ABSTRACT

Title of Thesis: DESIGN NOVELTY EVALUATION THROUGH ORDINAL EMBEDDING: COMPARISON OF NOVELTY AND TRIPLET ERRORS

Matthew Keeler
Master of Science, 2024

Thesis Directed by: Professor Mark Fuge
Department of Mechanical Engineering

A practical and well-studied method for computing the novelty of a design is to construct an embedding via a collection of pairwise comparisons between items (called triplets), and use distances within that embedding to compute which designs are farthest from the center. These triplet comparisons are posed in the form of "Is Design A closer to Design B or Design C?", and inform the placement of designs in the similarity-space embedding. This method of creating an embedding from non-metric relationship comparisons is known as ordinal embedding. Unfortunately, ordinal embedding methods can require a large number of triplets before their primary error measure—the proportion of violated triplet comparisons—converges. But if our goal is accurate novelty estimation, is it really necessary to fully minimize all triplet violations? Can we extract useful information regarding the novelty of all or some items using fewer triplets than existing convergence rates on the saturation of triplet violations might imply? This thesis addresses

this question by studying the relationship between triplet violation error and novelty score error when using ordinal embeddings.

We find that estimating the novelty of a set of items via ordinal embedding can require significantly fewer human-provided triplets than is needed to converge the triplet error, and that this effect is modulated by the type of triplet sampling method (random versus uncertainty-informed active sampling). Having learned this, we propose the use of a custom metric we call the 'Expected Model Change' (EMC) which we use to observe when novelty information in the embedding has stopped updating under newly labeled triplets, so that conservative bounding functions need not be used. Moreover, to avoid the dangers of low accuracy in selecting the dimension of the ordinal embedding, we propose use of the Expected Model Change for tuning the embedding dimension to an appropriate value. In this framework, we explore the convergence properties of ordinal embeddings reconstructed from triplets taken from a variety of synthetic and real-world design spaces.

DESIGN NOVELTY EVALUATION THROUGH ORDINAL EMBEDDING: COMPARISON OF NOVELTY AND TRIPLET ERRORS

by

Matthew Keeler

Thesis 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
Master of Science
2024

Advisory Committee:
Professor Mark Fuge, Chair/Advisor
Professor Shapour Azarm
Professor Nikhil Chopra

Foreword

Portions of this thesis were previously published as “Fewer Triplets Than You Think: Novelty Error Converges Faster Than Triplet Violations in Ordinal Embeddings” for the International Design Engineering Technical Conferences [1].

UNIVERSITY OF MARYLAND

2181 Glenn L. Martin Hall
College Park, MD 20742-3035
TEL: 301-405-2410
http://www.enme.umd.edu

4/19/2024

Re: Previously Published Materials appearing in Thesis or Dissertation

Dean of the Graduate School,

Matthew Keeler has:

☐ NOT INCLUDED any previously published works within their thesis or dissertation.

☒ INCLUDED one or more previously published works within their thesis or dissertation. This letter certifies that the examining committee for the student has determined that the student made a substantial contribution to the previously published work. The inclusion of the previously published work has the approval of the thesis or dissertation advisor and the Graduate Director.

Sincerely,

Peter Sandborn
Director of Graduate Studies
Department of Mechanical Engineering
University of Maryland

Sincerely,

Mark Fuge
Digitally signed by Mark Fuge
Date: 2024.04.24 11:08:30 -04'00'

Mark Fuge
Advisor for Matthew Keeler

Acknowledgments

There are many people who have been helpful in both my personal and professional journey that is culminated in this thesis. First, and most importantly, I would like to thank my advisor, Dr. Mark Fuge. Mark has had to bare with me for many hours of talks and meetings, and through all that time made himself constantly available, being the most powerful resource of knowledge that I have encountered. Mark's advice has both shaped my development as a person and researcher.

I would also like to thank my colleagues in the IDEAL lab as well as my predecessors, notably Dr. Faez Ahmed, whose resources and datasets surrounded much of my work in this thesis. Faez's writing prowess and meticulous documentation made for a smooth transition into this project. Although very busy and oweing me nothing, Faez also made himself available to discuss my continuations of his work, and for that I am grateful.

I would like to thank my parents and brother for being my emotional touchstones throughout my personal and academic journey. Without their support, I would not have made it through primary school, much less this Master's program.

Finally, I would like to thank the National Science Foundation through award #2155072 for their financial support, which has enabled me to complete this research.

Thank you again to every family member, colleague, and professor who has shaped the journey that has led to this thesis. I will forever be grateful for your time, knowledge, and support.

Table of Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

ABC	Example of Triplet Comparison
b	Number of Uncertainty Bootstrapped Embeddings
C_{emb}	Centroidal Distance Matrix (Embedded Space)
C_{true}	Centroidal Distance Matrix (True Space)
D	Pairwise-Distance Matrix
d	Ordinal Embedding Dimension
EMC	Expected Model Change
$\overline{ij}$	Euclidean-Distance
K	Gram Matrix
k_{ij}	Element of Gram Matrix
n	Total Number of Embedded Designs
Nov	Scored Novelty Error
T	Upper Triangle of Pairwise-Distance Matrix D
X	Matrix of All Embedding Coordinates
x_i	Embedding Coordinate of Design i
λ	Design Space Density/Sparsity Value
ξ_{ijk}	GNMDS Slack on Triplet Constraint 'ijk'
π_{ijk}	Uncertainty Score of Triplet 'ijk'
$\overline{\rho}$	Average Pairwise-Distance Matrix
$\overline{\sigma}^2$	Variance Pairwise-Distance Matrix
Φ	Normal Probability Density Function

CKL	Crowd-Kernel Learning
DbA	Design by Analogy
EMC	Expected Model Change
GNMDS	Generalized Non-Metric Multidimensional Scaling
NMDS	Non-Metric Multidimensional Scaling
STE	Stochastic Triplet Embedding

Chapter 1: Introduction

While the search for creative designs is the backbone of modern innovation, evaluating the creativity of a design remains a challenge. Creativity is a multi-faceted design metric that is comprised of sub-metrics including—but not limited to—design quality, variety, and novelty [3–5]. Most of these metrics are difficult to objectively score, but one suggested approach to computing design *novelty* involves using human-provided evaluations or comparisons to embed a pool of designs into a low-dimensional Euclidean space [2], and then using distances in that space to compute a design's relative novelty. This process involves receiving pairwise comparisons in the form of triplet queries—is design 'A' closer to design 'B' or 'C'—from human raters. It is comparatively easy to perform a simple similarity comparison task of finding the closer match to a reference design from only two candidates [6], compared to the much more difficult task of ranking all designs by novelty from a large pool of candidates. Given comparison triplets provided by human raters, we can construct an embedding from this non-metric data that projects designs into a latent space where similarity is expressed as a latent distance between designs—that is, similar designs are plotted closer together than dissimilar designs [7]. With this embedding of items which best satisfy the triplets received from human raters, prior work has shown that the most novel designs correlate with those furthest away from the centroid of all items in the embedding [2, 4].

Although this approach sounds reasonable, it has one major drawback: effectively reconstructing the true design embedding that satisfies human-provided triplets requires a large number of triplet comparisons. More precisely, the total number of possible triplets for an embedding of n items grows combinatorially: $n\frac{(n-1)!}{(n-3)!2!}$. As an example, for 10 designs there are a total of 360 possible triplet comparisons, which increases to 1365 triplets for 15 designs and 3420 triplets for 20 designs. Asking a human rater to exhaustively perform thousands of pairwise comparisons is expensive and mentally taxing, and is further exacerbated when considering that multiple human raters are needed to represent a general consensus on similarity groupings.

Fortunately, although an exhaustive enumeration of all triplets is needed to represent every similarity grouping in the embedding, in practice, we need significantly fewer than this. Research has shown that by *actively selecting* the triplets which carry the most information, one can reduce the number of triplets needed to reconstruct the latent space [8–11]. However, in existing theoretical work the number of triplet samples needed for a given pairwise comparison model provide only rough order relations (up to a constant scaling factor), so it is unclear exactly how many samples one would practically need to collect for a given application.

In addition, most existing research addresses creating the best embedding for *all items*—as in the embedding which satisfies the most triplets. If the goal of an embedding is to establish which items are *novel*, however, it is unclear if this level of embedding accuracy is needed. By contrast, if we just wish to discover which ideas are the most novel—and not the exact embedded coordinates of all items—then we may not need as many comparisons as existing theoretical bounds might imply.

Moreover, by nature, bounding functions are conservative. Using a bounding function to predetermine the amount of labeled triplets needed would result in unnecessary costs in human

time and labor. To be cost-effective, this method would require a procedure for evaluating the saturation of novelty information online—that is as triplets are labeled one-by-one. This way, human rating would cease immediately after the embedding has stopped responding to new labeled triplets.

For the purposes of convergence analysis, this thesis proposes the use of an alternative error metric that prioritizes the accuracy of an embedding for calculating the relative novelty of items, as opposed to solely reducing the violations of all triplet comparisons. This 'novelty error' metric leverages existing techniques for describing novelty detailed in [2] to describe how successful an embedding is at relaying novelty information.

With this alternative error metric, the rest of this thesis addresses the following questions in chapters 3-6 respectively:

- Does an ordinal embedding method—specifically Generalized Non-Dimensional Scaling (GNMDS)—discover the relative novelty of designs using fewer triplets than it takes to minimize triplet violations? How do these reconstruction accuracy indicators relate to one another?

- How would one identify that a sufficient number of triplet queries have been collected to effectively reconstruct the novelty information within the design space, before all possible triplet combinations are labeled? Could an online evaluation method be used in place of conservative bounding functions?

- What are the effects of poor dimension approximation for the embedding? How would one determine an appropriate dimension in which to create the embedding reconstruction of the unknown design space?

- How does the distribution of items in the design space affect the reconstruction properties of the ordinal embedding? Can the findings to the previously posed questions generalize to a variety of potential design space distributions?

We provide both empirical results across a range of synthetic and real-world examples, as well as a new diagnostic measure—Expected Model Change—that can be computed while collecting triplets to determine when sufficient triplets have been gathered.

Chapter 2: Related Work and Background

This chapter will provide an overview of existing work and concepts which are central to the contributions of this thesis. It will cover novelty estimation in engineering design as well as ordinal embeddings and their role in novelty estimation, as well as existing work around tuning ordinal embeddings.

2.1 Novelty in Engineering Design

In engineering design, novelty refers to originality or uniqueness with regard to a design's solution approach or structural composition [3]. Novelty is one of the primary sub-metrics for describing the creativity of a design—along with design variety and quality [4, 5]. Unlike most other agreed upon submetrics—which require domain-specific scoring criteria—novelty is often considered as being relative to other candidate designs, and so developing methods for generalizing design novelty evaluation is an active research interest. Specifically, the fields of design-by-analogy (DbA), design education, and AI-integrated design desire a method for identifying and evaluating novel designs.

2.1.1 Uses for Novelty Estimation

Design by analogy (DbA) refers to the practice of drawing inspiration from existing designs when designing something new [12, 13]. Existing studies have shown DbA is a powerful method for improving the diversity of design concepts in the initial brainstorming phases of design ideation [13, 14]. Additional research has also shown that when designers draw inspiration from expert-identified novel and creative solutions that this facilitates generation of new novel and creative solutions [12, 15]. With a generalizable method in place for evaluating the novelty of designs, DbA techniques can more effectively identify novel designs to be used as inspirational stimuli.

Because the development of novel and creative solutions is the backbone of engineering innovation, substantial research has been conducted to developing methods for teaching new engineers to produce creative designs [16, 17]. Existing methods for teaching creativity in engineering design mostly center on teaching new engineers effective brainstorming techniques for generating a diverse set of concepts [18, 19]. These methods encourage outside-the-box thinking to generate as many candidate designs as possible. However, quantity of concepts does not necessarily facilitate quality concept generation [19]. With a generalized method of evaluating design novelty, the success of idea generation tasks can be analyzed in educational environments. Effective novelty evaluation may also reveal new practices for generating unique solutions.

With modern advances in artificial intelligence (AI), recent work has explored capacities where AI can assist in engineering concept generation. Existing work has used AI in efforts to identify creative solutions and explain what might make them creative [4, 20]. Advances in generative AI have also motivated work that uses AI to model and develop creative design solutions

itself [21, 22]. With an effective and generalizable methods for describing design novelty, AI could be used to explain the tangible features behind high-novelty assessment and even be able to self-assess novelty to learn how to generate novel solutions on its own.

2.1.2 Methods for Novelty Evaluation

Because novel designs are designs which are rare or unique among a set, computation of design novelty first requires computation of pairwise similarity between designs. Traditionally, in fields like psychology, similarity is measured by defining a set of relevant features and then counting the number of shared features [23]. For comparing engineering designs, these computations require an exhaustive list of potentially relevant features to be identified across designs, and thus are particularly ineffective for comparing designs with widely different functional architectures. These methods also assume that all features in the predefined list are weighted equally when performing a similarity comparison. If design A has five shared features with design C, and design B only has three shared features with design C, then design A will be ruled as more similar regardless of how integral those features are in the design.

Due to these shortcomings in traditional feature count-based methods, for engineering design, similarity comparisons often require experts to predefine scoring criteria [5,24,25]. Experts must not only anticipate a pool of relevant features, but also need to assign weights on the importance of those features. Novelty can then be computed by counting the number of shared features and weighing their importance accordingly in a summation [25]. Still, however, these methods require the development of hyper-specifc scoring rubrics which rely heavily on the opinions of experts. These methods also assume that collective similarity can be reasonably measured in a

single direction based on feature co-occurence.

More recent work has proposed plotting designs in a spatial graph or embedding—where each dimensional axis describes a facet of the design to be considered when computing similarity [2, 13, 26]. Then the novelty of a design can be computed by first identifying the centroid, or averaged position among designs in the embedded space, and then computing the Euclidean-distance between each design and the centroid [2]. Designs which exist far from the centroid are designs which are highly dissimilar to the average design. In other words, these designs are unique and novel relative to the other considered designs in the embedding space. Similarity-space embeddings are powerful in that they allow both identification of novel items but also visualization into how each feature influences the reported similarity and clusters the designs. But how are these embeddings created? What features should describe the axes of the embedding and how should they be scored? These questions will be addressed in the next section.

2.2 Ordinal Embeddings

The biggest hurdle for implementing classical embedding-based approaches of modeling design similarities is developing scoring criteria for evaluating shared functional architectures across designs. Cataloging an exhaustive list of the features present in a pool of design ideas is expensive, and these methods assume that humans can reliably identify all design facets that distinguish any two designs. These methods may be practical for contrasting simple designs with nuanced differences in structure, but become near-impossible when comparing a dataset of complex designs.

Consider the colored-shape dataset found in Figure 2.1 from Ahmed *et al.* [2]. This dataset

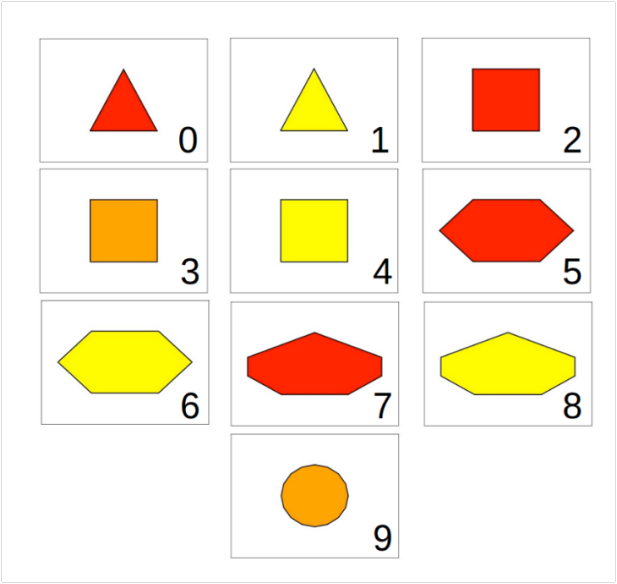

Figure 2.1: Shape Dataset from Ahmed *et al.* [2]

contains ten simple colored polygons. If one were to implement a classical embedding-based approach for describing the shared features, each shape could be described by its **shape** and its **color**. These two features are ubiquitous across all shapes and distinguish any shape from its neighbors. They also lend themselves nicely to metric evaluation.

In Figure 2.1, the shape dataset has been plotted on axes corresponding to the 'color' and 'number of side' features. Shapes with higher color intensity/brightness are sorted near the top and shapes with large number of sides are sorted near the right-side of the embedding. This groups all shapes in two directions of feature similarity. Using the centroidal distance method for novelty evaluation described in §2.1.2, shape 9 can be identified as the most-novel shape, as its features are largely distinguishable from the average shape. Shape 9 both has a unique number of sides that is much larger than any other shape, and also has a unique orange color shared only with shape 3. For these reasons, the classical embedding approach places it far from the other shapes. The remaining shapes can similarly be sorted based on their centroidal distance to obtain

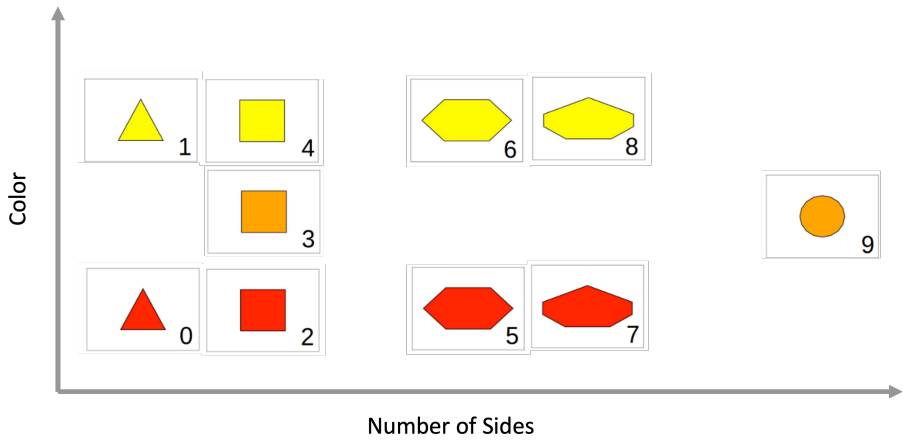

Figure 2.2: Classical Embedding for Shape Dataset

a novelty-ordered set.

For the colored-shape dataset, this classical approach was straightforward, as a small number of feature axes could be identified that generalized across all items and those features could be metrically evaluated. Now consider performing the same classical embedding procedure on the designs found in Figure 2.3. This dataset contains ten hand-drawn sketches for a device that froths a basin of milk [2]. There are many different types of designs in this dataset: some that stir, shake, spin, have motors, have a human operator, have air pressure... the list goes on. Whereas in the colored shape dataset it was easy to identify two generalize-able features that classified all shapes, in this dataset there are no immediately identifiable features that can categorize all designs. Even if a list of sorting criteria could be developed, many of those features would not easily lend themselves to quantification. For these reasons, implementing the same novelty-sorting procedure used on the colored shape dataset seems infeasible for this dataset.

But what if there was a way to create an embedding like we did for the shape dataset in Figure 2.2 without explicitly identifying any scoring criteria? While the milk-frother dataset does not lend itself to easy feature evaluation like the shape dataset, there are still similarities amongst

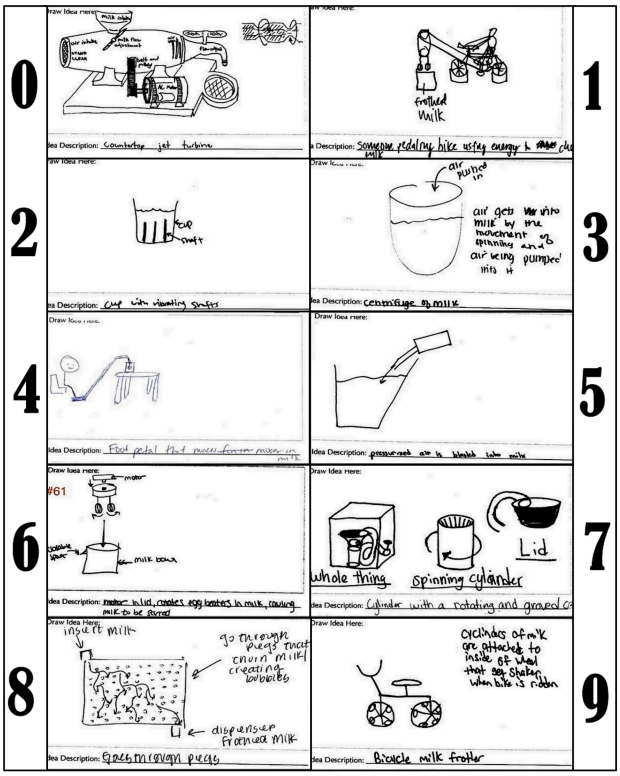

Figure 2.3: Milk Frother Dataset from Ahmed *et al.* [2]

designs that can be reasonably identified. These similarities can be observed when limiting comparisons to only pairs of designs. In the milk frother data, at first glance, designs 1 and 6 appear to have very similar functional architectures. Whereas it was difficult to identify features that generalize over the entire dataset, it is comparatively easy to identify featural similarities between this pair of designs. For example, both of these devices have an egg-beater mixer which is lowered into the milk basin to stir the milk. This, one could argue, would be evidence to claim designs 1 and 6 are more similar than say designs 1 and 8 are. This type of non-metric, comparison-based observation is an example of *ordinal data*.

2.2.1 What is Ordinal Data?

Ordinal data characterizes information solely in categorical terms. In essence, ordinal data is described based on relative classes which do not provide precise magnitude indications. An example of ordinal data is a **triplet** comparison, which compares the similarity of three items at once. In the context of design similarity, a triplet query might be posed as "is design A more similar to design B or design C?". The answer to this query is called a **triplet label**. A labeled triplet from this query might be 'ABC', which reads as "design A is closer to design B than design C". In this notation, the first letter refers to the reference design, the second letter refers to the more similar candidate, and the third letter is the less similar candidate. This triplet is ordinal, in that it only categorizes design pair 'AB' as being more similar than pair 'AC', but does not describe how much *more* similar the pair is. Ordering a triplet label does not require any metric information that describes the closeness of pairs 'AB' and 'AC', as long as a judgment can be made for which pair is more similar.

In contrast to scoring-based similarity descriptions that can only leverage numerical features of a design, triplets are powerful in that they allow us to compare designs that have largely different functional architectures. Comparing complex, multi-component designs with metric similarity scores is often difficult or impractical. It may be unclear what features should be represented in a similarity metric, or how those features should be aggregated into a single similarity score. Furthermore, defining the scoring criteria for similarity metrics is not an exact-science and impractical when comparing designs with different functional architectures. Triplet comparisons do not require pre-defined scoring criteria. In addition, existing research has demonstrated that humans can more reliably identify the closer match of a triplet comparison than they can

quantify the closeness of a pair [2, 6, 27]. Of course, not all humans will prioritize the same criteria when ordering a triplet comparison, but existing research has shown that collecting triplets from multiple human raters can reveal a general consensus for human-perceived similarity assessment [6, 27].

But how can ordinal triplet comparisons be used to metrically evaluate design novelty? Ordinal data must be converted into an embedded feature space like what was done for the shape dataset in Figure 2.2. This is known as **ordinal embedding**.

2.2.2 What do Ordinal Embeddings Do?

Ordinal embeddings work to generate a mapping of designs where similarity is expressed as a Euclidean distance between designs, but without explicitly identifying or metrically scoring features. Instead, ordinal embeddings are generated from an optimization framework. In this optimization, the coordinates of the designs in the embedding are design variables x_n where n is the index for a specific design. Ordinal triplet comparisons appear as constraints on the allowable placement of designs. To be included in this framework, triplet label 'ijk' can be rewritten as $\overline{ij} < \overline{ik}$ or:

$$||x_i - x_j||_2^2 < ||x_i - x_k||_2^2$$

This triplet imposes the constraint that the distance between design A and B in the embedding must be smaller than the distance between A and C in the embedding. Satisfying this triplet will satisfy the human-perceived similarity assessment in the embedding. As more triplet constraints are imposed on the embedding, the embedding will be sorted into a similarity feature-

space like the shape dataset, but without labels on the feature axes.

There are many objective function protocols for generating an embedding from these triplet constraints. This problem is known as the Non-Metric Multidimensional Scaling (NMDS) problem. In essence, it creates a mapping of items where magnitude 'scaling' can be approximated when the input is purely 'non-metric' ordinal data—which lacks magnitude information.

In general, there are two primary classes of optimization algorithms for generating an embedding from a pool of ordinal triplet comparisons. The first requires the development of a probabilistic model for describing the probability that a given triplet ordering is satisfied in a generated embedding [28]. Crowd Kernel Learning (CKL), and Stochastic Triplet Embedding (STE) are examples of these algorithms [6,27]. In these methods, optimization of the embedding placement is done by maximizing the predefined triplet likelihood function. The second class of algorithm involves directly solving for the Gram matrix by minimizing the degree of violation on every triplet constraint through slack variables. The most common algorithm in this class is Generalized Non-Metric Multidimensional Scaling (GNMDS) [29]. This optimization framework is more intuitive, as it does not require the development of a probabilistic model. GNMDS is the key ordinal embedding algorithm used in this thesis, and is described more thoroughly in the following section.

2.2.3 Generalized Non-Metric Multidimensional Scaling (GNMDS)

Generalized Non-Metric Multidimensional Scaling improves upon the major shortcomings of the traditional Shepard-Kruskal NMDS [30,31]. Notably, GNMDS employs a convex program and can create embeddings without requiring inequality comparisons on all pairwise distances.

All subsequent definitions in this section are transcribed from the orginal GNMDS paper by Agarwal *et al.* [29].

First, the GNMDS program recasts the problem of solving for the matrix of all embedding coordinates X as a problem that solves for the Gram matrix $K = X^T X$. With this distinction the triplet inequality constraints

$$||x_i - x_j||_2^2 \leq ||x_i - x_k||_2^2$$

Can be written in terms of the corresponding elements of the Gram matrix

$$k_{ii} - 2k_{ij} + k_{jj} \leq k_{ii} - 2k_{ik} + k_{kk}$$

Because paired comparisons can only provide relative spatial constraints, GNMDS must include protocols for maintaining translational and scale invariance in its solutions. Translational invariance is achieved by restricting the embedding to be centered at the origin. In other words

$$\sum_b x_b = 0$$

And because

$$(\sum_b x_b)^T (\sum_b x_b) = 0$$

Then

$$\sum_{ab} k_{ab} = 0$$

To prevent the embedding solution from collapsing onto the origin, a constant is added to the triplet inequality constraints to ensure that the constraint is met beyond a threshold margin. In

the original GNMDS paper, a constant of '1' was arbitrarily chosen. The triplet constraints now appear as

$$k_{ii} - 2k_{ij} + k_{jj} \leq k_{ii} - 2k_{ik} + k_{kk} + 1$$

To create an objective for this embedding optimization problem and to make triplet constraints soft in the cases where no feasible solution exists, a slack variable ξ is added to triplet constraints. The objective function is then defined as minimization on the slack of the triplet constraints. Rewriting the triplet constraints and imposing the protocols for translational and scale invariance, the GNMDS optimization becomes

$$\begin{aligned} \min_{K,\xi_{ijk}} \quad & \sum_{ijk} \xi_{ijk} \\ \text{s.t.} \quad & -2k_{ij} + k_{jj} + 2k_{ik} - k_{kk} \leq 1 + \xi_{ijk} \\ & \sum_{ab} k_{ab} = 0 \quad \xi_{ijk} \geq 0 \quad K \geq 0 \end{aligned}$$

After solving for the optimal Gram matrix K, the coordinate matrix can be obtained by performing an eigendecomposition $K = U\Lambda U^T$ and setting $X = U\Lambda^{\frac{1}{2}}$.

The GNMDS method for creating an embedding from ordinal triplet data will be used in all data collection experiments in this thesis. GNMDS was chosen due to its intuitive and straightforward approach for evaluating embedding accuracy. GNMDS not only aims to reduce triplet violations, but considers the severity of the violation and encourages large margins of label satisfaction. By first solving for the Gram matrix and then performing a spectral decomposition,

GNMDS can generate an embedding placement for designs which are not represented in triplet constraints. The embedding dimension can also be easily tuned by constraining the rank of the Gram matrix, which will be explored in §5.

2.3 Implementing Triplet Embeddings

Regardless of the algorithm, solving the NMDS problem to create an embedding of items from ordinal relations requires a large number of triplet comparison queries. The number of triplet comparisons that can be made from a pool of N items grows combinatorially at $n\frac{(n-1)!}{(n-3)!2!}$. Because triplet queries must be labeled by human raters, and multiple raters are needed to form a general consensus on the labels, there is a strong motivation to reduce the number of labeled triplets that are needed to generate a converged embedding.

To address this problem, past research has explored ways of actively selecting triplets prior to labeling [7, 10, 32]. As the embedding algorithm learns spatial trends in the triplet comparison data, some labeled triplets will challenge the embedding's existing assumptions and lead to large updates, while other triplets will already be satisfied prior to labeling. To identify these impactful triplets, existing research suggest implementing uncertainty sampling—where unlabeled triplets that the model has little confidence in predicting are prioritized for labeling [9, 33].

Capturing the uncertainty associated with a triplet query is typically done with either a Bayesian approach, which attempts to model triplet likelihood with a probability estimation [9], or a Bootstrap approach which creates multiple embeddings from subsets of a triplet pool to measure collective disagreement [33]. Because it is difficult to encode properties of embedding distances into a prior distribution [33] and there is debate over the structure for creating a like-

lihood estimate for ordinal embeddings, we implement the Bootstrap Approach in this paper for determining uncertainty. Our paper uses uncertainty sampling in experiments to better understand how many triplet queries might be needed to capture sufficient information in the true space and to determine which queries are the best at relaying specific desired information.

2.3.1 Bootstrap Uncertainty Sampling

This section will detail the bootstrap uncertainty sampling approach from Lohaus *et al.*which was used for actively selecting triplets in all experiments [33]. This approach uses bootstrap subsampling to generate b embeddings, each produced from a subset of the current pool of labeled triplets. In our experiments, a 75% bootstrapped subset was found to be appropriate to represent enough of the labeled pool while still allowing for variance across the bootstrapped embeddings for the uncertainty algorithm to work.

To model triplet label uncertainty, the Lohaus approach factors in the uncertainty in the pairwise distances contained in each triplet comparison. However, the ultimate score is created by considering how those pairwise distance uncertainties create uncertainty in the reported triplet label. First, a matrix of all pairwise Euclidean distances D_{ij} is created for all of the b bootstrapped embeddings. Then the sample mean $\overline{\rho}_{ij}$ and variance $\overline{\sigma}^2_{ij}$ of each pairwise distance across the bootstrapped embeddings are computed as

$$\overline{\rho}_{ij} = \frac{1}{b}\sum_{k=1}^{b} D_{ij}^{(k)}, \quad \overline{\sigma}^2_{ij} = \frac{1}{b-1}\sum_{k=1}^{b}(D_{ij}^{(k)} - \overline{\rho}_{ij})^2$$

Assuming the pairwise distance are normally distributed as $N(\overline{\rho}_{ij}, \overline{\sigma}^2_{ij})$, the uncertainty in

a triplet label π_{ijk} can be modeled as

$$\pi_{ijk} := \Phi(\frac{\overline{\rho}_{ij} - \overline{\rho}_{ik}}{\overline{\sigma}_{ij} - \overline{\sigma}_{ik}})$$

Here Φ denotes the normal cdf function. This uncertainty representation models how pairwise distance uncertainty impacts triplet uncertainty. When the absolute difference in the average pairwise distances is much larger than the absolute difference in the standard deviations, this produces a large ratio magnitude and represents a triplet that is confidently ordered across the bootstrap embeddings. When the pairwise distance averages are similar relative to the standard deviations, this produces a ratio near zero and indicates a triplet ordering with high uncertainty in its label. After the normal cdf, this means that π_{ijk} near 0.5 are the most uncertain, and π_{ijk} near 0 or near 1 represent triplets with confident labels across the embeddings.

Implementing this uncertainty representation in an active triplet sampling framework works as follows:

1. Randomly select and label initial triplets to seed the uncertainty algorithm. In our experiments this was done by randomly choosing two triplets per reference design to label. This was done to ensure the model had at least some information about the placement of every design prior to uncertainty sampling.

2. Subsample 75% of the labeled triplet pool b times and plug into an ordinal embedding algorithm like GNMDS to produce b embeddings. In our experiments $b = 10$ was found to be appropriate to explore the bootstrapped variance without significantly adding to the computational complexity of rerunning the GNMDS algorithm b times.

3. Create pairwise distance matrix D_{ij} for every bootstrapped embeddings and compute the sample average $\overline{\rho}_{ij}$ and variance $\overline{\sigma}^2_{ij}$ for each pairwise element.

4. Compute the cdf uncertainty score π_{ijk} for every triplet combination.

5. Sort all triplet combination uncertainty scores π_{ijk} by their squared difference from 0.5 (most uncertain). Choose the triplet combination closest to 0.5 to label, label it, add it to the pool.

6. Repeat steps 2 - 5 until embedding convergence or all triplets have been added to the pool.

2.3.2 Convergence of Triplet Embeddings

Although finding an exact relation between the number of items to be embedded and the minimum number of triplets to be sampled is application specific, research has been conducted toward finding a lower bound for the number of actively selected triplets needed to create an embedding [8, 28]. In fact, by examining the number of constraints that need to be satisfied by adding a single item to an embedding, logic indicates that for an embedding of n objects and d dimensions, at least $\Omega(d\log n)$ bits of information are needed to define an embedding [8]. Extending this logic to triplet queries indicates that the number of triplets needed to satisfy an embedding increases on $\mathcal{O}(dn\log n)$ [8]. These bounds, however, are quite conservative and describe converged embedding construction on the only basis of triplet violations—with no consideration into how the centroidal distances and novelty information are affected.

2.4 How This Thesis Extends on Past Work

While existing work has been performed which uses ordinal embeddings to describe human-perceived similarity, relatively little work has been conducted toward understanding the convergence properties of ordinal embeddings in the context of evaluating engineering designs. Any work that has been done on understanding the convergence properties of ordinal embeddings only proposes conservative bounds on the number of sampled triplets required for triplet violation convergence. These bounds are interesting for understanding the computation complexity growth rates for ordinal embeddings, but are not practical for defining general stopping criteria in a triplet sampling framework.

In this thesis, we are interested in how *novelty* information is captured through triplet sampling and reconstructed in the ordinal embedding. As such, our questions regarding convergence of the embedding are with regard to capturing the novelty information in the true space. Likewise, we are interested in how the accuracy of the embedding dimension approximation plays a role in the convergence properties of the novelty information. Existing work which explores the role of the embedding dimension does so with regard to triplet violations, and does not consider the impact of dimension approximation error on the reconstruction of centroidal distances needed for novelty estimation. This thesis also explores how the distribution of items in the true space can affect these convergence properties.

Chapter 3: Comparison of Triplet Violation and Novelty Error Metrics

In this chapter we detail: (1) the motivation behind our custom *novelty error* metric, (2) our method for generating a synthetic dataset of known items in the 'true' space to serve as an experiment baseline for computing the error of GNMDS-produced embeddings, (3) the sampling algorithms we use to select new triplets to label, and (4) how we compute the convergence behavior of the novelty error in our synthetic datasets.

3.1 Defining the Novelty Error

In existing applications of ordinal embeddings, the representation error of the embedding is typically expressed as a triplet violation loss—where the loss is defined as the fraction of the total triplet pool with labels that do not match the orientation of items in the embedding. This loss is intuitive and useful for internal embedding construction, as it directly measures the embedding's ability to adhere to the provided triplet constraints [10]. However, while the triplet violation loss might measure the overall quality level of the chosen embedding, it does not directly map to how that embedding is used to measure novelty. While it is true that all triplet constraints must be satisfied for a 1-to-1 mapping of the design space, if the goal is only to capture the general novelty of designs through their relative distance in the embedded space, an exact or perfect representation may not be needed and triplet violations might be unimportant if they do not cause

changes to a design's novelty score. To better investigate how novelty information saturates as a function of embedding quality, we created a custom error metric we call the *novelty error*.

Instead of looking at triplet violations, our custom metric Nov measures an error based on the deviation of the centroidal distances of all items in the embedding, compared to the true space orientation. We did this to mirror the novelty estimation method used by [2]. This error was calculated and measured as follows: First, the distance of each item C_{true} from the centroid of all items in the true space $\overline{C}_{true}$ was computed and normalized by the largest of these distances $max(C_{true})$. This created a reference array for how close to the centroid each item should be in the true space. Then the same was done for the reconstructed embedding and the reconstruction centroid to produce C_{emb} and $max(C_{emb})$. For GNMDS specifically, the centroid is defined to be the origin to allow for translational invariance between subsequent iterations. The difference in these two centroidal distance arrays was computed, and the L_1 norm was used to quantify the error. A multiplier of $\frac{1}{n}$ is added to scale the error by the number of plotted designs. In this way, the error can be interpreted as the average deviation of relative centroidal distances. For example, if the novelty error is 0.3, this means that average rank deviation in centroidal distances between the target and embedded spaces is three tenths relative to the largest centroidal distance.

$$Nov = \frac{1}{n}\left\| \frac{C_{true}}{max(C_{true})} - \frac{C_{emb}}{max(C_{emb})} \right\|_1 \tag{3.1}$$

$$C_{true} = \left\| X_{true} - \overline{X}_{true} \right\|_2 \tag{3.2}$$

$$C_{emb} = \left\| X_{emb} \right\|_2 \tag{3.3}$$

This error metric quantifies the embedding's ability to not only reconstruct the correct novelty

ordering of items, but also accounts for the magnitude of centroidal distances that may be unobservable when looking only at a novelty order. For example, a novelty order of A, B, C, D might tell us that item A is the most novel followed by B, C, and D, but it does not tell us how much *more* novel item A is. If we considered purely a novelty order as an error metric, it may be possible that we incur no loss for an embedding that correctly produced the ordering A, B, C, D but had item A twice as far from the centroid than the same item in the true mapping.

3.2 Generating a Synthetic Dataset

While in reality we would not know the true latent space of rated items, for us to properly display the convergence behavior of the produced embeddings in our below experiments, we had to generate a synthetic true space to serve as a baseline for both triplet error and computing item novelty. To generate the synthetic latent space, we assigned n items d-dimensional coordinates from a uniform distribution at random with values between -1 and 1. Because triplet queries only capture relative orderings of items, the embedding coordinates can only match the true coordinates up to transformations in scaling, rotation, and translation. Thus the precise scale, position, and rotation of the true synthetic space will not matter for our below analyses. We also tried other synthetic distributions apart from uniform, such as a single peaked Gaussian; the results appear qualitatively similar to the uniform distribution apart from some effects of clustering on the novelty error magnitude. These effects of the synthetic distribution will be covered in more detail in Chapter 6.

After we construct the 'true' latent space, we exhaustively compute all of the $n\frac{(n-1)!}{(n-3)!2!}$ possible triplet pairings and ordered them based on each item's position in the true space. For

example, if in the synthetic true space the distance between item ‘A’ and item ‘C’ was greater than the distance between item ‘A’ and item ‘B’, we record the triplet as ‘ABC’. Here the first letter refers to the reference item (sometimes called the anchor), the second letter refers to the closer item of the two candidates, and the third letter is the further item of the two candidates. These ‘true’ triplet orderings represent the ‘true’ triplet pool, which we hide from the embedding algorithm. A given sampling algorithm (See §3.3) will then choose one of these triplets to uncover and add to the training dataset for the embedding algorithm. This process simulates what would be done by human raters who would order the requested triplet comparisons based on design similarity.

As mentioned in §2.2.3, our experiments use the Generalized Non-Metric Dimensional Scaling (GNMDS) algorithm proposed in [29] to embed the items given training triplets. Compared to competing methods such as Crowd Kernel Learning [27] and Stochastic Triplet Embedding [6], GNMDS does not assume a probability model to quantify triplet likelihood in the embedding. Rather, the GNMDS algorithm solves a constrained optimization problem by minimizing slack variables for every similarity inequality that is expressed in the triplet pool. In this way, it can converge on an embedding even if all triplets cannot be exactly satisfied in a given embedding space. Although our synthetic dataset guarantees the existence of a true space that satisfies all provided triplets, in our later experiments where we allow fewer dimensions in the embedding space than are needed to model the true space, GNMDS will need to reconcile embeddings where perfectly satisfying all triplets is impossible. This could also occur in cases where there was labeling noise or disagreement between raters, which is common in practice.

3.3 Triplet Sampling Algorithms

To sample the next triplet to be labeled and added to the training pool, we compared two different sampling algorithms: Random Sampling and Uncertainty Sampling. Random sampling selects a random triplet to label from the $n\frac{(n-1)!}{(n-3)!2!}$ possible triplet combinations not already labeled.

Uncertainty Sampling uses bootstrapping similar to that in [33] and detailed in §2.3.1, where uncertain triplets are identified as triplets which contain distances with high variance across the bootstrapped embeddings. Specifically, to quantify the uncertainty associated with a particular triplet, GNMDS creates 10 embeddings by bootstrapping a 75% subset of the triplet pool. This ensures that each embedding is unique but also representative of the vast majority of triplets currently in the pool. For each of the 10 GNMDS embeddings we then compute the distances between every possible pair of items to produce a distance matrix. From the individual distance matrices an average distance matrix and variance matrix are computed across the 10 embeddings.

With the average distance matrix and variance matrix in hand, we then exhaustively identify all possible $n\frac{(n-1)!}{(n-3)!2!}$ triplet combinations. For each triplet combination, the following ratio pi_{ijk} is computed:

$$pi_{ijk} = \frac{\overline{d_{ij}} - \overline{d_{ik}}}{\sigma_{ij} - \sigma_{ik}} \tag{3.4}$$

Here $\overline{d_{ij}}$ refers to the average distance between items i and j and σ_{ij} refers to the standard deviation across the bootstrapped embeddings. This ratio is small when there is either large variance amongst the reference/candidate distances contained within the triplet combination, or

the average distance of each reference/candidate pair is nearly equal. It is not enough that a triplet contains distances which are highly variable, but the variance in the distances must cause high variance in the predicted ordering of the triplet query to be uncertain. Therefore, the uncertainty scoring ratio must capture both the center of the net distance Gaussian as well as the variance.

Each scoring ratio is then assigned a probability from the standard Gaussian cumulative distribution function and ordered by its closeness to 0.5. To receive a probability of 0.5, a triplet must have a scoring ratio pi_{ijk} of 0—which implies that either the distances contained in the triplet are equal or the net distance has near infinite variance. Both of these scenarios are indicative of a triplet with a highly uncertain ordering 'ijk' or 'ikj'. After the unlabeled triplet with probability closest to 0.5 is identified, its true label is uncovered and it is added to the training triplet pool.

Before we can sample any triplets, we need to initialize the training triplet pool with enough information to compute a starting GNMDS solution. To initialize the pool of sampled triplets, we randomly selected two unique triplet combinations for each possible reference coordinate to ensure that the initial GNMDS run had at least some information pertaining to the placement of every item contained in the embedding. This was especially important when selecting the next triplet using uncertainty sampling, which requires not only enough information to run GNMDS with the entirety of the working pool but also enough information to run GNMDS with several randomly-selected 75% partitions needed to compute bootstrap-based uncertainty estimates.

3.4 Matlab Implementation of GNMDS

To implement the GNMDS algorithm, a Matlab script developed by van der Maaten was used [34]. This GNMDS implementation is slightly different than the mathematical derivation

of GNMDS described in §2.2.3. Unchanged, however, is the cost function of GNMDS being the summation of all triplet violation slacks ξ_{ijk}. In the Matlab implementation this cost function is the same but implemented as a hinge loss on the violations, which is equivalent to the GNMDS optimization formulation that imposed a non-negativity constraint on the slack variables. The Matlab implementation is listed below:

```
% Evaluate violation slack on every triplet constraint
slack = max(D(sub2ind([N N], triplets(:,1), triplets(:,2))) + 1 -...
D(sub2ind([N N], triplets(:,1), triplets(:,3))), 0);


% Compute value of cost function
C = sum(slack(:))
```

Here C denotes the cost function which is computed as the sum of all triplet violation $slack$ magnitudes. The hinge loss to compute slack accesses the euclidean distances from the distance matrix D, and takes the difference between each anchor candidate pair in the triplet. The distance between the `triplets(:,1)` anchor and `triplets(:,2)` closer candidate pair should be smaller than the distance between the anchor and the `triplets(:,3)` further candidate pair. A constant of '1' added to prevent collapse on the origin as described in §2.2.3.

To compute the gradient of the hinge loss, the `accumarray` function is used to sum over the pairwise distances in all embedded points which result in a positive slack violation. This is done for each embedded dimension. The pairwise distances create the change in the triplet violation magnitude and the accumulation of them creates the change in the cost function. The Matlab code is detailed below:

```
% Compute gradient
G = zeros(N, no_dims);
viol = (slack > 0);
for i=1:no_dims
    G(:,i) = accumarray(
    [triplets(viol, 1); triplets(viol, 2); triplets(viol, 3)], ...
    [2 .* (X(triplets(viol, 1), i) - X(triplets(viol, 2), i)) ...
    - 2 .* (X(triplets(viol, 1), i) - X(triplets(viol, 3), i)); ...
    - 2 .* (X(triplets(viol, 1), i) - X(triplets(viol, 2), i)); ...
    2 .* (X(triplets(viol, 1), i) - X(triplets(viol, 3), i))], [N 1]);
end
dC = G(:);
```

Essentially this implementation identifies all triplets that are currently being violated and identifies the net directional 'pull' of each violation on an embedded data point—where the magnitude of each pull is the difference in the candidate-anchor distances for violated triplets. The total contribution of these 'pulls' determines the gradient of the embedding with respect to each plotting position.

Finally, with the gradient of the cost function computed, the embedding placements in X are updated through a gradient descent from gradient G. This is not the same as the traditional GNMDS in §2.2.3 which used the Gram matrix K in the optimization. Instead, the coordinate matrix is updated directly as shown below:

```
% Perform gradient update
```

```
X = X - (eta ./ no_triplets .* N) .* reshape(G, [N no_dims]);


% Update learning rate
if old_C > C + tol
    eta = eta * 1.01;
else
    eta = eta * .5;
end
```

Here `eta` define the learning rate or step size of the gradient descent. The learning rate is adaptive and updated to become larger by a multiplier of 1.01 if the cost function is decreased from the previous iteration, and becomes smaller with a multiplier of 0.5 if the cost function has increased. This is done in an effort to better avoid local minima in this non-convex optimization.

3.5 Analysis of Convergence Properties Between Error Metrics

The solid lines in Figure 3.1 show the convergence plots for randomly sampled triplets from a synthetically generated true space. The horizontal axis represents the number of triplets provided to the embedding algorithm—which although synthetically labeled in this example—would be labeled by human raters. Each line represents the average of 10 trials with varying initial triplet pools, while faded regions show the bootstrapped 90% confidence interval. Both the true space and the latent embedding space are two dimensional and contain 10 items for a total of 360 possible triplets. These triplets were all labeled from the same true space.

From the convergence plots, it appears that the GNMDS algorithm initially corrects for the

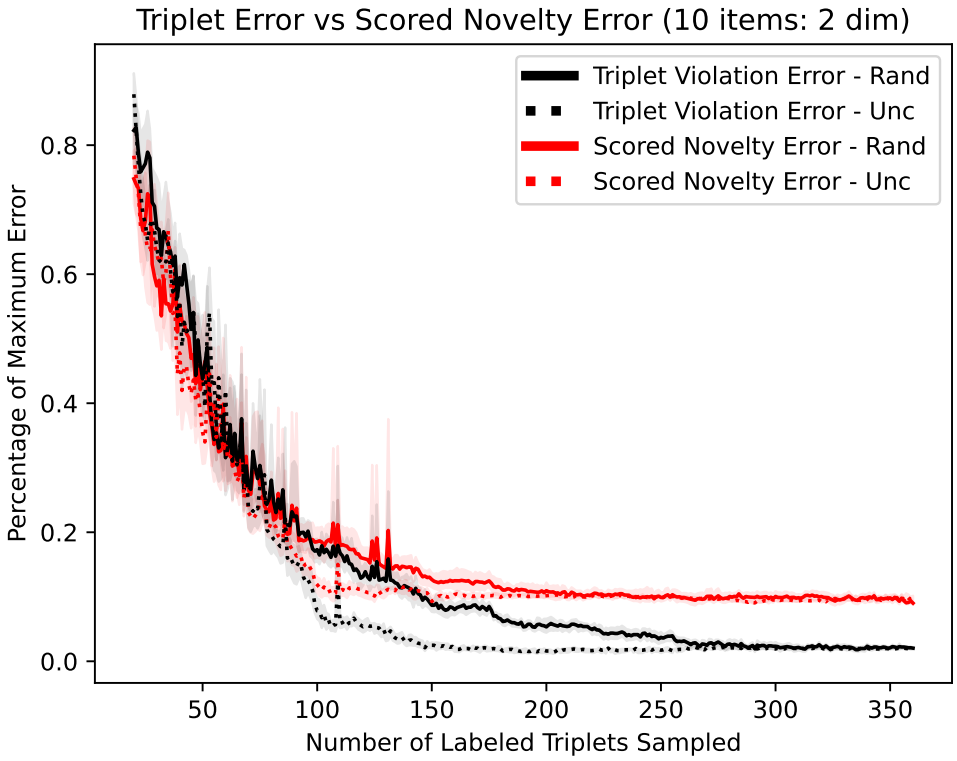

Figure 3.1: Convergence plots for triplets sampled with random and uncertainty sampling for a 2-dimensional true and embedded space

triplet violation error and novelty error at about the same rate. At about 150 triplets, there is some divergence between the two solid-line error metrics which relate to random sampled triplets. Here the scored novelty error is nearly at its final value, while the triplet violation error continues to drop until it settles at closer to 250 triplets sampled. These results support the hypothesis that the novelty information of the embedding can saturate before the triplet violations. Therefore, using triplet violations to signify embedding convergence will demand more labeled triplets than needed for novelty convergence. For randomly sampled triplets, our results show that if only novelty convergence is desired, this can reduce human effort by almost a factor of two which is the dominant cost in triplet embedding methods.

Figure 3.2 takes the same randomly-sampled data from Figure 3.1 where every embedded

instance is expressed by the two error metrics associated with that instance—this allows us to directly compare how changes in triplet violation error correlate to the corresponding novelty score error. From the linearly-scaled left plot, there appears to be a fairly strong linear correlation between the two error metrics. This correlation appears stronger for low error magnitudes, which corresponds to larger pools of sampled triplets, than for small triplet pools with more sparse information about the nature of the embedding space.

To quantify the correlation between the two error metrics under random sampling, the Pearson and Spearman Rank Correlation Coefficients were computed. The Pearson coefficient describes the strength of the linear relationship between two variables under the assumption that they are indeed linearly correlated. For this data, the Pearson coefficient was 0.9482, which is fairly close to 1, and indicates a strong positive linear relationship: when triplet error decrease novelty error decreases at a near-linear rate. The Spearman coefficient measures the strength of association between the two variables without assuming linearity initially. For this data the Spearman coefficient was 0.7908, which is moderately close to 1 and indicates a monotonic relationship between the errors. However, because it is smaller than the Pearson coefficient, this could be evidence that the monotonic relationship is not as linear as the Pearson coefficient implies, and the initial assumption of linearity used for Pearson computation is questionable. Regardless, these metrics do support the conclusion that decreasing triplet violations has a strong relationship with decreasing novelty error.

The right plot of Figure 3.2 shows the same plot but with logarithmically scaled axes in an effort to separate out the correlation of errors for triplet pools containing enough information to generate meaningful embeddings. The integer nature of the triplet violation error can be seen in the horizontal bands of points for small errors. For the smallest triplet violation error band,

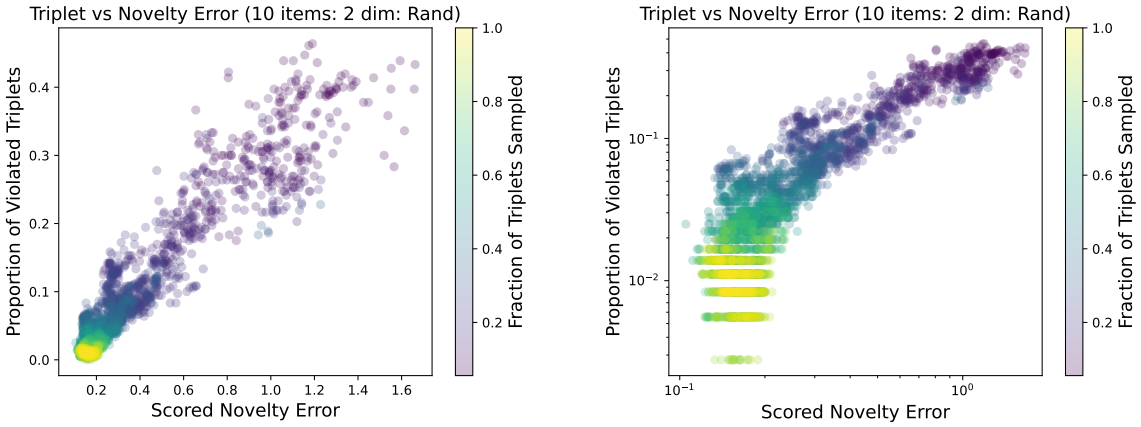

Figure 3.2: Correlation plots for comparison of triplet violation and scored novelty errors

there is still noticeable variance in the novelty error. Notably however, the novelty error sees its lowest values before the lowest triplet violation band and does not substantially decrease across the bands where the triplet violation error drops to its final value.

From the correlation plots for the two error metrics, it appears that as GNMDS is working to reduce the triplet violations in the embedding, the novelty information is learned at a near-equal rate. This is a promising sign for the method of using GNMDS to reproduce the novelty information of the design space. Novelty information is tightly linked to triplet queries, such that satisfying more labeled queries—even randomly chosen ones—will have a return on investment in reproducing the novelty in the design space.

The dotted lines of Figure 3.1 shows the error convergence plot for triplets sampled through uncertainty sampling. Unlike for random sampling which saw divergence in the novelty error and triplet error plot trajectories at around 150 triplets sampled, this divergence can be seen at about 100 triplets sampled with the uncertainty sampling method. Notably, the triplet violation error still takes more triplets to approximately converge to its final value, however unlike with random sampling both error plots are nearly settled at the point that they diverge. Whereas in random

sampling the triplet violation error benefited from gathering more triplets after the novelty error had stabilized, in uncertainty sampling the triplet violation error only finds slightly more triplets valuable after the novelty error has settled.

Both error metrics still appear to be learned at a near-equal rate for the bulk of the uncertainty sampling. By prioritizing informative triplets which are difficult to satisfy, GNMDS is able to identify the minimum number of triplet violations with fewer labeled constraints. These uncertain triplets appear to greatly inform the holistic layout of the embedding. Under random-sampling, GNMDS was forced to uncover triplets that it was fairly certain it already guessed correctly, and as a consequence required more labeled triplets until it's assumptions were challenged. With regard to the novelty scores, these 'hard-to-learn' triplets often require significant reordering to satisfy, thereby forcing GNMDS to prioritize exploration in the early-stages of sampling.

As for the smaller gap in saturation between the two error metrics than for the random-sampling counterpart, this is again likely thanks to saving the 'assumed-correct' triplets for the final stages of labeling. With 100 'hard-to-learn' triplets prioritized, the remaining triplets have little benefit to the triplet violation error, as most were already labeled correctly. While the embedding still learns the holistic novelty of designs using fewer triplets than the violation error, uncertainty sampling produces closer estimates in fewer samples.

Because GNMDS is defined by a minimization of the triplet violation error, the novelty error will converge in as many triplets as the triplet violations or fewer. In all of our testing, the novelty error saturated in fewer triplets than the triplet violation error. Additionally, the results of Figure 3.1 show that there exist design-space distributions for which the GNMDS-reconstructed novelty error saturates in *significantly* fewer triplets than the triplet violation error. Despite the

small number of samples tested—given that the novelty error convergence is bounded by the triplet violation error and that we have demonstrated cases where the gap between the two is large—we believe that this is enough evidence to motivate a new method for identifying the finite convergence behavior of reconstructed novelty in ordinal embeddings.

If existing conservative bounding functions would demand more triplet samples than needed for novelty convergence, is there a way to evaluate the degree of convergence of the novelty information as triplets are labeled one-by-one? This would require a metric that indicates novelty convergence of the embedding, but does not require existing knowledge of all labeled triplets to compute.

Chapter 4: Estimating Novelty via the Expected Model Change

While the novelty error is useful for capturing the saturation of design novelty scores in the embedding, it unfortunately requires complete knowledge of the position of designs in the design space. (The same is true for computing triplet violations on unseen triplets.) In practice, we do not know the true design space and the only insight we have toward the layout of the design space gradually comes in the form of the human-labeled triplet comparisons as they are ordered. The novelty error, therefore, cannot be directly used for online evaluation of the embedding saturation. Even the triplet violation error cannot be used in this context, as it requires all triplets to already be labeled to evaluate.

4.1 Defining the Expected Model Change

While it is true the exact magnitude of the novelty error cannot be computed without knowledge of the true-space orientation of designs, it is noteworthy that 'true' centroidal-distances used in the computation of the novelty error are fixed values, and so the change in the novelty error is defined only by the change in the embedding itself. From §3.1, if the centroidal distances of items in the true space C_{true} are fixed across triplet sampling iterations, only the centroidal distances of the embedding C_{emb} are causing change in the novelty error. In theory, if the embedding saturates to the point where an added triplet evokes little change in the centroidal distances within

the embedding, it follows that the true novelty error should also be saturated around the error for the fully-sampled triplet pool.

If monitoring change in C_{emb} is the key to understanding novelty information convergence, we need a protocol for how exactly a time history of C_{emb} values should be stored to measure the updates. The naive answer would be to store the C_{emb} values for the k-th iteration of triplet labeling and the $k+1$-th and measure the magnitude of the update $\Delta C_{emb} = C_{emb}^{(k+1)} - C_{emb}^{(k)}$. Unfortunately, the nature of triplet sampling sometimes means uncovering a triplet label that is already satisfied by the embedding, despite the triplet being unlabeled. If this happens, ΔC_{emb} might reach a value of 0 before saturation is actually occurring. One potential work-around then would be to measure the ΔC_{emb} of the last 5 or 10 updates such that it is unlikely that all of the updates in that window are the result of adding an already satisfied triplet. This could work, but it is unclear how large the window would have to be to ensure that triplets which challenge the current embedding are captured, and the larger the window, the less-effective this method is at minimizing labeled data and immediately identifying convergence.

Can we instead save on triplet labeling costs by instead of measuring change from previously labeled triplets, look at how future triplets will shape C_{emb}? Well, anytime the random or uncertainty sampler identifies a triplet to be labeled, there are only two possible labels for that triplet. If we envision the C_{emb} update for both triplets, we can anticipate how future triplet labeling will impact the novelty error before we invest the resources. Even better, it is guaranteed that one of those possible triplet outcomes challenges the current embedding orientation, and so one of them should produce a detectable ΔC_{emb}. We refer to this detectable update as the the **Expected Model Change** (EMC).

To compute the EMC, two embeddings are created each time an uncertain triplet is identi-

fied by the sampler: once for the true ordering of the triplet (*e.g.*, 'ABC') and once for the inverse of the true ordering (*e.g.*, 'ACB'). It is possible that when the true label 'ABC' is added to the triplet pool that the embedding already inadvertently satisfied 'ABC' without the triplet explicitly appearing as a constraint. If we only consider the model change for adding the true triplet, these instances would produce a model change of 0. By considering the model change for adding both the true and inverse triplet, it is guaranteed that the current embedding violates exactly one of these constraints and should produce a measurable model change in that instance. Although in the computation framework of the EMC we refer to the 'true' and 'inverse' ordering, this process is done before the label is obtained, and so it is unknown to the algorithm which triplet ordering is which. This will not be important for the calculation, and either triplet can be considered the 'true' or 'inverse' in the calculation framework as long as both outcomes are considered.

At this point there are 3 active embeddings produced from GNMDS: the current embedding of the training triplet pool, the embedding produced by adding the 'true' triplet, and the embedding produced by adding the 'inverse' triplet to the pool. For each embedding, the centroidal distances C_{emb} are computed and normalized by the largest distance for the embedding like in § 3.1. As a convention we refer to $C_{emb}^{(0)}$ as the centroidal distances from the currently labeled pool, $C_{emb}^{(1)}$ as the envisioned distances when the 'true' triplet label is added, and $C_{emb}^{(2)}$ as the envisioned distances when its inverse compliment triplet label is added.

We then compute ΔC_{emb} updates by taking the difference between the current novelty matrix and the novelty matrix after adding a new triplet. Here, $\Delta C_{emb}^{(1)}$ refers to the update from adding the true triplet and $\Delta C_{emb}^{(2)}$ the update from adding the inverse triplet.

$$\Delta C_{emb}^{(1)} = \frac{C_{emb}^{(0)}}{max(C_{emb}^{(0)})} - \frac{C_{emb}^{(1)}}{max(C_{emb}^{(1)})} \tag{4.1}$$

$$\Delta C_{emb}^{(2)} = \frac{C_{emb}^{(0)}}{max(C_{emb}^{(0)})} - \frac{C_{emb}^{(2)}}{max(C_{emb}^{(2)})} \tag{4.2}$$

If we have added a triplet that is already inadvertently satisfied by the training pool embedding, then we would expect that one of those model-change matrices should be comprised of near-zero updates. It is possible, however, that for small pools of triplet labels that the optimization is not constrained enough and there is a large update even if the triplet is already satisfied. For these reasons, we compute both updates instead of only computing the update for the unsatisfied triplet. This also allows us to store the predicted embedding results before the triplet is labeled and pull from the corresponding embedding after it is labeled without rerunning the GNMDS solver.

Finally, the L1 norm is taken for both ΔC_{emb} updates to represent the collective deviation in centroidal distances after the triplet is added. The EMC is then defined as the larger of the two updates—which should be from the triplet that is not already satisfied in the embedding. A multiplier of $\frac{1}{n}$ is included to make it an average update among designs.

$$EMC = \frac{1}{n} max(\left\|\Delta C_{emb}^{(1)}\right\|_1, \left\|\Delta C_{emb}^{(2)}\right\|_1) \tag{4.3}$$

When the EMC is near-zero, the average novelty update is small for both of the two possible triplet combinations. This means that the embedding has saturated such that new labeled triplets will not majorly impact the identified novelty information, regardless of the triplet label.

4.2 Using the EMC on Synthetic Data

In Figure 4.1 we explore the relationship between the expected model change and the novelty error by comparing the rate of saturation for each metric for 10 items under uncertainty sampling with the same data from Figure 3.1. The top plot shows the raw magnitude change in each metric after a single triplet is added to the working pool. As expected for both metrics, the embeddings are heavily influenced by newly added triplets when the triplet pool is small, but as the pool grows the embeddings begin to settle on an orientation that is robust in spite of newly added triplet constraints. More importantly, both error metrics appear to display this saturation at the same moment—shown by the metric change plots flat-lining at about 50% of the triplet pool sampled. Note again that the y-axis is not the novelty error and EMC themselves, but the change in each value between subsequent triplets.

Without scaling, it is apparent that the raw novelty error does not approach a magnitude change of zero even when nearly all triplets are sampled, and is thus never truly resilient to added triplets. We hypothesize that this offset is a consequence of the information loss imparted by using triplet comparisons to describe the true space. Even when all triplets are satisfied, GNMDS has the flexibility to slightly adjust the relative placement of items without violating any triplets. This would not be a problem to the internal triplet violation loss, but these allowable perturbations will result in slightly different true novelty errors. If this is the case, it would follow that an embedding with large number of n items would near a magnitude change of 0—as it is more difficult for the embedding to slightly perturb the placement of items without violating one of the many triplet constraints. The bottom plot of Figure 4.1 represents the same data with the raw magnitude change values scaled by the first-recorded error corresponding to the initial

triplet pool. In this scaled plot it is more apparent that when the expected model change stops updating that the true novelty error for the embedding has also stopped updating. In later sections of the paper, we modify the true versus embedding dimensions for GNMDS, and while we do not provide equivalent plots to Fig. 4.1 in those cases, we observed consistent behavior where the expected model change converges at or slightly after the novelty error.

If the results in Figure 4.1 hold true, then the saturation of the expected model change could serve as a proxy for the saturation of the true novelty error. A policy could be implemented to stop labeling triplets on the basis of online evaluation of the expected model change convergence curve. This policy may look something like computing the variance of the expected model change corresponding to the last k triplet entries and comparing that variance to a small stopping-criterion variance. Alternatively, because the EMC is expressed as an expected novelty update under newly sampled triplets, a cutoff tolerance value could be proposed. A tolerance value of say 0.1 for example would mean that when EMC magnitudes reach 0.1 that the deviation in relative novelty error is expected to be 0.1, which might be deemed 'good enough' to stop sampling.

With a method for evaluating the status of the embedding saturation online, determining conservative theoretical bounds like those found in [8] is unnecessary. However, up until now we have assumed that the dimension of the true space is the same as the embedding space. In reality, we can only guess at the dimension of the true space when providing it as a parameter for GNMDS. How do we choose this dimension and what are the consequences to this procedure if our approximation is bad? The next chapter explores this question.

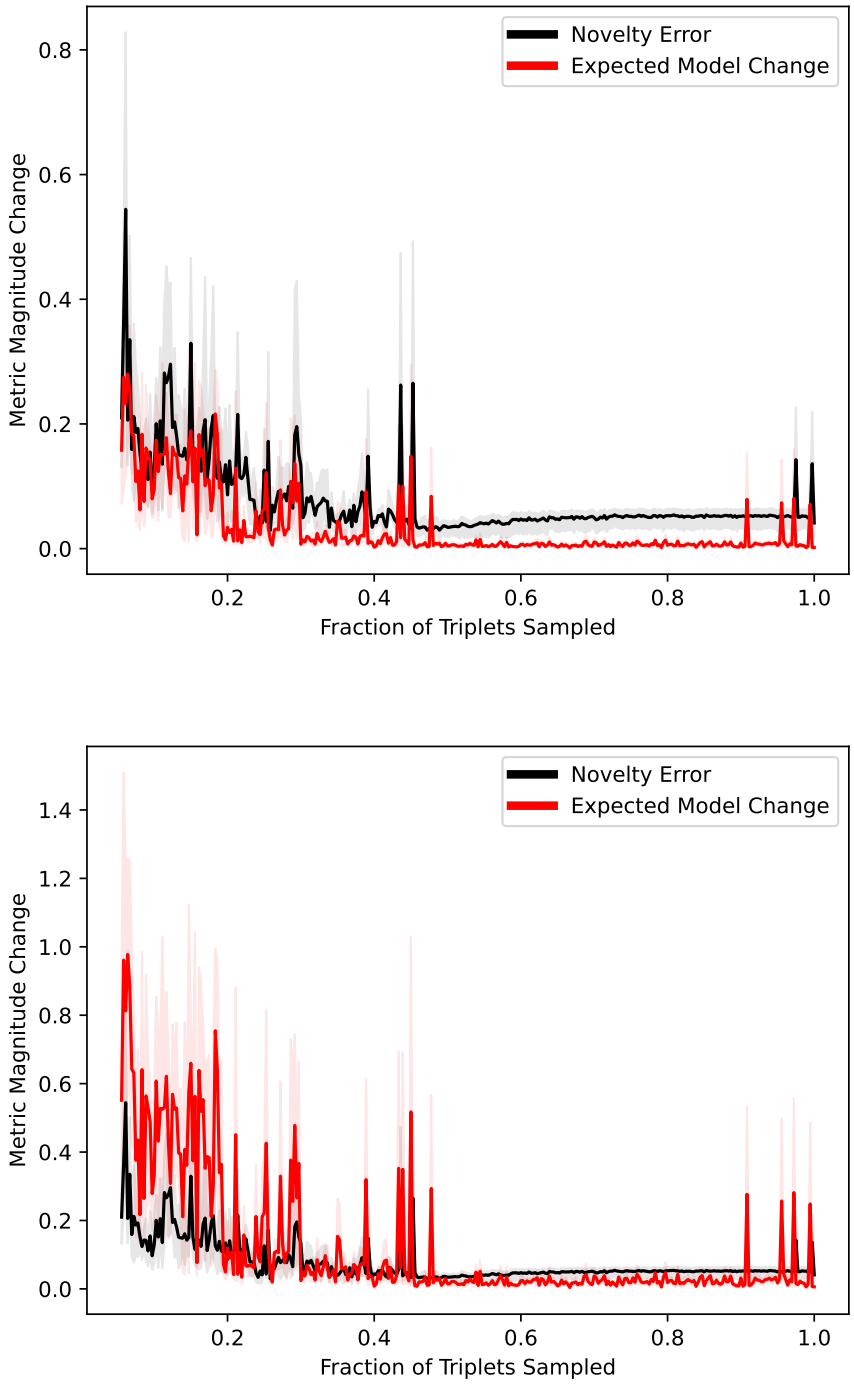

Figure 4.1: Raw (top) and scaled (bottom) magnitude change plots for the true novelty error and expected model change after a new triplet is added to the working pool. Scaled plot normalizes the raw magnitude change by the initial error value from the smallest triplet pool.

Chapter 5: Estimating the Embedding Dimension

With the expected model change, we can detect the saturation of novelty information without any knowledge of the design space apart from the labeled triplets. Additionally, however, GNMDS requires the user to specify an embedding dimension in order to create an embedding at all. This would not be a problem if the design space was of a known dimension, but with only ordered similarity comparisons it is unclear how the dimension of the design space would be derived—or even approximated.

In our previous tests which used synthetically generated data, the embedding dimension was specifically chosen to be equal to the known true space dimension. In reality, we would need some policy for choosing an embedding dimension. Poor approximation of the design space dimension could likely have disastrous consequences to the accuracy of the embedding's novelty scores. Given too few dimensions to work with, the embedding algorithm might have to tolerate a noticeable quantity of triplet violations which it cannot feasibly satisfy. Given too many dimensions to work with, the embedding algorithm might have no problem feasibly satisfying triplets, but with so much flexibility, it could end up with an embedding that cannot produce accurate novelty distances, compared to using the true design space dimension. Before developing a policy for choosing an appropriate embedding dimension, we need to better understand the consequences of a poor embedding dimension on the reported novelty scores.

5.1 Dimension Approximation on Triplet Violations

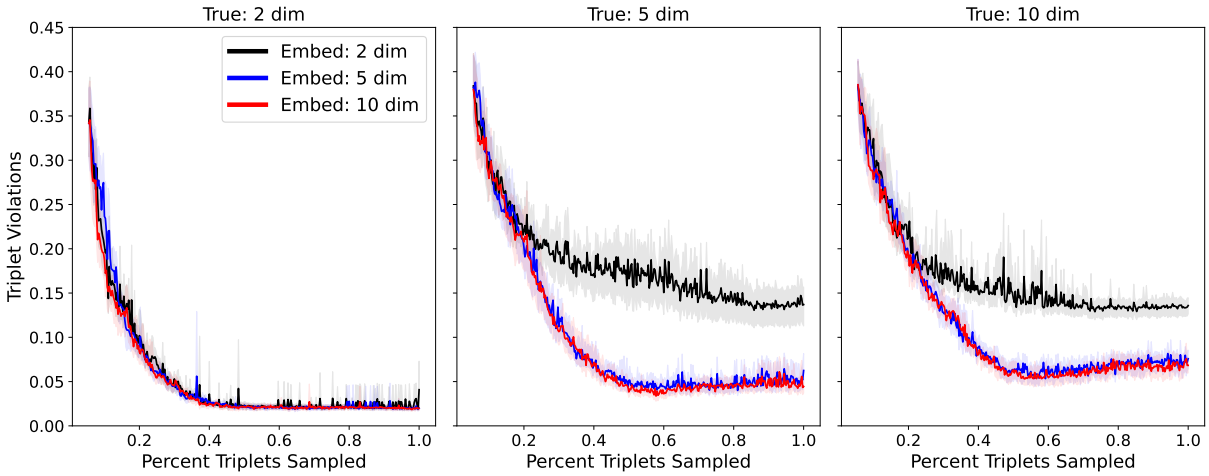

Figure 5.1: Triplet violation error curves under varied true and embedding dimensions.

In Figure 5.1, the average triplet violation curve for 10 trials of a 10-item true space is shown for various true and embedded dimensions. In each subplot, three curves are presented, with each curve representing a different embedding dimension given to GNMDS. All three curves share the same hidden triplet pool and correspond to the same synthetically-generated 'true' space. In addition to the 2, 5, and 10 dimensions shown in Figure 5.1, we also tested dimensions from 2 to 10 in unit increments, as well as 15 and 25 dimensions. Qualitatively those results mirrored the overall behavior of Figure 5.1, so we did not include them in the manuscript to make the figures more readable. This was also true for the 2, 5, and 10 dimensional subplots explored later in the manuscript.

In the left most subplot for a two-dimensional true space, it can be seen that all three curves are almost exactly in line with each other. The black curve represents the case where the embedding dimension exactly matches the true space dimension—in this case two dimensions. The blue and red curves both represent instances of over-approximation—using five or ten embedding di-

mensions when only two were needed. From this plot, it does not appear that over-approximating the dimensionality of the true space comes at any expense to the learning of triplet violations.

In the middle subplot of Figure 5.1, curves for a five-dimensional true space are shown, with the black curve representing an under-approximation case of using only two dimensions. While the exact and over-approximation curves look roughly identical, the under-approximation curve has large error bars and saturates at a much higher final triplet violation value. This result is consistent with our hypothesis. When fewer dimensions are provided to GNMDS than needed for an exact construction, the algorithm must tolerate a large proportion of triplets which cannot be feasibly satisfied without violating existing labeled triplets. This puts a limit on the minimum triplet-violation error which can be achieved by GNMDS—a limit that is noticeably larger than the converged error for an exact-approximation.

In the rightmost subplot of Figure 5.1, for a true-dimension of ten, the under-approximation case of using only two dimensions shares the above behavior. However, Figure 5.1 also features the under-approximation case of using only five-dimensions, and this curve appears to be in-line with the exact approximation case. While under-approximation of the true space can lead to a much larger triplet violation error than when an exact-approximation is used, the severity of data loss appears to be governed by the severity of the approximation error. Using only two dimensions to reproduce a ten-dimensional true space omits 80% of the needed dimensions, resulting in a significant handicap on the algorithm's ability to reproduce the triplet orderings in the true space. On the other hand, using five dimensions to reproduce the ten-dimensional true-space results in a more-palatable omission of 50% of the needed dimensions—which in the case of a ten item design space does not appear to be severe enough to hinder GNMDS's ability to satisfy triplet labels. Future research is needed to better determine when the dimension

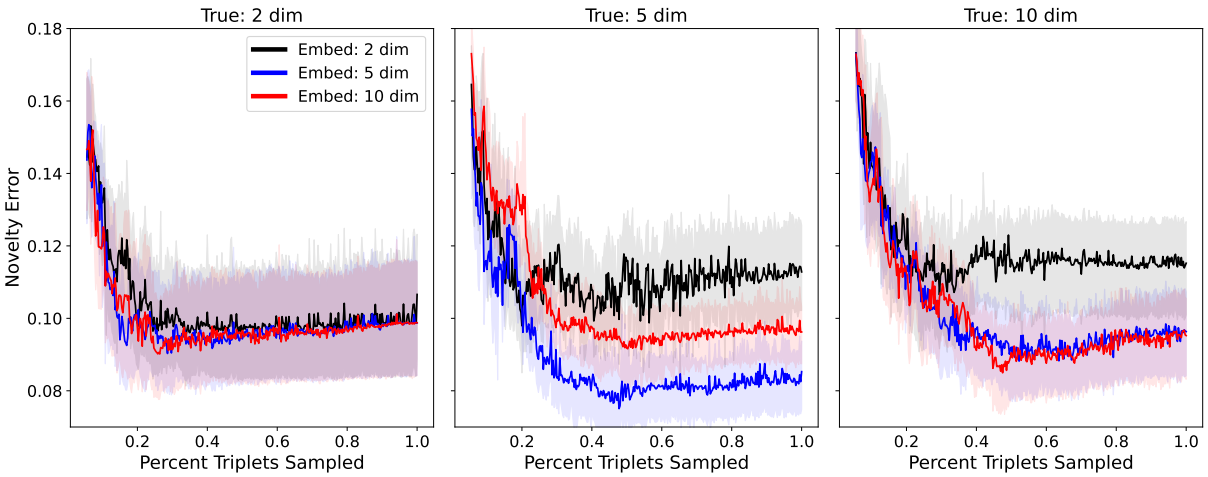

Figure 5.2: Novelty error curves under varied true and embedding dimensions.

approximation error is severe enough for the triplet violation error to be affected. The severity is likely linked to the ratio between the true and embedding dimensions, however the 2/5 ratio (40% of dimensions captured) in Figure 5.1 lead to significant data loss while the 5/10 ratio (50% of dimensions captured) did not. It therefore appears to be the case that larger dimensional true spaces are more resilient to the data loss from the projection onto a smaller embedding dimension.

5.2 Dimension Approximation on Novelty Error

Figure 5.2 represents the same trials used in Figure 5.1 but displays the scored novelty error rather than the triplet violation error. Unlike for the triplet violation error, in the two-dimensional and five-dimensional true-space subplots, there is noticeable separation between the exact and over-approximation novelty error curves. In the two-dimensional true space subplot, the exact approximation curve actually settles at a worse novelty error than the over-approximation cases, although the separation is not large enough to generalize provided the large error bars.

For the five-dimensional true space case however, the exact-approximation drastically out-

performs *both* the under *and* over-approximation cases. While over-approximating the dimension of the design-space did not appear to influence GNMDS's ability to satisfy triplet constraints, it does appear to negatively influence the accuracy of the converged embedding's novelty scores. With more dimensions to work with than needed, GNMDS is given a lot of slack in its effort to satisfy triplet constraints. This would not appear to be a problem if only the triplet violation error was used to assess embedding accuracy, however in terms of novelty error, GNMDS is now able to create item orientations that are drastically different than the true-space orientation *without* those perturbations resulting in violated triplet constraints. With large over-approximation error in the dimension selection of the embedding, the triplet violation error becomes a weaker metric for evaluating embedding accuracy.

It is noteworthy however that the over-approximation curve in the five-dimensional subplot still outperforms the under-approximation curve with regard to novelty accuracy. In this plot, although over-approximation loosens the restrictions imposed by the triplet constraints, loosened restrictions appear to be preferable to the data loss imparted through low-dimensional projection. In the ten-dimensional subplot, under-approximation of the true-space dimension continues to exhibit higher novelty error than using the exact number of dimensions. Given these results, a policy for choosing the embedding dimension might be to err on the side of over-approximation, as it produces a more palatable novelty error than under-approximation. However, over-approximation is still detrimental to the accuracy of the embedding, so a policy that chooses an embedding dimension close to the exact dimension would be valuable.

Although the true novelty error can never be observed in practice, online evaluation of the expected model change may be able to guide the tuning of the embedding dimension. In Figure 5.3 the expected model change is observed under various dimensions—this time with

each subplot corresponding to a specified embedding dimension. In the leftmost subplot, the red and blue curves correspond to the 2:5 and 2:10 cases of under-approximation. The average model change curve in these instances is both noisier and slower to settle than the 2:2 curve where dimensions exactly match. In the center subplot, the red curve represents the 5:10 case of under-approximation. This curve is mostly indistinguishable from the exact-dimension curve, much like the 5:10 case was in Figures 5.1 and 5.2. The expected model change appears to be effective at characterizing cases of extreme under-approximation—which were the cases of the largest true novelty and triplet violation errors in Figures 5.1 and 5.2.

5.3 Online Evaluation of Embedding Dimension

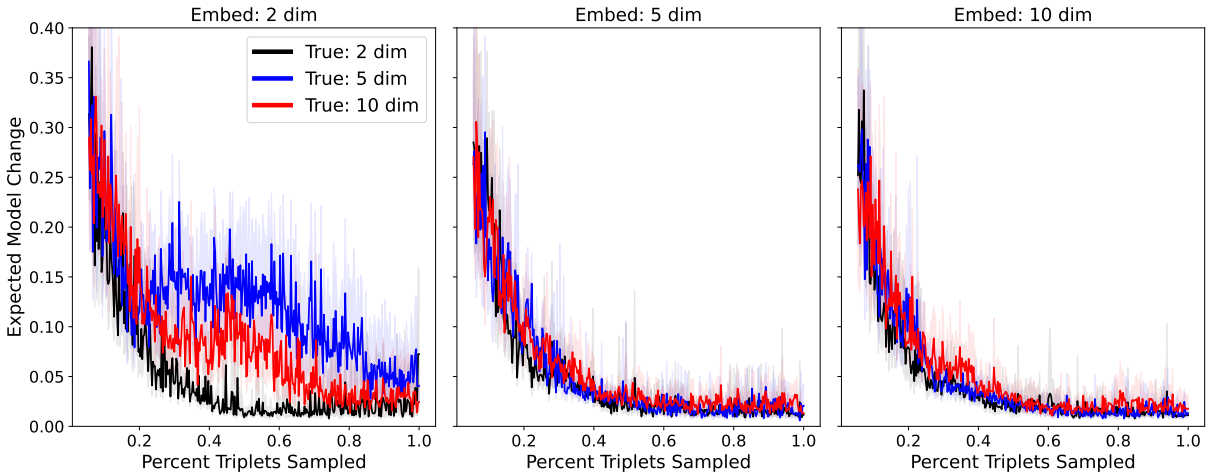

Figure 5.3: Expected model change curves under varied true and embedding dimensions.

In the center and rightmost subplots of Figure 5.3, cases of over-approximation appear indistinguishable from cases using the exact number of dimensions. We know from Figure 5.2 that the true novelty error for the 10:5 case of over-approximation yielded a larger average novelty error than the exact 5:5 case. This does not appear to be reflected in the expected model change.

The model change can only detect differences in the saturation rate of the embedding. In cases of over-approximation, triplets are satisfied at the same rate as the exact case, the exception being that the novelty error plateaus at a larger value. The expected model change will not be effective at identifying over-approximation of the unknown true dimension.

Provided these characteristics of the expected model change, a policy for tuning the embedding dimension may look like the following: (1) choose a large embedding dimension for GNMDS, (2) sample enough triplets until saturation tolerance has been reached for the EMC, (3) choose a smaller embedding dimension and give the already labeled pool back to GNMDS, (4) if the new EMC curve is noticeably slower to saturate than the previous EMC curve (a) record the previous dimension as the 'best' approximation, if not (b) repeat with a smaller dimension.

We acknowledge that this method of tuning the embedding dimension is only a rough heuristic, however, our results suggest that (1) the expected model change does not appear have its finite sample convergence proprieties change when the dimension is over-approximated and (2) gross under-approximations of the embedding dimension can manifest in the finite sample convergence. Choosing an appropriate embedding dimension for GNMDS remains and open problem, and future testing with the expected model change is needed. It is encouraging, however, that the expected model change—which governs triplet sampling in our method—does not appear to require more triplet sampling when the dimension is over-approximated. Thus, for our method, we recommend erring on the side of over-approximating the embedding dimension. We believe the expected model change could provide be a key feature in determining the error in dimension approximation, provided that expected model change continues to exhibit different behavior when grossly under-approximating the dimension.

An alternative method for approximating the embedding dimension might involve reserving

a portion of human-provided triplets as a validation set. These validation triplets would be used to test the model under various embedding dimensions, with the expectation being that models with lower validation scores should indicate under-approximation of the true dimension (which leads to loss of triplet information). While this method provides a more concrete protocol for choosing the embedding dimension than the rough heuristic method from the EMC, there are a couple of problems with this approach. Notably, the dominant cost of ordinal embeddings is the cost of triplet labeling, and this method requires a sizable amount of validation triplets to ensure fair representation in the triplet violation score. Additionally, it is unclear how this method would be used alongside the bootstrap-uncertainty triplet sampling, which informs triplet sampling from the current status of the embedding. Withholding validation triplets which were chosen from uncertainty sampling might cause leakage and prevent these triplets from being used to evaluate the dimension approximation.

While the above results were all conducted using synthetically generated data, the next chapter considers how the results might change if applied to more complex datasets including those found in the real world.

Chapter 6: How Do Our Results Extend to Real-World Design Spaces?

In all of our testing prior to this section, the triplet pool given to GNMDS was generated from a synthetic uniformly-random design space, and all observations from previous sections were specific to this type of synthetic data. In this chapter, we wish to explore the impact of design space distributions which are not uniformly-random to see if our observations generalize to a variety of potential design space distributions.

6.1 Effect of Design Space Sparsity and Clustering

To test the generalizability of our previous observations regarding the behavior of the novelty error and EMC, our initial goal was to create a sort of 'torture test' distribution that would best contrast the uniformly-random data. To meet this goal, it was important to reconsider what exactly triplet constraints impose on the embedding construction and what they reveal about the unknown true space.

Triplet labels and constraints only provide relative information about the true space. Through triplet labels, the embedding will be able to group similar designs together, but the effectiveness of the absolute coordinates is highly dependent on the construction of the embedding algorithm and the nature of the triplet constraints. Purely relative spatial reconstruction is fine for studying the minimization of triplet violations, but centroidal-distance-based novelty scores are influenced

by the absolute coordinate reconstruction accuracy. Poor absolute coordinate accuracy will result in worse converged novelty errors, and may even impact the convergence rate of the novelty error and EMC.

But what design spaces lead to more informative triplets on the absolute placement of designs? Well, from our studying of triplets and previous work from active sampling in §2.3.1, we know that the triplets which best inform the embedding are triplets which have high label uncertainty. In other words, informative triplets are those which contain two anchor-candidate pairwise distances that are close to each other. Suppose we have all 360 labeled triplets for a known design space of ten designs. If in the true space, all ten of these triplets have near-equal pairwise Euclidean distances, then these triplets impose high restrictions on the embedding in order to be satisfied. Because the triplets all represent small margins of similarity disagreement, slight perturbations in the placement of designs in the embedding will be more likely to result in violated triplets than for a pool of triplets with large disparity in the pairwise distances. Ordinal embedding algorithms like GNMDS can only evaluate their success on triplet violation accuracy. If an embedding contains many triplets which correspond to large disparity of pairwise distances in the true space, then pertubations in embedding placement are less likely to result in violated triplets, and the ordinal embedding algorithm might achieve perfect triplet reconstruction despite poor reconstruction of distance magnitudes. For these reasons, we believe that triplet pools which have many 'close' or '50/50' triplets should result more informed and less variable absolute construction accuracy in distance magnitudes.

But what does this mean practically for what true spaces should look like for better reconstruction of absolute distances through triplet embedding? If triplets which are 'close' and '50/50' when trying to satisfy are the most informative, then spaces with the maximum number of

these triplets should be more practical for absolute magnitude reconstruction. These correspond to sparse design spaces—where designs are near-uniformly scattered and have representation spread around the full design space. If this hypothesis is correct, then our previous experiments with uniformly-random synthetic design spaces are the 'best-case' for distance magnitude reconstruction, and spaces with dense clusters are more 'torturous' for absolute construction accuracy.

When the design space contains a dense region, it follows that there are larger areas without a single design represented then there would be for a uniformly-random space of the same design count. Dense regions in the design space results in a large quantity of 'loose' triplet constraints. For example, suppose a design space contains ten designs labeled 0-9. Now suppose that designs 0-8 are very similar and are plotted closely together in the true space, while design 9 is very dissimilar and plotted far from the other designs. Every triplet which compares a reference to design 9 would be a 'loose' triplet. In other words, these triplets will not inform the true position of design 9. Every triplet containing design 9 will tell GNMDS it requires the reference design plotted closer to the 'not-9' competing candidate. The embedding will quickly learn that 9 must be plotted far away from the other designs, and may even have no trouble labeling 9 as the 'most-novel' design, but it will have no idea *how far* from the other designs 9 should be plotted. All of these triplets will be satisfied if design 9 is plotted one unit from the nearest design, and they will still be satisfied if design 9 is plotted 1000 units from the nearest design. Because the novelty error is computed based on the magnitude of centroidal distances, this may result in large variance in the novelty error.

6.1.1 Developing Synthetic Design Spaces of Varying Sparsity

To test our hypothesis and study the impact of design-space sparsity on the embedding reconstruction accuracy metrics, a synthetic dataset of design spaces had to be generated. This dataset would have to explore varying degrees of design-space sparsity and categorize each design-space by its level of sparsity. This demanded a way of evaluating the sparsity of a design-space.

As mentioned in §6.1, sparse design-spaces are near-uniformly spread throughout the space. In the lense of pairwise distances, uniformly spread means that the distribution of pairwise distances has little variance. This is contrasted with a design-space with dense clusters. For these spaces, pairwise distances within clusters are small, while paiwise distances between clusters are large—resulting in large variance in pairwise distance.

Using this intuition, we define pseudo-code for calculating the space-density metric λ as follows:

1. Compute the matrix of pairwise distances in the design-space D
2. Take the upper triangle of the symmetric pairwise distance matrix D to avoid double counting distances. Trim the diagonal which should be zeros as this represents the distance of a design with itself. This reshaped distance matrix will be referred to as T.
3. Compute the standard deviation of the elements of the reshaped distance matrix T. This standard deviation is λ.

With a metric to define the density/sparsity of the design space, the next step was to generate many design spaces and order them based on their density λ. To do this, we generated

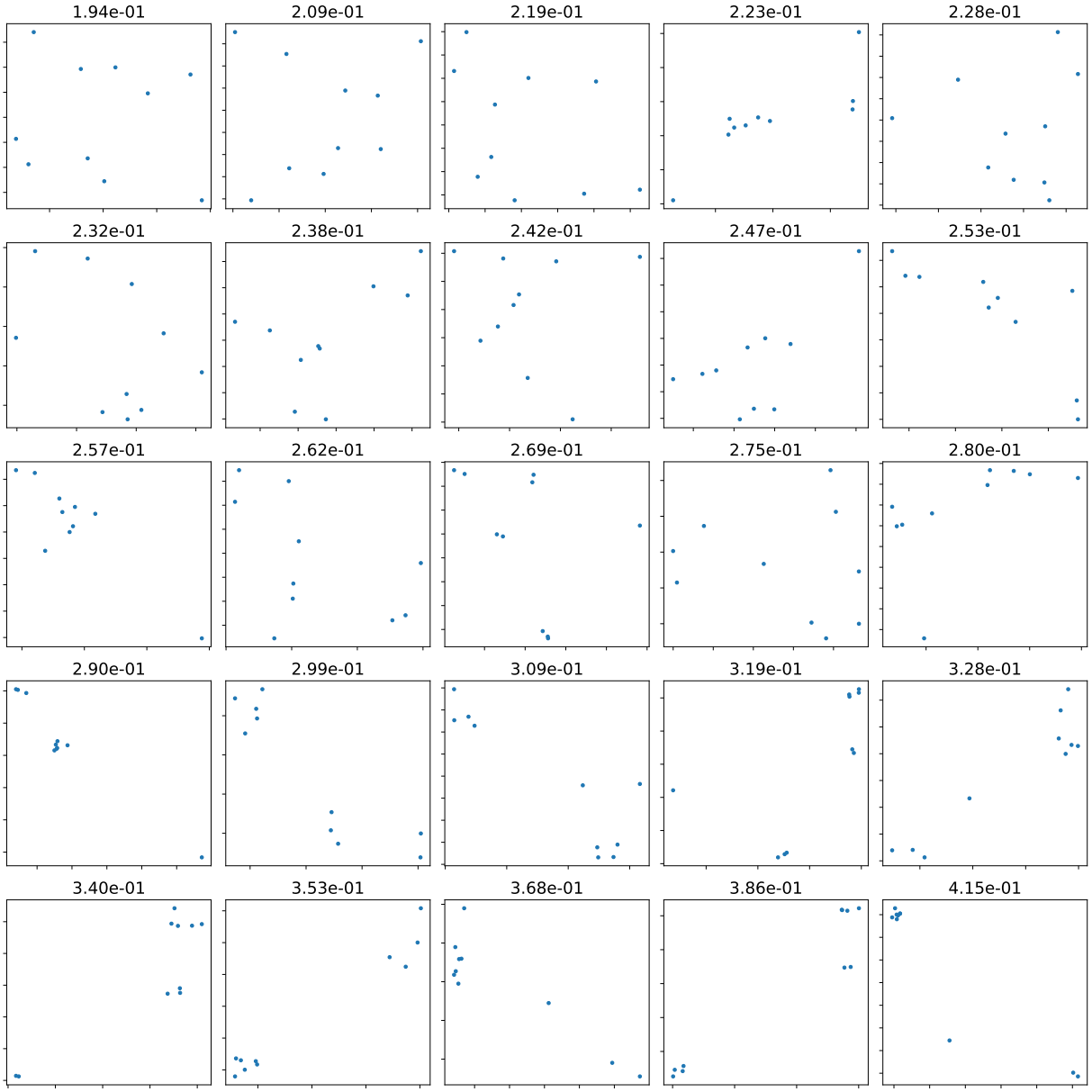

Figure 6.1: Sparsity-ordered synthetic design spaces. Spread value for each plot is shown at the top.

ten thousand synthetic spaces from uniform, normal, and exponential random generators. Each generator was also given a random number of cluster centers between one and five and a random cluster sampling standard deviation between 0.1 and 1. These ten thousand synthetic spaces were then sorted by increasing density value λ (decreasing sparsity), and 25 design spaces were selected at evenly spaced percentile values from the range of λ values. The 25 design spaces for the two-dimensional version of this process are shown in Figure 6.1. This was also repeated for 5 and 10 dimensions.

In the synthetic design spaces in Figure 6.1, it is clear that larger λ values decrease the sparsity of the space and increase clustering along the perimeter of the design space. It is noteworthy that no magnitudes are provided in the synthetic subplots. This is because the domain of the design space is defined by the designs furthest from the centroid both for the novelty error and the triplet representation. These subplots provide a visual for how the designs tend to occupy the design space with varying λ.

6.1.2 Design-Space Sparsity on Convergence Properties (Exact Dimension)

Now with the sparsity-ordered synthetic data in hand, we are able to observe the impact of design space sparsity on the embedding reconstruction metrics. In Figure 6.2, the error convergence plots are provided for all 25 sparsity-ordered plots in 2, 5, and 10 dimensions. The color bar on the right of each row of subplots relates to increasing density value λ. Light yellow-colored points correspond to spaces with dense clusters while dark-colored points correspond to sparse design spaces.

Overall, the convergence plots for the triplet violation error in Figure 6.2 appear fairly con-

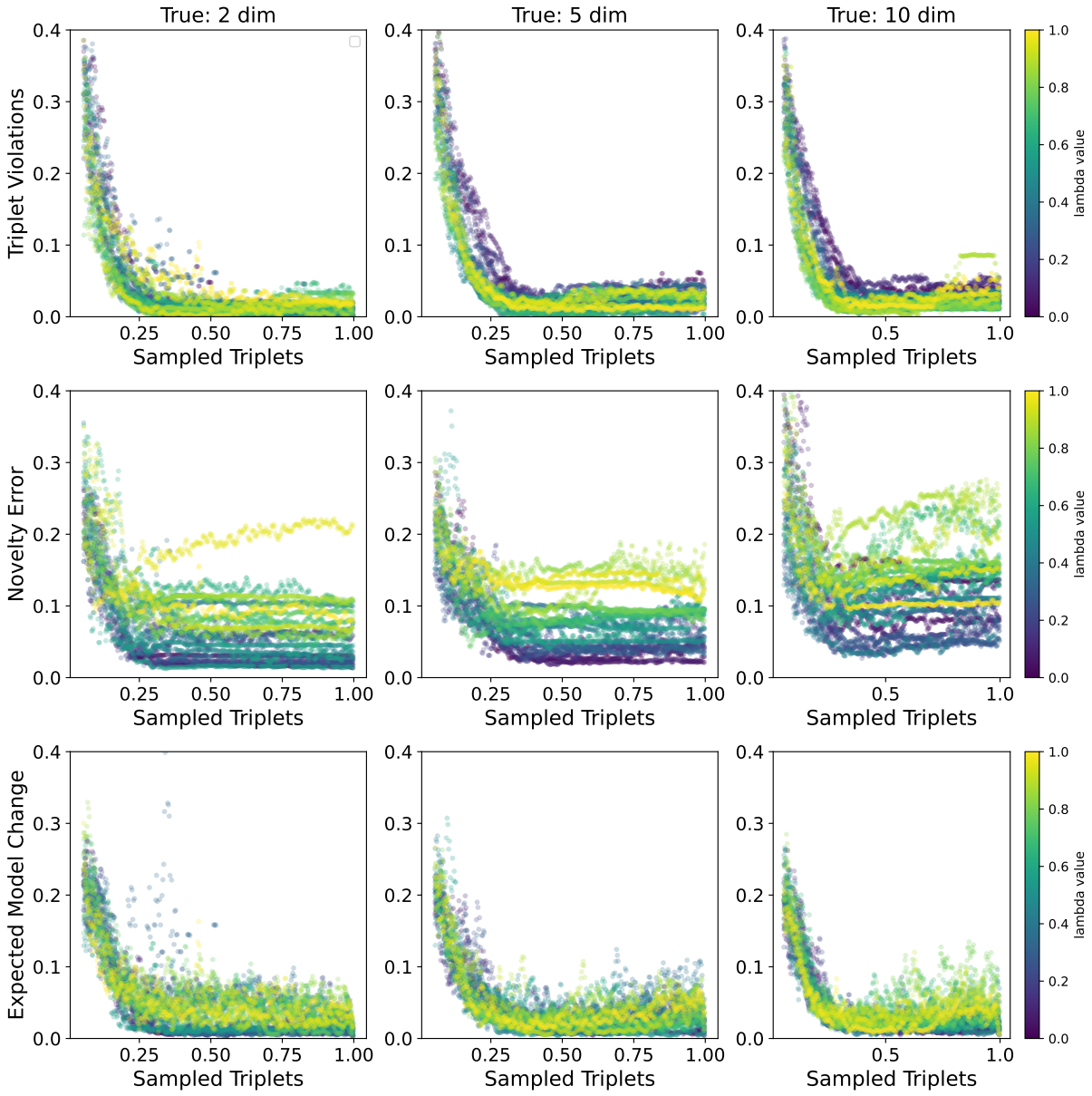

Figure 6.2: Exact dimension reconstruction errors for sparsity-ordered synthetic data.

sistent across sparsity/density values. In 10 dimensions, there is a slight color gradient that begins to form with the dark-colored sparse data saturating at slightly higher triplet violation scores. This gradient is not very pronounced, but we hypothesize that this is because sparse spaces have more uncertain/informative triplets. While having a large number of these '50/50' triplets was hypothesized to be beneficial for constraining the embedding to better capture distance magnitudes, harder to satisfy triplet constraints result in more violations at convergence. A space with dense clusters on the other hand has many 'easy-to-satisfy' triplets. When GNMDS discovers which designs belong to each cluster, all other triplets which compare designs *between* clusters will all be satisfied, making only the *in-cluster* triplets challenging to satisfy. In higher dimensions, the effects of sparsity/density are more pronounced as the domain of the design space is larger.

Concerns with the impact on the novelty reconstruction accuracy motivated exploration into the effects of design space sparsity. In Figure 6.2, the novelty error does show a fairly clear color gradient in all true-space dimensions. Generally, light-yellow colored plots saturate at worse converged novelty error than darker plots. Because the light-colored plots correspond to clustered data, this matches our hypothesis. When the design-space is densely clustered, there are less informative triplets to constrain the embedding, and the reconstruction accuracy of distance magnitudes is more variable. This, in turn, makes the novelty error—which is based on distance magnitudes—more variable. This high variability tends to result in the GNMDS-embedding converging on a worse novelty error for clustered spaces. However, there are some cases where darker, sparser plots saturate at worse novelty errors. Lighter, clustered plots have a larger feasible region for satisfying triplets, but this is not to say the GNMDS-embedding cannot still converge on an embedding with similar distance magnitudes as the true design space. Generally, however, loosely constrained problems from clustered data are more likely to converge on an embedding

with worse distance magnitude reconstructions.

However, while the magnitude of the novelty error appears to increase with more clustered design spaces in Figure 6.2, the number of sampled triplets required for convergence appears relatively unchanged across the color gradient. If anything, the plots which saturate in slightly less triplets tend to be lighter, clustered spaces. There is not enough of a discrepancy to claim that more clustered spaces generally require fewer triplets, however, this would make sense considering they contain fewer highly-informative triplets. When the embedding learns a handful of between-cluster triplets, the rest of the between-cluster triplets should already be satisfied.

The EMC plots from Figure 6.2 reflect the fact that the novelty error converges in a similar number of sampled triplets across all sparsity values. Overall, the trends in the EMC appear consistent across the sparsity index gradient. This is a promising sign, as it was unknown before experimenting how clustering would affect the convergence rate of the distance magnitudes and novelty error. These results suggest that our observations regarding the rate of novelty convergence should be consistent across design spaces with varying sparsity, and likewise the EMC should continue to be an effective measure for capturing the novelty convergence when the true space is unknown.

6.1.3 Design-Space Sparsity on Convergence Properties (Inexact Dimension)

All of the plots in Figure 6.2, however, were for *exact* embedding dimension reconstructions (*i.e.*, true and embedding dimensions were equal). Does varying sparsity in the design space impact the influence of inexact embedding dimension approximation differently than we observed for the uniform-random synthetic data? To address this question, the embedding met-

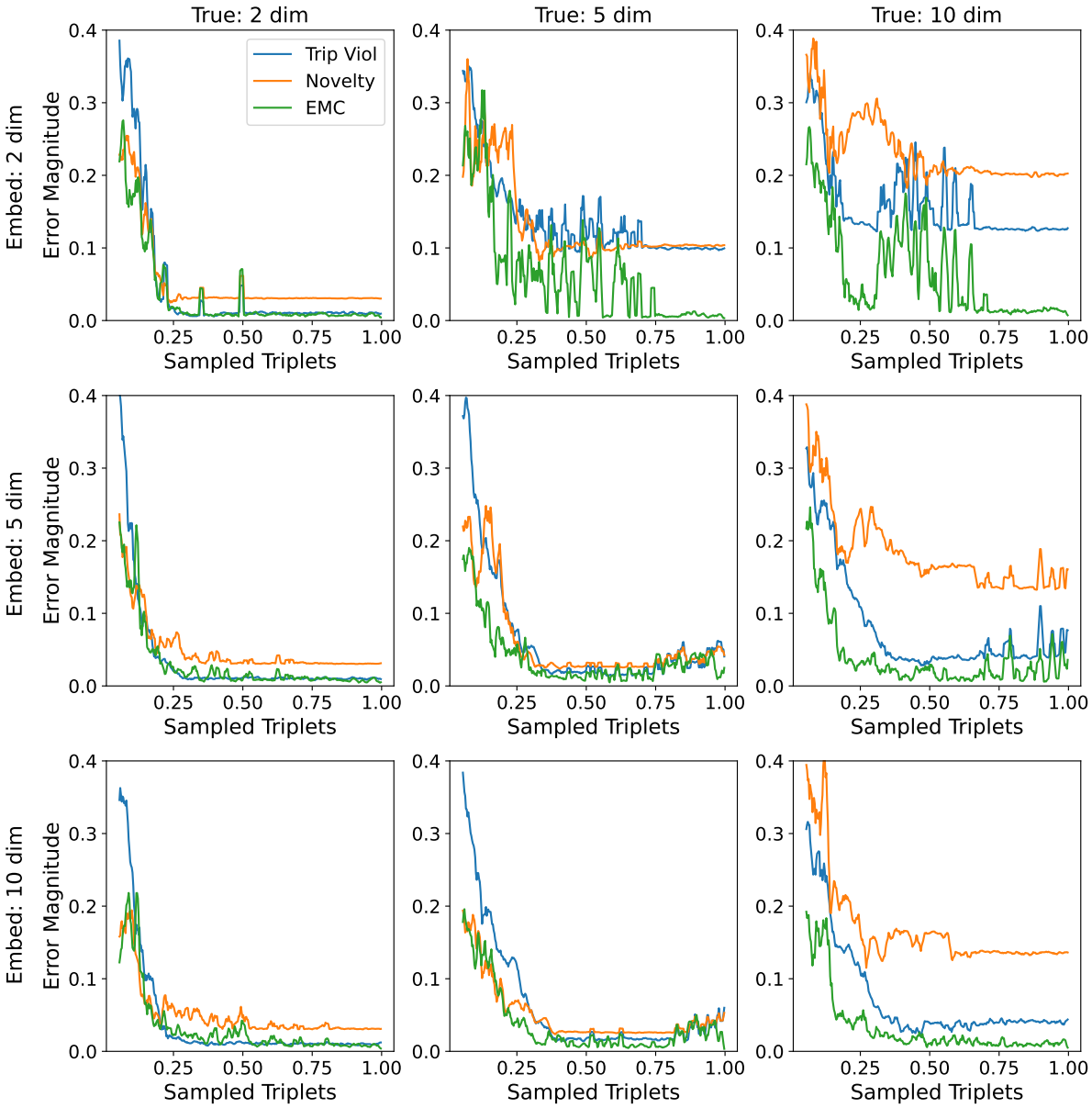

Figure 6.3: Embedding dimension reconstruction results for the lowest λ-value synthetic plot. This corresponds to a space with a sparse layout.

rics were plotted under different embedding dimensions for the least clustered (Figure 6.3) and most clustered (Figure 6.4) design spaces.

For the lowest λ-value least-clustered design spaces in Figure 6.3, the design space resembles that of the initial uniformly random synthetic data. Following the diagonal of the subplots show the performance for exact dimension approximation. The upper right triangle corresponds to under-approximated embedding dimension plots while the lower left triangle corresponds to over-approximated embbedding dimension plots. Like for the uniformly-random data, novelty and triplet errors suffer from dimension under-approximation both in terms of the converged value and the sampled triplets required for convergence. The characteristic decay shape for the EMC is also the same as was observed in the uniformly-random data. Also like the uniformly-random data, the least-clustered data shows minimal changes in the convergence error values when over-approximating the embedding dimension. There are slight observable increases in the amount of triplets sampled at convergence, but the effects of over-approximation remain minor compared to the effects of under-approximation. Overall, these results suggest the findings regarding the severity of poor dimension approximation are consistent between the least-clustered and uniformly-random data. It still appears preferable to err on the side of over-approximation, as the convergence criteria for the EMC do not require significantly more sampled triplets like it does for under-approximation.

On the other hand, for the highest λ-value most-clustered design spaces in Figure 6.4, there does not appear to be noticeable change in the converged error magnitudes across all degree of poor dimension approximation. This is most likely due to the fact that dense-clustered design spaces have many triplets with large disparity in the pairwise distances between clusters. When the true space contained many '50/50' uncertain triplets in the sparse-data, under-approximation

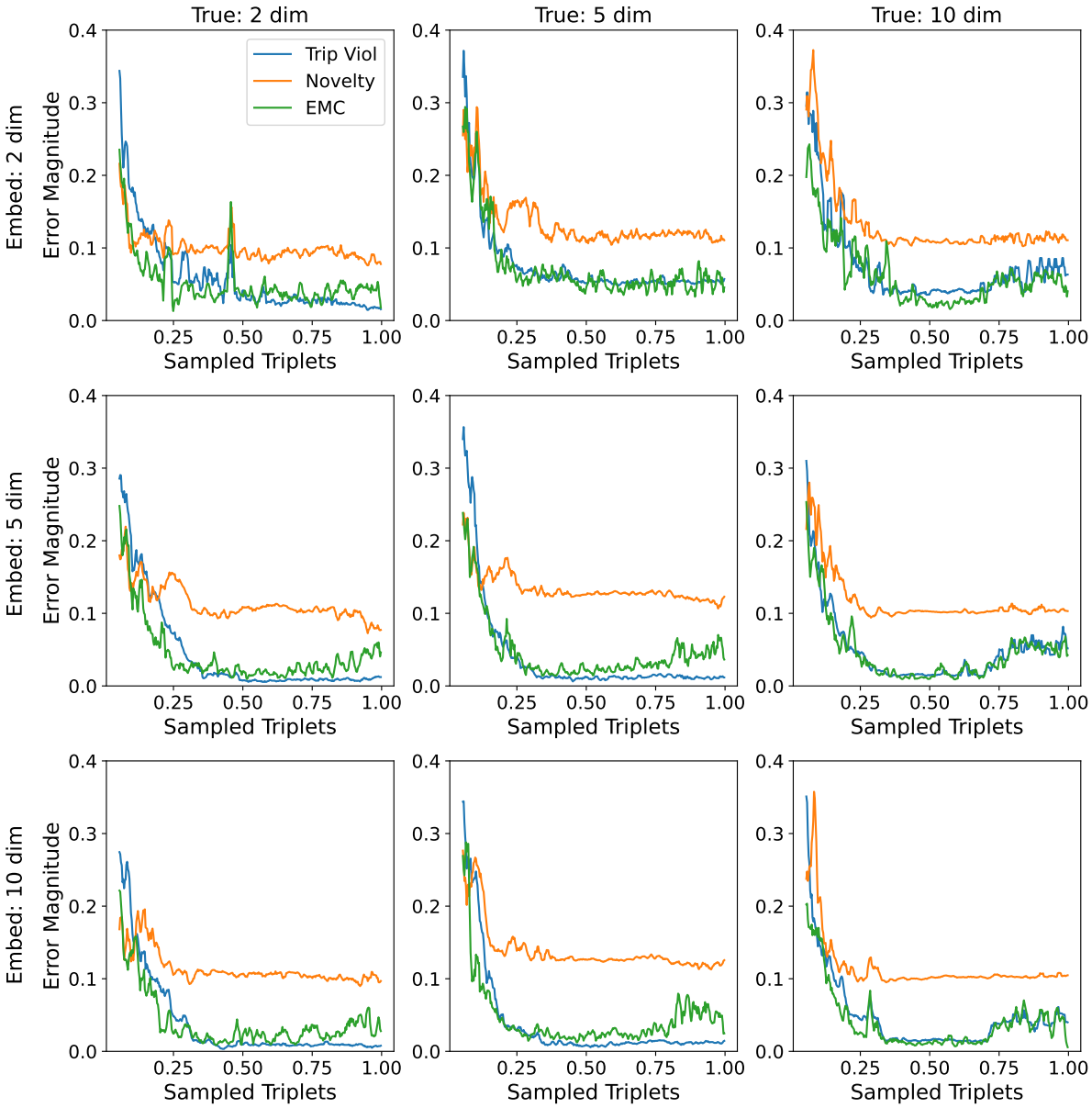

Figure 6.4: Embedding dimension reconstruction results for the highest λ-value synthetic plot. This corresponds to a space with dense clusters.

of the embedding dimension resulted in triplet labels that could not feasibly be satisfied. In the dense-clustered case, the separation of clusters and pairwise-distance disparity in triplets is large enough that projection onto a smaller dimension-space does not largely affect feasibility. Overall, the consistency across all degree of poor dimension approximation suggests that the method of using EMC to classify novelty convergence remains viable for dense-clustered spaces. However, the method of using the EMC to gauge the embedding dimension does not appear to be practical for design-spaces of this form. It is notable however, that because the error magnitudes and convergence behavior do not significantly change, that exact embedding dimension identification appears unnecessary for reconstructing dense-clustered spaces.

Overall, the results from exploring the sparsity-ordered synthetic dataset suggest that the observations found from §3-5 remain applicable for a wide variety of distributions. The EMC still appears to be an effective means of measuring novelty error convergence, despite the fact that densely-clustered spaces tend to result in worse novelty error reconstruction. It also does not appear that clustering greatly impacts the convergence rate of the novelty and triplet errors, and if anything, the convergence occurs in fewer triplets sampled. For sparse data, the adverse effects of under-approximation of the embedding dimension remain for the three error metrics, and the over-approximation plots suggest erring on the side of a larger embedding dimension. For the densely clustered data, the adverse effects of poor dimension approximation are mitigated by the fact that general clustering trends are preserved through projection.

6.2 Airfoil Dataset

While we believe the method of generating syntehtic design spaces of varying sparsity was effective at generalizing our observations to a variety of potential design similarity distributions, we were also interested in verifying those results on a real dataset of designs.

To observe the effect of creating an ordinal embedding for a design space with a dense center, we collected triplets from a dataset of airfoil geometries. This airfoil dataset contained 1222 airfoil geometries which were defined by a 384-dimensional image. These airfoils contain many slight perturbations of similar geometric profiles, and so are useful for understanding the level of clustering that might exist in a real-world design space. Principal component analysis (PCA) was used to reduce the dimension of the dataset to trials with 2, 5, or 10 components. After reconstructing the airfoil dataset in PCA-space, ten random airfoils were selected to define a single 'true space'. This subsampling was done ten times for each PCA dimension to produce 30 airfoil 'true-spaces' each containing ten designs. The full 2D PCA space of airfoils can be seen in Figure 6.5

Figure 6.6 shows the convergence behavior of the scored novelty error under the same conditions as Figure 5.2 but for the airfoil dataset rather than the synthetically generated dataset. In general, the airfoil novelty curves in Figure 6.6 saturate at larger novelty error magnitudes than those for the synthetic dataset in Figure 5.2. This is consistent with our hypothesis on the impact of spaces with heavy clustering. With too many 'loose' triplet constraints, the GNMDS algorithm has a lot of freedom on the placement of airfoils on the outer perimeter of the design space before any triplets are violated.

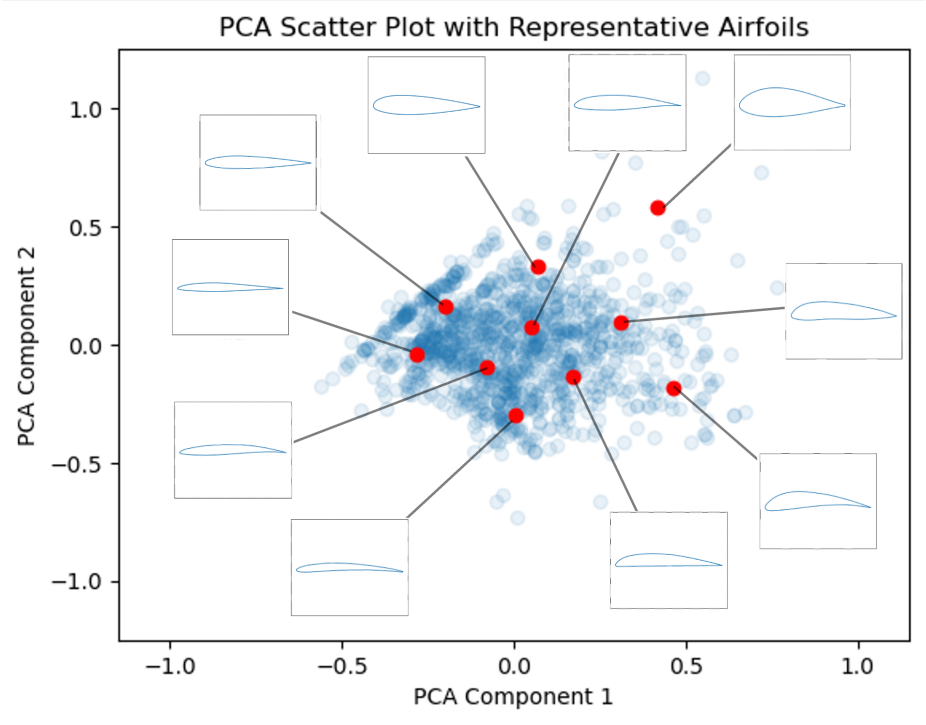

Figure 6.5: 2D PCA full design space for the airfoil dataset with representative design plots.

6.3 Airfoil Dataset

While the loosely constrained problem results in a larger converged novelty *magnitude* error than for the synthetic spaces of the same size in Figure 5.2, GNMDS has no problem satisfying triplet constraints as shown in Figure 6.7. This also provides support for our hypothesis. If GNMDS struggled to even capture the relative novelty ranking of designs, the embedding would produce a much larger number of triplet violations than for the synthetic dataset in Figure 5.1. This is not the case, as only the novelty magnitudes are larger. While design spaces with heavy concentrations may make it more difficult to interpret how much *more* novel a design is from another, the impact on the triplet violations and thus the novelty rank accuracy appears negligible.

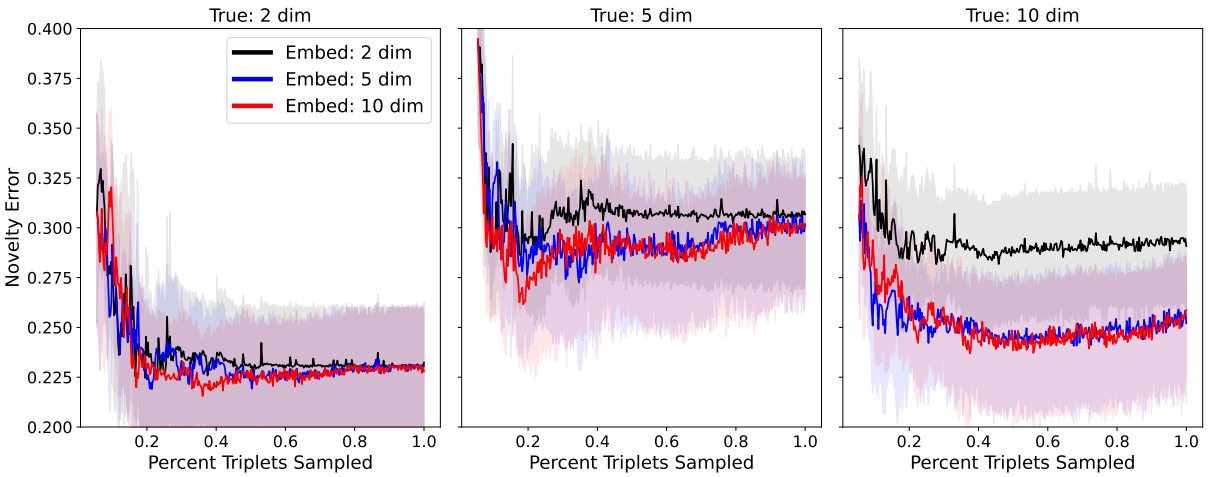

Figure 6.6: Novelty error curves for the 'airfoil geometry' dataset under varied true and embedding dimensions.

6.4 Milk-Frother Dataset

In the airfoil experiment, the effect of a design space with a dense center was observed, but the 'design space' in question was constructed though PCA. While it was a helpful tool for collecting a triplet pool with 'loose' constraints, the design space was not actually representative of a similarity space that human's produced. As such, we wanted to study the reconstruction error of a design space directly created through human judgments. Human raters may provide some self-contradictory triplet orderings which may affect the convergence rate and accuracy of the GNMDS-produced embedding. Unfortunately, synthetic spaces were used in testing because the true novelty error and triplet violation errors can not be computed without knowing the true orientation of designs. The two reconstruction error metrics cannot be used directly to observe the convergence behavior of an embedding created from human-provided triplets. Luckily, the expected model change metric proposed in §4 does not require a known true-space, and could provide insight on how the novelty information of the embedding saturates.

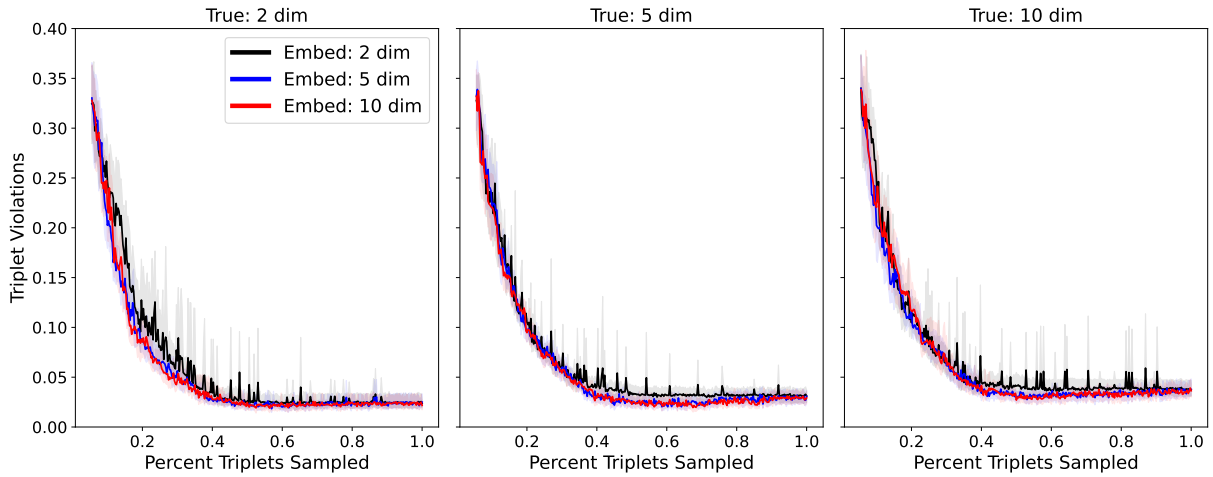

Figure 6.7: Triplet violation error curves for the 'airfoil geometry' dataset under varied true and embedding dimensions.

To conduct this experiment, the 'milk-frother' dataset from [2] was used. This dataset contains all 360 possible human-labeled triplet orderings for a ten-item design space. Each design depicts a unique device for frothing a basin of milk. Human raters were given sheets of paper with a hand-drawn image of each milk frother design as well as a short sentence description of the device. Raters were given these images in triplet groupings and asked to order them based on their perceived similarity. Because of slight disagreement between human-ordered triplets across raters, a majority vote decided which triplet ordering represented a consensus. The milk-frother sketches are shown in Figure 2.3. Figure 6.8 shows the resulting GNMDS-produced 2D embedding from the triplets collected from human raters.

Figure 6.9 shows the expected model change saturation plots for the milk-frother dataset. Ordered triplets from the human-labeled dataset were selected one-by-one under uncertainty sampling and given to GNMDS. Each plot displays the average EMC from ten unique initial triplet pools, with error bars showing the 90% confidence interval. The figure is divided into subplots corresponding to the embedding dimension provided to GNMDS. The true orientation of de-

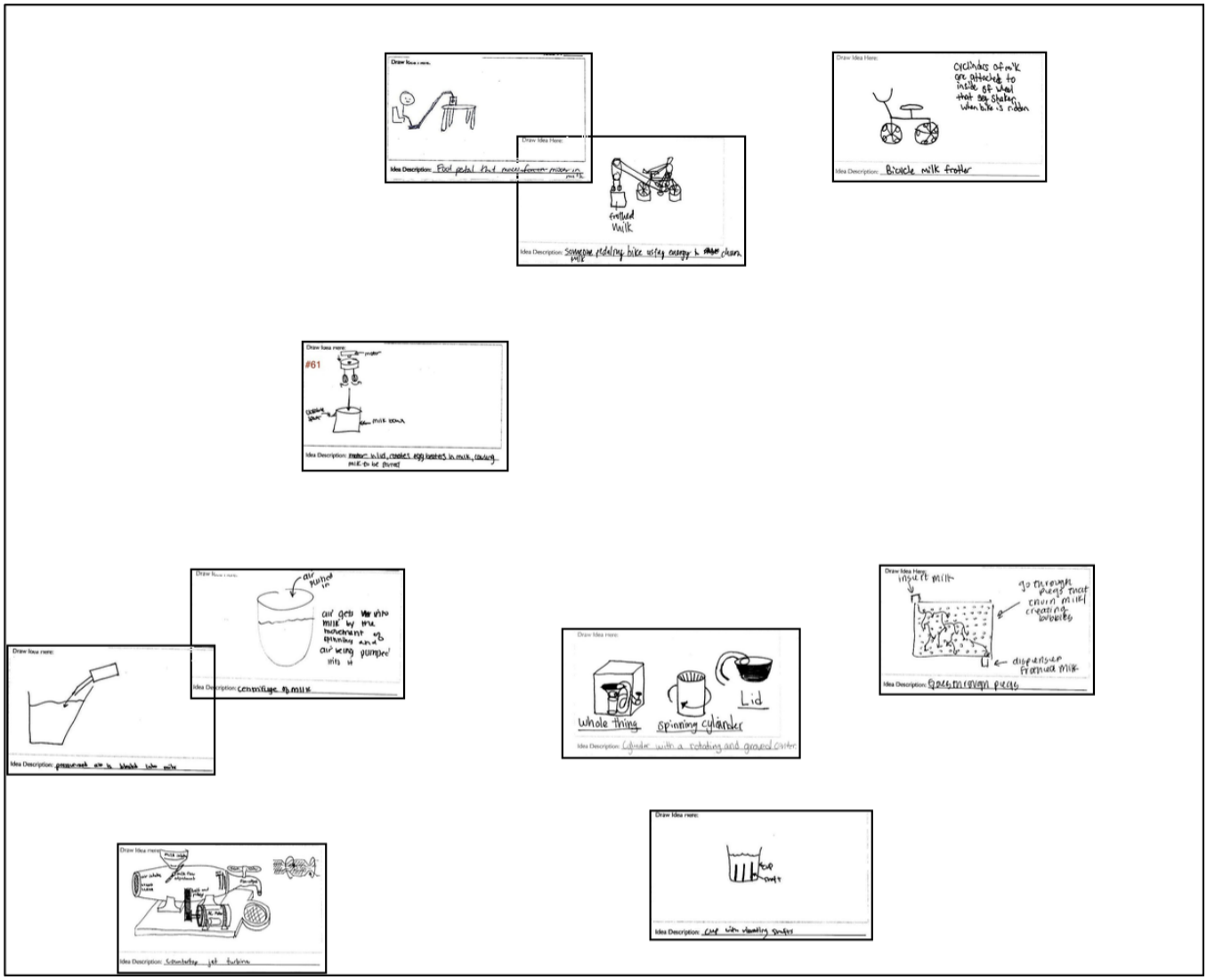

Figure 6.8: GNMDS-produced 2D embedding for the human-provided triplets collected from the milk-frother dataset.

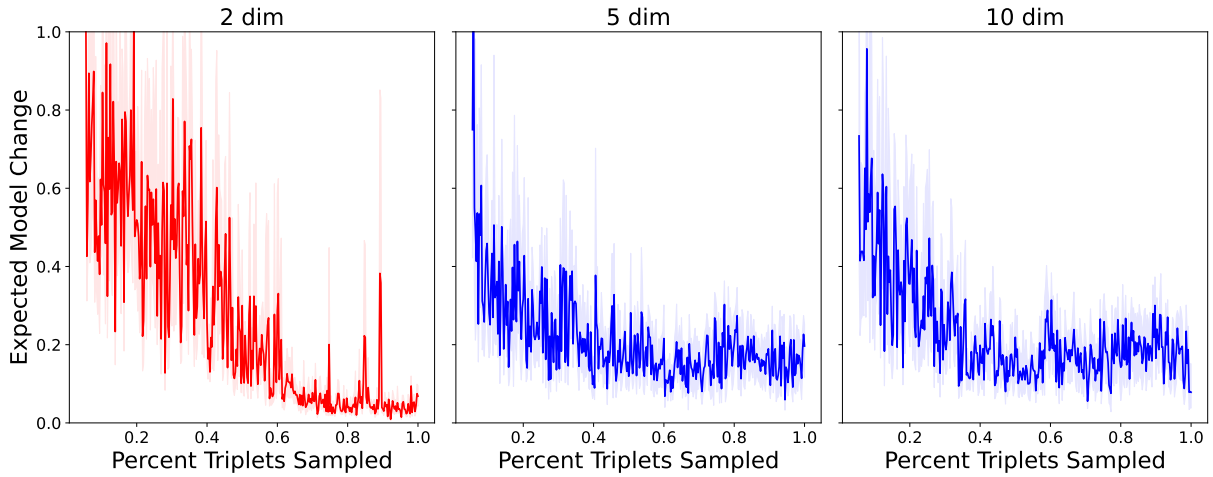

Figure 6.9: Expected model change saturation p lots from the milk frother dataset [2] in varied embedding dimensions under uncertainty sampling.

signs — and by extensions the true dimension of the design-space — is unknown.

In Figure 6.9, the expected model change curve for a 2-dimensional embedded space has a much different shape than the curves for a 5- or 10-dimensional embedded space. Whereas the 5- and 10-dimensional curves reach convergence at around 40% of the triplet pool labeled, the 2-dimensional curve does not reach convergence until about 60% of triplets are labeled. Using the results from Figure 5.3, the slower saturation for the 2-dimensional curve constitutes evidence for an under-approximation of the true dimension, and it is likely that the true dimension is closer to 5 or 10. It is noteworthy that the 2-dimensional curve does oscillate less after reaching convergence, however this is likely a consequence of fewer degrees of freedom GNMDS is given to construct an embedding.

Overall, it appears that the method detailed in §5.3 for tuning the embedding dimension, could be used. There is no way to ensure that the embedding dimension is accurate, but the expected model change seems to be exhibiting the same behavior for an under-approximated dimension as observed for the synthetic data in Figure 5.3. It also does not appear that the

human-labeled triplets — which contain some small degree of self-disagreement across each rater — greatly affect the convergence rate of novelty information. In both the uniform-random synthetically-generated plots from Figure 5.2 and the milk-frother plots from Figure 6.9, the novelty information saturates at about 40% of the triplet pool sampled.

Chapter 7: Conclusion

In this thesis, we explored the relationship between triplet constraints and centroidal distance magnitudes within an ordinal embedding to better understand how novelty information is captured through similarity comparisons. While the method of constructing an ordinal embedding through the collection of triplet queries is well-studied, to the best of our knowledge, ours is the only paper to study the behavior of the centroidal distances within the embedding — which can be used to identify novel items. Using the Generalized Non-Metric Multidimensional Scaling algorithm outlined in [29] and synthetically generated design-spaces, we observed the reconstruction accuracy of an embedding as triplets were labeled one-by-one.

In our testing, we found that the centroidal distances — and thus the novelty information — within an embedding can saturate before triplet violations are minimized. This implies that in applications where an ordinal embedding will be used for ranking design novelty, fewer triplets need to be labeled than theoretical bounds on the triplet violation accuracy would suggest. Given this phenomenon, we proposed a procedure for identifying when novelty information within an embedding has saturated to the point where additional triplet labeling has little benefit. This was done by tracking the deviation in centroidal distances after a triplet is labeled to create a metric we called the Expected Model Change (EMC). We further explored the dangers of low accuracy in selecting the dimension of the ordinal embedding and proposed use of the EMC to

tune the embedding dimension to an appropriate value. Each of these steps were taken to develop a pipeline for using an ordinal embedding to identify novel items in real-world design spaces.

7.1 Summary of Key Contributions

The following list is a summary of the key contributions of this thesis in the order they were presented.

- In studying the saturation of centroidal distances of an embedding reconstruction of a known synthetic true space, we compared the triplet violation and novelty error convergence properties. We displayed the monotonic relationship between these properties: showing that novelty error decreases with triplet sampling. However, despite this relationship early on, we showed the novelty error saturates faster than the triplet violation error. Existing work, such as work by Jamieson, have proposed bounding functions to define embedding saturation on the basis of triplet violation saturation [7, 8]. However, before this thesis it was unknown how the centroidal distances and thus novelty information converged under triplet sampling. These results suggest that existing growth bounds for triplet sampling in applications of novelty estimation will require more labeled queries than necessary for centroidal distance convergence.

- We presented the Expected Model Change (EMC) which measures changes in the novelty information of an embedding under continued triplet sampling. We showed the EMC closely follows the convergence properties of the true novelty error, and thus can be used as a proxy when the true space is unknown and novelty error cannot be computed. With the EMC convergence data, a policy can be developed to identify novelty convergence as

triplets are labeled one-by-one, thus removing the need for conservative bounds on the violations to determine how many triplets need labeled for a saturated embedding.

- We explored the effect of poor embedding dimension approximation on the novelty, triplet violation, and EMC metrics. We showed the best novelty reconstruction accuracy occurs in a 'goldilocks' area around the true dimension, however, we also showed that underapproximation of the embedding dimension has more sever effects on the novelty error magnitude and require triplets for convergence. We showed how the EMC can provide some insight into the accuracy of the embedding estimate, and that overapproximation of the embedding dimension has little effect on the ability for the EMC to detect convergence. While existing work has studied the effect of the embedding dimension on the amount of triplet queries needed for violation convergence, to the best of our knowledge, this thesis is the only one to consider the effects of embedding dimension on the ability for triplet labels to reconstruct centroidal distances.

- We explored the effect of the true space distribution on the convergence properties of the embedding. We showed that clustered true spaces have a larger feasible region in the embedding optimization, and thus converge on larger novelty error magnitudes than sparser spaces. We also showed that the information captured in triplets from clustered spaces is better preserved in a lower dimension projection than for a sparse space. These results were explored for synthetic and real-world data. To the best of our knowledge, this thesis is the only of its kind to consider the effects of clustering/sparsity in the design space on the ability for ordinal embedding methods to reconstruct centroidal distances.

7.2 Limitations and Future Work

To properly identify the success of ordinal embeddings at capturing novelty error from a true space distribution, synthetic data was needed with known centroidal distance magnitudes. We developed methods for testing different true space distributions and dimensions, but without knowing the exact types of distributions that would employ this type of novelty estimation, it was difficult to generalize our findings to any potential design space. Future work on design datasets with a known or conventional similarity space would better validate our results. We were also limited in computational resources, so most of our experiments were conducted on 10-design spaces. Future work and feasibility analysis on large datasets would better express the capabilities of ordinal embeddings for novelty estimation.

Our results and experiments had the primary purpose of displaying general behaviors of the novelty, triplet violation, and EMC reconstruction metrics under various conditions. While these general trends are insightful and provide evidence to support this papers claims, future work is needed to perform proper statistical tests on each hypothesis. The primary claims of the thesis, such as novelty error converging faster than triplet violations, EMC following the novelty error, and loss of information in underapproximated embedding dimensions, are reasoned to be universal in ordinal embeddings. However, what is unknown with statistical certainty is the magnitude of these effects. Future hypothesis testing, as well as further exploration into heuristic policies such as tuning the embedding dimension with the EMC, would go a long way in validating ordinal methods of novelty estimation.

All tests in this paper used the GNMDS algorithm, and most all outside of the initial testing used the bootstrap-uncertainty method of active triplet sampling. GNMDS was chosen as it is the

most methodically simple way of satisfying triplet constraints. While this method is reasonable and explored in existing literature, our results are limited to this subset of ordinal embeddings. We do not believe the the major trends of the novelty and triplet violation errors found in this thesis should vary across ordinal embedding methods—as these effects are largely governed by relaying metric information through ordinal triplet comparisons—but the magnitudes of these effects might vary across application of NMDS solver. Future work that attempts to generalize these findings to other ordinal embedding methods like CKL and STE or different policies of active triplet sampling would strengthen the claims of this thesis.

Additionally, apart from the EMC plots from the milk-frother triplets in Figure 6.9, all triplet pools tested were noiseless and guaranteed the existence of an embedding which satisfied all labels—assuming enough embedding dimensions were provided. To reflect cases where human raters may provide self-contradictory triplet orderings, future work should be conducted which introduces a varying degree of noise in the triplet pool and studies the resulting impact on the observations found in this thesis.

Bibliography

[1] Matthew Keeler and Mark Fuge. Fewer Triplets Than You Think: Novelty Error Converges Faster Than Triplet Violations in Ordinal Embeddings. American Society of Mechanical Engineers Digital Collection, November 2023.

[2] Faez Ahmed, Sharath Kumar Ramachandran, Mark Fuge, Sam Hunter, and Scarlett Miller. Unpacking subjective creativity ratings: Using triplet queries to find idea maps. page 16.

[3] Jami J Shah, Santosh V Kulkarni, and Noe Vargas-Hernandez. Evaluation of idea generation methods for conceptual design: effectiveness metrics and design of experiments. *J. Mech. Des.*, 122(4):377–384, 2000.

[4] M. L. Maher and D. H. Fisher. Using AI to Evaluate Creative Designs. *DS 73-1 Proceedings of the 2nd International Conference on Design Creativity Volume 1*, pages 45–54, 2012.

[5] P.-A. Verhaegen, D. Vandevenne, and J. R. Duflou. ORIGINALITY AND NOVELTY: A DIFFERENT UNIVERSE. *DS 70: Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia*, pages 1961–1966, 2012.

[6] Laurens van der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, September 2012. ISSN: 2378-928X.

[7] Kevin Jamieson. The Analysis of Adaptive Data Collection Methods for Machine Learning. page 230.

[8] Kevin G. Jamieson and Robert D. Nowak. Low-dimensional embedding using adaptively selected ordinal data. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1077–1084, Monticello, IL, September 2011. IEEE.

[9] Deniu He. Active learning for ordinal classification based on expected cost minimization. *Scientific Reports*, 12(1):22468, December 2022. Number: 1 Publisher: Nature Publishing Group.

[10] Gregory Canal, Stefano Fenu, and Christopher Rozell. Active Ordinal Querying for Tuplewise Similarity Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3332–3340, April 2020. Number: 04.

[11] Kuan-Hao Huang and Hsuan-Tien Lin. A Novel Uncertainty Sampling Algorithm for Cost-Sensitive Multiclass Active Learning. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 925–930, December 2016. ISSN: 2374-8486.

[12] Joel Chan, Katherine Fu, Christian Schunn, Jonathan Cagan, Kristin Wood, and Kenneth Kotovsky. On the Benefits and Pitfalls of Analogies for Innovative Design: Ideation Performance Based on Analogical Distance, Commonness, and Modality of Examples. *Journal of Mechanical Design*, 133:081004, August 2011.

[13] Pao Siangliulue, Kenneth C. Arnold, Krzysztof Z. Gajos, and Steven P. Dow. Toward Collaborative Ideation at Scale: Leveraging Ideas from Others to Generate More Creative and Diverse Ideas. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 937–945, Vancouver BC Canada, February 2015. ACM.

[14] Karen Leggett Dugosh and Paul B. Paulus. Cognitive and social comparison processes in brainstorming. *Journal of Experimental Social Psychology*, 41(3):313–320, May 2005.

[15] Richard L. Marsh, Joshua D. Landau, and Jason L. Hicks. How examples may (and may not) constrain creativity. *Memory & Cognition*, 24(5):669–680, 1996. Place: US Publisher: Psychonomic Society.

[16] Shanna R. Daly, Erika A. Mosyjowski, and Colleen M. Seifert. Teaching Creativity in Engineering Courses. *Journal of Engineering Education*, 103(3):417–449, July 2014.

[17] Elizabeth Starkey, Christine A. Toh, and Scarlett R. Miller. Abandoning creativity: The evolution of creative ideas in engineering design course projects. *Design Studies*, 47:47–72, November 2016.

[18] J. S. Linsey, E. F. Clauss, T. Kurtoglu, J. T. Murphy, K. L. Wood, and A. B. Markman. An Experimental Study of Group Idea Generation Techniques: Understanding the Roles of Idea Representation and Viewing Methods. *Journal of Mechanical Design*, 133(3):031008, March 2011.

[19] Akin Osman Kazakçi, Thomas Gillier, Gerald Piat, and Armand Hatchuel. Brainstorming versus creative design reasoning.

[20] Binyang Song, Scarlett Miller, and Faez Ahmed. Attention-Enhanced Multimodal Learning for Conceptual Design Evaluations. *Journal of Mechanical Design*, 145(041410), February 2023.

[21] Haibing Li and Roland Lachmayer. Generative Design Approach for Modeling Creative Designs. *IOP Conference Series: Materials Science and Engineering*, 408:012035, October 2018.

[22] Othman Sbai, Mohamed Elhoseiny, Antoine Bordes, Yann LeCun, and Camille Couprie. DesIGN: Design Inspiration from Generative Networks. pages 0–0, 2018.

[23] Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. Place: US Publisher: American Psychological Association.

[24] Daniel A. McAdams and Kristin L. Wood. A Quantitative Similarity Metric for Design-by-Analogy. *Journal of Mechanical Design*, 124(2):173–182, June 2002.

[25] Jami J. Shah, Steve M. Smith, and Noe Vargas-Hernandez. Metrics for measuring ideation effectiveness. *Design Studies*, 24(2):111–134, March 2003.

[26] Dongwook Hwang and Kristin Wood. Assessing the Novelty of Design Outcomes: Using a Perceptual Kernel in a Crowd-sourced Setting. December 2020.

[27] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively Learning the Crowd Kernel, June 2011. arXiv:1105.1033 [cs].

[28] Leena Chennuru Vankadara, Siavash Haghiri, Michael Lohaus, Faiz Ul Wahab, and Ulrike von Luxburg. Insights into Ordinal Embedding Algorithms: A Systematic Evaluation, October 2021. arXiv:1912.01666 [cs, stat].

[29] Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized Non-metric Multidimensional Scaling.

[30] Roger N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. Part I. *Psychometrika*, 27(2):125–140, 1962. Place: Germany Publisher: Springer.

[31] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2):115–129, June 1964.

[32] Kevin G Jamieson, Lalit Jain, Chris Fernandez, Nicholas J. Glattard, and Rob Nowak. NEXT: A System for Real-World Development, Evaluation, and Application of Active Learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[33] Michael Lohaus, Philipp Hennig, and Ulrike von Luxburg. Uncertainty Estimates for Ordinal Embeddings, June 2019. arXiv:1906.11655 [cs, stat].

[34] Laurens van der Maaten. Github repository of laurens van der maaten. `http://lvdmaaten.github.io`.