also to contrast between multiple formats of the same input. Finally, we discussed compositionality as a failure mode in visual understanding in Chapter 6. We replaced fixed-sized image patches with scene graphs of semantically meaningful visual tokens for transformers to learn better structured, dense, image representations.

In general, pre-training on large amounts of data to learn representations without strong priors, can lead of unforeseen side-effects that manifest as failures. Our research unraveled image representations to discover key aspects of data, architectures and training pipelines that govern these failures. They also open several directions for further research, some of which we discuss in Chapter 8.

Chapter 8

Future Work

8.1 Learning Visual Priors Alongside Representations

In Chapter 6, we presented our findings that rich atomic embeddings, structured as a scene graph can significantly improve multi-modal retrieval performance. This tokenization framework however, relies on the use of off-the-shelf models to identify tangible and intangible tokens. An important direction for further research is learning the full tokenization process during image-text pre-training. This involves training the scene-graph extraction process alongside the full representation alignment framework.

Similar to human perception, we believe that models should learn from complex scenes by decomposing them into atomic entities. A portion of these entities can be detected based on prior knowledge, similar to the way we used off-the-shelf segmenters and relation extractors. With densely annotated data (like MS-COCO), we can certainly train an end-to-end framework including a tokenizer that produces a scene-graph. However, a second portion of entities unknown to the model and not annotated in training data always exists. Utilizing self-supervision to learn these entities during end-to-end pre-training is an interesting research direction. Such approaches enable

models to update themselves with unseen entities, without forgetting known concepts, which can have a strong impact in continual/lifelong learning research communities.

8.2 Distributional Understanding in Generative Models

Deployment of large deep learning models can be hurdled by several safety constraints. These are generally dominated by model uncertainty to distribution shifts or out-of-scope inputs. Such examples may be provided either specifically or accidentally from distributions different from that of the training data. Deep models however, have little to no understanding of input likelihood or typicality, unless they are specifically optimized for it. When presented with inputs that are out-of-scope, models therefore resort to mis-predictions and hallucinations which, under safety-critical setups, can have serious consequences. It is imperative that inputs to deployed models are examined for their *typicality* i.e., closeness to a known training distribution, before being processed.

For a given training distribution (or input distribution, ID), out-of-distribution (OOD) examples can arise from infinitely many directions. Instead of designs that make assumptions on OOD statistics like outlier exposure [HMD19], we can instead *learn* ID data via state-of-the-art generative models. Autoregressive generative models like PixelCNN [Oor+16], Flow-based models like RealNVP [DSB17] and Diffusion Models like Variational Diffusion [Kin+23] which can generate photo-realistic images are also capable of density estimation in competitive benchmarks. However, prior works [KIW20] have shown that these models often surprisingly fail to distinguish between ID and OOD data. We hypothesize that generative capabilities are not necessarily optimized for semantic understanding which could lead to learning pixel correlations for realistic images rather than meaningful concepts from the distribution.

Our studies on representation models have shown that they produce features that respect semantically-relevant information rather than naive image compression. A direction for better density estimation and OOD detection is to utilize semantically-rich representations as conditioning for generative models. This method applies a granular control to image generation and sheds light on what concepts constitute as in-distribution and are suitable for further processing.

Appendix A

Supplementary Material - Chapter 2

A.1 Representation Space

In the top panel of Figure A.1, we visualize the representations of SimCLR pre-trained on ImageNet-1K [Rus+15b]. Each row denotes the representation vector ($\mathbf{h}_i$) of a random sample drawn from the the ImageNet-1K train set. There are 2048 columns corresponding to the representation size of a ResNet-50 [He+16] encoder.

We observe that the representations are all nearly sparse with a small number of strongly deviated coordinates. We verify this observation in the second panel of Figure A.1 where, we plot the distribution of all the SimCLR features of the same samples as the top panel, as well that of other self-supervised models including, DINO, SwaV, MoCo, VICReg and Barlow Twins. In each distribution, a very large number of features have a magnitude of 0 or very close to 0. In the zoomed version of the same plot, we can see a relatively small number of features that show strong activations.

A.2 Results on Other Datasets

As an extension to the results shown in Table 3.1, we include results on more datasets including CIFAR-10 [KNHa], CIFAR-100 [KNHb] and STL-10 [CLN] on 8 self-supervised

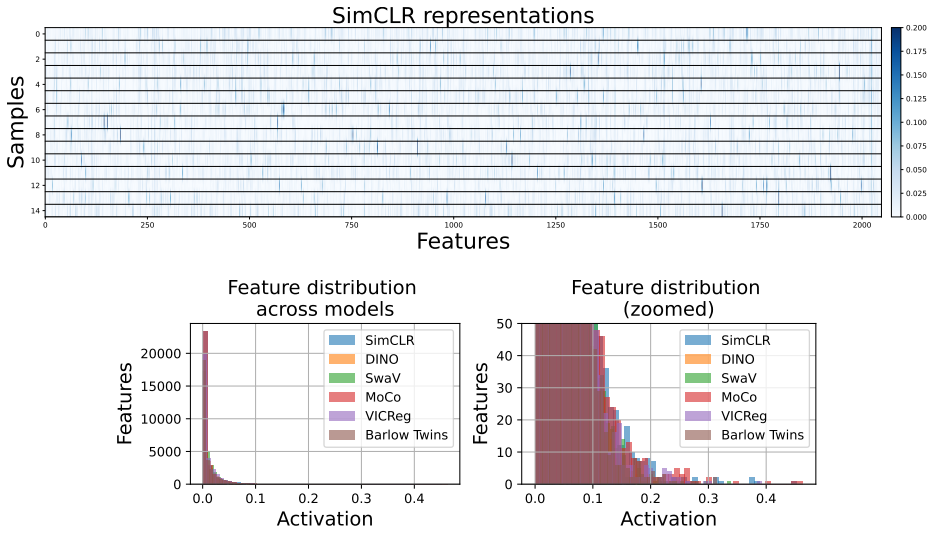


Figure A.1: **Visualizing the self-supervised representation space:** The top panel shows a heatmap of SimCLR representations of random ImageNet-1K samples. In the second panel, we plot the distribution of the features of the visualized samples for various models. We observe that representations are mostly sparse with a small number of strongly activated coordinates.

Table A.1: **Linear classification performance with Q-Score regularization (more datasets):** Similar to Table 3.1, we tabulate our results on CIFAR-10, CIFAR-100 and STL-10

Model	CIFAR-10			CIFAR-100			STL-10		
	Baseline	Lasso	Q-Score	Baseline	Lasso	Q-Score	Baseline	Lasso	Q-Score
SimCLR	90.83	89.33	**92.31**	65.82	68.21	**68.90**	76.42	75.59	**79.83**
SwaV	89.23	89.37	**90.03**	65.13	66.06	**66.52**	73.94	69.93	**75.03**
MoCo	92.95	90.59	**94.77**	70.12	67.23	**71.16**	73.21	72.65	**74.29**
BYOL	92.59	90.27	**92.82**	70.54	71.26	**72.71**	70.59	70.27	**74.47**
DINO	89.54	89.57	**89.85**	66.82	65.52	**67.49**	68.36	69.29	**69.38**
SimSiam	91.03	90.74	**92.48**	66.58	65.69	**69.03**	72.94	67.54	**73.52**
VICReg	92.69	91.83	**93.74**	68.81	66.75	**71.76**	70.76	70.61	**72.82**
Barlow	93.46	91.75	**93.87**	71.82	71.54	**71.91**	74.17	70.27	**74.32**

baselines when fine-tuned (further trained) with and without Q-Score regularization. In Table A.2, we observe that Q-Score regularization helps boost the performance of all state-of-the-art models across datasets.

A.3 Transfer Performance of Q-Score Regularization

In Table A.2, we tabulate the transfer learning performance (linear evaluation) of various unseen datasets [KNHa; KNHb; CLN; Maj+13; NZ08; BGV14b; Kra+13;

Cim+14] on 6 self-supervised models trained on ImageNet-1K with and without Q-Score regularization. We use frozen ResNet-50 representations for each transfer dataset (using actual image size) and perform linear evaluation using a classifier. We observe that the average accuracy of unseen datasets improves on all setups, especially on SimCLR, SwaV and MoCo.

In Figure A.2, we visualize the gradient heatmaps of some discriminative features discovered on SimCLR on ImageNet-1K on both ImageNet-1K and unseen datasets, Aircraft [Maj+13], Food [BGV14b] and Cars [Kra+13]. We observe that the physical meaning associated with each feature is consistent between both the training and unseen data. The heatmaps also correspond to informative features, strongly correlated with the ground truth. These gradients indicate that discriminative features are transferable across unseen datasets, which support the improvement we observe in Table A.2.

We also visualize the representations of correct and incorrect classifications of the Flowers [NZ08] dataset in Figure A.3. We use SimCLR pre-trained on ImageNet-1K (top panel) and the same model pre-trained with Q-Score regularization (bottom panel). We observe that the same properties as Figure 3.3 on ImageNet-1K (train dataset) transfer at test time to Flowers, an unseen dataset. Before regularization, representations, especially the mis-classified ones, do not contain highly activating discriminative features. These features get more enhanced after Q-Score regularization leading to improved top-1 accuracy as shown in Table A.2.

Table A.2: **Transfer learning performance of various state-of-the-art self-supervised models trained on ImageNet-1K with and without Q-Score regularization:** We observe that fine-tuning with Q-Score regularization improves the average transfer accuracy on all self-supervised models.

Transfer Dataset	SimCLR		SwaV		MoCo		BYOL		DINO		SimSiam		VICReg		Barlow Twins	
	Baseline	Q-Score Regularized	Baseline	Q-Score Regularized	Baseline	Q-Score Regularized	Baseline	Q-Score Regularized	Baseline	Q-Score Regularized	Baseline	Q-Score Regularized	Baseline	Q-Score Regularized	Baseline	Q-Score Regularized
CIFAR-10	70.13	**70.55**	71.27	**72.42**	72.39	**73.26**	71.36	**72.99**	**72.62**	70.33	72.62	**74.29**	73.32	**73.98**	71.23	**73.83**
CIFAR-100	40.23	**40.70**	42.52	**42.69**	**45.70**	44.11	**45.92**	45.36	43.32	**46.45**	**45.85**	42.93	**42.20**	42.04	41.01	**45.18**
STL-10	65.74	**65.77**	65.81	**65.89**	66.87	**67.03**	85.45	**86.07**	**80.07**	79.05	71.58	**72.76**	67.76	**74.95**	**67.38**	65.02
Aircraft	62.38	**68.13**	63.86	**73.08**	**69.34**	67.59	**63.76**	62.14	63.9	**73.9**	66.5	**72.75**	68.22	**65.69**	**64.3**	63.12
Flowers	**88.12**	85.19	86.35	**86.99**	89.19	**89.61**	87.37	**89.59**	87.97	**89.81**	**88.54**	88.25	85.09	**85.69**	85.86	**87.82**
Food	71.68	**74.2**	**77.23**	72.82	79.25	**79.56**	70.69	**72.78**	71.01	**77.87**	**74.55**	71.45	**78.03**	73.88	72.82	**76.82**
Cars	51.61	**54.26**	50.74	**53.05**	54.37	**54.87**	51.83	**54.85**	51.22	**51.91**	50.92	**50.64**	**52.29**	50.44	51.5	**53.51**
DTD	55.69	**56.06**	55.63	**57.18**	55.90	**57.12**	55.63	**56.06**	50.9	**53.77**	52.07	**53.24**	51.27	**52.63**	**54.07**	51.08
Average	63.12	**64.36**	64.18	**65.52**	66.62	**66.64**	66.50	**67.48**	65.13	**67.89**	65.33	**65.79**	64.77	**64.91**	63.52	**64.55**

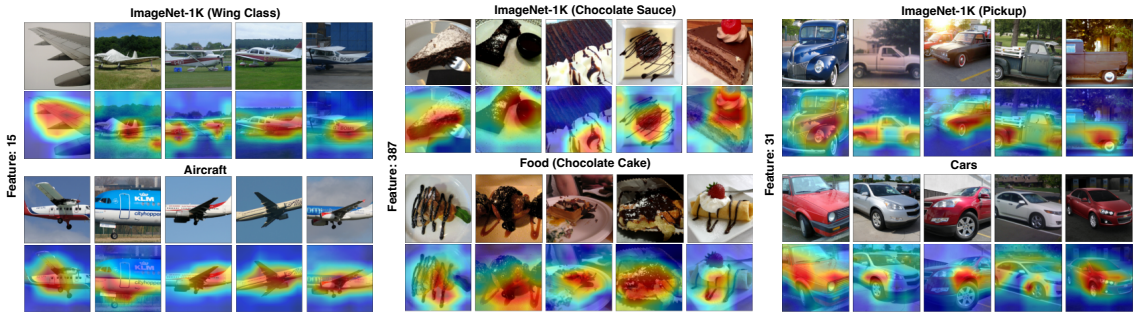


Figure A.2: **Discriminative features on unseen datasets:** We visualize the discriminative features discovered on ImageNet-1K classes on unseen datasets like Aircraft, Food and Cars. We observe that discriminative features correspond to the same physical attributes as the training data and are core and informative.

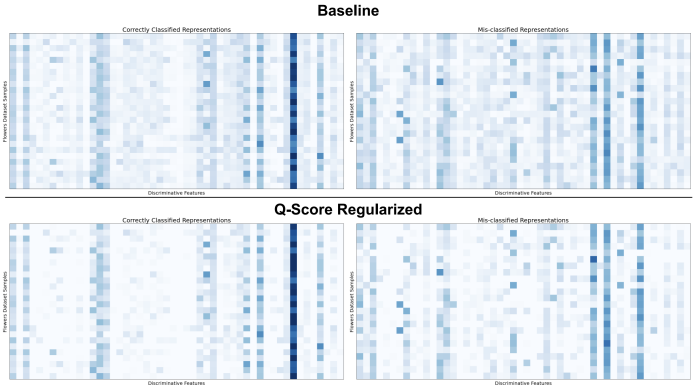


Figure A.3: **Comparing correct and mis-classified representations in Flowers dataset:** In these heatmaps, we visualize the discriminative features of several Flowers dataset samples. In the top panel, we display the correct (left) and incorrect (right) classifications of SimCLR (trained on ImageNet-1K) and in the bottom panel, we visualize the same when pre-trained using Q-Score regularization. Similar to the observations in Figure 3.3, we observe that the regularization enhances discriminative features, thereby leading to an improvement in performance.

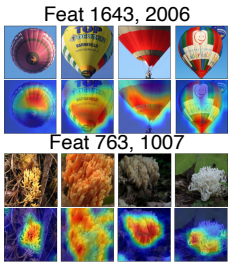


Figure A.4: **Feature groups:** Groups of features (non-axis-aligned) can still correspond to meaningful physical concepts that are associated with class labels, therefore, can also be discriminative.

A.4 Axis-Alignment and Principal Components

In our analysis, we select discriminative features independently and observe their heatmaps and activations across the population. Figure A.4 shows some examples of non-axis-aligned groups of features that can still correspond to meaningful concepts associated with class labels. With discriminative features, we attempt to collect all the conceptual information associated with class labels in the dataset. These concepts can be encoded by independent or groups of features and which strongly activate when the concept is present and can still lie in the middle portion of A.

To (partially) validate our selection method, we have also conducted a PCA analysis where we select principal components of feature representations and perform linear evaluation on top of them. In Figure 3.2, we observe that, until 40% reduction of the representation size, PCA and discriminative features perform comparably in terms of the linear classification accuracy while discriminative features significantly outperforms random features across the board. We also plot the gradients of the highly activating PCA features and compare them to discriminative features in the full representation space in Figure A.6. We observe that both sets of features activate the same portions of the images between both correct and incorrect classifications. These results indicate that discriminative features capture a fair amount of information in the feature representations and thus (partially) validating our underlying assumption.

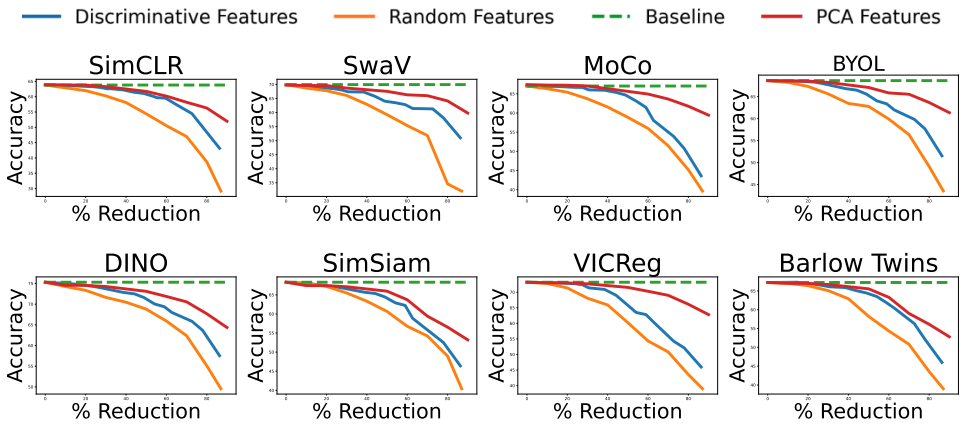


Figure A.5: **Linear classification accuracy on discriminative features:** Similar to Figure 3.2, we compare discriminative and random features to PCA features of matching sizes. Discriminative features also match the performance of PCA features to a certain extent showing that features can be considered as axis-aligned.

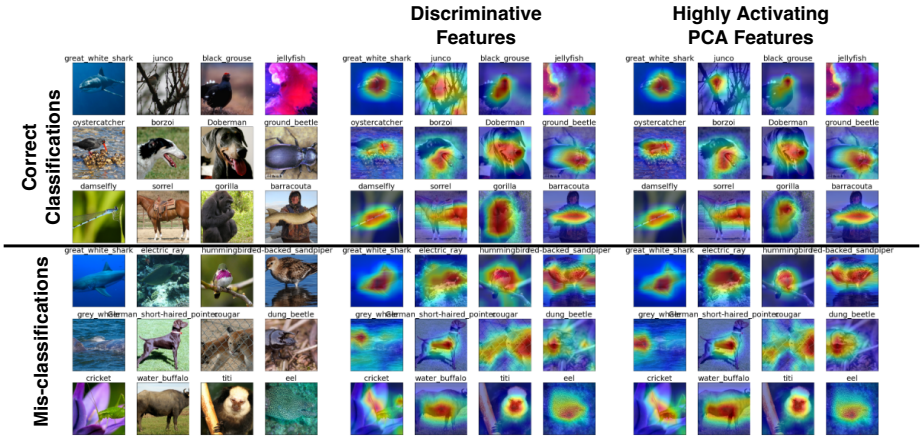


Figure A.6: **Comparing gradient heatmaps of discriminative features and PCA features:** In this figure, we plot the gradient heatmaps of the discriminative features of correct and incorrect classifications on ImageNet-1K trained on SimCLR. We also plot the discriminative PCA features for the same images. We observe that both sets of features activate the same portions of the images meaning that discriminative features can we viewed as axis-aligned.

A.5 Selecting Features from the Upper or Lower Tail of A

We discussed in the Discriminative Feature Section that we select discriminative features by increasing the lower limits (from 0 percentile) and decreasing the upper limits (from 100 percentile). In Figure A.7, we compare discriminative features with features selected from lower tail of A (lower percentile fixed at 0) and the upper tail (upper percentile fixed at 100). Our discriminative features outperform both selection methods up to 60% reduction in representation size.

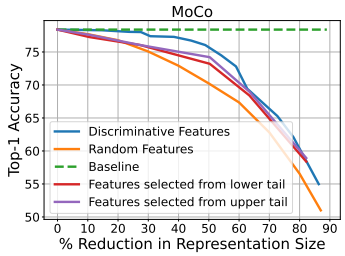


Figure A.7: **Selecting discriminative features:** We additionally include the results on MoCo, ImageNet-100 when selecting features either from the lower or upper tail of A which perform worse than the middle portion of A.

A.6 Ablation on Q-Score Loss Hyper-Parameters

In this section, we discuss how we can perform a hyper-parameter search on λ_1 and λ_2 to find the best performing pair of values. We take the baseline of SimCLR trained on ImageNet-1K and further train this model under the setup outlined in the Experimental Setup Section. We train keeping both $\lambda_1 = \lambda_2 = 0$ and run experiments by gradually increasing λ_1 to find the best performing value. Next, we search over λ_2 keeping the best performing value of λ_1. In these experiments we find that $\lambda_1 = \lambda_2 = 10^{-4}$ is the best performing pair. We find that this pair shows improved performance across most experiments. Due to lack of resources, we do not heavily tune these

hyper-parameters in our experiments, however, we can expect improved performance if tuning is performed.

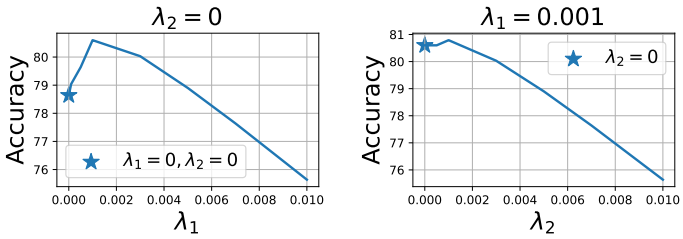


Figure A.8: **Hyper-parameter Search on λ_1 and λ_2:** We set $\lambda_2 = 0$ and search across various values of λ_1 to find the best performing experiment. Next, we set λ_1 to the best performing value and search over λ_2.

A.7 Q-Score on Supervised Learning

We note that we select discriminative features and compute Q-Score on self-supervised representations without using any label information. Thus, our study to show correlation between Q-score and classification outcome is non-trivial since self-supervised models learn without labels. Nevertheless, we have included an experiment in Figure A.9, where we analyze Q-score as a predictor of classification outcome (correct vs incorrect) on supervised ResNet-18 (ImageNet-100) and ResNet-50 (ImageNet-1K) representations as well as their robust versions (l2 threat model). Self-supervised representations generally perform better than supervised representations on Q-score indicating that the representational properties we have identified may be mainly prominent in self-supervised learning. We observe that non-robust supervised ResNet shows lower AUROC and AUPRC compared to robust ResNet on both ImageNet-100 and ImageNet-1K setups. This is in line with observations in [Eng+20] and [SF21] that show that robust models provide better axis-alignment of features.

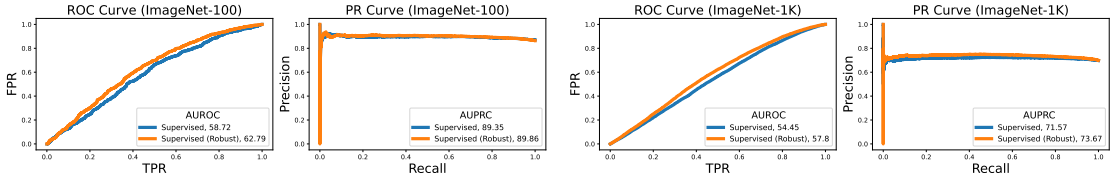


Figure A.9: **Precision-Recall and ROC curves of Q-Score on supervised setups:** In the first two plots, we compute the ROC and PR curves (similar to Figure 3.4) of Q-score on the representations of a supervised ResNet-18 model and a robust ResNet-18 trained on ImageNet-100. In the last two plots, we show the same for ResNet-50 trained on ImageNet-1K. We observe that robust ResNet performs better for Q-score when used as a predictor for correct or mis-classified representations.

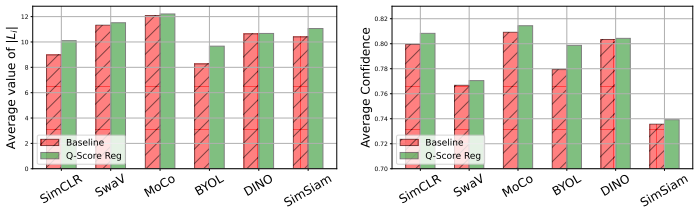


Figure A.10: **Average $|L_i|$ (left) and classification confidence (right) before and after regularization:** On the left we plot the average value of $|L_i|$ (number of highly activating features) and on the right we plot the average classification confidence over the population of ImageNet-1K. We observe that both the number of highly activating features and classification confidence consistently improve on every self-supervised baseline with Q-Score regularization. This improvement is due to the nature of Q-Score regularization which maximizes highly activating discriminative features over the course of pre-training leading to a higher number of such features and improved classification confidence.

A.8 Q-Score and Classification Confidence

In Figure A.10, we plot the mean of $|L_i|$ (left), i.e., number of highly activating features in the i^{th} sample, and the mean linear classification confidence (right) over the population for each self-supervised model pre-trained with and without Q-Score regularization. We observe an increase in the average number of highly activating features (L_i) and as a result, an improvement in classification confidence, due to more enhanced features.

A.9 More Gradient Heatmaps of SimCLR

In Figures A.11, A.12, A.13 and A.14, we plot more heatmaps of highly and lowly activating features of SimCLR for 4 different ImageNet-1K classes. We observe that the highly activating features correspond to unique physical properties that are correlated with the ground truth, whereas, lowly activating features, map to spurious portions that do not contribute to useful information.

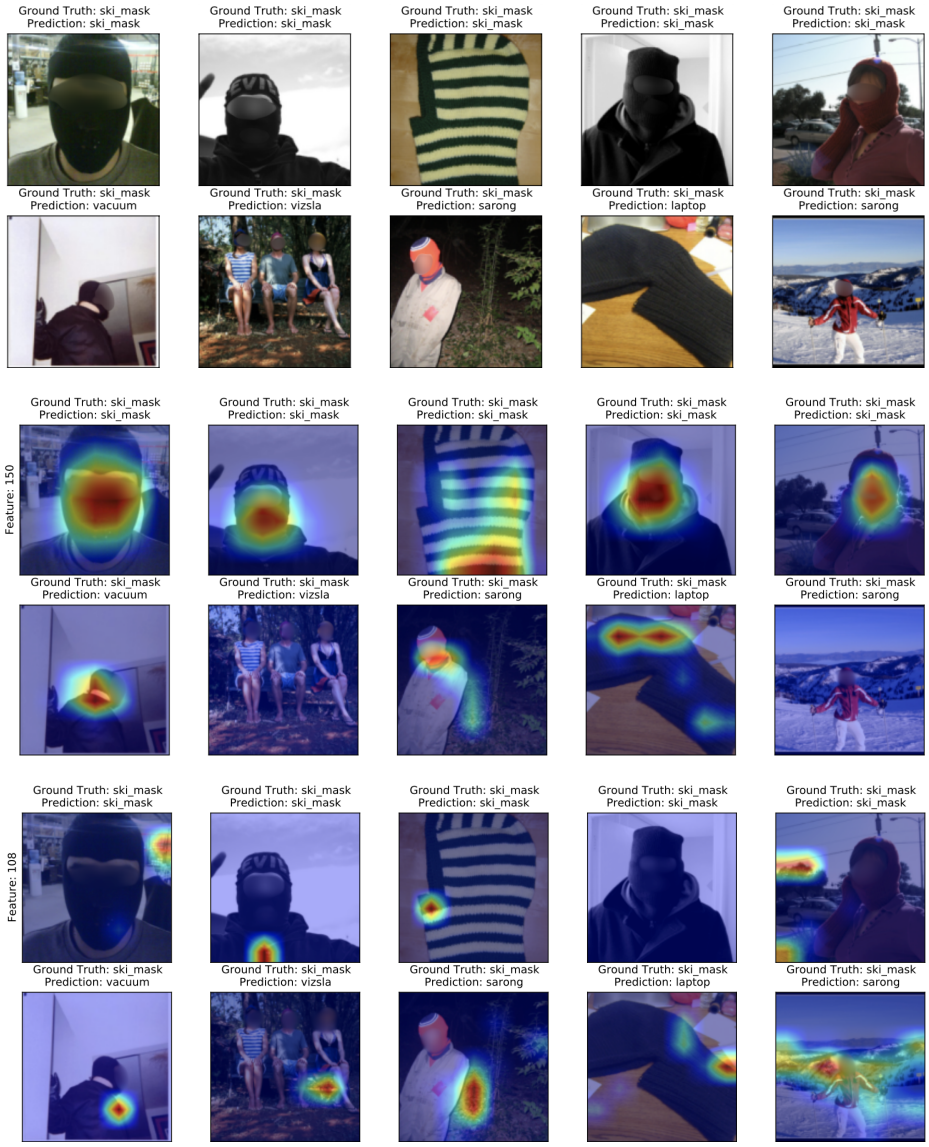


Figure A.11: **Heatmaps of discriminative and lowly activating features of SimCLR (Class - Ski Mask)**: We plot the gradient heat maps of the top activating discriminative feature (by magnitude) for the given class and a lowly activating feature of the same class. We observe that discriminative features are more correlated with ground truth labels compared to lowly activating features in both correct and incorrect classification. The discriminative feature in correct classifications correspond to a unique physical attribute that may not exist (or be obfuscated) in mis-classified images.

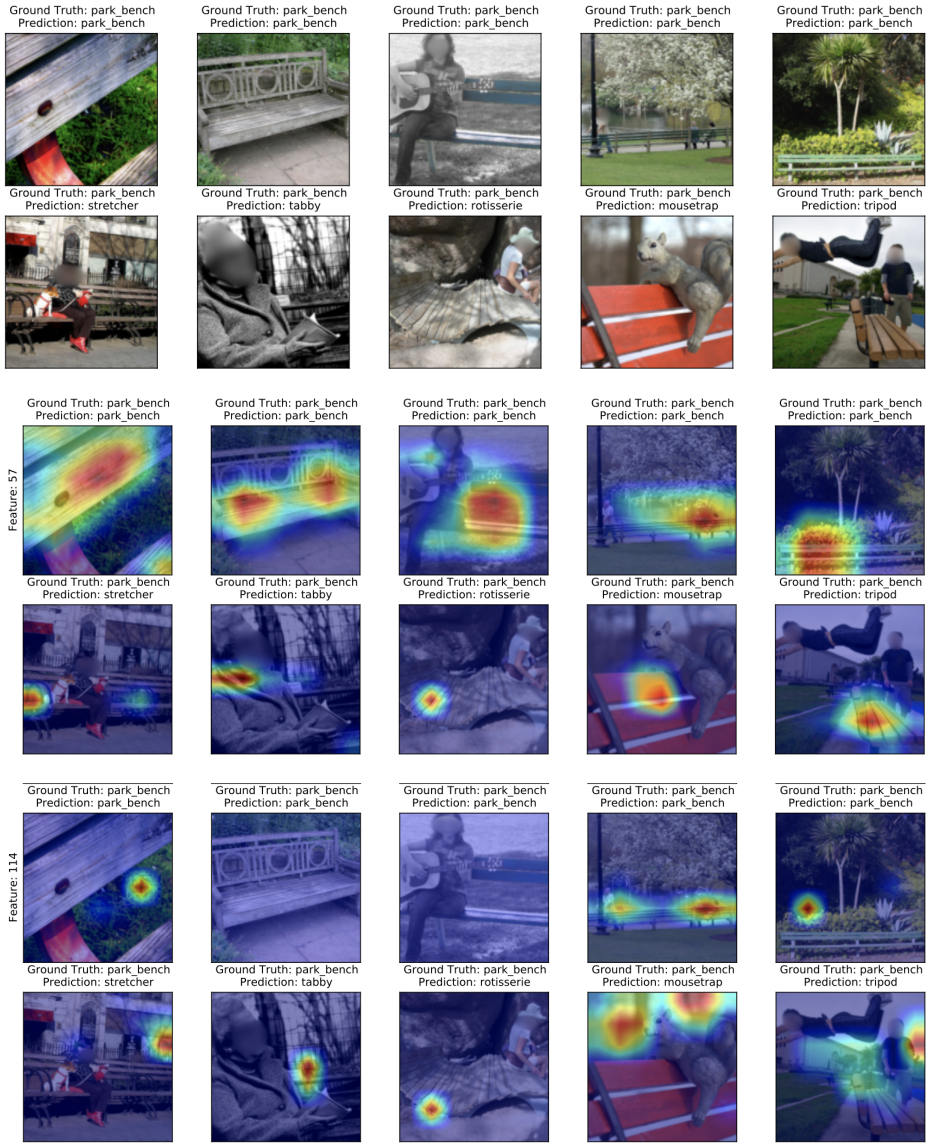


Figure A.12: **Heatmaps of discriminative and lowly activating features of SimCLR (Class - Park Bench)**: We plot the gradient heat maps of the top activating discriminative feature (by magnitude) for the given class and a lowly activating feature of the same class. We observe that discriminative features are more correlated with ground truth labels compared to lowly activating features in both correct and incorrect classification. The discriminative feature in correct classifications correspond to a unique physical attribute that may not exist (or be obfuscated) in mis-classified images.

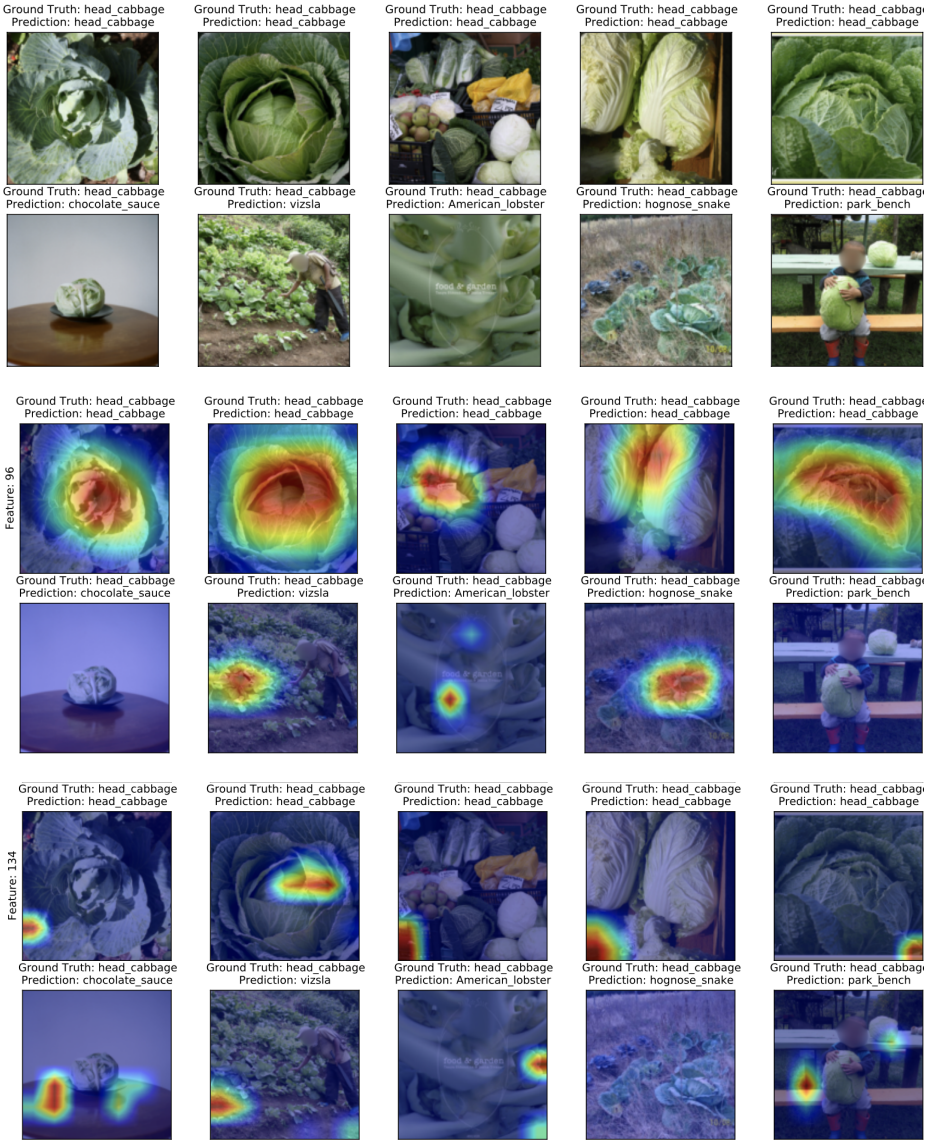


Figure A.13: **Heatmaps of discriminative and lowly activating features of SimCLR (Class - Head Cabbage)**: We plot the gradient heat maps of the top activating discriminative feature (by magnitude) for the given class and a lowly activating feature of the same class. We observe that discriminative features are more correlated with ground truth labels compared to lowly activating features in both correct and incorrect classification. The discriminative feature in correct classifications correspond to a unique physical attribute that may not exist (or be obfuscated) in mis-classified images.

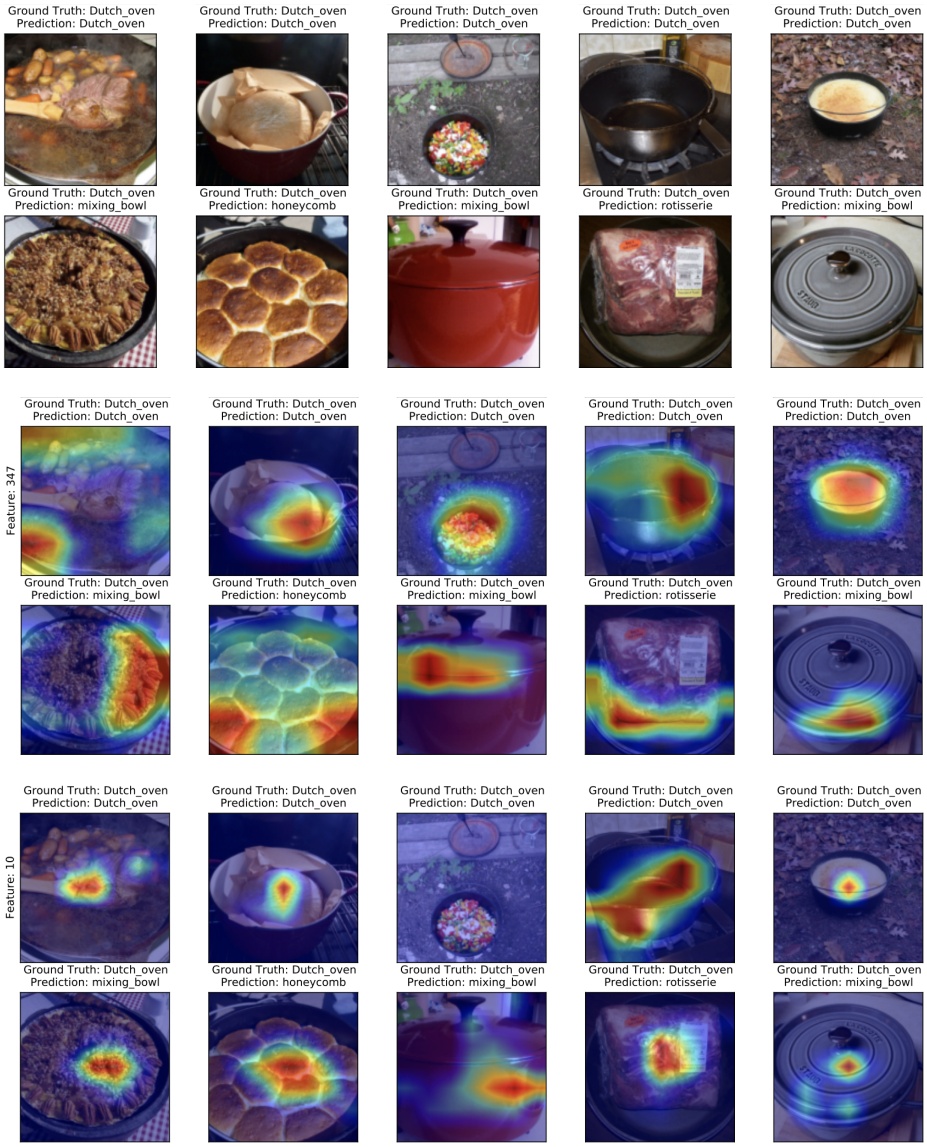


Figure A.14: **Heatmaps of discriminative and lowly activating features of SimCLR (Class - Dutch Oven)**: We plot the gradient heat maps of the top activating discriminative feature (by magnitude) for the given class and a lowly activating feature of the same class. We observe that discriminative features are more correlated with ground truth labels compared to lowly activating features in both correct and incorrect classification. The discriminative feature in correct classifications correspond to a unique physical attribute that may not exist (or be obfuscated) in mis-classified images.

Appendix B

Supplementary Material - Chapter 3

We use pre-trained models from the solo-learn package [Cos+22] and the official implementation of CLIP [Rad+21].

B.1 Analyzing FALCON Explanations Across Various Models

We have performed a global analysis comparing the FALCON concepts across various supervised and self-supervised models (ResNet-50 encoder). In Table B.1, we tabulate the number of interpretable feature groups identified from the final representation

Table B.1: **Feature groups and concepts for various models:** We tabulate the number of interpretable groups for each model and number of unique concepts extracted after explaining each group. We observe that many frequently occurring concepts are shared across models.

Model	# interpretable groups	# unique concepts	Most frequent concepts
SimSiam	249	578	'white', 'head', 'brown', 'eye', 'face'
SimCLR	293	676	'white', 'head', 'face', 'brown', 'blue'
MoCo	271	559	'white', 'head', 'face', 'eye', 'black'
SwaV	182	417	'head', 'brown', 'white', 'hand', 'black'
BYOL	281	477	'head', 'white', 'brown', 'eye', 'face'
ResNet-50 (Sup)	91	183	'brown', 'head', 'red', 'water', 'white'

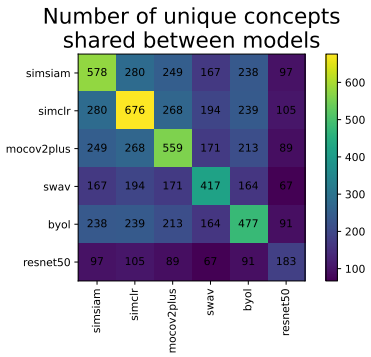


Figure B.1: **Shared concepts between models:** Among the unique concepts extracted using FALCON on the representation space of various models, we plot the number shared concepts between each pair of models.

layer, along with the total number of unique concepts extracted from FALCON for these groups. Note that each explanation consists of multiple conceptual words. In the last column, we also list the most frequently occurring concepts for each model. We observe that among all the models we study, the supervised ResNet-50 model has the least number of interpretable groups and unique concepts. The most frequent concepts among all the models are almost identical, including general attributes like various colors, face, eye, which frequently occur in the ImageNet dataset. We also compute the number of shared concepts between each pair of models in Figure B.1. We observe that each model shares roughly less than 50% of its total concepts with any other model. This indicates that although each model is trained on the same data i.e, ImageNet, their training paradigms can enable them to encode some unique properties that are missed by other models. We calculate the number of concepts in each model that are not shared with any other model - SimSiam 160, SimCLR 210, MoCo 159, SwaV 128, BYOL 116, ResNet-50 39. For example, these are some unshared (truly unique) concepts of ResNet-50 - ‘eel’, ‘disc’, ‘grip’, ‘shooter’, ‘tub’, ‘sink’, ‘weimaraner’, ‘decal’.

Table B.2: **Comparing FALCON used with CLIP and LAION-400M vs BLIP-2 zero-shot captioning:** We apply FALCON with BLIP 2 [Li+23] generated captions and ask participants to select the better explanation when compared with CLIP+LAION. BLIP captions underperform compared to CLIP+LAION.

Framework	% of times selected as best explanation
FALCON + CLIP + LAION	**58.12**
FALCON + BLIP 2 (OPT, caption COCO)	41.8

B.2 Employing a Captioning Model instead of CLIP

BLIP-2's [Li+23] zero-shot image captioning is a powerful tool to extract text captions out of highly activating images. One advantage of using a separate vocabulary with a vision-language model is the flexibility of controlling the expressiveness/specificity of the captioning dataset depending on the complexity of the target model. For example, to explain an MNIST-trained model, one may use a much smaller vocabulary whereas explaining a model like CLIP may require an equivalently large vocabulary. Moreover, the set of reference captions can be updated online, even after deployment without having to re-train any model. The similarity matrix allows us to extract multiple captions per image with a confidence score, allowing us to discard unreliable captions. Off the shelf captioning models may be domain-specific and could generate noisy captions with low expressiveness.

We compared FALCON + BLIP 2 with FALCON + CLIP + LAION in an MTurk evaluation over 91 features and asked participants to select the best describing explanation for the displayed set of images (See Table B.2. Explanations generated via our CLIP+LAION captioning outperforms BLIPs captioning, however, BLIP 2 is still a practical alternative given that it is trained on a large scale on LAION.

Table B.3: **Percentage of highly activating features in the ResNet-50 representation space:** For different model representations, we tabulate the percentage of features that activate at least 10 samples with a magnitude greater than α. We select α according to the mean of the distribution of the representation space (See Section 4.3 for more details).

Model	ResNet-50 (Supervised)	SimCLR	MoCo	DINO	BYOL	SimSiam	SwaV
% features that highly activate > 10 samples	0.68	21.92	17.70	16.66	25.63	8.00	6.00
α	0.27	0.34	0.34	0.14	0.24	0.32	0.31

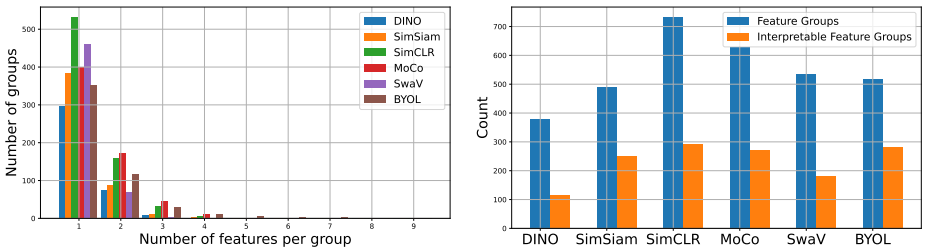


Figure B.2: **Distribution of feature groups:** For different self-supervised model representation spaces we compute the feature groups (from Algorithm 1. On the left we plot group sizes against the number of groups and on the right, we plot the number of interpretable groups among the discovered feature groups.

B.3 Interpretable Features in Various Models

We discuss in Section 4.3 that to discover potentially explainable features we can apply a strong value for α in $\mathcal{T}_i$, set of highly activating images. Since the distribution of each model representation space can be different, to be consistent we select $\alpha = mean(\mathbf{H}) + 16 \times std(\mathbf{H})$ (where $\mathbf{H}$ is the representation matrix). In Figure B.3, we tabulate the percentage of highly activating features in the final-layer representations where $|\mathcal{T}_i| > 10$. ResNet-50 has a particularly low number of highly activating features compared to self-supervised baselines. The remaining features in the representation space (or by relaxing α to be less rigorous), do not activate a resembling set of images, making such features harder to explain (Figure 4.4). Some more examples of such features are shown in Figure B.3.

We also discussed in Section 4.3 that simply thresholding by α does not guarantee explainability as the top activating images can still be unrelated. A larger portion of

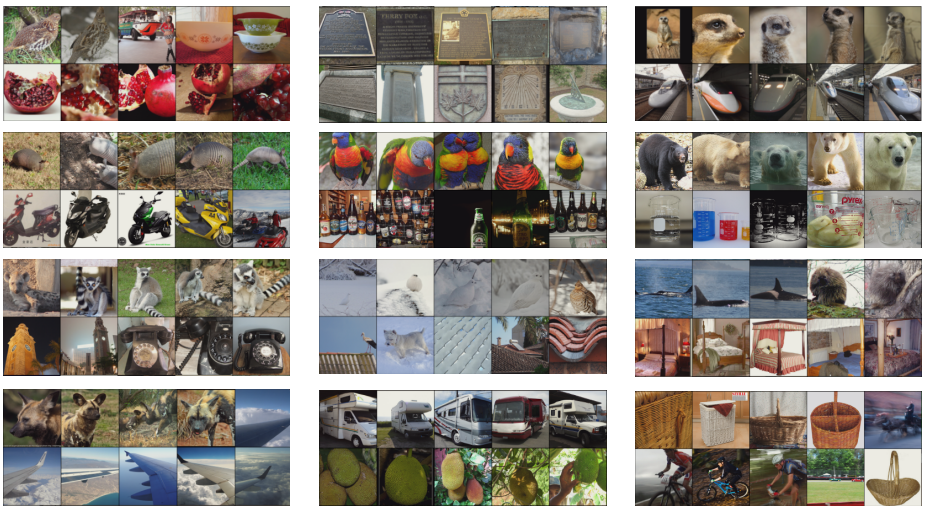


Figure B.3: **Examples of top activating images of some un-explainable independent features:** We provide more examples of top activating images of some independent features of DINO [Car+21] (ResNet-50 [He+16]) representations. The image sets are not correlated in any sense, making it hard to discover shared concepts for these features.

the representation space can be explained with feature groups. Using the Algorithm 1, we discover feature groups and interpretable feature groups for various self-supervised models. In Figure B.2, on the left, we show the distribution of feature groups and their size. All the identified groups contain at least 10 highly activating images. A large percentage of feature groups contain 1-2 features per group, however, there also exist feature groups that contain up to 9 features. On the right, we compare the feature groups and the *interpretable* feature groups, according to Algorithm 1. The interpretable groups activate samples that are more similar (based on CLIP cosine similarity) and are therefore easy to explain with shared natural language concepts.

B.4 Human Study to Evaluate Concepts

Eliminating malicious and inadequate responses: In our studies, we only select participants that have a HIT approval rate of greater than 90 and the number of HIT approvals is $>$ 500 in the past. Each task is active for 30 minutes allowing

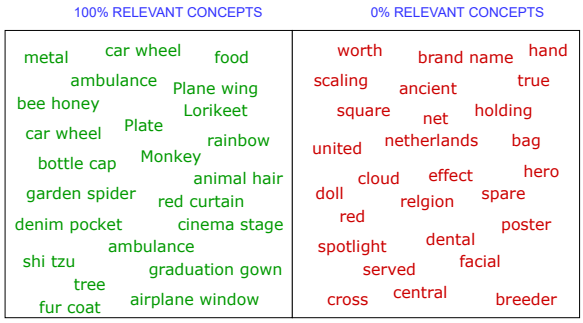


Figure B.4: **Comparing most and least relevant concepts based on AMT study:** We display the concepts with 100% relevancy agreement on the left and the concepts with 0% relevancy agreement on the right.

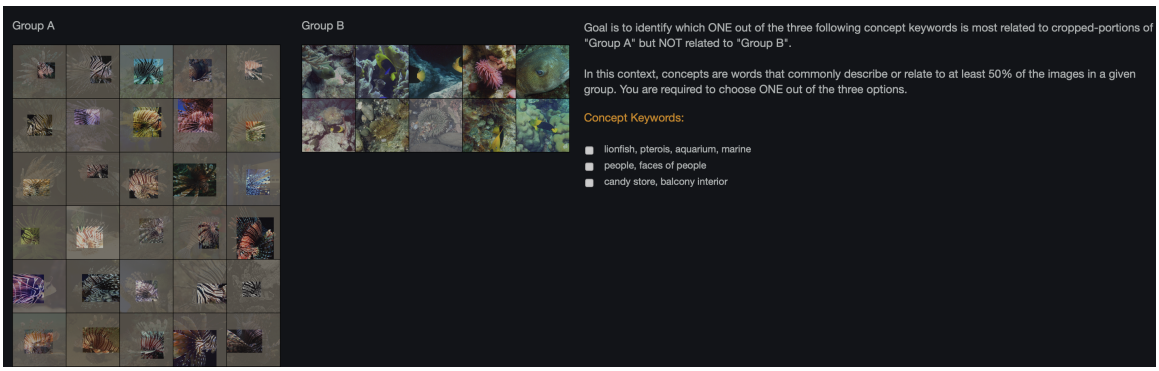


Figure B.5: **Amazon Mechanical Turk user study template:** A template of our user study where we display two groups of images for a target feature and ask the users to select the best explanation among 3 options.

the participants ample time to make their selections. We did not explicitly include control questions, however, we identified a small number of tasks which had low-quality concept sets which we used to verify the reliability of the participants. We also approve and pay the participants only after verifying their annotation quality.

In Figure B.5, we show a template of our user study where we display two groups i.e., highly activated cropped images (Group A) and lowly activating images (Groups B). In this example, we compare FALCON concepts to that of MILAN and Net-Dissect. As discussed in Section 4.4, we also evaluate top 6 FALCON concepts on their relevancy. We define the *agreement of relevancy* between workers as the percentage of workers that believe a concept is relevant. This, averaged for all concepts in a feature, is plotted in Figure B.6. We observe that, for 93 features, up to 86% of them

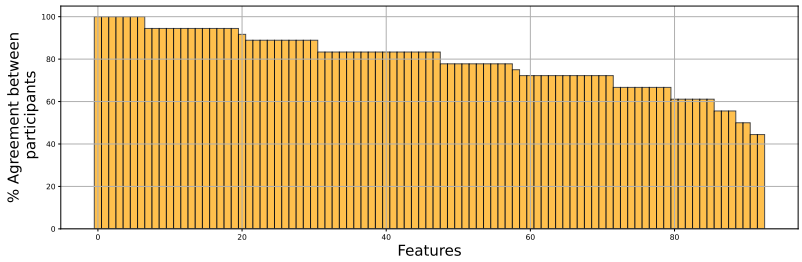


Figure B.6: **Average agreement between participants for each feature:** We plot the agreement of relevancy for each concept averaged by the feature for 93 features we perform human study on.

are agreed to be relevant among at least 66% of the workers. We also visualize the concepts where the agreement of relevancy is 100% (left) and 0% (right) in Figure B.4. We observed that the irrelevant concepts have a very low average CLIP score of 0.067. This is likely because there were other, more specific concepts for that feature, or the concepts were out-of-context for the displayed images. In contrast, the concepts with 100% relevancy have a relatively higher average CLIP score of 0.284 (unsurprisingly) and are strongly correlated with the displayed images.

B.5 Transferring Concepts to Unseen Data

In Section 4.6, we study a non-trivial setup of transferring concepts from one interpretable model to another. In this Section we study a simpler scenario of transferring concepts to unseen datasets. Essentially, we evaluate if our extracted concepts (on ImageNet validation data), generalizes to new datasets. In Figure B.7, for several DINO features, we display the highly (cropped) and lowly activating images, as well as the highly activating images in STL-10 [CLN] which is an unseen dataset. We extract concepts using FALCON and MILAN to compare the quality. We observe that STL-10 images for each feature closely resemble that of ImageNet and more importantly, correlate with most of FALCON concepts. FALCON also generally provides more explicit concepts covering multiple aspects, compared to MILAN. This confirms that

extracted concepts generalize well to unseen or unknown data.

B.6 Explaining Supervised Representations and Early-Layer Features

To further confirm the generalizability of our concept extraction framework, FALCON, we extract concepts from different layers of supervised pre-trained ResNet-50 (using ImageNet and LAION). Our results are shown in Figure B.8. We observe the initial layer features, activate very primitive type concepts like color or geometric patters. FALCON extracts this information in its concepts based on the cropped images. As we move closer to the final layer, the feature crops become larger and concepts become more descriptive. We thus confirm that FALCON can be applied to explain any neuron in any vision model, supervised or unsupervised.

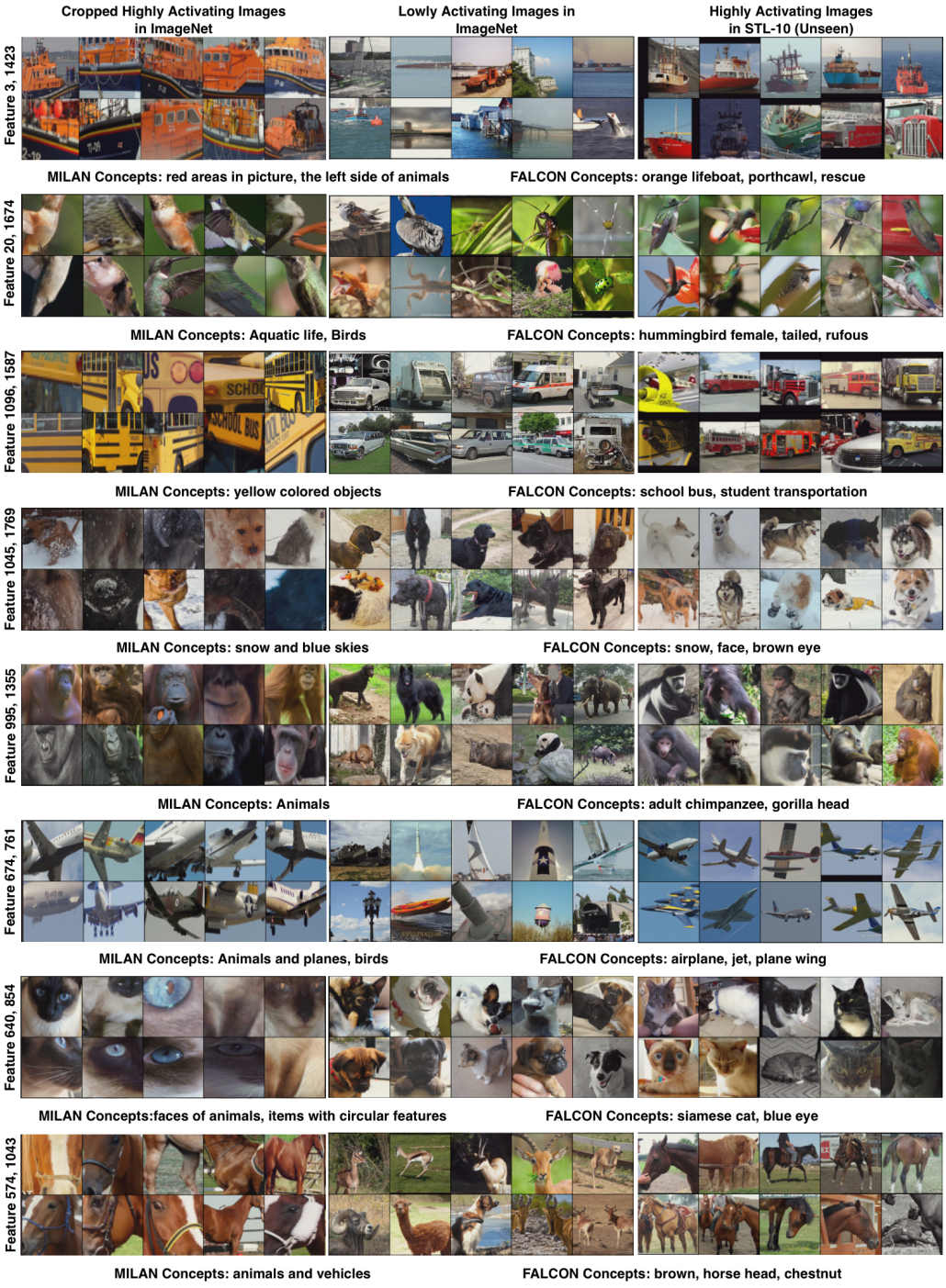


Figure B.7: **Generalization of concepts to unseen data:** We extract concepts from various features of DINO [Car+21] representations (using ImageNet) and verify if they generalize to STL-10 [CLN], an unseen dataset. In all features, the STL-10 images closely resemble the ImageNet images and contain the concepts described by FALCON.

Figure B.8: **Concepts for features of various layers of supervised ResNet-50:** We extract concepts from random features of layers of supervised pre-trained ResNet-50. We compare FALCON concepts with MILAN concepts.

Appendix C

Supplementary Material - Chapter 4

	SimCLR	**BYOL**	**MoCov2**	**SimSiam**
Encoder	ResNet-50	ResNet-50	ResNet-50	ResNet-50
Zero init residual	False	False	False	True
Projection model features	MLP (4096, 256)	MLP (4096, 256)	MLP (4096, 256)	MLP (2048, 2048, 2048)
Prediction model features	N/A	MLP (4096, 256)	N/A	MLP (512, 2048)
Momentum encoder (for right encoder)	False	True	True	True
Stop-grad (for right encoder)	False	True	True	True
Contrastive loss temperature	0.1	N/A	0.1	N/A
Optimizer	LARS	LARS	LARS	LARS
Learning rate	0.2	0.2	0.2	0.2
Weight decay	1.5×10^{-6}	1.5×10^{-6}	1.5×10^{-6}	1.5×10^{-6}
Learning rate schedule	cosine decay	cosine decay	cosine decay	cosine decay
Epochs	1000	1000	1000	1000
Linear probe epochs	90	90	90	90
Linear probe learning rate	0.3	0.3	0.3	0.3
Linear probe optimizer	SGD	SGD	SGD	SGD
Linear probe learning rate schedule	cosine decay	cosine decay	cosine decay	cosine decay

Table C.1: **Training setup for each model**: We provide the specific architecture and training setup for each encoder for reproducibility.

Augmentation	**Hyper-parameter Values**	**Probability (Left View)**	**Probability (Right View)**
Random Resized Crop	224×224, min area: 0.08, max area: 1.0, min aspect: 3/4, max aspect: 4. / 3., aspect dist: log, resize method: bicubic	1.0	1.0
Color jitter	contrast: 0.4, brightness: 0.4, saturation: 0.2, hue: 0.1	0.8	0.8
Grayscale	N/A	0.2	0.2
Horizontal flip	N/A	0.5	0.5
Gaussian blur	min sigma: 0.1, max sigma: 2.0, kernel size: 23	1.0	0.1
Amplitude rescale	$m = 0.8$, $n = 1.75$	0.2	0.0
Phase shift	$p = 0.4$, $q = 0.7$	0.2	0.0
Random frequency mask	$k \sim [0.01, 0.1)$	0.5	0.0
Gaussian mixture mask	$c = 20$, $\sigma \sim [10, 15)$	0.2	0.0

Table C.2: **Augmentation hyperparameters:** We provide the parameters used for each augmentation, both image and FDA along with the probability.

C.1 Training Setup

We provide all our implementation details for each baseline - SimCLR, BYOL, MoCov2 and SimSiam in Table C.1. We also include linear probing hyperparameters for full reproducibility.

C.2 Augmentation Hyperparameters

We provide the parameters used for each image and FMA augmentation along with the probability in the left and right view in Table C.2.

Bibliography

[Aga18] Abien Fred Agarap. *Deep Learning using Rectified Linear Units (ReLU)*. 2018. arXiv: 1803.08375 [cs.NE].

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. *Wasserstein GAN*. 2017. arXiv: 1701.07875 [stat.ML].

[Aro+19] Sanjeev Arora et al. *A Theoretical Analysis of Contrastive Unsupervised Representation Learning*. 2019. arXiv: 1902.09229 [cs.LG].

[BHB19] Philip Bachman, R Devon Hjelm, and William Buchwalter. "Learning Representations by Maximizing Mutual Information Across Views". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/ddf354219aac374f1d40b7e760ee5bb7-Paper.pdf.

[Bae+20] Alexei Baevski et al. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. 2020. arXiv: 2006.11477 [cs.CL].

[Bal+23] Randall Balestriero et al. *A Cookbook of Self-Supervised Learning*. 2023. arXiv: 2304.12210 [cs.LG].

[BPL21] Adrien Bardes, Jean Ponce, and Yann LeCun. *VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning*. 2021. arXiv: 2105.04906 [cs.CV].

[Bau+17a] David Bau et al. *Network Dissection: Quantifying Interpretability of Deep Visual Representations*. 2017. arXiv: 1704.05796 [cs.CV].

[Bau+17b] David Bau et al. "Network Dissection: Quantifying Interpretability of Deep Visual Representations". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). DOI: 10.1109/cvpr.2017.354. URL: http://dx.doi.org/10.1109/CVPR.2017.354.

[Bau+16] Miguel A Bautista et al. "CliqueCNN: Deep Unsupervised Exemplar Learning". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/file/65fc52ed8f88c81323a418ca94cec2ed-Paper.pdf.

[BJ17] Piotr Bojanowski and Armand Joulin. "Unsupervised Learning by Predicting Noise". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 517–526. URL: http://proceedings.mlr.press/v70/bojanowski17a.html.

[BBV21] Florian Bordes, Randall Balestriero, and Pascal Vincent. *High Fidelity Visualization of What Your Self-Supervised Representation Knows About*. 2021. arXiv: 2112.09164 [cs.LG].

[BGV14a] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. "Food-101 – Mining Discriminative Components with Random Forests". In: *European Conference on Computer Vision*. 2014.

[BGV14b] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. "Food-101 – Mining Discriminative Components with Random Forests". In: *European Conference on Computer Vision*. 2014.

[BM67] E. O. Brigham and R. E. Morrow. "The fast Fourier transform". In: *IEEE Spectrum* 4.12 (1967), pp. 63–70. DOI: 10.1109/MSPEC.1967.5217220.

[Cai+21] Mu Cai et al. "Frequency Domain Image Translation: More Photo-realistic, Better Identity-preserving". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021). DOI: 10.1109/iccv48922.2021.01367. URL: http://dx.doi.org/10.1109/ICCV48922.2021.01367.

[Car+18a] Mathilde Caron et al. "Deep Clustering for Unsupervised Learning of Visual Features". In: *Lecture Notes in Computer Science* (2018), pp. 139–156. ISSN: 1611-3349. DOI: 10.1007/978-3-030-01264-9_9. URL: http://dx.doi.org/10.1007/978-3-030-01264-9_9.

[Car+18b] Mathilde Caron et al. "Deep Clustering for Unsupervised Learning of Visual Features". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[Car+19] Mathilde Caron et al. "Unsupervised Pre-Training of Image Features on Non-Curated Data". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[Car+20a] Mathilde Caron et al. *Unsupervised Learning of Visual Features by Contrasting Cluster Assignments*. 2020. arXiv: 2006.09882 [cs.CV].

[Car+20b] Mathilde Caron et al. "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 9912–9924. URL: https://proceedings.neurips.cc/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

[Car+21] Mathilde Caron et al. "Emerging Properties in Self-Supervised Vision Transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9650–9660.

[Che+20a] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1597–1607. URL: http://proceedings.mlr.press/v119/chen20j.html.

[CH21] Xinlei Chen and Kaiming He. "Exploring Simple Siamese Representation Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 15750–15758.

[Che+20b] Xinlei Chen et al. *Improved Baselines with Momentum Contrastive Learning*. 2020. arXiv: 2003.04297 [cs.CV].

[Che+21] Zhiyang Chen et al. "DPT: Deformable Patch-based Transformer for Visual Recognition". In: *Proceedings of the 29th ACM International Conference on Multimedia*. MM '21. ACM, Oct. 2021. DOI: 10.1145/3474085.3475467. URL: http://dx.doi.org/10.1145/3474085.3475467.

[Cim+14] M. Cimpoi et al. "Describing Textures in the Wild". In: *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[CLN] Adam Coates, Honglak Lee, and Andrew Y. Ng. "Stanford STL-10 Image Dataset". In: (). URL: https://cs.stanford.edu/~acoates/stl10/.

[Cos+22] Victor Guilherme Turrisi da Costa et al. "solo-learn: A Library of Self-supervised Methods for Visual Representation Learning". In: *Journal of Machine Learning Research* 23.56 (2022), pp. 1–6. URL: http://jmlr.org/papers/v23/21-1155.html.

[DSB17] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. *Density estimation using Real NVP*. 2017. arXiv: 1605.08803 [cs.LG]. URL: https://arxiv.org/abs/1605.08803.

[Dos+14] Alexey Dosovitskiy et al. "Discriminative Unsupervised Feature Learning with Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper/2014/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf.

[Dos+16] Alexey Dosovitskiy et al. "Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.9 (2016), pp. 1734–1747. DOI: 10.1109/TPAMI.2015.2496141.

[Dos+21] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].

[Dwi+21] Debidatta Dwibedi et al. "With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021). DOI: 10.1109/iccv48922.2021.00945. URL: http://dx.doi.org/10.1109/iccv48922.2021.00945.

[Elh+22] Nelson Elhage et al. *Toy Models of Superposition*. 2022. arXiv: 2209.10652 [cs.LG].

[Eng+20] Logan Engstrom et al. *Adversarial Robustness as a Prior for Learned Representations*. 2020. URL: https://openreview.net/forum?id=rygvFyrKwH.

[EGH21a] Linus Ericsson, Henry Gouk, and Timothy M. Hospedales. *Why Do Self-Supervised Models Transfer? Investigating the Impact of Invariance on Downstream Tasks*. 2021. arXiv: 2111.11398 [cs.CV].

[EGH21b] Linus Ericsson, Henry G. R. Gouk, and Timothy M. Hospedales. "Why Do Self-Supervised Models Transfer? Investigating the Impact of Invariance on Downstream Tasks". In: *ArXiv* abs/2111.11398 (2021). URL: https://api.semanticscholar.org/CorpusID:260495154.

[FXT19] Zeyu Feng, Chang Xu, and Dacheng Tao. "Self-Supervised Representation Learning From Multi-Domain Data". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 3244–3254.

[FV18] Ruth Fong and Andrea Vedaldi. "Net2Vec: Quantifying and Explaining How Concepts are Encoded by Filters in Deep Neural Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018). DOI: 10.1109/cvpr.2018.00910. URL: http://dx.doi.org/10.1109/CVPR.2018.00910.

[Gar+22] Quentin Garrido et al. *RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank*. 2022. arXiv: 2210.02885 [cs.LG].

[Gho+19] Amirata Ghorbani et al. *Towards Automatic Concept-based Explanations*. 2019. arXiv: 1902.03129 [stat.ML].

[GSK18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. "Unsupervised Representation Learning by Predicting Image Rotations". In: *ArXiv* abs/1803.07728 (2018). URL: https://api.semanticscholar.org/CorpusID:4009713.

[Gri+21] Tom George Grigg et al. *Do Self-Supervised and Supervised Methods Learn Similar Visual Representations?* 2021. arXiv: 2110.00528 [cs.CV].

[Gri+20] Jean-Bastien Grill et al. "Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 21271–21284. URL: https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

[Gul+17] Ishaan Gulrajani et al. *Improved Training of Wasserstein GANs*. 2017. arXiv: 1704.00028 [cs.LG].

[Han+22] Kai Han et al. *Vision GNN: An Image is Worth Graph of Nodes*. 2022. arXiv: 2206.00272 [cs.CV].

[HH07] Bruce C Hansen and Robert F Hess. "Structural sparseness and spatial phase alignment in natural scenes". In: *JOSA A* 24.7 (2007), pp. 1873–1885.

[HW79] J. A. Hartigan and M. A. Wong. "A k-means clustering algorithm". In: *JSTOR: Applied Statistics* 28.1 (1979), pp. 100–108.

[He+15] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[He+16] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[He+18] Kaiming He et al. *Mask R-CNN*. 2018. arXiv: 1703.06870 [cs.CV].

[He+20] Kaiming He et al. "Momentum Contrast for Unsupervised Visual Representation Learning". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 9726–9735. DOI: 10.1109/CVPR42600.2020.00975.

[He+21] Kaiming He et al. *Masked Autoencoders Are Scalable Vision Learners*. 2021. arXiv: 2111.06377 [cs.CV].

[He+22] Kaiming He et al. "Masked Autoencoders Are Scalable Vision Learners". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022). DOI: 10.1109/cvpr52688.2022.01553. URL: http://dx.doi.org/10.1109/CVPR52688.2022.01553.

[HJB84] M. Heideman, D. Johnson, and C. Burrus. "Gauss and the history of the fast fourier transform". In: *IEEE ASSP Magazine* 1.4 (1984), pp. 14–21. DOI: 10.1109/MASSP.1984.1162257.

[HMD19] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. *Deep Anomaly Detection with Outlier Exposure*. 2019. arXiv: 1812.04606 [cs.LG]. URL: https://arxiv.org/abs/1812.04606.

[Her+22a] Evan Hernandez et al. *Natural Language Descriptions of Deep Visual Features*. 2022. arXiv: 2201.11114 [cs.CV].

[Her+22b] Evan Hernandez et al. *Natural Language Descriptions of Deep Visual Features*. 2022. arXiv: 2201.11114 [cs.CV].

[Hor+18] Grant Van Horn et al. *The iNaturalist Species Classification and Detection Dataset*. 2018. arXiv: 1707.06642 [cs.CV].

[Hua+19] Jiabo Huang et al. "Unsupervised Deep Learning by Neighbourhood Discovery". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2849–2858. URL: http://proceedings.mlr.press/v97/huang19b.html.

[HYZ21] Weiran Huang, Mingyang Yi, and Xuyang Zhao. *Towards the Generalization of Contrastive Self-Supervised Learning*. 2021. arXiv: 2111.00743 [cs.LG].

[Jin+21] Li Jing et al. "Understanding Dimensional Collapse in Contrastive Self-supervised Learning". In: *ArXiv* abs/2110.09348 (2021).

[Jin+22] Li Jing et al. "Understanding Dimensional Collapse in Contrastive Self-supervised Learning". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=YevsQ05DEN7.

[Kal+22] Neha Kalibhat et al. *Measuring Self-Supervised Representation Quality for Downstream Classification using Discriminative Features*. 2022. arXiv: 2203.01881 [cs.LG].

[Kal+23] Neha Kalibhat et al. *Identifying Interpretable Subspaces in Image Representations*. 2023. arXiv: 2307.10504 [cs.CV].

[KLF22] Priyatham Kattakinda, Alexander Levine, and Soheil Feizi. *Invariant Learning via Diffusion Dreamed Distribution Shifts*. 2022. arXiv: 2211.10370 [cs.CV].

[Kho+20] Prannay Khosla et al. *Supervised Contrastive Learning*. 2020. arXiv: 2004.11362 [cs.LG].

[Kim+20] Donghyun Kim et al. *Cross-domain Self-supervised Learning for Domain Adaptation with Few Source Labels*. 2020. arXiv: 2003.08264 [cs.CV].

[Kim+21] Donghyun Kim et al. "CDS: Cross-Domain Self-Supervised Pre-Training". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9123–9132.

[Kin+23] Diederik P. Kingma et al. *Variational Diffusion Models*. 2023. arXiv: 2107.00630 [cs.LG]. URL: https://arxiv.org/abs/2107.00630.

[KW17] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG].

[KIW20] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. *Why Normalizing Flows Fail to Detect Out-of-Distribution Data*. 2020. arXiv: 2006.08545 [stat.ML]. URL: https://arxiv.org/abs/2006.08545.

[Kir+19] Alexander Kirillov et al. *Panoptic Segmentation*. 2019. arXiv: 1801.00868 [cs.CV].

[Kir+23] Alexander Kirillov et al. *Segment Anything*. 2023. arXiv: 2304.02643 [cs.CV].

[Koh+21] Pang Wei Koh et al. *WILDS: A Benchmark of in-the-Wild Distribution Shifts*. 2021. arXiv: 2012.07421 [cs.LG].

[KZB19] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. "Revisiting Self-Supervised Visual Representation Learning". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 1920–1929. DOI: 10.1109/CVPR.2019.00202.

[KTP21] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. "Mean Shift for Self-Supervised Learning". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021). DOI: 10.1109/iccv48922.2021.01016. URL: http://dx.doi.org/10.1109/iccv48922.2021.01016.

[Kot+22] Divya Kothandaraman et al. *FAR: Fourier Aerial Video Recognition*. 2022. arXiv: 2203.10694 [cs.CV].

[Kra+13] Jonathan Krause et al. "3D Object Representations for Fine-Grained Categorization". In: *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia, 2013.

[Kri+16] Ranjay Krishna et al. *Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations*. 2016. arXiv: 1602.07332 [cs.CV].

[KNHa] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "CIFAR-10 (Canadian Institute for Advanced Research)". In: (). URL: http://www.cs.toronto.edu/~kriz/cifar.html.

[KNHb] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "CIFAR-100 (Canadian Institute for Advanced Research)". In: (). URL: http://www.cs.toronto.edu/~kriz/cifar.html.

[Küg+21] J. von Kügelgen* et al. "Self-supervised learning with data augmentations provably isolates content from style". In: *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*. *equal contribution. Dec. 2021.

[LY15] Ya Le and X. Yang. "Tiny ImageNet Visual Recognition Challenge". In: 2015.

[Lee+21] Hankook Lee et al. *Improving Transferability of Representations via Augmentation-Aware Self-Supervision*. 2021. arXiv: 2111.09613 [cs.LG].

[LEP22] Alexander C. Li, Alexei A. Efros, and Deepak Pathak. *Understanding Collapse in Non-Contrastive Siamese Representation Learning*. 2022. arXiv: 2209.15007 [cs.LG].

[Li+21] Bo Li et al. *Invariant Information Bottleneck for Domain Generalization*. 2021. arXiv: 2106.06333 [cs.LG].

[Li+17] Da Li et al. "Deeper, Broader and Artier Domain Generalization". In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017). DOI: 10.1109/iccv.2017.591. URL: http://dx.doi.org/10.1109/ICCV.2017.591.

[Li+22] Junnan Li et al. *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. 2022. arXiv: 2201.12086 [cs.CV].

[Li+23] Junnan Li et al. *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. 2023. arXiv: 2301.12597 [cs.CV].

[Li+19] Liunian Harold Li et al. *VisualBERT: A Simple and Performant Baseline for Vision and Language*. 2019. arXiv: 1908.03557 [cs.CV].

[Lin+15] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].

[Liu+21] Ze Liu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. arXiv: 2103.14030 [cs.CV].

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2015. arXiv: 1411.4038 [cs.CV].

[LH19] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

[Ma+24] Chuofan Ma et al. *Groma: Localized Visual Tokenization for Grounding Multimodal Large Language Models*. 2024. arXiv: 2404.13013 [cs.CV].

[MH08] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

[Maj+13] S. Maji et al. *Fine-Grained Visual Classification of Aircraft*. Tech. rep. 2013. arXiv: 1306.5151 [cs-cv].

[Mar+15] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[MA20] Jesse Mu and Jacob Andreas. *Compositional Explanations of Neurons*. 2020. arXiv: 2006.14032 [cs.LG].

[NZ08] Maria-Elena Nilsback and Andrew Zisserman. "Automated Flower Classification over a Large Number of Classes". In: *Indian Conference on Computer Vision, Graphics and Image Processing*. 2008.

[NF16] Mehdi Noroozi and Paolo Favaro. "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles". In: *ArXiv* abs/1603.09246 (2016). URL: https://api.semanticscholar.org/CorpusID:187547.

[OW22] Tuomas Oikarinen and Tsui-Wei Weng. *CLIP-Dissect: Automatic Description of Neuron Representations in Deep Vision Networks*. 2022. arXiv: 2204.10965 [cs.CV].

[OMS17] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. "Feature Visualization". In: *Distill* (2017). https://distill.pub/2017/feature-visualization. DOI: 10.23915/distill.00007.

[Ola+20] Chris Olah et al. “Zoom In: An Introduction to Circuits”. In: *Distill* (2020). https://distill.pub/2020/circuits/zoom-in. DOI: 10.23915/distill.00024.001.

[OLV18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. 2018. arXiv: 1807.03748 [cs.LG].

[OLV19] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. 2019. arXiv: 1807.03748 [cs.LG].

[Oor+16] Aaron van den Oord et al. *Conditional Image Generation with PixelCNN Decoders*. 2016. arXiv: 1606.05328 [cs.CV]. URL: https://arxiv.org/abs/1606.05328.

[Opp+79] A. Oppenheim et al. “Phase in speech and pictures”. In: *ICASSP '79. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 4. 1979, pp. 632–637. DOI: 10.1109/ICASSP.1979.1170798.

[OL81] A.V. Oppenheim and J.S. Lim. “The importance of phase in signals”. In: *Proceedings of the IEEE* 69.5 (1981), pp. 529–541. DOI: 10.1109/PROC.1981.12022.

[Oqu+23] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2023. arXiv: 2304.07193 [cs.CV].

[Oqu+24] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2024. arXiv: 2304.07193 [cs.CV].

[Pen+19] Xingchao Peng et al. “Moment matching for multi-source domain adaptation”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 1406–1415.

[PC82] Leon N Piotrowski and Fergus W Campbell. “A Demonstration of the Visual Importance and Flexibility of Spatial-Frequency Amplitude and Phase”. In: *Perception* 11.3 (1982). PMID: 7167342, pp. 337–346. DOI: 10.1068/p110337. eprint: https://doi.org/10.1068/p110337. URL: https://doi.org/10.1068/p110337.

[Prz+23] Marcin Przewięźlikowski et al. “Augmentation-aware Self-supervised Learning with Guided Projector”. In: *ArXiv* abs/2306.06082 (2023). URL: https://api.semanticscholar.org/CorpusID:259129503.

[PG20] Senthil Purushwalkam and Abhinav Kumar Gupta. “Demystifying Contrastive Self-Supervised Learning: Invariances, Augmentations and Dataset Biases”. In: *ArXiv* abs/2007.13916 (2020). URL: https://api.semanticscholar.org/CorpusID:220830871.

[Rad+21] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[Rom+21] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. arXiv: 2112.10752 [cs.CV].

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[Rus+15a] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[Rus+15b] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[Sae+21] Aaqib Saeed et al. "Federated Self-Supervised Learning of Multisensor Representations for Embedded Intelligence". In: *IEEE Internet of Things Journal* 8.2 (2021), pp. 1030–1040. ISSN: 2372-2541. DOI: 10.1109/jiot.2020.3009358. URL: http://dx.doi.org/10.1109/JIOT.2020.3009358.

[Sal+21] Zohaib Salahuddin et al. *Transparency of Deep Neural Networks for Medical Image Analysis: A Review of Interpretability Methods*. 2021. DOI: 10.48550/ARXIV.2111.02398. URL: https://arxiv.org/abs/2111.02398.

[Sca+09] Franco Scarselli et al. "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[Sch+19] Steffen Schneider et al. "wav2vec: Unsupervised Pre-Training for Speech Recognition". In: *Interspeech 2019* (2019). DOI: 10.21437/interspeech.2019-1873. URL: http://dx.doi.org/10.21437/interspeech.2019-1873.

[Sch+21] Christoph Schuhmann et al. *LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs*. 2021. arXiv: 2111.02114 [cs.CV].

[Sel+19] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128.2 (2019), pp. 336–359. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

[She+17] Jian Shen et al. *Wasserstein Distance Guided Representation Learning for Domain Adaptation*. 2017. arXiv: 1707.01217 [stat.ML].

[Sin+22] Amanpreet Singh et al. "FLAVA: A Foundational Language And Vision Alignment Model". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022. DOI: 10.1109/cvpr52688.2022.01519. URL: http://dx.doi.org/10.1109/CVPR52688.2022.01519.

[SF21] Sahil Singla and Soheil Feizi. *Salient ImageNet: How to discover spurious features in Deep Learning?* 2021. arXiv: 2110.04301 [cs.LG].

[SL22] Yiyou Sun and Yixuan Li. *OpenCon: Open-world Contrastive Learning*. 2022. arXiv: 2208.02764 [cs.LG].

[Thr+22] Tristan Thrush et al. "Winoground: Probing Vision and Language Models for Visio-Linguistic Compositionality". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022. DOI: 10.1109/cvpr52688.2022.00517. URL: http://dx.doi.org/10.1109/CVPR52688.2022.00517.

[Tia+20a] Yonglong Tian et al. "What makes for good views for contrastive learning". In: *ArXiv* abs/2005.10243 (2020). URL: https://api.semanticscholar.org/CorpusID:218719252.

[Tia+20b] Yonglong Tian et al. "What Makes for Good Views for Contrastive Learning?" In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 6827–6839. URL: https://proceedings.neurips.cc/paper/2020/file/4c2e5eaae9152079b9e95845750bb9ab-Paper.pdf.

[Tib96] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[TKH21] Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. *Contrastive learning, multi-view redundancy, and linear models*. 2021. arXiv: 2008.10150 [cs.LG].

[Tou+20] Hugo Touvron et al. *Training data-efficient image transformers and distillation through attention*. 2020. arXiv: 2012.12877 [cs.CV].

[Vas+23] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[WH20] Bram Wallace and Bharath Hariharan. "Extending and Analyzing Self-supervised Learning Across Domains". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 717–734. ISBN: 978-3-030-58574-7.

[WO21] Luyu Wang and Aaron van den Oord. *Multi-Format Contrastive Learning of Audio Representations*. 2021. arXiv: 2103.06508 [cs.SD].

[Wan+20] Zeyu Wang et al. "Towards Unique and Informative Captioning of Images". In: *Lecture Notes in Computer Science* (2020), pp. 629–644. ISSN: 1611-3349. DOI: 10.1007/978-3-030-58571-6_37. URL: http://dx.doi.org/10.1007/978-3-030-58571-6_37.

[Wan+22] Zifeng Wang et al. *MedCLIP: Contrastive Learning from Unpaired Medical Images and Text*. 2022. arXiv: 2210.10163 [cs.CV].

[WAG22] Olivia Wiles, Isabela Albuquerque, and Sven Gowal. *Discovering Bugs in Vision Models using Off-the-shelf Image Generation and Captioning*. 2022. arXiv: 2208.08831 [cs.CV].

[Wu+18] Zhirong Wu et al. “Unsupervised Feature Learning via Non-Parametric Instance Discrimination”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[Xia+23] Zhuofan Xia et al. *DAT++: Spatially Dynamic Vision Transformer with Deformable Attention*. 2023. arXiv: 2309.01430 [cs.CV].

[Xia+21] Tete Xiao et al. “What Should Not Be Contrastive in Contrastive Learning”. In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=CZ8Y3NzuVzO.

[Xu+21] Qinwei Xu et al. “A Fourier-based Framework for Domain Generalization”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021). DOI: 10.1109/cvpr46437.2021.01415. URL: http://dx.doi.org/10.1109/CVPR46437.2021.01415.

[Yan+22a] Haiyang Yang et al. *Domain Invariant Masked Autoencoders for Self-supervised Learning from Multi-domains*. 2022. arXiv: 2205.04771 [cs.CV].

[Yan+22b] Jianwei Yang et al. *Focal Modulation Networks*. 2022. arXiv: 2203.11926 [cs.CV].

[Yan+22c] Jingkang Yang et al. “Panoptic Scene Graph Generation”. In: *ECCV*. 2022.

[YS20] Yanchao Yang and Stefano Soatto. “FDA: Fourier Domain Adaptation for Semantic Segmentation”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020). DOI: 10.1109/cvpr42600.2020.00414. URL: http://dx.doi.org/10.1109/cvpr42600.2020.00414.

[YCA20] Asano YM., Rupprecht C., and Vedaldi A. “Self-labelling via simultaneous clustering and representation learning”. In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=Hyx-jyBFPr.

[YGG17a] Yang You, Igor Gitman, and Boris Ginsburg. “Large Batch Training of Convolutional Networks”. In: *arXiv: Computer Vision and Pattern Recognition* (2017).

[YGG17b] Yang You, Igor Gitman, and Boris Ginsburg. *Large Batch Training of Convolutional Networks*. 2017. arXiv: 1708.03888 [cs.CV].

[Yu+22] Jiahui Yu et al. *CoCa: Contrastive Captioners are Image-Text Foundation Models*. 2022. arXiv: 2205.01917 [cs.CV].

[Yuk+23] Mert Yuksekgonul et al. *When and why vision-language models behave like bags-of-words, and what to do about it?* 2023. arXiv: 2210.01936 [cs.CV].

[Zbo+21a] Jure Zbontar et al. *Barlow Twins: Self-Supervised Learning via Redundancy Reduction*. 2021. arXiv: 2103.03230 [cs.CV].

[Zbo+21b] Jure Zbontar et al. *Barlow Twins: Self-Supervised Learning via Redundancy Reduction*. 2021. arXiv: 2103.03230 [cs.CV].

[ZZL22] Yan Zeng, Xinsong Zhang, and Hang Li. *Multi-Grained Vision Language Pre-Training: Aligning Texts with Visual Concepts*. 2022. arXiv: 2111.08276 [cs.CL].

[ZIE16] Richard Zhang, Phillip Isola, and Alexei A. Efros. "Colorful Image Colorization". In: *European Conference on Computer Vision*. 2016. URL: https://api.semanticscholar.org/CorpusID:50698.

[Zha+18] Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 2018. arXiv: 1801.03924 [cs.CV].

[Zha+21] Ruihan Zhang et al. "Invertible Concept-based Explanations for CNN Models with Non-negative Concept Activation Vectors". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.13 (2021), pp. 11682–11690. ISSN: 2159-5399. DOI: 10.1609/aaai.v35i13.17389. URL: http://dx.doi.org/10.1609/aaai.v35i13.17389.

[Zha+22a] Xiang Zhang et al. *Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency*. 2022. arXiv: 2206.08496 [cs.LG].

[Zha+22b] Xingxuan Zhang et al. "Towards Unsupervised Domain Generalization". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022). DOI: 10.1109/cvpr52688.2022.00486. URL: http://dx.doi.org/10.1109/CVPR52688.2022.00486.

[Zha+20] Yuhao Zhang et al. *Contrastive Learning of Medical Visual Representations from Paired Images and Text*. 2020. arXiv: 2010.00747 [cs.CV].

[Zho+17] Bolei Zhou et al. "Places: A 10 million Image Database for Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[Zho+18] Bolei Zhou et al. *Semantic Understanding of Scenes through the ADE20K Dataset*. 2018. arXiv: 1608.05442 [cs.CV].

[Zou+22] Xueyan Zou et al. *Generalized Decoding for Pixel, Image, and Language*. 2022. arXiv: 2212.11270 [cs.CV].

[Zou+23] Xueyan Zou et al. *Segment Everything Everywhere All at Once*. 2023. arXiv: 2304.06718 [cs.CV].