LEARNING WITH AND WITHOUT HUMAN FEEDBACK

A Dissertation
Presented to
The Academic Faculty

By

Austin Shiyi Xu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2024

LEARNING WITH AND WITHOUT HUMAN FEEDBACK

Thesis committee:

Dr. Mark Davenport
Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Justin Romberg
Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Christopher Rozell
Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Zsolt Kira
School of Interactive Computing
Georgia Institute of Technology

Dr. Ashwin Pananjady
Electrical and Computer Engineering and
Industrial and Systems Engineering
Georgia Institute of Technology

Date approved: April 15, 2024

ACKNOWLEDGMENTS

The core of any graduate school experience starts and ends with one's advisor. As such, I must begin my acknowledgements with my advisor, Mark Davenport. From the beginning, Mark has simultaneously given me the space I needed to set my own research agenda and pursue areas I found interesting, while being available to provide both technical input and writing guidance. Thank you for fostering a research environment which gave me the freedom to pick my own problems (often after a bit of aimless wandering) and for making sure I took time to take care of myself outside of work by encouraging and supporting my distance running hobby. While I have little hope in matching his running abilities, I hope to one day become a mentor of his caliber.

I cannot understate my appreciation for Ashwin Pananjady. While Ashwin's incredibly sharp technical thinking is unmatched, I will always remember his endless patience mentoring me. I started working with Ashwin as a novice statistician with only a shallow appreciation of Hoeffding's inequality. In the subsequent three years, Ashwin has shaped everything from how I pick interesting problems to work on ("this needs a bit more sauce") to my technical depth to my standards in technical writing and communication.

My time at Georgia Tech has been accentuated by a wealth of excellent professors, collaborators, and labmates. I must thank the other members of my thesis committee: Christopher Rozell, Justin Romberg, and Zsolt Kira. In particular, I would like to thank Justin Romberg and Cheng Mao for their excellent courses in statistical inference and machine learning which laid the foundation of much of this thesis. My collaboration with Namrata Nadagouda on active learning would not have succeeded without her perseverance and guidance. Andrew McRae's deep knowledge and sharp insights in statistical inference made many proofs sharper (and pages shorter). Jingyan Wang's high standards in experimentation and technical writing played no small role in turning a roughly written document into a polished conference paper.

When not working, I was likely "wasting time" with my fellow 11th floor Coda labmates. I have to thank the Children of the Norm community for making day-to-day life bearable: Jihui Jin, Santhosh Karnik, Matthew O'Shaughnessy, Andrew McRae, Nauman Ahad, Namrata Nadagouda, Peimeng Guan, Yuqin Yang, Brighton Ancelin, Alex Saad-Falcon, Chiraag Kaushik, Belen Urcelay, and many others that I undoubtedly missed, thank you. I'll cherish the times we spent naming papers, walking to DCT, getting Covid tested for the purposes of getting a free cookie, or even watching the repavement of a neighboring parking deck. I hope you share my opinion that the time was not wasted, but well spent.

During my PhD, I have been fortunate to intern at both Amazon and Duolingo. At Amazon, Arjun Seshadri's mentorship has shaped almost all aspects of my professional life, from research outlook to career prospects. During a crossroads during my internship, Arjun nudged me in the right direction: "Do you want to keep working on this, or do you want to do something interesting?" What emerged out of "interesting" was a successful first venture into computer vision. Thank you for putting up with my deluge of Slack messages – I hope I wasn't *too* annoying. I would also like to thank Mariya Vasileva, Achal Dave, and the rest of the Style team for your insightful comments and help. At Duolingo, Will Monroe's guidance and patience was instrumental in my first foray into natural language processing. Will's never-ending technical knowledge, ranging from NLP to recommendation systems to Git to internal microservices were foundational in shaping both my research and technical abilities. I also must thank Stephen Mayhew, Ali Malik, and the members of the Personalization Lab at Duolingo for the insightful conversations, sparkling water runs, and our weekly FIKA.

I am also grateful for my friends from both Michigan. I do not think I could have survived lockdown without our video calls and game nights. Our group reunions, whether watching Michigan beat Notre Dame under pouring rain or scaling mountains in Colorado, were delightful reprieves from the day-to-day grind of research. My DC friends have played an equally important role, from coordinating calls across three different timezones to our yearly trip to Shenandoah to getting attacked by wasps in the National Arboretum. Thank you for making every trip to DC worth looking forward to.

I will forever be grateful for my parents, Jianying and Xiaochu. Mom and dad, I cannot repay you for all of the sacrifices you've made, from fighting through the challenges as an American immigrant to raising two unruly boys to tolerating both our decisions to pursue PhDs. Thank you for teaching me the importance of hard work, discipline, and ambition. Mom, I know I'll always be able to talk about research with you, and Dad, I know you'll always be there to remind us there are more interesting things to discuss. I also have to thank my brother Alec for being my go-to for banter. Thank you for introducing me to the latest vocabulary and corrupting my English with Alec-isms. Our video calls and holiday breaks were always a welcome break.

Finally, I must thank my wonderful partner, Jenny Liu. Without your unending love, kindness, and patience along with that signature sense of humor, the work in this thesis would not have been remotely possible. Thank you for putting up with the long nights before deadlines, the numerous constraints when planning visits, and my endless complaints about the conference review process. I'm excited for our future together.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The development of contemporary machine learning (ML) models is driven, in part, by the availability and volume of labeled training data. Labels provided by humans play a central role in this training pipeline, offering models ground truth annotations from which to extract patterns. However, collecting such feedback from humans is a challenging and time-consuming task. As a result, practitioners must be intentional both in how they choose to query humans for feedback and in the problem settings for which they request feedback.

This thesis explores learning from human feedback along two fundamental directions. The first part of the thesis focuses on how we can more effectively learn from and collect human feedback from a mathematically grounded perspective. In Chapter 2, we consider the paired comparison, a simple mechanism for collecting human feedback, and show that paired comparison responses are capable of estimating a much richer parametrization of user preferences than previously established [1]. In Chapter 3, we propose a new mechanism for collecting human feedback called the perceptual adjustment query [2] designed to balance informativeness and cognitive burden. We apply perceptual adjustment queries to a human perception model parametrized by a low-rank metric and rigorously prove estimation error bounds.

The second part focuses on how we can leverage pretrained models to avoid collecting additional human feedback. In Chapter 4, we consider the cold-start phase of a recommender system, where no user relevance feedback is available to train a retrieval model. Using the generative abilities of large language models, we design a retrieval framework capable of retrieving relevant text for users without any human relevance feedback [3]. In Chapter 5, we improve synthetic image dataset generation by removing the need for humans-in-the-loop [4]. Existing methods require human annotators to repeatedly label synthetically generated images; our proposed framework leverages tools from image editting to re-use existing labeled images, bypassing the need for human annotators.

CHAPTER 1

INTRODUCTION AND BACKGROUND

Past industrial revolutions, from rapid mechanization in the 1760s to the birth of the modern digital age in the 1960s, share a common theme: the continued advancement in the partnership between humans and machines. The advent of the machine learning (ML) boom marks the first development of *intelligent* machine partners. These machine partners, however, must often be trained with large amounts of *training data*, a subset of which is presented to humans for feedback, such as labels or preferences statements.

Recent empirical breakthroughs ML have resulted largely from increasing model capacity and training data size. While it would be infeasible to hand annotate the petabytes of data needed to train models with billions of parameters, human feedback still plays a key role in adapting trained models for specific tasks and enhancing the quality of outputs. This is perhaps best highlighted in the development process of large language models (LLMs), which are first pretrained on copious amounts of unlabeled text data on a general purpose task, such as next word prediction. To adapt the pretrained model to a specific task, such as text sentiment classification or document summarization, *supervised finetuning* is employed with human annotated datasets. After supervised finetuning, the model undergoes *human preference finetuning* to steer the model towards "better" (factually correct, non-harmful, creative, etc.) outputs that align with human preferences, often provided in the form of paired comparisons between model outputs. In these latter two stages, humans play a critical role in curating model outputs.

These large foundational models are emerging as intelligent machine partners for human users. As with any partnership, clear two-way communication is essential for the success of this new paradigm of human-machine collaboration; it is crucial that learning systems be able to comprehend and learn from human perception and judgement. As such, modern ML has relied heavily on human feedback, encompassing everything from labeled images to relative preference orderings of items to manually summarized documents. However, human feedback may be difficult or even impossible to collect in large quantities. For example, in the medical image domain, dedicated medical experts are required to invest long periods of time identifying diseases on images. Even after the initial time investment, disagreements among experts results in additional time and monetary investment before a final labeled dataset can be produced. Therefore, a crucial component in this emerging partnership is to understand how we can best obtain human feedback. This requires both re-examining existing mechanisms for feedback and the development of new querying mechanisms. On the other hand, by training such models to encode and reflect human perception and judgement hints at opportunities to avoid collecting additional human feedback. Another crucial component of human-machine partnerships is to explore the role of using intelligent machines to re-use or avoid collecting human feedback in new application areas. This thesis explores these two fundamental components:

How can we more effectively learn ***with*** *human feedback and when can we use foundational tools to learn* ***without*** *additional feedback*

1.1 Background

This exploration of learning from human feedback brings together a variety of disciplines, ranging from high-dimensional statistics to contemporary generative models. The goal of this section is to provide the reader with a high-level overview of these fields.

1.1.1 A statistical toolbox: structure in high dimensions.

Contemporary ML problems are defined not only by the volume of data, but also the high-dimensional nature of such data; modern systems are being asked on to process large volumes of text (represented by high-dimensional embedding vectors) and high definition images and video. In order to effectively learn in these settings, conventional wisdom states that a massive amount of data is necessary. However, real-world data often contains hidden *low-dimensional structure*, which leads to more tractable learning solutions. To improve our understanding of various forms of human feedback in high-dimensional settings, we turn to the broad and rich field of high-dimensional statistics (e.g., [5, 6]) for mathematical tools for our analysis. The section aims to provide a brief overview of the statistical benefits of assuming low-dimensional structure is hidden in our high-dimensional data.

The goal of learning from human feedback often to recover a potentially large set of unknown parameters from human responses. For example, Chapter 2 and Chapter 3 consider parametrized models for human perception and preference and explicitly aim to estimate the unknown parameters. More broadly, modern recommender systems leverage deep neural networks to predict user actions, with (potentially implicit) human feedback used to train such models (i.e., recover/learn unknown model parameters.) The process for finetuning LLMs can be viewed similarly. To perform estimation of some unknown parameter $\boldsymbol{\theta}^\star$ from feature-label pairs $(\boldsymbol{x}_i, y_i), i = 1 \ldots, n$, we typically solve an optimization program of the form

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ell(\boldsymbol{\theta}; \boldsymbol{x}_i, y_i) + \lambda\, r(\boldsymbol{\theta}). \tag{1.1}$$

Above, ℓ is some loss function that measures how well the parameter $\boldsymbol{\theta}$ fits the data and r is a regularizer that enforces structure in our estimated parameter, with regularization parameter λ controlling the degree to which structure is enforced.

As a concrete example, we consider the problem of linear regression, where we receive n noisy measurements $y_1, \ldots, y_n$ of the form

$$y_i = \langle \boldsymbol{x}_i, \boldsymbol{\theta}^\star \rangle + \epsilon_i,$$

where $\boldsymbol{x}_i \in \mathbb{R}^d$ are random "sensing vectors", $\boldsymbol{\theta}^\star \in \mathbb{R}^d$ is the parameter we are trying to recover, and ϵ_i is random noise. Classical statistical results advise that $n \gtrsim d$ measurements are needed to faithfully estimate $\boldsymbol{\theta}^\star$. Concretely, this means that when the number of measurements n satisfies $n \gtrsim d$, the squared estimation error $\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\|_2^2$ scales like ${}^{d}\!/\!{}_{n}$. Requiring at least order d measurements is reasonable for small-scale problems, but becomes overly restrictive as the dimension of the problem grows. If, however, the parameter $\boldsymbol{\theta}^\star$ contains low-dimensional structure, then there is hope that the number of measurements (and error) scales with the ambient dimension of the problem.

One common example of structure in linear regression is *sparsity*, that is, only s entries of $\boldsymbol{\theta}^\star$ are non-zero, with $s \ll d$. In this setting, we should expect that the number of measurements needed to recover $\boldsymbol{\theta}^\star$ now scales with s instead of d. While intuitive, the mathematical machinery needed to prove such a result forms the basis of a rich field of study called *compressed sensing* [7, 8, 9]. In the presence of sparsity, we can deploy the regularized least-squares estimator

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (y_i - \langle \boldsymbol{x}_i, \boldsymbol{\theta} \rangle)^2 + \lambda \|\boldsymbol{\theta}\|_1$$

to estimate $\boldsymbol{\theta}^\star$. The ℓ_1 norm is chosen as a convex regularizer that enforces sparsity. Results from compressed sensing dictate that $n \gtrsim s$ measurements are needed to faithfully estimate $\boldsymbol{\theta}^\star$ (i.e., have squared estimation error scale like ${}^{s}\!/\!{}_{n}$, ignoring logarithmic factors.) This result holds under a random sensing model with particular choices of distribution for $\boldsymbol{x}_i$ and ϵ_i. One such distribution choice is that $\boldsymbol{x}_i$ and ϵ_i are independent from each other and are both drawn i.i.d from a standard Gaussian of correct dimension.

Analogous results hold in the matrix setting, which is considered in Chapter 3. When trying to estimate high-dimensional matrices from linear (trace) measurements, a natural form of structure is to assume that they are low-rank. Concretely, suppose we want to estimate a square matrix $\boldsymbol{\Sigma}^\star \in \mathbb{R}^{d \times d}$. Without structural assumptions, there are d^2 degrees of freedom in the problem, necessitating $n \gtrsim d^2$ measurements to achieve an squared error $\|\boldsymbol{\Sigma}^\star - \widehat{\boldsymbol{\Sigma}}\|_F^2$ that scales like ${}^{d^2}\!/\!{}_{n}$. However, if we assume that $\boldsymbol{\Sigma}^\star$ is rank r, with $r \ll d$, then the degrees of freedom is reduced to roughly rd (to see this, consider the singular value decomposition of $\Sigma^\star$.) As a result, if the number of measurements satisfies $n \gtrsim rd$, the squared estimation error scales like $^{rd}/_{n}$. These specific results again depend on specific distributional assumptions of random design and measurement noise, with one such satisfactory distribution being standard Gaussian [10, 11]. To enforce low-rank structure, we can utilize the nuclear norm as a form of regularization. The nuclear norm is computed by taking the ℓ_1 norm of the singular values of a matrix, so when used as a regularizer, it promotes sparsity in the singular values, inducing low-rank matrix estimates.

1.1.2 Learning from relational queries.

Mechanisms for collecting feedback from humans largely fall in two categories: *cardinal* queries or *ordinal* queries. Cardinal queries elicit numerical responses from users that answer "how much" of a particular quantity exists, whereas ordinal queries elicit relational responses that indicate how particular items are ordered. These two query categories occupy two different sides of a fundamental expressiveness-cognitive burden trade-off. On one hand, ordinal queries have been shown to be more efficient in terms of cognitive load (i.e., they are easy to respond to), but at the expense of expressiveness per query response [12, 13]. This limited expressiveness is fundamental to ordinal queries. For example, saying one item is "better" than another reveals no information about the absolute quality of both items, nor does it reveal how close in quality the two items are. On the other hand, cardinal responses elicit more expressive responses at the cost of cognitive burden. Obtaining absolute information about an item's quality is fundamentally a much richer response [14]. However, conveying precise scores is both individual-specific and cognitively burdensome: different people may have different scales [15] and their scales can drift over time [16, 17]. Such drawbacks make aggregating responses across many users extremely difficult.

Because training large-scale models relies on aggregating human responses, cardinal queries are often eschewed for their ordinal counterparts. This is best illustrated with the standard procedure of finetuning language models with ranked human feedback as opposed to absolute scores. As a result, this thesis primarily focuses on ordinal queries. Humans are presented with a set of items and asked about how items in the set related to each other. There exists an expressiveness-cognitive load trade-off within the class of ordinal queries: queries can be made more complex by increasing the size of the set of items presented to the human, asking the human to respond to a more complex relation, or both. Naturally, increasing the complexity of a query increases the amount of information the human is able to convey in one interaction. However, this increase in information richness does not come free; increasing the number of items users must consider increases the burden imposed on the human [18].

A simple and popular relational query is the paired comparison, where the human is presented two items and asked to respond with which item is more preferred. Such queries appear in a variety of contexts, ranging from recommender systems [19, 20] to finetuning language models [21, 22, 23, 24]. While easy to respond to, paired comparisons reveal relatively little information per query. As a result, more complex extensions of the paired comparison have been developed to extract richer responses. Examples include triplet queries [25, 26] ("Which of items i_1 or i_2 is most similar to item j?"), the nearest neighbor query [27] ("Which of items $i_1, \dots, i_n$ is most similar to item j?"), ranking queries [28] ("Rank order items $i_1, \dots, i_n$ in relation to item j").

Chapter 2 and Chapter 3 consider learning from relational queries using parametric models of human preference or perception. To illustrate learning in these settings, we consider the problem of learning from paired comparisons. Here, the goal is to estimate some unknown parameter $\boldsymbol{\theta}^\star \in \mathbb{R}^d$ that characterizes human preferences or perception. For convenient shorthand, we refer to this parameter $\boldsymbol{\theta}^\star$ as a user's *preference vector*. To perform statistical estimation, we typically assume that responses to paired comparisons follow a known random response model. One such example is the Bradley-Terry model [29], which dictates that a paired comparison response $y_i \in \{0, 1\}$ between two items, described by feature vectors $\boldsymbol{x}_{i_1}$ and $\boldsymbol{x}_{i_2}$, are Bernoulli distributed with mean parameterized by $\boldsymbol{\theta}^{\star}$:

$$y_i \sim \text{Ber}\left(\frac{\exp\left(f(\boldsymbol{x}_{i_1};\boldsymbol{\theta}^{\star})\right)}{\exp\left(f(\boldsymbol{x}_{i_1};\boldsymbol{\theta}^{\star})\right)+\exp\left(f(\boldsymbol{x}_{i_2};\boldsymbol{\theta}^{\star})\right)}\right).$$

Above f is some function that characterizes how "preferred" an item $\boldsymbol{x}_{i_j}$ is. Popular choices for f include the Euclidean inner product $\langle \boldsymbol{x}_{i_j}, \boldsymbol{\theta}^{\star}\rangle$ and distance $\|\boldsymbol{x}_{i_j} - \boldsymbol{\theta}^{\star}\|_2$. From this response model, we can estimate the preference vector $\boldsymbol{\theta}^{\star}$ via maximum-likelihood estimation:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log\left(1+\exp\left(f(\boldsymbol{x}_{i_1};\boldsymbol{\theta}^{\star}) - f(\boldsymbol{x}_{i_2};\boldsymbol{\theta}^{\star})\right)\right) - y_i\left(f(\boldsymbol{x}_{i_1};\boldsymbol{\theta}^{\star}) - f(\boldsymbol{x}_{i_2};\boldsymbol{\theta}^{\star})\right).$$

When f is chosen to be the standard Euclidean inner product, estimation from paired comparisons under the Bradley-Terry model coincides with logistic regression. As such, this problem setting is well-studied, with a long line of work focused on deriving statistical error bounds (e.g., [30, 31, 32, 33].) More generally, studying random response models that better reflect human behavior has been the focus of recent efforts [34, 35, 36, 37, 38].

1.1.3 Generative language and image models.

Pretrained models with both rich learned representations and strong generative abilities are prime candidates to augment the learning process, potentially allowing us to circumvent collecting additional human feedback. This section aims to provide a brief overview of contemporary generative models for natural language and images.

Natural language. The generative ability of large language models (LLMs) have demonstrated strong abilities in generating and understanding natural language. As a result, models such as GPT [39] or Llama [40, 41] are being deployed in many application areas from summarizing documents to code generation to language translation. The work of Chapter 4 utilizes the strong generative abilities of LLMs to better align how users interact with an information retrieval system to the content within the retrieval system. As a result, the retrieval system is capable of finding more relevant text without asking users for relevance feedback. LLMs are the byproduct of scaling: they contain billions (or even trillions) of parameters while trained on billions of tokens[1]. Such models are typically autoregressive in nature: When generating the next word in a sequence, the model is only conditioned on the previous words in the sequence. This modeling choice results in models that utilize decoder-only transformer architectures [42].

In order to achieve such strong empirical performance, LLMs follow the standard pre-train then finetune paradigm. During the pretraining stage, the emphasis is on data volume. Billions of tokens of text are scraped from a wide variety of sources and used to train the LLM in a self-supervised manner using a next token prediction objective. The aim of pretraining is to endow the model with a general understanding of the patterns and statistics that occur in natural language. From there, the model is adapted to specific tasks via supervised finetuning, where the model is trained on higher quality, labeled, and task-specific data. A final step in the current LLM development paradigm is to align models with human preferences [22]. The LLM is asked to generate multiple outputs for a given input, and humans are asked to rank the outputs based on specific desirable criteria, such as factfulness, non-toxicity, or helpfulness. Models are then finetuned using reinforcement learning to learn a preference scoring function [21, 22] or directly optimizing the language model [24].

Images. Recent advancements in generative image modeling have yielded models capable of synthesizing extremely realistic images. The work of Chapter 5 relies on the expressive power of generative adversarial networks (GANs) [43]. GANs consist of a generator that synthesizes new images and a discriminator that attempts to distinguish between synthesized images and real images. These two models are trained in an adversarial manner, with the

[1] Tokens are chunks of text that models take as input or generate. They are typically shorter than words; tokens typically contain approximately four English characters.

generator attempting to “trick” the discriminator and the discriminator attempting to “catch” the generator’s fake images. While this training procedure yields a generator capable of synthesizing extremely realistic images, training is often unstable.

Early research efforts focused on training stability [44, 45]. Later work focused on synthesizing larger and higher resolution images, leading to methods that progressively grow an image during generation [46]. This line of work lead to the popular StyleGAN variants [47, 48, 49], which utilized a novel *style block* architecture to allow for control over generated style and attributes. Due to the control over generated attributes, image editing with GANs has grown in popularity. To edit a real image, one must be able to find a representation of that image in the GAN’s latent space, leading to increased efforts in the field of *GAN inversion* [50]. Parallel efforts in large scale synthesis lead to BigGAN [51], which is capable of class conditional synthesis at ImageNet scale. More recent efforts have equipped GANs with the ability to generate images from user text inputs [52].

Diffusion models have become an increasingly popular type of generative image model due to the quality of synthesized images [53]. Such models learn to transform a simple noise distribution, such as Gaussian noise, to a target distribution, like natural images. Diffusion models are applied incrementally, gradually removing noise step-by-step until realistic samples of the target distribution are generated. This denoising process can occur in the pixel space [54, 55, 56] or in the latent space of a pretrained autoencoder [57]. In contrast to GANs, which exhibit fast inference speeds at the cost of synthesized image quality, diffusion models are generally capable of producing higher quality and more diverse images.

1.2 Thesis overview

The technical content of this thesis is organized into two parts. Part I investigates how to better learn from human feedback, whereas Part II studies how to circumvent human feedback in two concrete problem settings. Here, we provide a brief overview of the technical content in each of the two sections.

1.2.1 Learning with human feedback

Part I contributes to improving how we learn from human feedback in two distinct ways. In Chapter 2, we study an established human feedback mechanism, the paired comparison, and show that one can learn a much richer characterization of user preferences than previously thought. In Chapter 3, we propose a novel mechanism for collecting human feedback which balances informativeness per query and cognitive load. While not appearing in this thesis, I have also contributed to the development of a novel type of relational query, named the *nearest neighbor query* to bridge the gap between active classification and metric learning. We do not present this work in this thesis, but instead refer the interested reader to [27].

Simultaneous preference and metric learning from paired comparisons. In this chapter, I start with a very simple existing query, the paired comparison, and show that from binary responses, one can learn both a user's preferences and *how* they make their preference judgements. Specifically, under a distance based model for human preference, paired comparison responses are capable of localizing a user's preference point while also learning the *distance metric* under which users make their comparison judgments. Existing results from learning from paired comparisons have exclusively focused on learning either the user's preference point or a Mahalanobis distance metric. The work in this chapter reveals that the expressive power of paired comparisons has been underestimated in such prior work: *joint estimation* of preference points and distance metrics is possible from just paired comparison responses. The main contributions of this chapter are (1) the derivation of an estimator for this joint problem and (2) validation of this estimator on both synthetic and real-world data.

Perceptual adjustment queries and an inverted measurement paradigm for low-rank metric learning. In this chapter, I propose the *perceptual adjustment query* (PAQ), a novel mechanism for collecting human feedback. Users are presented with a reference item and a continuous set of points emanating from the reference item, and asked to select the first item along the path that is different from the reference. This query construction leverages the continuous nature of both human perception and features spaces used in contemporary machine learning models to balance *cognitive burden* and *informativeness*. Specifically, we study the problem of learning a low-rank Mahalanobis distance metric from PAQs in a high-dimensional setting. This problem gives rise to a novel type of measurement scheme for the typical low-rank matrix sensing problem, leading to the development of new statistical estimators. We rigorously prove sample complexity bounds under and validate our results with synthetic experiments.

1.2.2 Circumventing human feedback

Part II studies two concrete problems where human feedback cannot be collected. In order to circumvent using or collecting human feedback, both problems leverage the abilities of pretrained generative models. In Chapter 4, we study a large language model augmented retriever for the problem of personalized educational content retrieval. In Chapter 5, we leverage existing labeled images to remove a human-in-the-loop component of a synthetic dataset generation framework.

Large language model augmented exercise retrieval for personalized language learning. In this chapter, I study the problem of exercise retrieval for online language learners. Due *cold-start* constraints, user data is not often available at the quantity or quality necessary to train a recommendation system end-to-end. Despite this, language learning is a highly personalized setting, meaning strong personalization tools are necessary, even during this cold-start phase. Towards more reliable retrieval in this setting, we propose a zero-shot retrieval framework that utilizing the generative capabilities of large language models (LLMs). Our investigations reveal a fundamental semantic gap between how users express what they want to learn and the actual exercise content. In lieu of manually collecting relevance data and learning a gap-aware representation space, we propose using an off-the-shelf embedding model along with the generative abilities of an LLM. Coupled with a contrastive pretraining step that exploits inherent structure in exercise content, our framework is capable of outperforming several competitive existing retrieval systems, highlighting how the generative abilities of LLMs can be used to circumvent the need for collecting human relevance data.

Labeled dataset generation with no additional human annotations. In this chapter, I study the problem of synthetic dataset generation without humans-in-the-loop in the image domain. Existing frameworks for generating synthetic datasets with pixel level labels (e.g., semantic segmentation masks) require humans to *manually* annotate *images produced by a generative image model*. Such a requirement imposes harsh practical and methodological constraints: not only does one incur heavy start-up costs in establishing labeling infrastructure, one needs to invest financial resources paying annotators and auditing results. Relying on a human-in-the-loop imposes additional bottlenecks in experimentation: labeling must be complete before experiments can begin, training data is limited to images that have been labeled, and continuous valued labels, such as depth maps, cannot be collected from human annotators. Labeling labeled synthetic images further limits the usefulness of labeled data in different applications. We propose a framework to re-use *existing labels* of *real images* to train a synthetic dataset generation framework. We highlight how unsupervised machine learning techniques, specifically from the field of GAN inversion coupled with rich representations learned by generative models, can remove the need for humans in the loop entirely. Using a small number of labeled images (<50), we are able to achieve state-of-the-art performance in few-shot semantic segmentation, keypoint detection, and depth estimation.

Part I

Learning with human feedback

CHAPTER 2

LEARNING FROM SIMPLE HUMAN QUERIES: SIMULTANEOUS PREFERENCE AND METRIC LEARNING FROM PAIRED COMPARISONS.

In this chapter[1], we show that the expressiveness of an extremely simple query, the paired comparison, is much greater than established in previous work. In the context of human preference learning, previous work has established that localizing a user's preference point from paired comparisons is feasible. We show that from the same comparison responses, *joint* estimation of a user's preference point and a distance metric under which preference judgements are made is possible. Specifically, we present a novel approach to estimate the user's ideal point $\boldsymbol{u}$ and the Mahalanobis metric from paired comparisons of the form "item i is preferred to item j." This can be viewed as a special case of a more general metric learning problem where the location of some points are unknown *a priori*. We conduct extensive experiments on synthetic and real-world datasets to exhibit the effectiveness of our algorithm.

2.1 Introduction

Personalized recommendation and ranking algorithms have become increasingly important in recent years, influencing not only the items a user buys and movies he or she watches, but also potentially influencing which job candidates are interviewed, which college applicants are admitted, and even the matching behavior of online dating services. While there are a number of approaches to developing personalized recommendation systems, a particularly common approach uses a classical model for user preference known as the *ideal point model* [58]. In this model a user's preferences are represented as a point $\boldsymbol{u} \in \mathbb{R}^d$ that is embedded in the same space as a set of items $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^d$ (movies, shoes, food, etc). The key

[1] The work in this chapter appears in [1]

model assumption is that the closer an item $\boldsymbol{x}_j$ is to $\boldsymbol{u}$, the more the user will prefer item $\boldsymbol{x}_j$. We note that the ideal point is not necessarily a specific item $\boldsymbol{x}_i$, but rather represents the combination of features that the user most prefers. The ideal point model is an intuitive and interpretable way to model preferences and has been empirically shown to exhibit superior performance compared to other models of preference [59, 60].

In a practical system, the main challenge is to learn the latent $\boldsymbol{u}$ that represents a particular user's preferences. Given a precise quantification of a user's preferences for a number of items, one could infer the distances from $\boldsymbol{u}$ to those items and then easily estimate a good embedding $\boldsymbol{u}$. In practice, however, users find *paired comparison* queries of the form "do you prefer item i or item j" to be far easier to answer [18, 12]. As a result, a number of approaches to learning to rank from such paired comparisons have been proposed in recent years [61, 62, 63, 64, 30, 31, 65, 66, 67, 68, 69]. In the specific context of ideal point models, such queries allow the user to reveal which of the two items is closer to their ideal point. There is now a range of both practical algorithms for estimating $\boldsymbol{u}$ from such queries as well as theoretical treatments analyzing the performance of these algorithms in terms of error bounds and/or sample complexity guarantees [70, 64, 71, 72, 73, 74, 75, 26, 76, 77, 78].

While the problem of learning from paired comparisons in the ideal point setting is now well-understood, the vast majority of past work has only examined the case where the user makes judgements under the standard Euclidean distance metric. Assuming a Euclidean metric imposes two main limitations. First, it does not allow for features to interact. In practice, features often complement or compensate for each other. For example, consider the process of purchasing shoes. Each shoe can be described in terms of features such as color, price, materials, etc. An individual may prefer a cost of \$50 and a particular material. However, if the price was set instead to \$200, the user's preferred material may change to reflect the change in price – an effect that cannot be accommodated by a Euclidean (isotropic) metric. Second, the Euclidean metric assumes that all features are of equal importance to the user, which is often not the case. In the shoe purchasing example, a price conscious consumer may prioritize finding the best “bang for their buck,” in which case a lower price and higher quality of material would be prioritized over aesthetic features such as color.

To overcome these limitations, we consider the case where the user makes comparisons between items under a *Mahalanobis distance*. Specifically, let $\boldsymbol{\Sigma}^\star \in \mathbb{R}^{d \times d}$ be a symmetric positive definite matrix and set $\|\boldsymbol{x}\|_{\boldsymbol{\Sigma}^\star} = \sqrt{\boldsymbol{x}^T \boldsymbol{\Sigma}^\star \boldsymbol{x}}$. Then $\|\boldsymbol{x} - \boldsymbol{y}\|_{\boldsymbol{\Sigma}^\star}$ defines a Mahalanobis distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. This metric captures both feature interactions and the relative significance of those feature interactions via the eigenvalue decomposition $\boldsymbol{\Sigma}^\star = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$. The eigenvectors specify how features can interact to jointly affect preferences, and the eigenvalues allow for different combinations of features to play a larger or smaller role. See Section 2.4.2 below for a concrete example.

While a Mahalanobis metric allows for more flexible and powerful models of preference, the appropriate choice of $\boldsymbol{\Sigma}^\star$ will in general be unknown *a priori*. In this chapter, we develop a novel method to jointly learn both the ideal point and Mahalanobis metric from paired comparisons, which to the best of our knowledge represents the first approach for solving these problems simultaneously. By leveraging the structure of paired comparisons, we develop a simple convex optimization program that estimates $\boldsymbol{\Sigma}^\star$ and can then directly solve for $\boldsymbol{u}$. In the process, we also effectively learn the user’s ranking of the items. We also explore the possible benefits of a more sophisticated alternating scheme that iteratively refines the estimates of $\boldsymbol{\Sigma}^\star$ and $\boldsymbol{u}$. We demonstrate the effectiveness of our approach through experiments on both synthetic and real-world datasets.

2.2 Related work

Our work naturally builds on the existing literature on learning from paired comparisons, taking particular inspiration from the convex optimization approach to non-metric multidimensional scaling of [70] and the approaches in [64, 71, 74, 78] to developing algorithms for the ideal point setting. We also build on the extensive prior work on *metric learning*.

Learning a metric from paired comparisons was introduced in [79], where the authors assume the distance is parametrized by a known matrix $\boldsymbol{A}$ and a weighting matrix $\boldsymbol{W}$ with non-negative diagonal entries. $\boldsymbol{W}$ is learned via a convex program by manipulating the form of diagonal matrix multiplication. Setting $\boldsymbol{A}$ to $\boldsymbol{I}$ does not allow for feature interactions, whereas picking more complex $\boldsymbol{A}$ without overfitting the training data is non-trivial. Similarly in [80], the authors minimize the squared-hinge loss of differences in distances of pairs of items. However, the user is presented with two pairs of items must pick which pair of items is more similar, which is a more complex querying scheme for the user to answer. The same query type is used to learn a metric for images in [81]. The performance of nearest-neighbor-based classifiers have also benefited from learning a Mahalanobis metric that enhances class separation [82], where here class membership is used to inform the learning process.

Metric learning has also been explored in prior work on recommendation systems. For example, [83], [84], and [85] all learn Mahalanobis metrics for ranking given a known reference point and sets of similar and dissimilar items. Using sets of positive and negative items for each user, [86] learns a personalized projection operator for each user and estimates user preference in the learned latent space. In a similar setting, [87] learns transformed ideal points and items directly before learning a metric. Finally, [38] assumes each user has an ideal feature vector where user preference is measured by the inner product of this ideal feature vector with an item's feature vector and develops a feature selection scheme to account for intransitivity in noisy comparison outcomes while learning an ideal feature vector.

Our work differs from the above in that it uniquely assumes that *both* the metric and ideal point are unknown. Thus, it can be viewed as a more generalized metric learning problem where some of the data are missing. Most existing metric learning papers avoid the problem of knowing a user's preference by assuming a known reference point or utilizing more difficult queries (asking the user to compare two pairs of items). Based on this prior work, it might be unclear if simultaneous recovery of an unknown metric and ideal point is even feasible, but we show that it is indeed possible.

2.3 Observation model and estimation strategy

2.3.1 Observation model

For simplicity, we will begin by considering the noiseless observation model where the user always prefers the item closest to the user's ideal point $\boldsymbol{u}$ under the Mahalanobis distance metric induced by $\boldsymbol{\Sigma}^\star$, where $\boldsymbol{\Sigma}^\star$ is a symmetric positive definite matrix. To be more concrete, we let $\boldsymbol{\gamma} \in \mathbb{R}^n$ be the vector with entries $\gamma_i = \|\boldsymbol{x}_i - \boldsymbol{u}\|_{\boldsymbol{\Sigma}^\star}^2$. We let $\boldsymbol{y} \in \mathbb{R}^m$ denote our observations, where the k^{th} element of $\boldsymbol{y}$ denotes the outcome of the k^{th} comparison (between items $\boldsymbol{x}_{i_k}$ and $\boldsymbol{x}_{j_k}$) and is given by

$$y_k = \text{sign}(\gamma_{i_k} - \gamma_{j_k}). \tag{2.1}$$

For now we assume that the set of indices $\Omega = \{(i_1, j_1), \ldots, (i_m, j_m)\}$ corresponding to the items compared contains each pair of indices at most once, although our methods could easily be adapted to the case where Ω is a multiset. We will assume throughout our treatment that the embedding of the items $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ is fixed and known, as in a mature recommendation system. This embedding may correspond to known and interpretable features or be learned from other side information (or even paired comparisons, following a strategy along the lines of [88]).

Before we describe our estimation strategy, a few observations are in order. First, note that $\boldsymbol{y}$ consists of (1-bit) quantized samples of the $n \times n$ matrix $\boldsymbol{\Delta} = \boldsymbol{\gamma}\mathbf{1}_n^T - \mathbf{1}_n\boldsymbol{\gamma}^T$, where $\mathbf{1}_n \in \mathbb{R}^n$ is the vector of all ones. It will also be useful to work with the vectorized version of $\boldsymbol{\Delta}$, which we will denote by $\boldsymbol{\delta}$ and can be written as[2]

$$\boldsymbol{\delta} = (\mathbf{1}_n \otimes \boldsymbol{I}_n - \boldsymbol{I}_n \otimes \mathbf{1}_n)\boldsymbol{\gamma},$$

where $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix and $\otimes$ denotes the Kronecker product. For conciseness, we will let $\boldsymbol{Q} = \mathbf{1}_n \otimes \boldsymbol{I}_n - \boldsymbol{I}_n \otimes \mathbf{1}_n$.

To index into $\boldsymbol{\delta}$, we map every $(i_k, j_k) \in \Omega$ to a linear index between 1 and n^2 defined as $\Gamma = \{(i_k - 1)n + j_k : (i_k, j_k) \in \Omega\}$. We can equivalently write our observation model in (Equation 2.1) as

$$\boldsymbol{y} = \text{sign}(\boldsymbol{\delta}_\Gamma) = \text{sign}(\boldsymbol{Q}_\Gamma \boldsymbol{\gamma}), \tag{2.2}$$

where the notation $\boldsymbol{\delta}_\Gamma$ and $\boldsymbol{Q}_\Gamma$ indicates the vector or matrix obtained by selecting only the indices/rows indexed by Γ.

2.3.2 Estimation from unquantized observations

To gain some insight into this problem, we will temporarily ignore the quantization and suppose that we have direct access to $\boldsymbol{\delta}_\Gamma$ – in this case, how might we go about estimating $\boldsymbol{\Sigma}^\star$ and $\boldsymbol{u}$?

Consider $\gamma_{i_k} = \|\boldsymbol{x}_{i_k} - \boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star}$ and $\gamma_{j_k} = \|\boldsymbol{x}_{j_k} - \boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star}$ for any $(i_k, j_k) \in \Omega$. Then, for the linear index p corresponding to (i_k, j_k), $\boldsymbol{\delta}_p = \|\boldsymbol{x}_{i_k} - \boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star} - \|\boldsymbol{x}_{j_k} - \boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star}$. Observe that when we expand these terms we can cancel the coupled term $\boldsymbol{u}^T\boldsymbol{\Sigma}^\star\boldsymbol{u}$, greatly simplifying our subsequent analysis:

$$\begin{aligned}
\boldsymbol{\delta}_p &= \|\boldsymbol{x}_{i_k} - \boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star} - \|\boldsymbol{x}_{j_k} - \boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star} \\
&= \boldsymbol{x}_{i_k}^T\boldsymbol{\Sigma}^\star\boldsymbol{x}_{i_k} - 2\boldsymbol{x}_{i_k}^T\boldsymbol{\Sigma}^\star\boldsymbol{u} + \boldsymbol{u}^T\boldsymbol{\Sigma}^\star\boldsymbol{u} - (\boldsymbol{x}_{j_k}^T\boldsymbol{\Sigma}^\star\boldsymbol{x}_{j_k} - 2\boldsymbol{x}_{j_k}^T\boldsymbol{\Sigma}^\star\boldsymbol{u} + \boldsymbol{u}^T\boldsymbol{\Sigma}^\star\boldsymbol{u}) \\
&= \boldsymbol{x}_{i_k}^T\boldsymbol{\Sigma}^\star\boldsymbol{x}_{i_k} - \boldsymbol{x}_{j_k}^T\boldsymbol{\Sigma}^\star\boldsymbol{x}_{j_k} - 2(\boldsymbol{x}_{i_k} - \boldsymbol{x}_{j_k})^T\boldsymbol{\Sigma}^\star\boldsymbol{u}
\end{aligned} \tag{2.3}$$

[2] Note that this follows from the general identity vec$(\boldsymbol{ABC}) = (\boldsymbol{C}^T \otimes \boldsymbol{A})vec(\boldsymbol{B})$.

If we define

$$\boldsymbol{R} = \begin{bmatrix} — & (\boldsymbol{x}_{i_1} - \boldsymbol{x}_{j_1})^T & — \\ — & (\boldsymbol{x}_{i_2} - \boldsymbol{x}_{j_2})^T & — \\ & \vdots & \\ — & (\boldsymbol{x}_{i_m} - \boldsymbol{x}_{j_m})^T & — \end{bmatrix} \qquad \boldsymbol{S} = \begin{bmatrix} — & (\boldsymbol{x}_{i_1} + \boldsymbol{x}_{j_1})^T & — \\ — & (\boldsymbol{x}_{i_2} + \boldsymbol{x}_{j_2})^T & — \\ & \vdots & \\ — & (\boldsymbol{x}_{i_m} + \boldsymbol{x}_{j_m})^T & — \end{bmatrix}$$

then we can write (Equation 2.3) more concisely as

$$\boldsymbol{\delta}_\Gamma = \operatorname{diag}(\boldsymbol{S}\boldsymbol{\Sigma}^\star\boldsymbol{R}^T) - 2\boldsymbol{R}\boldsymbol{\Sigma}^\star\boldsymbol{u}, \tag{2.4}$$

where $\operatorname{diag}(\boldsymbol{A})$ returns the diagonal of $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ as a column vector. For brevity, let $\boldsymbol{a}_{\boldsymbol{\Sigma}^\star} = \operatorname{diag}(\boldsymbol{S}\boldsymbol{\Sigma}^\star\boldsymbol{R}^T)$. We now observe that if we observed $\boldsymbol{\delta}_\Gamma$ directly *and* already knew $\boldsymbol{\Sigma}^\star$, then we could estimate $\boldsymbol{u}$ by solving a standard least squares problem, resulting in the estimate

$$\widehat{\boldsymbol{u}} = \frac{1}{2}\left(\boldsymbol{\Sigma}^\star\right)^\dagger \boldsymbol{R}^\dagger(\boldsymbol{a}_{\boldsymbol{\Sigma}^\star} - \boldsymbol{\delta}_\Gamma). \tag{2.5}$$

Plugging this estimate into (Equation 2.4), we obtain a simple system of equations that is linear in $\boldsymbol{\Sigma}^\star$:

$$\begin{aligned} \boldsymbol{\delta}_\Gamma &= \boldsymbol{a}_{\boldsymbol{\Sigma}^\star} - 2\boldsymbol{R}\boldsymbol{\Sigma}^\star(\frac{1}{2}\left(\boldsymbol{\Sigma}^\star\right)^\dagger \boldsymbol{R}^\dagger)(\boldsymbol{a}_{\boldsymbol{\Sigma}^\star} - \boldsymbol{\delta}_\Gamma) \\ &= \boldsymbol{a}_{\boldsymbol{\Sigma}^\star} - \boldsymbol{R}\boldsymbol{\Sigma}^\star\left(\boldsymbol{\Sigma}^\star\right)^\dagger \boldsymbol{R}^\dagger(\boldsymbol{a}_{\boldsymbol{\Sigma}^\star} - \boldsymbol{\delta}_\Gamma) \\ &= \boldsymbol{a}_{\boldsymbol{\Sigma}^\star} - \boldsymbol{R}\boldsymbol{R}^\dagger(\boldsymbol{a}_{\boldsymbol{\Sigma}^\star} - \boldsymbol{\delta}_\Gamma), \end{aligned}$$

where the last equality follows from the fact that $\boldsymbol{\Sigma}^\star$ is assumed to be positive definite (and hence full-rank). Rearranging terms, we obtain the more convenient expression

$$\boldsymbol{0} = (\boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^\dagger)(\boldsymbol{a}_{\boldsymbol{\Sigma}^\star} - \boldsymbol{\delta}_\Gamma). \tag{2.6}$$

2.3.3 Single-step estimation from quantized observations

Given direct observations of $\boldsymbol{\delta}_\Gamma$, we could immediately estimate $\boldsymbol{\Sigma}^\star$ using (Equation 2.6). However, our observations as in (Equation 2.2) are (1-bit) quantized. In this case, using (Equation 2.6), we can instead formulate a constrained optimization problem to jointly estimate an $\boldsymbol{\Sigma}^\star$ and a set of distances $\boldsymbol{\gamma}$ (and hence $\boldsymbol{\delta}_\Gamma$) that are consistent with both our observations $\boldsymbol{y}$ and (Equation 2.6). Specifically, we will aim to find a solution that satisfies (Equation 2.6) while minimizing $\ell(\boldsymbol{\gamma})$, where $\ell(\boldsymbol{\gamma})$ is a loss function that encourages $\boldsymbol{\gamma}$ to be such that that the signs of entries of $\boldsymbol{\delta}_\Gamma = \boldsymbol{Q}_\Gamma\widehat{\boldsymbol{\gamma}}$ are consistent with the observed comparisons. For example, one could set $\ell(\boldsymbol{\gamma})$ to be the *hinge loss*:

$$\ell(\boldsymbol{\gamma}) = \sum_{k=1}^{m} \max(0, 1 - y_k(\boldsymbol{Q}_\Gamma\boldsymbol{\gamma})_k), \tag{2.7}$$

where $(\boldsymbol{Q}_\Gamma\boldsymbol{\gamma})_k$ denotes the k^{th} element in the vector $\boldsymbol{Q}_\Gamma\boldsymbol{\gamma}$. In the remainder of this chapter and in our experiments, we use the hinge loss, but our framework could easily be extended to accommodate any convex loss. Finally, in our proposed approach we also introduce slack variables $\boldsymbol{\zeta} \in \mathbb{R}^m$ to loosen the constraint (Equation 2.6) to improve stability and robustness to noise, and also introduce terms to the objective function to allow for a small amount of regularization on both $\boldsymbol{\Sigma}^\star$ and $\boldsymbol{\gamma}$:

$$\begin{aligned}(\widehat{\boldsymbol{\Sigma}}, \widehat{\boldsymbol{\gamma}}, \widehat{\boldsymbol{\zeta}}) = &\mathop{\arg\min}_{\boldsymbol{\Sigma}^\star, \boldsymbol{\gamma}, \boldsymbol{\zeta}} \ \ell(\boldsymbol{\gamma}) + \beta_1\|\boldsymbol{\zeta}\|_1 + \beta_2\|\boldsymbol{\Sigma}^\star\|_F^2 + \beta_3\|\boldsymbol{\gamma}\|_2^2 \\ &\text{s.t.} \quad -\boldsymbol{\zeta} \le (\boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^\dagger)(\operatorname{diag}(\boldsymbol{S}\boldsymbol{\Sigma}^\star\boldsymbol{R}^T) - \boldsymbol{Q}_\Gamma\boldsymbol{\gamma}) \le \boldsymbol{\zeta} \\ &\qquad \boldsymbol{\zeta} \ge \boldsymbol{0}, \quad \boldsymbol{\Sigma}^\star \succcurlyeq \boldsymbol{0}.\end{aligned} \tag{2.8}$$

The first two constraints aim to enforce (Equation 2.6), while the final constraint enforces that $\widehat{\boldsymbol{\Sigma}}$ is symmetric positive semi-definite. The constants $\beta_1, \beta_2, \beta_3$ are parameters set by the user to control the amount of regularization. The above formulation is a convex (semi-definite) program and can be solved by standard tools such as CVX [89, 90].

With $\widehat{\boldsymbol{\Sigma}}$ in hand, we can then immediately solve for $\widehat{\boldsymbol{u}}$ via (Equation 2.5). However, since we do not expect our estimate $\boldsymbol{\Sigma}^\star$ to be perfect, we instead use the regularized estimate:

$$\widehat{\boldsymbol{u}} = \frac{1}{2}(\widehat{\boldsymbol{\Sigma}}\boldsymbol{R}^T\boldsymbol{R}\widehat{\boldsymbol{\Sigma}} + \alpha\boldsymbol{I})^{-1}\widehat{\boldsymbol{\Sigma}}\boldsymbol{R}^T(\boldsymbol{a}_{\widehat{\boldsymbol{\Sigma}}} - \boldsymbol{Q}_\Gamma\widehat{\boldsymbol{\gamma}}), \tag{2.9}$$

where α is a regularization parameter set by the user. We will see in Section 2.4 that this simple single-step estimation procedure of estimating $\boldsymbol{\Sigma}^\star$ followed by $\boldsymbol{u}$ is surprisingly effective.

2.3.4 Noise considerations

A common noise setting is when paired comparison outcomes are made with respect to differences in distances corrupted by additive i.i.d noise. Such noise may arise from an imperfectly learned embedding or as a way of modeling response errors. While we pose the problem in a noiseless environment for simplicity, the estimation strategy outlined above can be adapted to accommodate such noise by replacing the loss function in (Equation 2.7) with the negative log-likelihood of observing comparison outcomes given a noise model, provided that the log-likelihood function is concave. For example, suppose we assume paired comparison outcomes follow the *Bradley-Terry model* [29], i.e.,

$$\mathbb{P}(\boldsymbol{x}_i \succ \boldsymbol{x}_j) = \frac{e^{-\|\boldsymbol{x}_i-\boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star}}}{e^{-\|\boldsymbol{x}_i-\boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star}} + e^{-\|\boldsymbol{x}_j-\boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star}}}, \tag{2.10}$$

where $\boldsymbol{x}_i \succ \boldsymbol{x}_j$ denotes that item $\boldsymbol{x}_i$ is preferred to item $\boldsymbol{x}_j$. Let y_k denote the outcome of the k^{th} comparison, with $y_k = -1$ if $\boldsymbol{x}_i \succ \boldsymbol{x}_j$ and $y_k = +1$ if $\boldsymbol{x}_j \succ \boldsymbol{x}_i$. Then, we can replace the loss function in (Equation 2.7) with the negative log-likelihood of observing the m paired comparison outcomes:

$$\ell(\boldsymbol{\gamma}) = \sum_{k=1}^{m} \log(1 + e^{-y_k(\boldsymbol{Q}_\Gamma\boldsymbol{\gamma})_k}), \tag{2.11}$$

where $(\boldsymbol{Q}_\Gamma\boldsymbol{\gamma})_k$ denotes the k^{th} entry of the vector $\boldsymbol{Q}_\Gamma\boldsymbol{\gamma}$

2.3.5 Alternating estimation

While the single-step approach described above is appealing due to its simplicity, the process of dividing the problem into first estimating $\boldsymbol{\Sigma}^\star$ and then estimating $\boldsymbol{u}$ suggests a natural extension of then taking our estimate of $\boldsymbol{u}$ and refining our estimate of $\boldsymbol{\Sigma}^\star$, and then iteratively alternating between these two problems to (potentially) improve our estimates. Specifically, after obtaining $\widehat{\boldsymbol{\Sigma}}$ and $\widehat{\boldsymbol{u}}$ using (Equation 2.8) and (Equation 2.9), we can set $\widehat{\boldsymbol{\Sigma}}^{(0)} = \widehat{\boldsymbol{\Sigma}}$ and $\widehat{\boldsymbol{u}}^{(0)} = \widehat{\boldsymbol{u}}$. Then, for iteration $k > 0$, we re-estimate $\boldsymbol{\Sigma}^\star$ by solving the following optimization problem that replaces the constraint from (Equation 2.6) with the constraint from (Equation 2.4) to allow us to directly incorporate our previous estimate of $\boldsymbol{u}$:

$$\begin{aligned}(\widehat{\boldsymbol{\Sigma}}^{(k)}, \widehat{\boldsymbol{\gamma}}^{(k)}, \widehat{\boldsymbol{\zeta}}^{(k)}) = &\underset{\boldsymbol{\Sigma}^\star,\boldsymbol{\gamma},\boldsymbol{\zeta}}{\arg\min}\ \ell(\boldsymbol{\gamma}) + \beta_1^{(k)}\|\boldsymbol{\zeta}\|_1 + \beta_2^{(k)}\|\boldsymbol{\Sigma}^\star\|_F^2 + \beta_3^{(k)}\|\boldsymbol{\gamma}\|_2^2 \\ &\text{s.t.} \quad -\boldsymbol{\zeta} \le \operatorname{diag}(\boldsymbol{S}\boldsymbol{\Sigma}^\star\boldsymbol{R}^T) - \boldsymbol{Q}_\Gamma\boldsymbol{\gamma} - 2\boldsymbol{R}\boldsymbol{\Sigma}^\star\widehat{\boldsymbol{u}}^{(k-1)} \le \boldsymbol{\zeta} \\ &\qquad\quad \boldsymbol{\zeta} \ge \boldsymbol{0}, \quad \boldsymbol{\Sigma}^\star \succcurlyeq \boldsymbol{0}.\end{aligned} \tag{2.12}$$

We can then update our estimate of $\boldsymbol{u}$ as before:

$$\widehat{\boldsymbol{u}}^{(k)} = \frac{1}{2}(\widehat{\boldsymbol{\Sigma}}^{(k)}\boldsymbol{R}^T\boldsymbol{R}\widehat{\boldsymbol{\Sigma}}^{(k)} + \alpha^{(k)}\boldsymbol{I})^{-1}\widehat{\boldsymbol{\Sigma}}^{(k)}\boldsymbol{R}^T(\boldsymbol{a}_{\widehat{\boldsymbol{\Sigma}}^{(k)}} - \boldsymbol{Q}_\Gamma\widehat{\boldsymbol{\gamma}}^{(k)}).$$

Note that we allow the regularization parameters to change across iterations. In practice we fix $\beta_1^{(k)}$, $\beta_2^{(k)}$, $\beta_3^{(k)}$, and $\alpha^{(k)}$ for all iterations $k \ge 1$, but we do consider an alternative set of parameters for the initialization ($k = 0$). This is somewhat natural since the initialization step actually involves solving a slightly different optimization problem.

2.3.6 Identifiability of the metric and ideal point

We conclude our description of our approach with a brief discussion of the degree to which the ideal point $\boldsymbol{u}$ and metric $\boldsymbol{\Sigma}^\star$ are potentially identifiable.

Proposition 1. *For a fixed $\boldsymbol{\Sigma}^\star \in \mathbb{R}^{d \times d}$, the ideal point $\boldsymbol{u}$ is identifiable if and only if $\boldsymbol{\Sigma}^\star$ is (strictly) positive definite.*

The proof is provided in the supplementary material and is similar to the proof of Proposition 3 in [38]. This result is not surprising, as if $\boldsymbol{\Sigma}^\star$ is rank-deficient, any part of $\boldsymbol{u} \in \ker(\boldsymbol{\Sigma}^\star)$ will be not recoverable. We note that, since we desire our constraint set to be closed, our estimation procedure enforces a positive *semi*-definite constraint. In practice, if $\boldsymbol{\Sigma}^\star$ is ill-conditioned, we may estimate a solution which is rank-deficient (which ignores the eigenvectors corresponding to relatively small eigenvalues), and thus the portion of $\boldsymbol{u}$ in the span of these eigenvectors may be extremely difficult to estimate. Note, however, that due to the influence of $\boldsymbol{\Sigma}^\star$, the unidentifiable portion of $\boldsymbol{u}$ also plays little role in determining the underlying preferences. In recognition of this, we typically quantify our estimation performance in terms of $\|\widehat{\boldsymbol{u}} - \boldsymbol{u}\|_{\boldsymbol{\Sigma}^\star}$.

We also note that, at least in the noise-free setting considered in this chapter, even if the metric is fully identifiable, it will only be so up to a constant scaling factor. However, note that a constant scaling factor does not change the learned ideal point or ranking of items. To see why this is true, note that for any $\boldsymbol{\Sigma}^\star$ and $\boldsymbol{\gamma}$ satisfying the constraint in (Equation 2.4), rescaling $\boldsymbol{\Sigma}^\star$ and $\boldsymbol{\gamma}$ (and hence $\boldsymbol{\delta}_\Gamma$) by an arbitrary constant $c > 0$ will yield another valid solution. However, it is relatively easy to show that for arbitrary scaling of $\boldsymbol{\Sigma}^\star$ and $\boldsymbol{\delta}_\Gamma$, the estimate provided by (Equation 2.5) of $\widehat{\boldsymbol{u}}$, as well as the resulting ranking of the items, remains unchanged.

Finally, we also note that when recovering $\boldsymbol{\Sigma}^\star$, if a subset of the eigenvalues of $\boldsymbol{\Sigma}^\star$ are equal or relatively close, it becomes impossible, or at least more difficult, to distinguish among the specific eigenvectors. In this case, our estimate may swap the order of the eigenvectors or learn different eigenvectors that span a similar space to the original, but can be quite different. As with scaling, this has little impact on estimating $\boldsymbol{u}$ or in terms of the resulting rankings, but plays an important factor in determining the appropriate evaluation metrics.

2.4 Experiments

2.4.1 Synthetic experiments

In this section, we demonstrate the effectiveness of the joint estimation on synthetically generated data. We assume *a priori* knowledge of an existing embedding of items and estimate $\boldsymbol{u}$ and $\boldsymbol{\Sigma}^\star$. In each simulation, n items $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are generated uniformly on the hypercube $[-2, 2]^d$ and one user $\boldsymbol{u}$ is generated uniformly on $[-1, 1]^d$. A positive definite matrix $\boldsymbol{\Sigma}^\star$ is generated by $\boldsymbol{\Sigma}^\star = \boldsymbol{L}^T\boldsymbol{L}$, where the entries of $\boldsymbol{L} \in \mathbb{R}^{d \times d}$ are drawn from the standard normal distribution. Comparisons are chosen uniformly without repetition and used to estimate the metric and ideal point.

Certain conditions are imposed on the matrix $\boldsymbol{\Sigma}^\star$: 1) The Frobenius norm of $\boldsymbol{\Sigma}^\star$ exceeds a small chosen threshold ϵ_F, 2) The smallest singular value of $\boldsymbol{\Sigma}^\star$ is larger than a small chosen threshold ϵ_S, and 3) The fraction $\|\boldsymbol{\Sigma}^\star\boldsymbol{u}\|_2/\|\boldsymbol{u}\|_2$ exceeds a small chosen threshold threshold ϵ_P. ϵ_F and ϵ_S are imposed to guard against numerical instabilities while ϵ_P is necessary to ensure that $\boldsymbol{u}$ is identifiable. For all synthetic experiments, the chosen values were $\epsilon_F = 0.5$, $\epsilon_S = 0.25$, and $\epsilon_P = 0.2$.

We define the user's ideal point reconstruction (UR) error as $\|\widehat{\boldsymbol{u}} - \boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star}/\|\boldsymbol{u}\|^2_{\boldsymbol{\Sigma}^\star}$. Letting the eigendecompositions of $\boldsymbol{\Sigma}^\star$ and $\widehat{\boldsymbol{\Sigma}}$ be $\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ and $\widehat{\boldsymbol{V}}\widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{V}}^T$, respectively, we define the weighted eigenstructure reconstruction (WER) error as $\|\boldsymbol{\Lambda} \odot |\boldsymbol{V}^T\widehat{\boldsymbol{V}}| - \boldsymbol{\Lambda}\|^2_F/\|\boldsymbol{\Lambda}\|^2_F$, where $\odot$ denotes element-wise multiplication and $|\boldsymbol{A}|$ takes the element-wise absolute value of $\boldsymbol{A}$. When $\widehat{\boldsymbol{\Sigma}}$ is recovered to be a scaled version of $\boldsymbol{\Sigma}^\star$, we expect the diagonal elements of $|\boldsymbol{V}^T\widehat{\boldsymbol{V}}|$ to be 1. In all cases when the WER error is small, $\boldsymbol{\Sigma}^\star$ is recovered well. However, there are instances in which a high value of the WER error does not imply a poor estimate

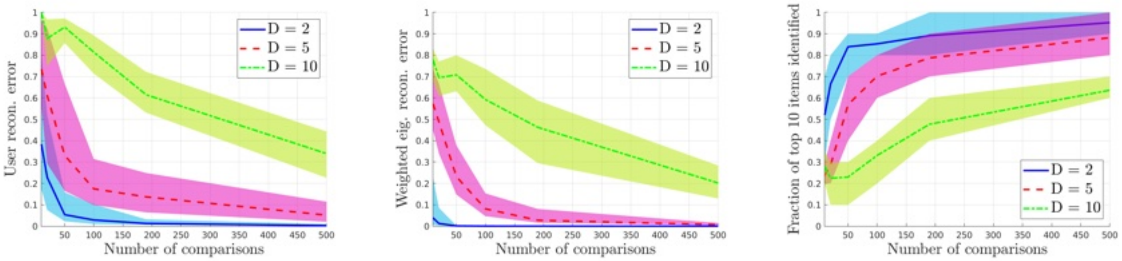


Figure 2.1: Median UR error, WER error, and interpolated median fraction of top 10 items identified over 100 trials, plotted with 25% and 75% quantiles. As the number of comparisons grows, both UR and WER error decrease to 0 as the fraction of top 10 items increases to 1 for all d. Regularization parameters: $\beta_1 = 2, \beta_2 = 0.002, \beta_3 = 0.001, \alpha = 1$.

of $\boldsymbol{\Sigma}^\star$. For example, (large) repeated eigenvalues in $\boldsymbol{\Sigma}^\star$ would result in a large WER error if the eigenvectors in $\widehat{\boldsymbol{V}}$ differed, but spanned the same space. Our synthetic data avoids this, but care is needed to quantify performance in general.

Single-step estimation In the first simulation, we show the improvement in estimation as the number of comparisons increases. For a fixed number of comparisons, we perform 100 trials and report the median UR error and WER error, and interpolated median of the fraction of the top 10 closest items to $\boldsymbol{u}$ identified for $d = 2, 5,$ and 10. Since the fraction of the top 10 items is discrete, we utilize the interpolated median in place of the usual median. In all cases, we include the 25% and 75% quantiles. For each trial, we generate a new metric, ideal point, and $n = 100$ items.

As shown in Figure 2.1, when a small number of comparisons are used for joint estimation, the UR and WER error are large, while the fraction of top 10 items correctly identified is small. As the number of comparisons increases from 10 to 500, the UR and WER errors decrease rapidly, while the fraction of top 10 items increases rapidly.

In the second simulation, we compare the performance of our algorithm against two algorithms that assume Euclidean distance to estimate the ideal point. **Euclidean Algorithm 1** is an adaptation of our single-step algorithm to solve for only the distances γ_e between

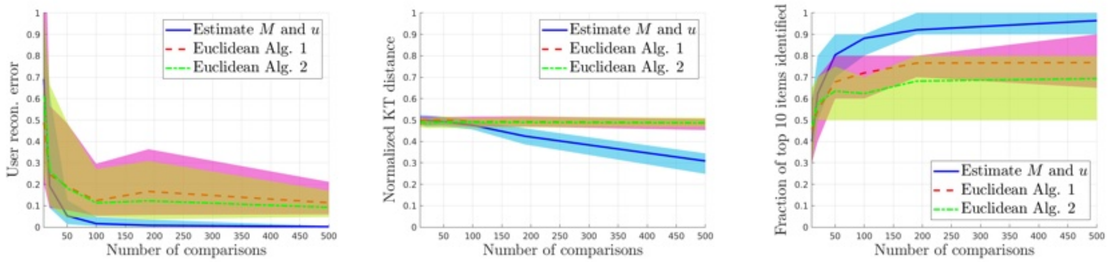


Figure 2.2: Comparison of singe-step estimation against Euclidean Algorithms 1 and 2 when the true distance metric is $\boldsymbol{\Sigma}^\star \neq \boldsymbol{I}$. Regularization parameters: $\beta_1 = 2, \beta_2 = 0.002, \beta_3 = 0.001, \alpha = 1$.

items and the ideal point:

$$\begin{aligned}(\widehat{\boldsymbol{\gamma}}_e, \widehat{\boldsymbol{\zeta}}) = &\arg\min_{\boldsymbol{\gamma},\boldsymbol{\zeta}}\ \ell(\boldsymbol{\gamma}_e) + \beta_1\|\boldsymbol{\zeta}\|_1 + \beta_2\|\boldsymbol{\gamma}_e\|_2^2 \\ &\text{s.t.} \quad -\boldsymbol{\zeta} \leq (\boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^\dagger)(\mathrm{diag}(\boldsymbol{S}\boldsymbol{R}^T) - \boldsymbol{Q}_\Gamma \boldsymbol{\gamma}_e) \leq \boldsymbol{\zeta}, \qquad \boldsymbol{\zeta} \geq \boldsymbol{0}.\end{aligned} \tag{2.13}$$

From here, we can solve for $\widehat{\boldsymbol{u}}$ by replacing replacing $\widehat{\boldsymbol{\Sigma}}$ with $\boldsymbol{I}$ in (Equation 2.9). **Euclidean Algorithm 2** is the approach in [74], which directly solves a convex program for $\boldsymbol{u}$ from the paired comparisons.

We sweep the performance for all three algorithms for $d = 2$ over different numbers of comparisons between 10 and 500. For a fixed number of comparisons, we perform 100 trials and report the median (or interpolated median) and 25% and 75% quantile for UR error, normalized Kendall's Tau distance, and the fraction of top 10 items identified. For each trial, we generate a new metric and ideal point and $n = 100$ new items. As seen in Figure 2.2, our algorithm outperforms both algorithms that assume a Euclidean distance metric by recovering a more accurate ideal point, ranking of items, and fraction of top K items. The same experiment was performed when $\boldsymbol{\Sigma}^\star = \boldsymbol{I}$ for all trials with very little loss in performance by using our algorithm (see the supplementary material for further details).

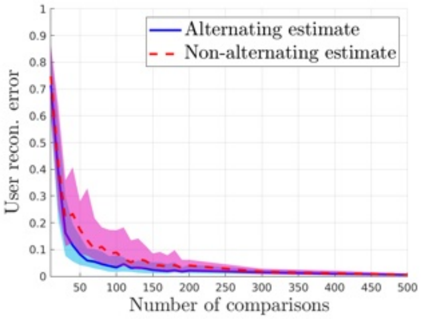


Figure 2.3: UR error for single-step and alternating estimation. Regularization parameters: $\beta_1^{(0)} = 2, \beta_2^{(0)} = 0.002, \beta_3^{(0)} = 0.0001, \alpha^{(0)} = 1$; $\beta_1^{(k)} = \frac{2}{3}, \beta_2^{(k)} = \frac{1}{15}, \beta_3^{(k)} = \frac{7}{1500}, \alpha^{(k)} = \frac{1}{2}$ for $k \geq 1$.

Alternating estimation We now explore the potential improvements that can be attained by our alternating estimation procedure. For $d = 5$, we fix an ideal point, metric, and a set of $n = 100$ items, and vary the number of comparisons. For a fixed number of comparisons m, we run 100 trials, where we select m new comparisons at random. We then run the alternating descent until the difference in the user reconstruction error between successive iterations is less than 10^{-3}, with a maximum number of iterations set to 100. We report the median and 25% and 75% quantiles for the initial and final UR error in Figure 2.3. We observe that alternating estimation does not improve the estimate of $\boldsymbol{u}$ much when the number of comparisons is small (< 40) or large (> 200). In the first regime, the comparisons do not reveal enough information to reliably recover $\boldsymbol{u}$, while in the second regime, the number of comparisons is sufficient to make the single-step estimation very accurate. The alternating method offers steady improvement in the intermediate regime, and is able to successfully reduce the error nearly 60%.

2.4.2 Graduate admissions dataset

We now apply our models to two PhD program admissions datasets from Georgia Tech School of Electrical and Computer Engineering. The *Unranked Candidates* dataset consists of over 3,000 applicants in three categories: 1) admitted with fellowship, 2) admitted, and 3) denied admission. The applicants are not ranked, so the only paired comparisons we can form are across categories. We assume that fellowship recipients are preferred to admitted candidates, who are preferred to denied candidates, so for n_F fellowship recipients, n_A admitted candidates, and n_D denied candidates, we can form at most $n_F(n_A + n_D) + n_A n_D$ comparisons. For each applicant, we have access to five features: GPA, GRE quantitative, verbal, and analytical writing scores, and a letter of recommendation (LoR) score. Each candidate's GPA is normalized to a 4.0 scale. The GRE verbal and quantitative scores are integers between 130 and 170, inclusive, while the GRE writing score is from 0 to 6 in 0.5 increments. Each candidate submitted at most three letters of recommendation, each of which is scored on a scale of 0 to 3. The scores are averaged and then exponentiated to obtain a LoR score between 1 and $e^3 \approx 20.09$.

The *Ranked Candidates* dataset consists of 88 applicants who are scored on a scale of 1 to 10, with 1 being the most preferred and 10 being least preferred. The top 11 candidates are uniquely rank ordered, and the rest of the candidates are sorted into 8 bins of various sizes. We only form comparisons between candidates with different scores, so two candidates with the same score are not compared. For each applicant, we have access to the same features except for letter of recommendation scores.

Unranked Candidates We begin by noting that the features being used in this model are inherently restrictive. Applicants are evaluated on many criteria beyond the features included, which can lead to occasional unexpected results. For instance, there exist large subsets of denied candidates whose average GRE scores are higher than those of a some fellowship recipients, which might indicate that a lower GRE score is more favorable, occasionally leading to rather unusual ideal point placement. In reality, we would expect that the optimal set of features should be the maximum value for all possible features. Furthermore, of the five features, we suspect that the GRE verbal score should likely matter the least, followed by the GRE quantitative score, as applicants from across the categories score similarly on

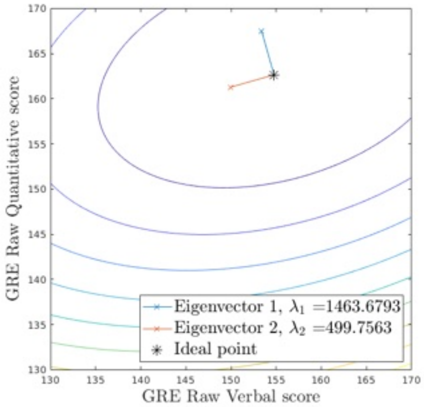


Figure 2.4: Level sets for learned metric for *Unranked Candidates* GRE verbal and quantitative scores. Regularization parameters: $\beta_1 = \frac{1}{650}, \beta_2 = \frac{1}{6500}, \beta_3 = \frac{2}{65} \cdot 10^{-6}, \alpha = 1$.

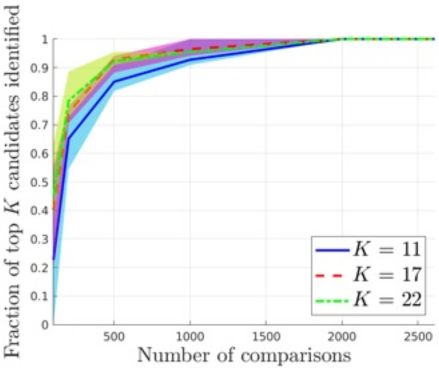


Figure 2.5: Fraction of top 11, 17, and 22 of ranked candidates identified. Regularization parameters: $\beta_1 = \frac{3}{800}, \beta_2 = \frac{1}{8000}, \beta_3 = \frac{5}{8} \cdot 10^{-11}, \alpha = 1$.

these two GRE sections. Our expectation is that the most significant features should be some combination of GRE writing, GPA, and LoR. With this in mind, we use our algorithm to learn relevant feature interactions and confirm our hypothesized ordering of the importance of features via the learned metric. We take $n_F = 33$, $n_A = 33$, and $n_D = 34$, form all 3333 possible comparisons, and learn the ideal point $\boldsymbol{u}$ and metric $\boldsymbol{\Sigma}^\star$ using a subset of all features.

When all five features are used to learn $\widehat{\boldsymbol{u}}$ and $\widehat{\boldsymbol{\Sigma}}$, our hypothesized ordering of importance for the features is correct. The three most significant features are a weighted difference between GPA and GRE writing, a weighted sum of GPA and GRE writing, and the LoR score. The learned ideal scores are 158.08 GRE verbal, 162.50 GRE quantitative, 4.68 GRE writing, 4.06 GPA, 15.28 LoR score. As seen in Figure 2.4, when the GRE verbal and quantitative scores features are used, the learned feature interactions are a weighted difference (eigenvector 1) and sum (eigenvector 2) of GRE verbal and quantitative scores. The structure of the learned metric seems to make intuitive sense, indicating that in order to compensate for a slightly lower GRE quantitative score, one must score significantly higher on the GRE verbal section.

Ranked Candidates Since ranking information is partially available in this dataset, we record the fraction of top $K = 11, 17$, and 22 candidates correctly identified as the number of comparisons increases using all four features. For a fixed number of comparisons, we perform 20 trials and report the mean and standard deviation of the fraction of the top K candidates correctly identified in Figure 2.5. The fraction of the top K candidates correctly identified for $K = 11, 17$, and 22 increases rapidly as the number of comparisons increases. With less than 20$\%$ of the total number of comparisons, we can identify over 90$\%$ of the top 22 and 17 candidates and over 80$\%$ of the top 11 candidates correctly.

2.5 Conclusions

In this chapter, we develop a method for jointly learning a user's ideal point and an underlying distance metric from paired comparisons. The metric captures feature interactions and their relative significance to users, neither of which are captured by the traditional Euclidean metric. We demonstrate our algorithm can correctly identify the ideal point and metric and can correctly rank graduate admission candidates and determine feature interactions on real-world data. We conclude by noting that in the Euclidean setting, adaptive querying schemes have been shown to enable dramatic reductions in the required number of comparisons [64, 91]. We expect similar gains are possible in our setting. Developing novel methods for adaptively selecting comparisons to maximize the amount of information collected about both $\boldsymbol{u}$ as well as $\boldsymbol{\Sigma}^\star$ is an important avenue for future research.

CHAPTER 3
DESIGNING MECHANISMS FOR RICHER HUMAN FEEDBACK: PERCEPTUAL ADJUSTMENT QUERIES AND AN INVERTED MEASUREMENT PARADIGM FOR LOW-RANK METRIC LEARNING

In this chapter[1], we introduce a new type of query mechanism for collecting human feedback, called the perceptual adjustment query (PAQ). Being both informative and cognitively lightweight, the PAQ adopts an inverted measurement scheme, and combines advantages from both cardinal and ordinal queries. We showcase the PAQ in the metric learning problem, where we collect PAQ measurements to learn an unknown Mahalanobis distance. This gives rise to a high-dimensional, low-rank matrix estimation problem to which standard matrix estimators cannot be applied. Consequently, we develop a two-stage estimator for metric learning from PAQs, and provide sample complexity guarantees for this estimator. We present numerical simulations demonstrating the performance of the estimator and its notable properties.

3.1 Introduction

Should we query cardinal or ordinal data from people? This question arises in a broad range of applications, such as in conducting surveys [92, 93, 94], grading assignments [95, 96], evaluating employees [97], and comparing or rating products [98, 99], to name a few. *Cardinal* data are numerical scores. For example, teachers score writing assignments in the range of 0-100, and survey respondents express their agreement with a statement on a scale of 1 to 7. *Ordinal* data are relations between items, such as pairwise comparisons (choosing the better item in a pair) and rankings (ordering all or a subset of items). There is no free lunch, and both cardinal and ordinal queries have pros and cons.

[1] The work in this chapter appears in [2]

On the one hand, collecting ordinal data is typically more efficient in terms of worker time and cognitive load [13], and surprisingly often matches or exceeds the accuracy of cardinal data [92, 13]. The information contained in ordinal queries, however, is fundamentally limited and lacks expressiveness. For example, pairwise comparisons elicit binary responses where two items are compared against each other, but the absolute placement of these items with respect to the entire pool is lost. On the other hand, cardinal data are more expressive [14]. For example, assigning two items scores of 1 and 2 conveys a very different message from assigning them scores of 9 and 10, or 1 and 10, although all yield the same pairwise comparison outcome. However, the expressiveness of cardinal data often comes at the cost of miscalibration: Prior work has shown that different people have different scales [15], and even a single person's scale can drift over time (e.g., [16, 17]). These inter-person and intra-person discrepancies make it challenging to interpret and aggregate raw scores effectively.

The goal of this chapter is to study whether one can combine the advantages of cardinal and ordinal queries to achieve the best of both worlds. Specifically, we pose the research question:

> *Can we develop a new paradigm for human data elicitation that is expressive, accurate, and cognitively lightweight?*

Towards this goal, we extract key features of both cardinal and ordinal queries, and propose a new type of query scheme that we term the *perceptual adjustment query* (PAQ). As a thought experiment, consider the task of learning an individual's preferences between modes of transport. The query can take the following forms:

- **Ordinal:** Do you prefer a \$2 bus ride that takes 40 minutes or a \$25 taxi that takes 10 minutes?

- **Cardinal:** On a scale of 0 to 1, how much do you value a \$2 bus ride that takes 40 minutes?

- **Proposed approach:** To reach the same level of preference for a \$2 bus trip that takes 40 minutes, a taxi that takes 10 minutes would cost \$$x$.

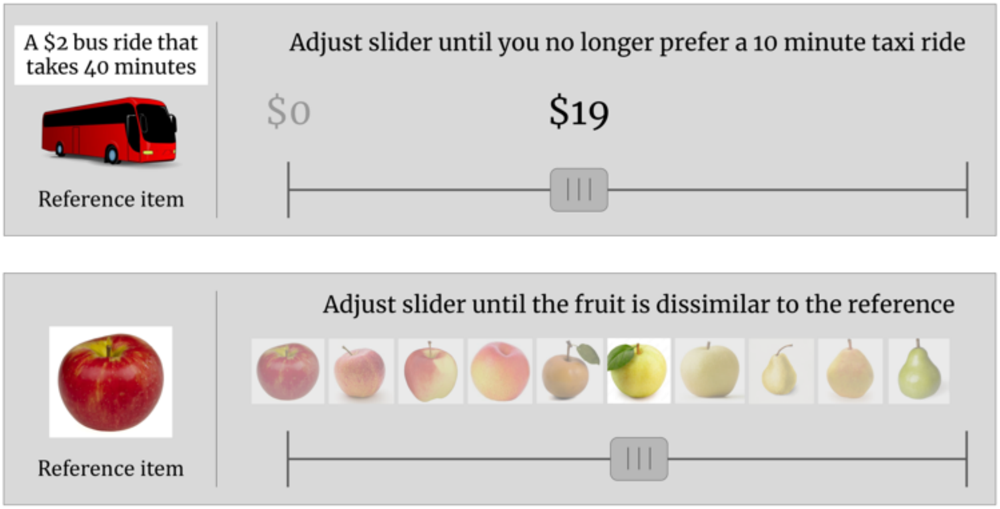

A $2 bus ride that takes 40 minutes
Reference item
Adjust slider until you no longer prefer a 10 minute taxi ride
$0
$19
Reference item
Adjust slider until the fruit is dissimilar to the reference

Figure 3.1: The user interface for perceptual adjustment query (PAQ) for preference learning (top) and similarity learning (bottom).

A user interface for the proposed approach is shown in Figure 3.1 (top). We present the user a reference item (a \$2 bus ride that takes 40 minutes), and a sliding bar representing the number of dollars (x) for the 10-minute taxi cost. As the user adjusts the slider, the value of x starts with 0 and gradually increases on a continuous scale. The user is instructed to place the slider at a point where they equally prefer a \$2 bus ride and a taxi ride of x dollars.[2] The PAQ thus combines ordinal and cardinal elicitation in an intuitive fashion: We obtain ordinal information by asking the user to make cognitive judgments in a relative sense by comparing items, and cardinal information can be extracted from the location of the slider. The ordinal reasoning endows the query with accuracy and efficiency, while the cardinal output enables a more expressive response. Moreover, this cardinal output mitigates miscalibration, because instead of asking the user to rate on a subjective and ambiguous

[2] The ordinal component is crucial in our proposed perceptual adjustment query— we provide a reference item and instruct people to make a relational judgment of the target item compared to the reference item. Hence, the perceptual adjustment query is distinct from sliding survey questions that elicit purely cardinal responses.

notion (i.e., preference), we provide the user a reference object (i.e., the $2 bus ride) to anchor their rating scale.

This combination of high per-response information and low cognitive burden makes the deployment of PAQs appealing in a variety of problem settings. For example:

- Learning human preferences. As illustrated in the taxi and bus example in Figure 3.1, one can ask users to pinpoint the cost at which a taxi ride is equally preferred to the bus ride. In a more complex setting, such as housing preferences, moving the slider can change multiple attributes, such as price, square footage, maintenance fees, proximity to employment, etc. User responses to PAQs yield information-dense statements about how features jointly impact human preferences.

- Learning a model for color perception. Imagine a user with red-green color blindness, the extent of which we wish to learn. We can present the user with an image of a red square and a sequence of colors that slowly transitions from red to green, and ask them to drag the slider until they perceive a difference in colors. In such a setting, PAQs present users with context (the full sequence of colors and the reference color) to help them indicate their color sensitivity: At what point can you start distinguishing the two colors?

- Studying generative models. Imagine we wish to characterize how the semantic characteristics of synthesized items (e.g., images) change along different directions of a given generative model. By traversing a continuous path in the model's latent space and generating a corresponding item for each point, PAQs present users with a sequence of items. Using an item at the beginning of the sequence as the reference item, we ask users to mark the first item along the sequence that is semantically different in a meaningful way. For example, to characterize how different directions in the latent space impact breed for a model trained to synthesize images of dogs, we ask users to mark the first image where the breed clearly changes.

Beyond combining the strengths of cardinal and ordinal queries, PAQs have additional advantages that are well illustrated with the example in Figure 3.1 (bottom). First, PAQs provide users with the *context* of a specific (continuous) dimension along which items vary. For example, consider a pairwise comparison between the reference item and the “yellow apple ” selected in Figure 3.1. They have similar shapes, but different colors. If these two items are shown to the user in isolation, the user lacks context to judge whether they should be considered similar or dissimilar. In contrast, the full spectrum provided in PAQs tells the user that the similarity judgment is apples vs. pears. The access to such context improves self-consistency in user responses [28]. Second, PAQs provide “hard examples” by design and thus enable effective learning. Consider Figure 3.1 (bottom): Items on the left of the spectrum are apples (clearly similar to the reference), and items on the right are pears (clearly dissimilar to the reference), and only a small subset of items in the middle appear ambiguous. PAQs collect information precisely about “confusing” items in this ambiguous region. On the other hand, if ordinal queries are constructed by selecting uniformly at random from the items shown, an item in the ambiguous region will rarely be presented to the user.

In this chapter, we apply the PAQ scheme in the framework of metric learning for human perception. In this problem, items are represented by points in a (possibly high-dimensional) space, and the goal is to learn a distance metric such that a smaller distance between a pair of items means that they are semantically and perceptually closer, and vice versa. Figure 3.1 (bottom) presents a PAQ for collecting similarity data for metric learning, where the user is instructed to place the slider at the precise point where the object appears to transition from being similar to dissimilar.

To construct a sequence of images as shown in Figure 3.1 (bottom), one can traverse a path in the latent space of a generative model — given a latent feature vector, the generative model synthesizes a corresponding image. In other settings, such as the taxi example in Figure 3.1 (top) or the housing preference task mentioned above, a sequence of items can be

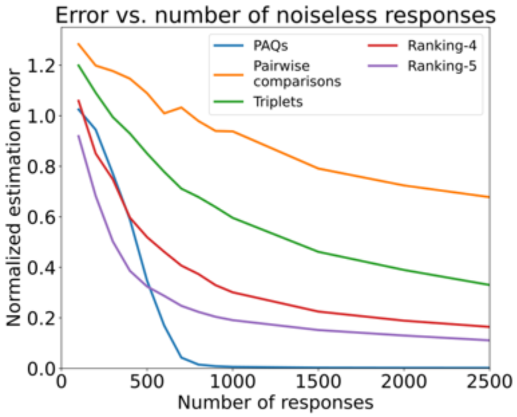


Figure 3.2: Simulation comparing performance of noiseless responses to PAQs and various ordinal queries when applied to low-rank metric learning. Ranking-k denotes that k items are ranked in terms of similarity to a reference item. For each query type, we plot the mean and standard error of the mean (shaded regions, not visible) of the normalized estimation error $\|\mathbf{\Sigma}^\star - \widehat{\mathbf{\Sigma}}\|_F / \|\mathbf{\Sigma}^\star\|_F$ over 10 independent trials.

formed by gradually changing the value of interpretable features, such as price and square footage.

Do PAQs improve upon ordinal queries? A simulation vignette. Consider the problem of Mahalanobis metric learning, which forms the focus of this chapter. In this setting, items are represented as points in the vector space $\mathbb{R}^d$, which is in turn endowed with a Mahalanobis metric parametrized by a symmetric positive semidefinite matrix $\mathbf{\Sigma}^\star \in \mathbb{R}^{d\times d}$. The (dis-)similarity of two items is determined by their distance under the metric: The larger the (squared) distance $\|\boldsymbol{x} - \boldsymbol{x}'\|^2_{\mathbf{\Sigma}^\star} = (\boldsymbol{x} - \boldsymbol{x}')^\top \mathbf{\Sigma}^\star (\boldsymbol{x} - \boldsymbol{x}')$ between two items $\boldsymbol{x}$ and $\boldsymbol{x}'$ is, the more dissimilar the items are. We are particularly interested in the setting in which $\mathbf{\Sigma}^\star$ is *low-rank*, which covers several important settings. For example, a user may make preference judgements using a small number of interpretable features [1, 100]. For another example, it has been shown that a small number of linear directions capture a vast majority of semantic changes in the latent space of a popular generative model, StyleGAN2 [101].

Established approaches in metric learning use ordinal queries, such as pairwise comparisons ("Are items $\boldsymbol{x}$ and $\boldsymbol{x}'$ similar?") [102, 103, 104, 105], triplet comparisons [25] ("Which of the two items $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is closer to reference item $\boldsymbol{x}_0$?"), and ranking queries ("Given a reference item $\boldsymbol{x}_0$, rank the set of items $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ in terms of similarity to $\boldsymbol{x}_0$") [28]. In Figure 3.2, we simulate the performance of such queries in a toy metric learning setup against the performance of PAQs.

In particular, we choose a random low-rank matrix $\boldsymbol{\Sigma}^\star$ in dimension 50 with rank 10 (see Section B.1 for our precise construction, which resembles the setup of [106]) and use the models of [25, 28] to produce standard pairwise, triplet, and ranking-k queries. We also use state-of-the-art algorithms to estimate the low-rank metric from these types of queries [25, 28]. In addition to these ordinal queries, we simulate PAQ responses under the model presented in Section 3.3 and use our algorithm (see Section 3.4) to generate a metric estimate. To simplify the example, all queries responses are generated in a *noiseless* fashion—for example, the triplet comparison always returns the closer item to the reference.

We present our results in Figure 3.2, which illustrates a significant gap in information richness between PAQs and a variety of ordinal queries. The number of PAQ responses needed to attain a reasonable normalized error levels is dramatically lower than those of typical ordinal queries. For example, to achieve a normalized error of 0.2, one needs at minimum 1,000 of any of the ordinal queries but only approximately 600 PAQ responses. Overall, Figure 3.2 quantitatively illustrates that PAQs can greatly improve upon the performance of existing ordinal queries on metric learning. The rest of the chapter explores this opportunity: It aims to make the deployment of PAQs theoretically grounded by designing provable methodology for learning a low-rank metric from PAQ responses.

Our contributions and organization. In addition to introducing the *perceptual adjustment query* (PAQ), we demonstrate its applicability to metric learning under a Mahalanobis metric. We first present a mathematical formulation of this estimation problem in Section 3.3. We then show that the sliding bar response can be viewed as an *inverted measurement* of the metric matrix that we want to estimate, which allows us to restate our problem as that of estimating a low-rank matrix from a specific type of trace measurement (Section 3.4).

However, our PAQ formulation differs from classical matrix estimation due to two technical challenges: (a) the sensing matrices and noise are correlated, and (b) the sensing matrices are heavy-tailed. As a result, standard matrix estimation algorithms give rise to *biased estimators*. We propose a query procedure and an estimator that overcome these two challenges, and we prove statistical error bounds on the estimation error (Section 3.5). The unconventional nature of the sensing model and estimator causes unexpected behaviors in our error bounds; in Section 3.6, we present simulations verifying that these behaviors also appear in practice.

Notation. For two real numbers a and b, let $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$. Given a vector $\boldsymbol{x} \in \mathbb{R}^d$, denote $\|\boldsymbol{x}\|_1$ and $\|\boldsymbol{x}\|_2$ as the ℓ_1 and ℓ_2 norm, respectively. Denote $\mathcal{S}^{d-1} := \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$ to be the set of d-dimensional vectors with unit ℓ_2 norm. Given a matrix $\boldsymbol{A} \in \mathbb{R}^{d_1 \times d_2}$, denote $\|\boldsymbol{A}\|_F$, $\|\boldsymbol{A}\|_*$, and $\|\boldsymbol{A}\|_{\mathrm{op}}$ as its Frobenius norm, nuclear norm, and operator norm, respectively. We denote $\mathbb{S}^{d \times d} = \{\boldsymbol{A} \in \mathbb{R}^{d \times d} : \boldsymbol{A} = \boldsymbol{A}^\top\}$ to be the set of symmetric $d \times d$ matrices. Denote $\boldsymbol{A} \succeq \boldsymbol{0}$ to mean that $\boldsymbol{A}$ is symmetric positive semidefinite. For $\boldsymbol{A} \succeq \boldsymbol{0}$, define the (pseudo-) norm $\|\boldsymbol{x}\|_{\boldsymbol{A}} = \sqrt{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}$. For matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d_1 \times d_2}$, denote by $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \operatorname{tr}\left(\boldsymbol{A}^\top \boldsymbol{B}\right)$ the Frobenius inner product. For two sequences indexed by x, we use the notation $f(x) \lesssim g(x)$ to mean that there exists some absolute positive constant $c > 0$, such that $f(x) \leq c \cdot g(x)$ for all x. We use the notation $f(x) \gtrsim g(x)$ when $g(x) \lesssim f(x)$.

3.2 Related work

We now discuss related work in metric learning, along with the statistical techniques we use for our algorithm and analysis.

Metric learning. In metric learning [107], prior work considers using paired comparisons (of the form "are these two items similar or dissimilar?") [102, 103, 104, 105] and triplet comparisons (of the form "which of the two items $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is more similar to the reference item $\boldsymbol{x}_0$?") [25]. The metric learning from triplets problem is generalized by [1] to consider an *unknown* reference point (referred to as an "ideal point") that captures different individual preferences. Sample complexity guarantees for simultaneous estimation of a metric and individual ideal points are established in [100]. Tuple queries [28] extend triplets to ranking more than two items with respect to a reference item. The PAQ can be viewed as extending this set of items to a continuous spectrum, which is natural when one uses a generative model such as a GAN [43, 48]. However, the goal of tuple queries is to rank the items, whereas in PAQ the ranking is provided by the feature space and we ask people to identify a transition point (similar vs. dissimilar) in this ranking.

Statistical techniques. In our theoretical results, we apply techniques from the high-dimensional statistics literature. Our theoretical formulation (presented in Section 3.4) resembles the problem of low-rank matrix estimation from trace measurements (e.g., [108, 10, 109, 110, 111, 112]; see [113] for a more complete overview), and in particular, when the sensing matrix is of rank one and random [114, 106, 115, 116, 117]. However, as discussed in Section 3.4, our model results in two important departures from prior literature. In our case, the sensing matrices are both heavy-tailed and correlated with the measurement noise, and the latter issue results in estimation bias for standard matrix estimation procedures. In addition, our heavy-tailed matrices violate the assumptions of much prior work that relies on sub-Gaussian or sub-exponential assumptions on the sensing matrices. Prior work has attempted to address the challenge of heavy tails with methods such as robust loss functions [118, 119] or the "median-of-means" approach [120, 121, 122], which partitions the data, constructs an estimator for each partition, and then forms one estimator based on some robustness criteria. We draw particular inspiration from [123], which applies truncation to control heavy-tailed behavior in a number of problem settings. However, in the low-rank matrix estimation setting, the paper [123] only analyzes the case of heavy-tailed noise under a sub-Gaussian design, meaning that their methodology and results are not applicable to our

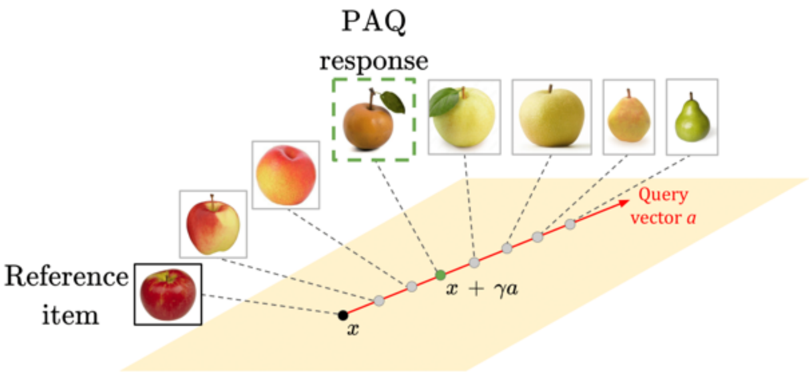


Figure 3.3: The perceptual adjustment query. Given a reference item $\boldsymbol{x}$ and a query vector $\boldsymbol{a}$, a continuous path of items is formed $\{\boldsymbol{x} + \gamma\boldsymbol{a} : \gamma \in [0, \infty)\}$. Then, a user is asked to pick the first item along this path that is dissimilar to the reference item, denoted by $\boldsymbol{x} + \gamma\boldsymbol{a}$.

problem setting.

3.3 Formal model

In this section, we present our model for the perceptual adjustment query (PAQ) in the context of its application to metric learning.

3.3.1 Mahalanobis metric learning

We consider a d-dimensional feature space where each item is represented by a point in $\mathbb{R}^d$. The distance metric model for human similarity perception posits that there is a metric on $\mathbb{R}^d$ that measures how dissimilar items are perceived to be. A recent line of work [1, 100] has modeled the distance metric as a Mahalanobis metric. If $\boldsymbol{\Sigma}^\star \in \mathbb{R}^{d\times d}$ is a symmetric positive semidefinite (PSD) matrix, the squared Mahalanobis distance with respect to $\boldsymbol{\Sigma}^\star$ between items $\boldsymbol{x}$ and $\boldsymbol{x}' \in \mathbb{R}^d$ is $\|\boldsymbol{x} - \boldsymbol{x}'\|^2_{\boldsymbol{\Sigma}^\star} := (\boldsymbol{x} - \boldsymbol{x}')^\top \boldsymbol{\Sigma}^\star (\boldsymbol{x} - \boldsymbol{x}')$. The distance represents the extent of dissimilarity between items $\boldsymbol{x}$ and $\boldsymbol{x}'$: If we further have a perceptual boundary value $y > 0$, this model posits that items $\boldsymbol{x}, \boldsymbol{x}'$ are perceived as similar if $\|\boldsymbol{x} - \boldsymbol{x}'\|^2_{\boldsymbol{\Sigma}^\star} < y$ and dissimilar if $\|\boldsymbol{x} - \boldsymbol{x}'\|^2_{\boldsymbol{\Sigma}^\star} \geq y$. We adopt a high-dimensional framework and, following existing work [25, 100], assume that the matrix $\boldsymbol{\Sigma}^\star$ is low-rank.

Note that if the goal is to predict whether two items are similar or dissimilar via computing the relation $\|\boldsymbol{x}-\boldsymbol{x}'\|_{\boldsymbol{\Sigma}^\star}^2 \gtrless y$, then this problem is scale-invariant, in the sense that two items are predicted as similar (or dissimilar) according to $(\boldsymbol{\Sigma}^\star, y)$, if and only if they are predicted as similar (or dissimilar) according to $(c_{\mathsf{scale}}\boldsymbol{\Sigma}^\star, c_{\mathsf{scale}}y)$ for any scaling factor $c_{\mathsf{scale}} > 0$. We are thus interested in finding the equivalence class of solutions $\{(c_{\mathsf{scale}}\boldsymbol{\Sigma}^\star, c_{\mathsf{scale}}y) : c_{\mathsf{scale}} > 0\}$.

Remark 1 (Choice of y)**.** *Since the goal is to learn (dis-)similarity between items, one can set the boundary value to be any positive scalar y, and estimate the matrix $\boldsymbol{\Sigma}^\star$ corresponding to this value of y. Indeed, our theoretical results proving error bounds on $\|\widehat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}^\star\|_F$ exhibit a natural scale-equivariant property (see Section 3.5, Scale Equivariance).*

3.3.2 The perceptual adjustment query (PAQ)

We assume that every point in our feature space $\mathbb{R}^d$ corresponds to some item. Recall from Figure 3.1 that a PAQ collects similarity data between a pair of items, where a reference item is fixed, and a spectrum of target items is generated from a one-dimensional path in the feature space. Denote the reference item by $\boldsymbol{x} \in \mathbb{R}^d$. The target items can be generated by any path in $\mathbb{R}^d$, but for simplicity, we consider straight lines. For any vector $\boldsymbol{a} \in \mathbb{R}^d$, we construct the line $\{\boldsymbol{x} + \gamma\boldsymbol{a} : \gamma \in [0,\infty)\}$. We call this vector $\boldsymbol{a}$ the *query vector*. As shown in Figure 3.3, the user moves the slider from left to right, and the value of γ increases proportionally to the distance traversed by the slider. Note that the value γ is *dimensionless*.

The user is instructed to stop the slider at the transition point where the target item transitions from being similar to dissimilar with the reference item. According to our model, this transition point occurs when the $\boldsymbol{\Sigma}^\star$-Mahalanobis distance between the target item and the reference item is y. The (noiseless) transition point, denoted by $\gamma_\star$, satisfies the equation

$$y = \|\boldsymbol{x} - (\boldsymbol{x} + \gamma_\star\boldsymbol{a})\|_{\boldsymbol{\Sigma}^\star}^2 = \gamma_\star^2 \boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}. \tag{3.1}$$

Note that the ideal PAQ response $\gamma_\star$ does not depend on the specific reference item $\boldsymbol{x}$ but rather only on the query direction $\boldsymbol{a}$ and the (unknown) metric matrix $\boldsymbol{\Sigma}^{\star}$. When querying users with PAQs, the practitioner has control over how the query vectors $\boldsymbol{a}$ are selected. We discuss how to select $\boldsymbol{a}$ in Section 3.4.2.

3.3.3 Noise model

We model the noise in human responses as follows: In the PAQ response (Equation 3.1), we replace the boundary value y by $y+\eta$, where $\eta \in \mathbb{R}$ represents noise. Thus the user provides a noisy response γ whose value satisfies $\gamma^2 \boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a} = y + \eta$. Substituting in (Equation 3.1), we have

$$\gamma^2 = \gamma_\star^2 + \frac{\eta}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}. \tag{3.2}$$

If $\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}$ is large, then in the user interface Figure 3.1 (bottom), the semantic meaning of the item changes rapidly as the user moves the slider along the direction $\boldsymbol{a}$, and the slider stops at a position that is close to the true transition point. On the other hand, if $\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}$ is small, then the image changes slowly as the user moves the slider. It is hard to distinguish where exactly the transition occurs, so the slider ends up in a larger interval around the transition point. Recall that the scaling γ is proportional to the distance traversed by the slider. This model (Equation 3.2) thus captures such variation in the noise level, where the noise term $\frac{\eta}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}$ is small when $\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}$ is large, and vice versa.

3.4 Methodology

In this section, we formally present the statistical estimation problem for metric learning from noisy PAQ data, and we develop our algorithm for estimating the true metric matrix $\boldsymbol{\Sigma}^\star$.

3.4.1 Statistical estimation

Assume we collect N PAQ responses, using N query vectors $\{\boldsymbol{a}_i\}_{i=1}^N$ that we select[3]. Denote the noise associated with these queries by random variables $\eta_1, \ldots, \eta_N \in \mathbb{R}$. We obtain PAQ responses, denoted by $\gamma_1, \ldots, \gamma_N$, that satisfy

$$\gamma_i^2 \boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i = y + \eta_i, \;\; i = 1, \ldots, N. \tag{3.3}$$

We assume the noise variable η is independent[4]of the query $\boldsymbol{a}$, has zero mean and variance ν_η^2, and is bounded, with $-y \leq \eta \leq \eta^\uparrow$ for some constant $\eta^\uparrow \geq 0$. Note that we must have $\eta + y \geq 0$ since $\gamma^2 \geq 0$; in addition, we place an upper bound $\eta^\uparrow$ on the noise.

Given the query directions $\{\boldsymbol{a}_i\}_{i=1}^N$ and the PAQ responses $\{\gamma_i\}_{i=1}^N$, we want to estimate the matrix $\boldsymbol{\Sigma}^\star$. We first rewrite our measurement model as follows: Recall that the matrix inner product is denoted by $\langle \boldsymbol{A}, \boldsymbol{B} \rangle := \operatorname{tr}\left(\boldsymbol{A}^\top \boldsymbol{B}\right)$ for any two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of compatible dimension. Then from (Equation 3.3), we write

$$\gamma^2 = \frac{y + \eta}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}. \tag{3.4}$$

Plugging (Equation 3.4) once more into (Equation 3.3), we have

$$y + \eta = \langle \boldsymbol{A}^{\mathsf{inv}}, \boldsymbol{\Sigma}^\star \rangle,$$

where

$$\boldsymbol{A}^{\mathsf{inv}} := \gamma^2 \boldsymbol{a}\boldsymbol{a}^\top = \frac{y + \eta}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \boldsymbol{a}\boldsymbol{a}^\top. \tag{3.5}$$

Hence, our problem resembles trace regression, and, in particular, low-rank matrix estimation

[3] In the sequel, we use the terms "responses"/"measurements" interchangeably for γ, and the terms "query vector"/"sensing vector" interchangeably for $\boldsymbol{a}$.

[4] This could be relaxed by placing conditions on the *conditional* distributions of η given $\boldsymbol{a}$ (and even the reference point $\boldsymbol{x}$), but we omit this for simplicity.

from rank-one measurements (because the matrix $\boldsymbol{A}^{\mathsf{inv}}$ has rank 1) [114, 106, 115, 116]. We call $\boldsymbol{A}^{\mathsf{inv}}$ the sensing matrix, and $\boldsymbol{a}$ the sensing vector. Classical trace regression assumes that we make (noisy) observations of the form $y = \langle \boldsymbol{A}, \boldsymbol{\Sigma}^\star \rangle + \epsilon$ where $\boldsymbol{A}$ is fixed before we make the measurement; in our problem, the sensing matrix $\boldsymbol{A}^{\mathsf{inv}}$ depends on our observed response γ and associated sensing vector $\boldsymbol{a}$. Hence, the process of obtaining a PAQ response can be viewed as an *inversion* of the standard trace measurement process. The inverse nature of our problem makes estimator design more challenging, as we discuss in the following section.

3.4.2 Algorithm

As our first attempt at a procedure to estimate $\boldsymbol{\Sigma}^\star$, we follow the literature [10, 116] and consider randomly sampling i.i.d. vectors $\boldsymbol{a}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$. We then use standard least-squares estimation of $\boldsymbol{\Sigma}^\star$. Since we expect $\boldsymbol{\Sigma}^\star$ to be low-rank, we add nuclear-norm regularization to promote low rank. In particular, we solve the following program:

$$\min_{\boldsymbol{\Sigma} \succeq \boldsymbol{0}} \frac{1}{N} \sum_{i=1}^{N} \left(y - \langle \boldsymbol{A}_i^{\mathsf{inv}}, \boldsymbol{\Sigma} \rangle \right)^2 + \lambda_N \|\boldsymbol{\Sigma}\|_*, \tag{3.6}$$

where $\lambda_N > 0$ is a regularization parameter. This is a convex semidefinite program and can be solved with standard off-the-shelf solvers.

However, the inverted form of our measurement model creates two critical issues when naïvely using (Equation 3.6):

- **Bias of standard matrix estimators due to dependence.** Note that the sensing matrix (Equation 3.5) depends on the noise η. Quantitatively, we have $\mathbb{E}\left[\eta \boldsymbol{A}^{\mathsf{inv}}\right] \neq \boldsymbol{0}$ (see Section B.3.1). Standard trace regression analyses require that this quantity be zero, typically assuming (at least) that η is zero-mean conditioned on the sensing matrix $\boldsymbol{A}$. The failure of this to hold in our case introduces a bias that does not decrease with the sample size N.

- **Heavy-tailed sensing matrix.** The factor $\frac{1}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}$ in $\boldsymbol{A}^{\mathsf{inv}}$ (see (Equation 3.5)) makes $\boldsymbol{A}^{\mathsf{inv}}$ heavy-tailed in general. When $\boldsymbol{a}$ is Gaussian, the term $\frac{1}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}$ is an inverse weighted chi-square random variable, whose higher-order moments are infinite (and the number of finite moments depends on the rank of $\boldsymbol{\Sigma}^\star$). This makes error analysis more difficult, as standard analyses require the sensing matrix $\boldsymbol{A}$ to concentrate well (e.g., be sub-exponential).

To overcome these challenges, we make two key modifications to the procedure (Equation 3.6).

Step 1: Bias reduction via averaging. First, we want to mitigate the bias due to the dependence between the sensing matrix $\boldsymbol{A}^{\mathsf{inv}}$ and the noise η. The bias term $\mathbb{E}\left[\eta \boldsymbol{A}^{\mathsf{inv}}\right]$ scales proportionally to $\mathbb{E}\left[\eta(y+\eta)\right] = \mathbb{E}\left[\eta^2\right]$. Therefore, to reduce this bias in the least-squares estimator (Equation 3.6), we need to reduce the noise variance. We reduce the effective noise variance (and hence the bias) by *averaging* i.i.d. samples. Operationally, instead of obtaining N measurements from N distinct sensing vectors $\{\boldsymbol{a}_i\}_{i=1}^N$, we draw n sensing vectors $\{\boldsymbol{a}_i\}_{i=1}^n$, and collect m measurements, denoted by $\{\gamma_i^{(j)}\}_{j=1}^m$, corresponding to each sensing vector $\boldsymbol{a}_i$. We refer to n as the number of (distinct) sensing vectors. To keep the total number of measurements constant, we set $n = \frac{N}{m}$, where the value of m is specified later. For each sensing vector $\boldsymbol{a}_i$, we compute the empirical mean of the m measurements:

$$\bar{\gamma}_i^2 := \frac{1}{m}\sum_{j=1}^m (\gamma_i^{(j)})^2 = \frac{1}{m}\sum_{j=1}^m \frac{y+\eta_i^{(j)}}{\boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i} = \frac{y+\bar{\eta}_i}{\boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i}, \tag{3.7}$$

where we define the average noise by $\bar{\eta}_i := \frac{1}{m}\sum\limits_{j=1}^m \eta_i^{(j)}$. This averaging operation reduces the effective noise variance from $\operatorname{var}(\eta_i) = \nu_\eta^2$ to $\operatorname{var}(\bar{\eta}_i) = \frac{\nu_\eta^2}{m}$. If n is small, we may have large error due to an insufficient number of query vectors $\boldsymbol{a}_i$. On the other hand, a small m leads to a large bias. Therefore, we set the value of m carefully to balance these two effects. This is studied theoretically in Section 3.5 and demonstrated empirically in Section 3.6.

Step 2: Heavy tail mitigation via truncation. Next, we need to control the heavy-tailed behavior introduced by the $\frac{1}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}$ term in the sensing matrix $\boldsymbol{A}^{\mathsf{inv}}$. Note that the sample averaging procedure (Equation 3.7) does not mitigate this problem. We adopt the approach in [123] and truncate the observations. Specifically, we truncate the averaged measurements $\bar{\gamma}_i^2$ by τ:

$$\widetilde{\gamma}_i^2 := \bar{\gamma}_i^2 \wedge \tau = \frac{y + \bar{\eta}_i}{\boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i} \wedge \tau, \tag{3.8}$$

where $\tau > 0$ is a truncation threshold that we specify later. We then construct the truncated sensing matrices

$$\widetilde{\boldsymbol{A}}_i = \widetilde{\gamma}_i^2 \boldsymbol{a}_i \boldsymbol{a}_i^\top = \left(\frac{y + \bar{\eta}_i}{\boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i} \wedge \tau \right) \boldsymbol{a}_i \boldsymbol{a}_i^\top, \quad i = 1, \dots, n. \tag{3.9}$$

While truncation mitigates heavy-tailed behavior, it also introduces additional bias in our estimate. The truncation threshold τ therefore gives us another tradeoff, and in our analysis to follow, we carefully set the value of τ to balance the effects of heavy-tailedness and bias.

Final algorithm. Before presenting our final optimization program, we summarize our assumptions and sensing model below.

Assumption 1 (Zero-mean, bounded noise)**.** *The observed noise values η_i are i.i.d copies of the random variable η, which is independent of the random sensing vector $\boldsymbol{a}$. The random noise satisfies*

- *Zero-mean:* $\mathbb{E}[\eta] = 0$
- *Bounded: There exists a positive constant $\eta^\uparrow$ such that $-y \le \eta \le \eta^\uparrow$ with probability* 1.

We choose the sensing vector distribution to be the standard multivariate normal distribution and collect, average, and truncate N PAQ responses following Algorithm 1. This

Algorithm 1 Inverted measurement sensing, averaging, and truncation.

```
Input: number of total measurements $N$, averaging parameter $m$ (that divides $N$), truncation threshold $\tau$, measurement value $y$
Compute the number of sensing vectors $n = \frac{N}{m}$
for $i = 1$ to $n$ do
    Draw sensing vector $\boldsymbol{a}_i$ from standard multivariate normal distribution
    Obtain $m$ PAQ measurements $(\gamma_i^{(1)})^2, \ldots, (\gamma_i^{(m)})^2$ of the form
        $(\gamma_i^{(j)})^2 = \frac{y + \eta_i^{(j)}}{\boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i},$
    where $\eta_i^{(j)}$ is an i.i.d. copy of the random noise $\eta$ for all $i$ and $j$
end for
for $i = 1$ to $n$ do
    Bias elimination via averaging: compute averaged response
        $\bar{\gamma}_i^2 = \frac{1}{m} \sum_{j=1}^{m} (\gamma_i^{(j)})^2.$
    Heavy tail mitigation via truncation: compute truncated response
        $\widetilde{\gamma}_i^2 = \bar{\gamma}_i^2 \wedge \tau.$
end for
Output: truncated responses $\widetilde{\gamma}_1^2, \ldots \widetilde{\gamma}_n^2$
```

process yields n truncated responses $\widetilde{\gamma}_1^2, \ldots \widetilde{\gamma}_n^2$. We then use these truncated responses to form the averaged and truncated matrices $\{\widetilde{\boldsymbol{A}}_i\}_{i=1}^n$, which we substitute into the original least-squares problem (Equation 3.6). To estimate $\boldsymbol{\Sigma}^\star$, we solve

$$\widehat{\boldsymbol{\Sigma}} \in \operatorname*{arg\,min}_{\boldsymbol{\Sigma} \succeq \mathbf{0}} \frac{1}{n} \sum_{i=1}^{n} \Big(y - \langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{\Sigma} \rangle \Big)^2 + \lambda_n \|\boldsymbol{\Sigma}\|_*, \tag{3.10}$$

where, again, λ_n is a regularization parameter that we specify later.

Practical considerations. In the averaging step, we collect m measurements for each sensing vector $\boldsymbol{a}_i$. These measurements could be collected from m different users. Furthermore, recall from Section 3.3.2 that the measurements do not depend on the reference item $\boldsymbol{x}$. As a result, one may also collect multiple responses from the same user by presenting them the same query vector $\boldsymbol{a}_i$ but with different reference items $\boldsymbol{x}$. In addition, recall from Section 3.3.1 that user responses are scale-invariant. Practitioners are hence free to set the boundary y to be any positive value of their choice without loss of generality, and the noise variance ν_η^2 scales accordingly with y. The user interface does not depend on the value of y.

3.5 Theoretical results

We now present our main theoretical result, which is a finite-sample error bound for estimating a low-rank metric from inverted measurements with the nuclear norm regularized estimator (Equation 3.10). Our error bound is generally stated, and depends on the averaging parameter m and the truncation threshold τ.

Recall that ν_η^2 denotes the variance of η. We define the quantities $y^\uparrow := y + \eta^\uparrow$ and $\mu_y = y + \texttt{median}(\eta)$. We further denote by $\sigma_1 \geq \cdots \geq \sigma_r > 0$ the non-zero singular values of $\boldsymbol{\Sigma}^\star$.

Theorem 1. *Suppose $\boldsymbol{\Sigma}^\star$ is rank r, with $r > 8$. Assume that we choose the sensing vector distribution the be the standard multivariate normal distribution, that Assumption 1 holds on the noise, and that we collect, average, and truncate measurements following Algorithm 1. Further, assume that the truncation threshold τ satisfies $\tau \geq \frac{\mu_y}{\operatorname{tr}(\boldsymbol{\Sigma}^\star)}$. Then there are positive constants c, C, C_1, and C_2, such that if the regularization parameter and the number of sensing vectors satisfy*

$$\lambda_n \geq C_1 \left[y^\uparrow \left(\frac{y^\uparrow}{\sigma_r r} \sqrt{\frac{d}{n}} + \frac{d}{n}\tau + \left(\frac{y^\uparrow}{\sigma_r r} \right)^2 \frac{1}{\tau} \right) + \frac{1}{\sigma_r r} \frac{\nu_\eta^2}{m} \right] \quad \textit{and} \quad n \geq C_2 r d, \tag{3.11}$$

then any solution $\widehat{\boldsymbol{\Sigma}}$ to the optimization program (Equation 3.10) *satisfies*

$$\|\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_F \leq C \left(\frac{\operatorname{tr}(\boldsymbol{\Sigma}^\star)}{\mu_y} \right)^2 \sqrt{r} \lambda_n \tag{3.12}$$

with probability at least $1 - 4\exp(-d) - \exp(-cn)$.

The proof of Theorem 1 is presented in Section 3.7. The two sources of bias discussed in Section 3.4.2 appear in the expression (Equation 3.11) for the regularization parameter λ_n (and consequently in the error bound (Equation 3.12)). The term scaling as $1/\tau$ corresponds to the bias induced by truncation, and decreases as the truncation gets milder (i.e., as the threshold τ gets larger). The term scaling as ν_η^2/m corresponds to the bias arising from dependence between the noise and sensing matrix. As discussed in Section 3.4.2, in this model, m-averaging results in a bias that scales like $1/m$.

Given the dependence of the estimation error bound on the parameters m and τ, we carefully set these parameters to obtain a tight bound as a function of the number of *total measurements* $N = mn$. These choices for m and τ, along with the final estimation error, are presented below in Corollary 1.

Corollary 1. *Recall that* $N = mn$. *Assume that the conditions of Theorem 1 hold, and set the values of the constants* (c, C, C_1, C_2) *according to Theorem 1. Suppose that the number of total measurements satisfies*

$$N \geq \left\{ 2C_2^{3/2} \frac{\nu_\eta^2}{(y^\dagger)^2} r^{3/2} d \right\} \vee \left\{ C_2 r d \right\}. \tag{3.13}$$

Set the averaging parameter m *and truncation threshold* τ *to be*

$$m = \left\lceil \left(\frac{\nu_\eta^2}{(y^\dagger)^2} \right)^{2/3} \left(\frac{N}{d} \right)^{1/3} \right\rceil \quad \text{and} \quad \tau = \frac{y^\dagger}{\sigma_r r} \sqrt{\frac{N}{md}}, \tag{3.14}$$

and set λ_n *equal to its lower bound in* (Equation 3.11). *With probability at least* $1 - 4\exp(-d) - \exp(-cN/m)$, *we have:*

(a) If $\frac{\nu_\eta^2}{(y^\dagger)^2} > \sqrt{\frac{d}{N}}$, *then any solution* $\widehat{\boldsymbol{\Sigma}}$ *to the optimization program* (Equation 3.10)

satisfies

$$\|\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_F \leq C' \, \frac{\sigma_1^2}{\sigma_r} \frac{(y^\uparrow)^{4/3} (\nu_\eta^2)^{1/3}}{\mu_y^2} \, r^{3/2} \left(\frac{d}{N} \right)^{1/3} . \tag{3.15}$$

(b) If $\frac{\nu_\eta^2}{(y^\uparrow)^2} \leq \sqrt{\frac{d}{N}}$*, then any solution* $\widehat{\boldsymbol{\Sigma}}$ *to the optimization program* (Equation 3.10) *satisfies*

$$\|\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_F \leq C' \, \frac{\sigma_1^2}{\sigma_r} \left(\frac{y^\uparrow}{\mu_y} \right)^2 r^{3/2} \left(\frac{d}{N} \right)^{1/2} . \tag{3.16}$$

In both cases, $C' = 3C \cdot C_1$.

The proof of Corollary 1 is provided in Section 3.8. A few remarks are warranted about our error bounds (Equation 3.15) and (Equation 3.16).

Error rates and noise regimes. Under the standard trace measurement model, it is known that if the measurement matrices are i.i.d. according to some sub-Gaussian distribution and the number of measurements satisfies $N \gtrsim rd$, then nuclear norm regularized estimators achieve an error that scales like $\sqrt{\frac{rd}{N}}$(e.g., [10, 109]). Such a result is also known to be minimax optimal [109]. Allowing heavier-tailed assumptions on the sensing matrices, such as sub-exponential [115, 11] or bounded fourth moment [123], typically results in additional $\log d$ factors but does not impact the exponent $1/2$ in the error rate. However, a crucial assumption in these results is that $\mathbb{E}\left[\eta \boldsymbol{A}^{\mathsf{inv}}\right] = \mathbf{0}$, and thus there is no bias due to measurement noise. Our inverted measurement sensing matrix is not only heavy-tailed but also leads to bias (see Lemma 1 in Section B.3.1). Nevertheless, we are able to reduce the bias and trade it for variance, ensuring consistent estimation in all regimes.

In Corollary 1, there are two distinct cases for error rate which correspond to two different noise regimes induced by the quantity $\frac{\nu_\eta^2}{(y^\uparrow)^2}$, which captures the noise level in our measurements. In particular, the two cases in Corollary 1 correspond to two regimes with distinct bias behavior:

(a) High-noise regime: In this setting, the bias due to measurement noise is non-negligible. As a result, we employ averaging with large m, which results in the rate scaling as $(d/N)^{1/3}$.

(b) Low-noise regime: In this setting, the measurement noise bias is dominated by the variance, and thus has negligible impact on the estimation error. As a result, we are able to achieve a rate of order $(d/N)^{1/2}$, which is consistent with established results for low-rank matrix estimation.

Sample complexity. Since the degrees of freedom in a rank-r matrix of size $d \times d$ is of order rd, one expects that the minimum number of measurements to identify a rank-r matrix is of order rd. This is reflected in Theorem 1, which assumes that the number of *distinct* sensing vectors $\{\boldsymbol{a}_i\}$ satisfies $n \gtrsim rd$. In the high-noise regime, from (Equation 3.14) in Corollary 1, we have that m scales like $(N/d)^{1/3}$. Thus, the total number of measurements is $N = mn \gtrsim (N/d)^{1/3} \cdot rd \gtrsim N^{1/3}d^{2/3}r$, and hence $N \gtrsim r^{3/2}d$. Given that the rank is assumed to be relatively small compared to the dimension, the extra factor of $\sqrt{r}$ is a relatively small price to pay to obtain consistent estimation. In the low-noise regime, it can be verified that $m = 1$ in (Equation 3.14) due to the low-noise condition $\frac{\nu_\eta^2}{(y^\dagger)^2} \leq \sqrt{\frac{d}{N}}$. No averaging is needed, and we only require $N = n \gtrsim rd$.

Dependence on rank. When compared to standard results, Corollary 1 differs in its dependence on rank. First, the matrix $\boldsymbol{\Sigma}^\star$ is assumed to have rank $r > 8$. This prevents the term $\frac{1}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}$ from making the sensing matrices so heavy-tailed that even truncation does not help. We empirically show that the assumption of $r > 8$ is necessary in Section 3.6. Second, there is an additional factor of r in our rate for both noise regimes. To interpret this, note that if $\boldsymbol{\Sigma}^\star$ has non-zero singular values in a fixed range, then $\mathbb{E}\left[\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}\right] = \operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right) \approx r$. Since the "magnitude" of the sensing matrix $\boldsymbol{A}^{\mathsf{inv}}$ is inversely proportional to $\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}$, increasing

r decreases the magnitude of $\boldsymbol{A}^{\mathsf{inv}}$ and thus also (for a fixed noise level) the signal-to-noise ratio.

Scale equivariance As discussed in Section 3.3.1, the metric learning from PAQs problem aims to find an equivalence class $\{(c_{\mathsf{scale}}\boldsymbol{\Sigma}, c_{\mathsf{scale}}y) : c_{\mathsf{scale}} > 0\}$, and the ground-truth $\boldsymbol{\Sigma}^\star$ is defined with respect to a particular choice of y. Accordingly, our error bounds are scale-equivariant: If we instead replaced y with $c_{\mathsf{scale}}y$, the bounds (Equation 3.15) and (Equation 3.16) would scale linearly in c_{scale}. This fact is precisely verified in Section B.2 and relies on the fact that the noise also scales appropriately in c_{scale}. As alluded to in Remark 1, practitioners may simply set y to be *any* positive number to estimate a metric that reflects item (dis-)similarity.

3.6 Numerical simulations

In this section, we provide numerical simulations investigating the effects of the various problem and estimation parameters. For all results, we report the normalized estimation error $\|\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_F / \|\boldsymbol{\Sigma}^\star\|_F$ averaged over 20 trials. Shaded areas (sometimes not visible) represent standard error of the mean. For all experiments, we follow [25] and generate the ground-truth metric matrix as $\boldsymbol{\Sigma}^\star = \frac{d}{\sqrt{r}}\boldsymbol{U}\boldsymbol{U}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{d\times r}$ is a randomly generated matrix with orthonormal columns. The noise η is sampled from a uniform distribution on $[-\eta^\uparrow, \eta^\uparrow]$ (where $\eta^\uparrow \leq y$). We set the regularization parameter, truncation threshold, and averaging parameter in a manner consistent with our theoretical results (see (Equation 3.11) and (Equation 3.14)), cross-validating to choose the constant factors. We solve the optimization problem using `cvxpy` [124, 125].

Effects of dimension and rank. Our first set of experiments characterizes the effects of dimension d and matrix rank r. For all experiments, unless we are sweeping a specific parameter, we set $y = 200$, $d = 50$, $r = 15$, and $\eta^\uparrow = 10$. Figure 3.4a shows the performance for varying values of d plotted against the normalized sample size N/d. For all dimensions

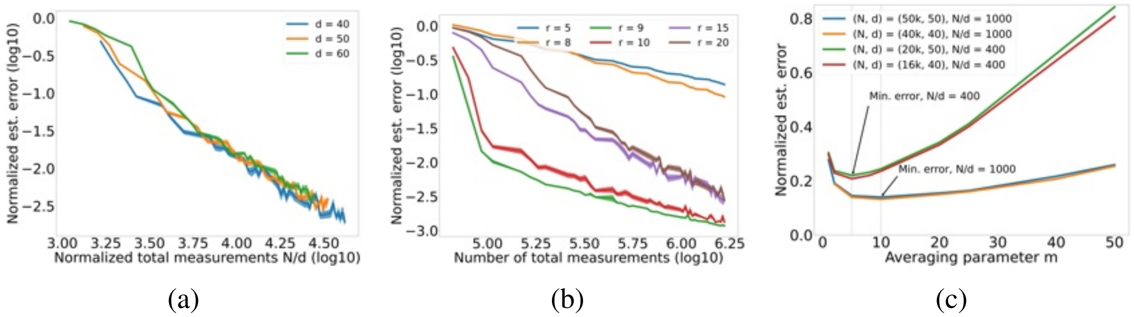


Figure 3.4: Simulations quantifying the effect of dimension d, rank r, and averaging parameter m on estimation error. Shaded areas correspond to standard error of the mean but sometimes not visible.

d, the error decays to zero as the total number of measurements N increases. Furthermore, the error curves are well-aligned when the sample size is normalized by d with fixed r, empirically aligning with Corollary 1. Figure 3.4b shows the performance for varying values of rank r. Recall that for our theoretical results we assume $r > 8$ to ensure that the quadratic term $\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}$ in the denominator of our sensing matrices does not lead to excessively heavy-tailed behavior. When $r > 8$, the number of measurements required for the same estimation error increases as the rank increases. A clear phase transition occurs at $r = 8$. The error still decreases with N for $r \leq 8$, but at a markedly slower rate than when $r > 8$. This empirically demonstrates that when $r \leq 8$, the sensing matrix tails are too heavy to be mitigated by truncation.

Effect of averaging parameter m. (Equation 3.14) suggests that the averaging parameter m should scale proportionally to $(N/d)^{1/3}$. To test this, we set $y = 200$, $d = 50$, $r = 9$, and $\eta^\uparrow = 200$. We vary values of m for different choices of the (N, d) pair, as shown in Figure 3.4c. The empirically optimal choice of m is observed to be the same when N/d is fixed, regardless of the particular choices of N or d (the green and red curves overlap, and the blue and orange curves overlap). Moreover, the optimal m is smaller when $N/d = 400$ compared to when $N/d = 1000$.

3.7 Proof of Theorem 1

Recall that we assume we collect N measurements under the inverted measurements sensing model presented in Algorithm 1 with standard Gaussian sensing vectors and bounded noise, mean-zero noise (Assumption 1).

We first introduce a restricted strong convexity (RSC) condition that our proof relies on. Since the matrix $\mathbf{\Sigma}^\star$ is assumed to be symmetric positive semidefinite matrices and of rank r, we follow [10] and consider a restricted set on which we analyze the behavior of the sensing matrices $\widetilde{\boldsymbol{A}}_i$. We call this set the "error set", defined by:

$$\mathcal{E} = \left\{ \boldsymbol{U} \in \mathbb{S}^{d\times d} : \|\boldsymbol{U}\|_* \leq 4\sqrt{2r}\|\boldsymbol{U}\|_F \right\}, \tag{3.17}$$

where recall that $\mathbb{S}^{d\times d}$ denotes the set of symmetric $d \times d$ matrices. We say that our shrunken sensing matrices $\{\widetilde{\boldsymbol{A}}_i\}_{i=1}^n$ satisfy a restricted strong convexity (RSC) condition over the error set $\mathcal{E}$, if there exists some positive constant $\kappa > 0$ such that

$$\frac{1}{n}\sum_{i=1}^{n} \langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{U}\rangle^2 \geq \kappa\|\boldsymbol{U}\|_F^2 \qquad \text{for all } \boldsymbol{U} \in \mathcal{E}. \tag{3.18}$$

The following proposition shows that the estimation error, when the sensing matrices satisfy the RSC condition and the regularization parameter is sufficiently large.

Proposition 2 ([123, Theorem 1] with $q = 0$)**.** *Suppose that $\mathbf{\Sigma}^\star$ has rank r and the shrunken sensing matrices satisfy the restricted strong convexity condition* (Equation 3.18) *with positive constant $\kappa > 0$. Then if the regularization parameter satisfies*

$$\lambda_n \geq 2\left\| \frac{1}{n}\sum_{i=1}^{n} y\widetilde{\boldsymbol{A}}_i - \frac{1}{n}\sum_{i=1}^{n}\langle \widetilde{\boldsymbol{A}}_i, \mathbf{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}_i \right\|_{\mathrm{op}}, \tag{3.19}$$

any optimal solution $\widehat{\mathbf{\Sigma}}$ *of the optimization program* (Equation 3.10) *satisfies*

$$\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}^{\star}\|_F \le \frac{32\sqrt{r}\lambda_n}{\kappa}.$$

This theorem is a special case of Theorem 1 in [123], which is in turn adapted from Theorem 1 in [10] (see [10] or [123] for the proof). Proposition 2 is a deterministic and nonasymptotic result and provides a roadmap for proving our desired upper bound. First, we show that the operator norm (Equation 3.19) can be upper bounded with high probability, allowing us to set the regularization parameter λ_n accordingly. Second, we show that the RSC condition (Equation 3.18) is satisfied with high probability. We begin by bounding the operator norm (Equation 3.19) in the following proposition.

Proposition 3. *Let* $y^{\uparrow} = y + \eta^{\uparrow}$*. Suppose that* $\mathbf{\Sigma}^{\star}$ *has rank* r*, with* $r > 8$*. Then there exists a positive absolute constant* C_1 *such that*

$$\left\| \frac{1}{n}\sum_{i=1}^{n} y\widetilde{\boldsymbol{A}}_i - \frac{1}{n}\sum_{i=1}^{n}\langle \widetilde{\boldsymbol{A}}_i, \mathbf{\Sigma}^{\star}\rangle \widetilde{\boldsymbol{A}}_i \right\|_{\mathrm{op}} \le C_1 \left[y^{\uparrow}\left(\frac{y^{\uparrow}}{\sigma_r r}\sqrt{\frac{d}{n}} + \frac{d}{n}\tau + \left(\frac{y^{\uparrow}}{\sigma_r r}\right)^2 \frac{1}{\tau} \right) + \frac{1}{\sigma_r r}\frac{\nu_\eta^2}{m} \right] \tag{3.20}$$

with probability at least $1 - 4\exp(-d)$.

The proof of Proposition 3 is provided in Section B.4. Next, we show that the RSC condition (Equation 3.18) is satisfied with high probability, as is done in the following proposition.

Proposition 4. *Let* μ_y *be the median of* $y + \bar{\eta}$*. Suppose that the truncation threshold* τ *satisfies* $\tau \ge \frac{\mu_y}{\mathrm{tr}(\mathbf{\Sigma}^{\star})}$*. Then there exist positive absolute constants* $\kappa_{\mathcal{L}}$*,* c*, and* C *such that if the number of sensing vectors satisfy*

$$n \ge Crd$$

then we have

$$\frac{1}{n}\sum_{i=1}^{n}\langle\widetilde{\boldsymbol{A}}_i,\boldsymbol{U}\rangle^2 \geq \kappa_{\mathcal{L}}\left(\frac{\mu_y}{\operatorname{tr}\left(\boldsymbol{\Sigma}^{\star}\right)}\right)^2\|\boldsymbol{U}\|_F^2 \tag{3.21}$$

simultaneously for all matrices $\boldsymbol{U}\in\mathcal{E}$ with probability at least $1-\exp(-cn)$, where $\mathcal{E}$ is the error set defined in (Equation 3.17).

The proof of Proposition 4 is provided in Section B.5. We now utilize the results of Proposition 2, Proposition 3 and Proposition 4 to derive our final error bound. By Proposition 3, the operator norm (Equation 3.19) can be upper bounded with high probability. We set the regularization parameter λ_n to satisfy

$$\lambda_n \geq C_1\left[y^{\uparrow}\left(\frac{y^{\uparrow}}{\sigma_r r}\sqrt{\frac{d}{n}}+\frac{d}{n}\tau+\left(\frac{y^{\uparrow}}{\sigma_r r}\right)^2\frac{1}{\tau}\right)+\frac{1}{\sigma_r r}\frac{\nu_\eta^2}{m}\right],$$

where C_1 is the constant in Proposition 3. Furthermore, by Proposition 4, we have that there exist universal constant $C_2>0$ such that if the number of sensing vectors satisfies $n\geq C_2 rd$, the RSC condition (Equation 3.18) holds for constant $\kappa=\kappa_{\mathcal{L}}\left(\frac{\mu_y}{\operatorname{tr}(\boldsymbol{\Sigma}^{\star})}\right)^2$ with high probability. Taking a union bound, we have that Proposition 3 and Proposition 4 hold simultaneously with probability at least $1-4\exp(-d)-\exp(-cn)$. Invoking Proposition 2, we have

$$\begin{aligned}\|\widehat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}^{\star}\|_F &\leq 32\sqrt{r}\cdot\frac{\lambda_n}{\kappa_{\mathcal{L}}\left(\frac{\mu_y}{\operatorname{tr}(\boldsymbol{\Sigma}^{\star})}\right)^2}\\ &\lesssim\left(\frac{\operatorname{tr}\left(\boldsymbol{\Sigma}^{\star}\right)}{\mu_y}\right)^2\sqrt{r}\lambda_n\end{aligned}$$

with probability at least $1-4\exp(-d)-\exp(-cn)$, as desired.

3.8 Proof of Corollary 1

We proceed by considering two cases. For each case, the proof consists of two steps. We first verify that the choices of the averaging parameter m and truncation threshold τ,

$$m = \left\lceil \left[\left(\frac{\nu_\eta^2}{(y^\uparrow)^2} \right)^2 \frac{N}{d} \right]^{1/3} \right\rceil \quad \text{and} \quad \tau = \frac{y^\uparrow}{\sigma_r r} \sqrt{\frac{N}{md}}, \tag{3.22}$$

satisfy the assumptions of Theorem 1, namely $n \geq C_2 rd$ and $\tau \geq \frac{\mu_y}{\mathrm{tr}(\boldsymbol{\Sigma}^\star)}$. We then invoke Theorem 1.

3.8.1 Case 1: high-noise regime

In this case, we have $\frac{\nu_\eta^2}{(y^\uparrow)^2} > \sqrt{\frac{d}{N}}$, which means by setting m according to (Equation 3.22), we have $m \geq 2$. As a result, the bound

$$\left\lceil \left[\left(\frac{\nu_\eta^2}{(y^\uparrow)^2} \right)^2 \frac{N}{d} \right]^{1/3} \right\rceil \leq 2 \left[\left(\frac{\nu_\eta^2}{(y^\uparrow)^2} \right)^2 \frac{N}{d} \right]^{1/3} \tag{3.23}$$

holds in the high-noise regime.

Verifying the condition on n. Recall that $n = \frac{N}{m}$. We have

$$\begin{aligned}
n = \frac{N}{m} &\overset{\text{(i)}}{\geq} \frac{N}{2} \left(\left(\frac{\nu_\eta^2}{(y^\uparrow)^2} \right)^2 \frac{N}{d} \right)^{-1/3} \\
&= \frac{1}{2} \left(N^2 d \left(\frac{(y^\uparrow)^2}{\nu_\eta^2} \right)^2 \right)^{1/3} \\
&\overset{\text{(ii)}}{\geq} \left(C_2^3 \left(\frac{(y^\uparrow)^2}{\nu_\eta^2} \right)^2 \left(\frac{\nu_\eta^2}{(y^\uparrow)^2} \right)^2 r^3 d^3 \right)^{1/3} \\
&= C_2 rd,
\end{aligned}$$

where step (i) is true by plugging in the choice of m from (Equation 3.22) and applying the bound (Equation 3.23), and step (ii) is true by substituting in the assumption $N \geq 2C_2^{3/2}\left(\frac{\nu_\eta^2}{(y^\uparrow)^2}\right)^2 r^{3/2}d$. Thus the condition $n \geq C_2 rd$ of Theorem 1 is satisfied.

Verifying the condition on τ. For the term $\sqrt{\frac{N}{dm}}$ in the expression of τ in (Equation 3.22), note that, by the previous point, $\frac{N}{m} = n \gtrsim rd$ (with a constant that is greater than 1). Thus $\sqrt{\frac{N}{dm}} \geq \sqrt{r} > 1$. Therefore, to verify the condition $\tau \geq \frac{\mu_y}{\operatorname{tr}(\boldsymbol{\Sigma}^\star)}$, it suffices to verify that

$$\frac{y^\uparrow}{\sigma_r r} \geq \frac{\mu_y}{\operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)}. \tag{3.24}$$

By definition, we have $y^\uparrow \geq \mu_y$. Furthermore, since $\boldsymbol{\Sigma}^\star$ is symmetric positive semidefinite, its eigenvalues are all non-negative and are identical to its singular values, and hence $\operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right) \geq \sigma_r r$, verifying the condition (Equation 3.24).

Invoking Theorem 1. By setting λ_n to its lower bound in (Equation 3.11) and substituting in $n = \frac{N}{m}$ and our choice of τ from (Equation 3.22), we have

$$\lambda_n = C_1\left(3\frac{(y^\uparrow)^2}{\sigma_r r}\sqrt{\frac{md}{N}} + \frac{\nu_\eta^2}{m}\right) \tag{3.25}$$

Substituting this expression of λ_n to the error bound (Equation 3.12), then substituting in our choice of m from (Equation 3.22) to (Equation 3.25) and defining $C' = 3C \cdot C_1$, we have

$$\|\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_F \leq C'\left(\frac{\operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)^2}{\sigma_r r}\right)\frac{(y^\uparrow)^{4/3}(\nu_\eta^2)^{1/3}}{\mu_y^2}\sqrt{r}\left(\frac{d}{N}\right)^{1/3}.$$

Using the fact that $\operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right) \leq \sigma_1 r$, we have

$$\|\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_F \leq C'\,\frac{\sigma_1^2}{\sigma_r}\frac{(y^\uparrow)^{4/3}(\nu_\eta^2)^{1/3}}{\mu_y^2}\,r^{3/2}\left(\frac{d}{N}\right)^{1/3}$$

as desired.

3.8.2 Case 2: low-noise regime

In this case, we have $\frac{\nu_\eta^2}{(y^\uparrow)^2} \leq \sqrt{\frac{d}{N}}$, which means by setting m according to Equation (Equation 3.22), we have $m = 1$. As a result, no averaging occurs.

Verifying the condition on n. Because $m = 1$ in this case, we have that $n = N$. By assumption, we have that $N \geq C_2 rd$, satisfying the condition $n \geq C_2 rd$ in Theorem 1.

Verifying the condition on τ. By the same analysis as in Case 1, we have that the condition $\tau \geq \frac{\mu_y}{tr\mathbf{\Sigma}^\star}$ in Theorem 1.

Invoking Theorem 1. By setting λ_n to its lower bound in (Equation 3.11), substituting in our choice of τ from (Equation 3.22) and noting $m = 1$, we have

$$\lambda_n = C_1 \left(3\frac{(y^\uparrow)^2}{\sigma_r r}\sqrt{\frac{d}{n}} + \frac{1}{\sigma_r r}\nu_\eta^2 \right). \tag{3.26}$$

We define $C' = 3C \cdot C_1$ and note that $n = N$ under Case 2. Substituting this expression of λ_n in (Equation 3.26) to the error bound (Equation 3.12), then using the fact that under Case 2, the bound $\nu_\eta^2 \leq (y^\uparrow)^2\sqrt{\frac{d}{N}}$ holds, we have

$$\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}^\star\|_F \leq C' \left(\frac{\mathrm{tr}\,(\mathbf{\Sigma}^\star)^2}{\sigma_r r} \right) \left(\frac{y^\uparrow}{\mu_y} \right)^2 \sqrt{\frac{rd}{N}}.$$

Using the fact that $\mathrm{tr}\,(\mathbf{\Sigma}^\star) \leq \sigma_1 r$, we have

$$\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}^\star\|_F \leq C' \, \frac{\sigma_1^2}{\sigma_r} \left(\frac{y^\uparrow}{\mu_y} \right)^2 \sqrt{\frac{r^3 d}{N}},$$

as desired.

3.9 Conclusion

We introduce the perceptual adjustment query, a cognitively lightweight way to obtain expressive human responses. We specifically investigate using PAQs for human perceptual similarity learning. Learning models of human perception or preference has a range of applications, including recommendation systems, interrogating generative models, and quantifying perceptual conditions such as color blindness or hearing loss. We use a Mahalanobis distance-based model for human similarity perception and use PAQs to estimate the unknown metric. This setup gives rise to a new inverted measurement scheme for high-dimensional low-rank matrix estimation which violates commonly held assumptions for existing estimators. We develop a two-stage estimator and provide corresponding sample complexity guarantees.

This work lays the foundation for future work in two directions: (1) practical deployment of PAQs and (2) theoretical characterization of learning from inverted measurements. One important aspect of deploying PAQs in practice is how to select the most informative query directions. While this work considers a random query direction scheme that is amenable for theoretical analysis, targeted selection of query directions may reduce the number of responses needed in practice. Conducting user studies to collect data from human responses will also bring additional insights into how the theoretical guarantees translate into practice.

Along theoretical lines, one key direction is to characterize the optimal rate for this problem by deriving information-theoretic lower bounds. It is possible that there exists a fundamental trade-off between the variance and the bias that arises from the measurement scheme; it is also possible that more sophisticated techniques are capable of overcoming such bias.

Part II

Circumventing human feedback

CHAPTER 4
PERSONALIZING WITHOUT HUMAN FEEDBACK: LARGE LANGUAGE MODEL AUGMENTED EXERCISE RETRIEVAL FOR PERSONALIZED LANGUAGE LEARNING

In this chapter[1], we study the cold-start phase of a recommender system, where minimal human feedback is available, and utilize the generative abilities of large language models in lieu of collecting human relevance labels. Specifically, we study the problem of zero-shot exercise retrieval in the context of online language learning, to give learners the ability to explicitly request personalized exercises via natural language. Using real-world data collected from language learners, we observe that vector similarity approaches poorly capture the relationship between exercise content and the language that learners use to express what they want to learn. This semantic gap between queries and content dramatically reduces the effectiveness of general-purpose retrieval models pretrained on large scale information retrieval datasets like MS MARCO [126]. We leverage the generative capabilities of large language models to bridge the gap by synthesizing hypothetical exercises based on the learner's input, which are then used to search for relevant exercises. Our approach, which we call mHyER, overcomes three challenges: (1) lack of relevance labels for training, (2) unrestricted learner input content, and (3) low semantic similarity between input and retrieval candidates. mHyER outperforms several strong baselines on two novel benchmarks created from crowdsourced data and publicly available data.

4.1 Introduction

Modern personalized education systems typically leverage the power of machine learning models to estimate user skill levels [127] and adaptively serve exercises to learners [128,

[1] The work in this chapter appears in [3]

129, 130]. Adaptivity, while a critical part of any personalized education system, is a *passive* form of personalization from the learner's point of view: While exercises are tailored to an estimate of the learner's skill level, this customization occurs behind the scenes, with no opportunity for learners to take initiative in shaping the learning process. In this chapter, we study a complementary form of *learner initiated* personalization in the context of *online language learning*. In particular, learners are given the ability to *explicitly* request learning content from an education system, which returns relevant exercises from a fixed catalog for the learner to do.

This type of learner initiated personalization can be viewed as a form of *self-directed learning*, where learners take initiative over the learning process. Self-directed learning has been shown to increase learner performance across multiple topics [131, 132, 133, 134], improve learner motivation [135], and create more cohesive learner environments [136]. Online language learning is a natural setting for self-directed learning, as people learn languages for very personal reasons: Some learn for fun, while others have specific goals, such as preparing for an international trip or developing language skills for business. Different reasons for learning lead to different needs for exercise content: Someone learning to write in a business setting may want extra practice with grammar or politeness, whereas the learner learning for a vacation may want exercises about hotels or transportation. Beyond highly personalized motivations for learning, online language learners do not have immediate access to instructors who can plan learning material to target weaknesses. As such, there is an inherent need for online language learners to have some degree of self-direction in order to get the most out of their learning experience.

With the goal of allowing language learners to tailor an online learning experience to their own needs, we formalize the task of **exercise retrieval for learner directed language learning** and evaluate machine learning models for this task. The goal of this task (Figure 4.1) is to retrieve relevant exercises from a set of existing exercises based on a learner's input. In this setting, collecting relevance labels (i.e., pairs of learner inputs and relevant exercises) is particularly challenging, as learners will typically be presented with only a small number of exercises for any given input. As a result, we consider the *zero-shot* setting, where we do not have access to relevance labels for training. While many off-the-shelf models exist for text-based retrieval, we show that direct similarity search (i.e., retrieving exercises that are the most similar to the user input in the representation space) with these models suffers from a semantic similarity gap between how users describe their learning objectives and exercise content. To overcome this gap, we leverage structure inherent to exercises and the generative capabilities of large language models. Specifically, we make the following contributions.

- We propose the new task of exercise retrieval for learner directed language learning in Section 4.3.1 and discuss how learner inputs give rise to a fundamental challenge in this task.
- We present our zero-shot retrieval approach, mHyER, in Section 4.4, and illustrate how augmenting retrieval with LLMs helps overcome the pitfalls of direct similarity search.
- With no existing benchmarks for this task, we create two novel benchmarks in exercise retrieval with both crowd-sourced data from learners of a popular language learning app and publicly available Tatoeba data. We evaluate our method against several strong dense retrieval baselines in Section 4.5 and empirically show that mHyER outperforms relevant baselines by a significant margin.

4.2 Related work

Exercise retrieval is naturally connected to the broad field of information retrieval, and in particular, dense retrieval [137, 138]. Dense retrieval focuses on retrieving documents based on similarity measured in a learned representation space. Zero-shot retrieval, or retrieval without training on task-specific relevance information, is of particular relevance to our

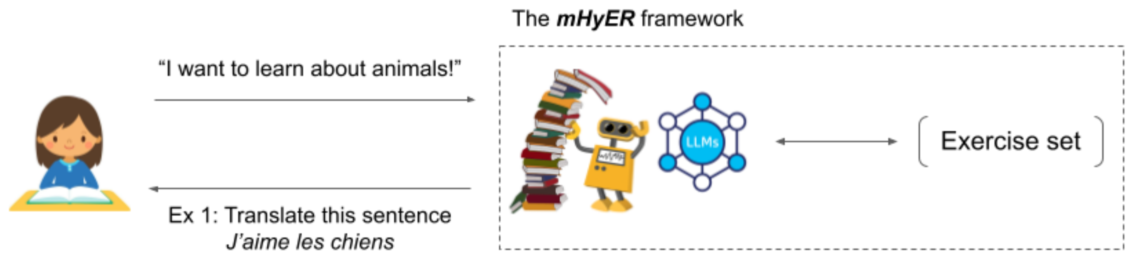

The *mHyER* framework
"I want to learn about animals!"
Ex 1: Translate this sentence
J'aime les chiens
LLMs
Exercise set

Figure 4.1: Exercise retrieval for learner directed language learning and our proposed solution, multilingual Hypothetical Exercise Retriever (mHyER). At a high level, learners are allowed to provide *any* natural language input, and the goal is to retrieve exercises relevant to that input. Our method utilizes large language models to do zero-shot retrieval.

task. Such methods typically rely on a supervised pretraining stage [139, 140, 141], where models are trained on large scale retrieval datasets, such as MS MARCO [126]. However, such supervised pretraining ultimately depends on the existence of suitable labeled datasets, which are not always readily available [142]. The rise of large language models (LLMs) with strong zero/few-shot performance in new domains has resulted in a line of research integrating LLMs into the retrieval pipeline. Such approaches typically rely on some combination of specialized prompting and synthesizing retrieval datasets to retrain retrieval models [143, 144, 145, 146]. Our approach takes particular inspiration from HyDE [147], which utilizes a LLM to synthesize a hypothetical document, which is used then used with a pretrained encoder to retrieve documents via nearest neighbors.

A fundamental step in any retrieval method is the representation space used for similarity comparisons. For the task of exercises retrieval, we focus on learning sentence embeddings, where pretrained language models such as BERT [148] or RoBERTa [149] serve as strong foundations. Contrastively learning sentence representations, which leverage techniques used in the image domain [150, 151], has become especially popular. The goal of contrastive learning is to learn a representation space where similar items ("positive pairs") are pulled close together while dissimilar items ("negative pairs") are pulled far apart in an unsupervised manner. In the image domain, positive pairs are formed by applying *data augmentation*, such as cropping or rotating an image. Such techniques are not directly transferable to natural language, resulting in a long line of methods [142, 152, 153, 154, 155] studying contrastive sentence embeddings. Of particular interest to the language learning setting is multilingual contrastive learning [156], where positive pairs can be taken as the same sentence in two different languages. In all, mHyER can be viewed as a combination of multilingual contrastive learning [156] and HyDE [147].

Personalized education systems often gauge a learner's skill level via Knowledge Tracing [127] in order to tailor exercise difficulty level. As a result, a variety of contemporary machine learning methods [157, 158, 159, 160, 161, 162] have been developed to track learner skill level from historical data. Such methods demonstrate strong empirical success and thus have been leveraged to adaptively recommend exercises to learners [128, 129] or even generate new exercises based on skill level [130]. This adaptivity can be viewed as a complementary piece to the problem of exercise retrieval directed language learning that we study in this chapter: learner initiated personalization can leverage existing tools from adaptivity to ensure exercises are both relevant and at the right skill level. On the other hand, adaptive systems can benefit from explicit learner direction. For example, we can view the learner input of "past tense verbs" as the learner explicitly saying they are not comfortable with past tense verbs, and use this information in skill estimation.

4.3 Problem setup

The goal of exercise retrieval for learner directed language learning is to retrieve relevant exercises for a learner given a text input from the learner describing what they want to learn. In particular, we assume that learner is taking a language learning course, which consists of two languages: the "first language" (i.e., a language they already know) and the "second language" (i.e., the language they are learning), which we refer to as L1 and L2, respectively.[2] The learner completes *exercises*, which are drawn from a fixed set of N exercises $\mathcal{E} = \{e_1, \ldots, e_N\}$ that are at an appropriate level for the learner. We can view

[2] These labels are an imprecise shorthand; L1 need not be the learner's first or native language, and L2 may be a third language or beyond.

this set of N exercises as samples from an unknown *exercise distribution*, which captures characteristics (style, length, content, etc) of exercises. For simplicity, we limit our attention to translation exercises, in which a learner translates an L1 sentence $e_i^{(\texttt{L1})}$ to the L2, with one correct L2 answer $e_i^{(\texttt{L2})}$ available as an example of a correct translation. The learner provides some input t, and our objective is to retrieve the K (unique) exercises that are the most relevant based on input t in a zero-shot manner. That is, without using any labeled relevance data for training, we want to retrieve K unique exercises $e_1^\star, \ldots, e_K^\star$ that maximize probability p that an exercise is relevant conditioned on learner input t:

$$e_1^\star, \ldots, e_K^\star = \underset{\substack{e_1, \ldots e_K \in \mathcal{E} \\ e_i \neq e_j \quad \forall i,j,\, i \neq j}}{\arg\max} \quad \prod_{i=1}^{K} p(e_i | T = t). \tag{4.1}$$

4.3.1 Learner inputs.

The core of the personalized experience in this problem setting is allowing learners to provide an input describing what they want to learn with *no restrictions on input content*, resulting in large number of potential input types. For example:

- **Topics:** Learners can request exercises that teach vocabulary relevant to a particular topic. Inputs such as "words about animals" or "countries" are such examples.
- **Grammar:** Learners can request exercises teaching grammatical concepts, such as "non-present tenses" or "irregular plurals".
- **Culture:** Learners can request to review culture-specific aspects of language, such as idioms, slang, or region-specific quirks. For example, a learner learning Spanish may want to learn about "voseo", a region-specific grammatical concept in South America.
- **Learning process:** Learners can request exercises that help with particular parts of the process of learning a language, such as "words that are hard to spell" or "sentences for first-year students".

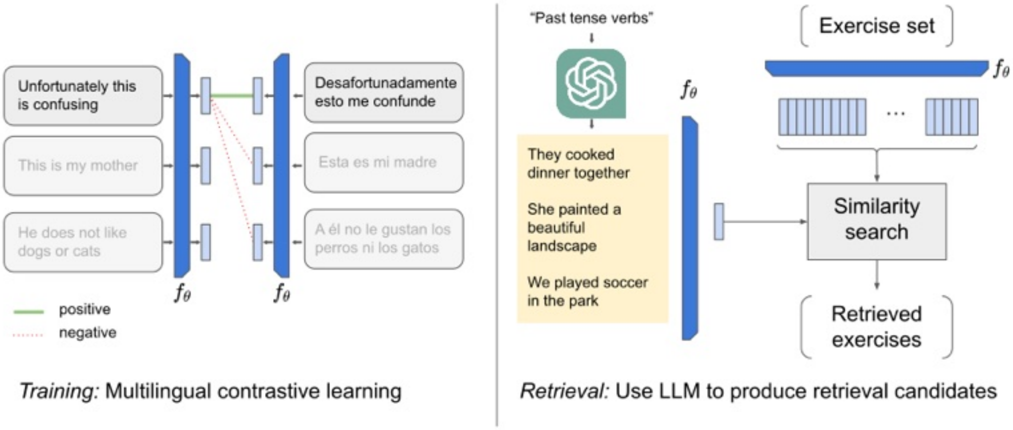

Unfortunately this is confusing
This is my mother
He does not like dogs or cats
Desafortunadamente esto me confunde
Esta es mi madre
A él no le gustan los perros ni los gatos
f_θ
f_θ
positive
negative
Training: Multilingual contrastive learning
"Past tense verbs"
They cooked dinner together
She painted a beautiful landscape
We played soccer in the park
f_θ
Exercise set
f_θ
...
Similarity search
Retrieved exercises
Retrieval: Use LLM to produce retrieval candidates

Figure 4.2: mHyER consists of two stages. Contrastive finetuning (left) is employed as a training stage to optimize our semantic similarity space for multilingual exercises. Then at retrieval time (right), a large language model is employed to synthesize hypothetical retrieval candidates. These retrieval candidates are then used in direct similarity search to retrieve exercises.

Learner inputs of these types result in what we call a *referential similarity gap*: Under modern text-based retrieval models, how learners express their learning objectives (i.e., the learner input t) is not considered similar to what it is referring to, i.e., the content of the exercises $e^{(\texttt{L1})}$ and $e^{(\texttt{L2})}$. We explore this gap in greater detail in Section 4.4.3.

4.4 Method

In this section, we present multilingual Hypothetical Exercise Retriever (mHyER), our zero-shot exercise retrieval framework, and show that it overcomes the pitfall of direct similarity search in learned representation spaces.

4.4.1 Baseline: direct search with similarity spaces.

The backbone of text-based retrieval is a vector space representation of text that reflects some notion of similarity between different pieces of text. Forming these representation spaces remains a core part of text-based retrieval, with early methods such as BM25 [163] formed representations via word frequency. Such methods struggle to generalize as their representation spaces are formed based on counting exact or near text matches. To improve generalization, contemporary methods for text-based retrieval typically train a model f_θ (parametrized by θ) that maps natural language inputs (from the space of all text inputs $\mathcal{T}$) to some d-dimensional vector space: $f_\theta : \mathcal{T} \to \mathbb{R}^d$. Such models are referred to as *encoders*, and learn representations of text called *embeddings*. That is, if $t \in \mathcal{T}$ is some text, then $f_\theta(t)$ is its embedding representation. Because exercises are typically short sentences or sentence fragments, we focus on encoders specifically geared towards learning sentence embeddings in this work.

Harnessing the vast availability of text data, contemporary encoders are typically neural networks trained such that texts with similar content are more similar in the representation space under some measure, like cosine similarity. That is, if $t_1, t_2 \in \mathcal{T}$ are similar in content, then $\text{sim}(f_\theta(t_1), f_\theta(t_2))$ is large (and positive). This similarity space suggests a natural approach for retrieving exercises: Pass each exercise e_i through the model f_θ to obtain its embedding representation $f_\theta(e_i)$.[3] Then, when a learner provides an input t, pass t through the model to obtain $f_\theta(t)$ and return the K exercises with largest cosine similarity $\text{sim}(f_\theta(e_i), f_\theta(t))$. As we see in Section 4.4.3, direct similarity search often retrieves sentences featuring "language about language", which are often irrelevant to the learner's input. This leads us to leverage the generative abilities of LLMs, as we discuss next.

4.4.2 mHyER: augmenting direct search with generative capabilities.

If large quantities of relevance data were available, we could train a model to approximate the relevance probability in (Equation 4.1) by learning a representation space where learner inputs and relevant exercises are considered similar and then performing direct search. However, input relevance data is unlikely to be available at the scale necessary to train such a model. Instead, we propose mHyER, visualized in Figure 4.2, which after a multilingual contrastive training stage, retrieves exercises in a two-step manner. First, we sample a set of

[3] We slightly abuse notation here and write $f_\theta(e_i)$ to mean either $f_\theta(e_i^{(\texttt{L1})})$ or $f_\theta(e_i^{(\texttt{L2})})$. The choice to compare against the representation of the L1 or L2 sentence is explored in Section 4.5.

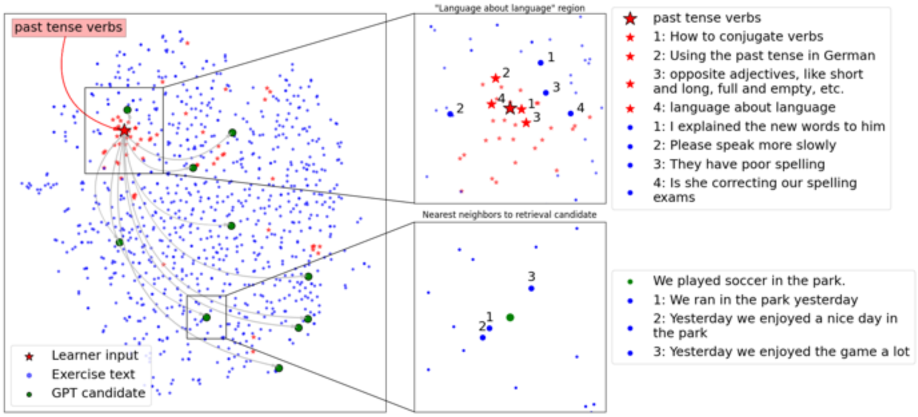


Figure 4.3: TSNE visualization of exercise, learner input, and GPT-4-synthesized retrieval candidate representations in the representation space of a trained mBERT encoder (left). Learner inputs concentrate in the language about language region (top right), making direct similarity search sub-optimal. Retrieval candidates bridge the referential similarity gap between learner inputs and exercise text and are close in similarity to exercises that meet the learner's specifications (bottom right).

N_c hypothetical exercises from the exercise distribution *conditioned on the learner input*. We call these sampled exercises our *retrieval candidates*. In principle, we do not have access to this exact distribution, but we can efficiently approximate sampling via LLM. Second, we use the retrieval candidates to perform similarity search via K-nearest neighbors. mHyER is inspired by two complementary methods: the multilingual contrastive learning approach of [156], and the HyDE retrieval method of [147]. We now discuss both the training and retrieval stages in greater detail.

Stage 1: Learning a multilingual similarity space. While we operate in a setting where no explicit learner relevance data is provided, the multilingual nature of our exercises implies that a certain structure should exist in our representation space. Namely, the sentence $e_i^{(\mathtt{L1})}$ in L1 should be similar to its translation $e_i^{(\mathtt{L2})}$ in L2. To ensure this structure is reflected in our representation space, we take inspiration from [156] and utilize multilingual contrastive learning, an unsupervised approach that aims to learn a representation where similar items (called *positive* pairs) are closer together and dissimilar items (called *negative* pairs) are far apart. For exercise e_i, the contrastive loss $\mathcal{L}_i$ with a mini-batch of N_B sentence pairs is

$$\mathcal{L}_i = -\log \frac{\exp\left(\text{sim}\left(f_\theta(e_i^{(\texttt{L1})}), f_\theta(e_i^{(\texttt{L2})})\right)/\tau\right)}{\sum_{j=1}^{N_B} \exp\left(\text{sim}\left(f_\theta(e_i^{(\texttt{L1})}), f_\theta(e_j^{(\texttt{L2})})\right)/\tau\right)}, \tag{4.2}$$

where τ is the user-set temperature parameter and $\text{sim}\,(\cdot,\cdot)$ is the cosine similarity:

$$\text{sim}\left(f_\theta(t_1), f_\theta(t_2)\right) = \frac{\langle f_\theta(t_1) f_\theta(t_2)\rangle}{\|f_\theta(t_1)\|_2 \|f_\theta(t_2)\|_2}. \tag{4.3}$$

In this work, rather than train a sentence encoder from scratch, we follow the commonly accepted practice of initializing our encoder with existing BERT-based checkpoints and contrastively finetuning these checkpoints on exercise data.

Stage 2: Sampling retrieval candidates and exercise retrieval. A core component of mHyER is sampling from the exercise distribution conditioned on the learner input. While we cannot sample directly from this distribution, we can approximate sampling with a LLM. In particular, we prompt the LLM with a fixed a description of the exercise distribution and instruct the LLM to synthesize *hypothetical* exercises based on this description and based on a learner's input. Crucially, we can synthesize exercises *without requiring any labeled examples*, i.e., we do not embed examples of inputs and relevant exercises in the prompt. To retrieve exercises, the LLM synthesizes K_h hypothetical exercises, which we denote $\tilde{e}_1, \ldots, \tilde{e}_{K_h}$. We then encode these hypothetical exercises via f_θ to obtain K_h vectors $f_\theta(\tilde{e}_1), \ldots, f_\theta(\tilde{e}_{K_h})$. To retrieve exercises, we retrieve the K exercises that have the highest similarity score compared to the average of the K_h vectors: $\frac{1}{K_h}\sum_{i=1}^{K_h} f_\theta(\tilde{e}_i)$. We use GPT-4 [39] in this work, but in practice, any LLM of sufficient capacity can be used.

4.4.3 Bridging the referential similarity gap with mHyER.

In an effort to better understand the task of retrieving exercises from learner inputs, we crowdsourced a small dataset of learner inputs from users of Duolingo, a popular language learning app. We then contrastively finetune mBERT with roughly 40,000 real exercises from the app, spanning 4 different language courses. To get a sense of how contrastively learned similarity spaces reflect learner inputs and exercise text, we visualize our collected data, along with a subsample of the exercises, via TSNE in Figure 4.3. This visualization reveals **a fundamental referential similarity gap between learner inputs and exercise text**: How learners describe what they want to learn occupies a distinct part of the representation space, characterized by explicit use of words or phrases about language (e.g., "verbs", "past tense", "adjectives"). We refer to this region as the "language about language" region. As a result, direct similarity search yields exercises that similarly contain words explicitly about language. As shown in Figure 4.3, the input "past tense verbs" is most similar to exercises about language (e.g., "I explained the new words to him"). Figure 4.3 also highlights how synthesizing retrieval candidates helps bridge this referential similarity gap by "translating" the learner's input (which is typically expressed in "language about language") to a hypothetical in-distribution exercise whose content satisfies the learner input. We provide concrete examples of learner inputs and synthesized retrieval candidates in Table 4.1.

4.5 Datasets and experimental results

In this section, we first give an overview of two novel datasets specifically for the task of learner directed language learning. We then compare mHyER against a variety of baselines on these datasets.

Table 4.1: Examples of collected learner inputs and retrieval candidates synthesized based on the learner input via GPT-4. For a variety of input types, GPT-4 is able to bridge the referential similarity gap by synthesizing text that closely resembles real exercise text while incorporating the concept that the learner wants to learn.

Input	Synthesized retrieval candidates	
Past tense	They went to the concert last night. Did you finish your project on time? We didn't have any coffee this morning.	She cooked a delicious meal for us. He had never seen such a beautiful sunset. Were they able to solve the problem?
Future tense	She will be moving to France next year. I won't attend the party tonight. When will you finish the project?	They'll be studying for the exam tomorrow. In five years, I'll have my own business. We're going to plant a garden this summer.
Present progressive verbs	Are you studying for the test? She's preparing dinner for tonight. They're practicing their dance routine.	He's not listening to the lecture. I'm writing a letter to my friend. The cat is chasing its tail.
Idiomatic syntax	It's raining cats and dogs! Don't put all your eggs in one basket. He's feeling under the weather.	She has a heart of gold. I'm on cloud nine. Keep your chin up!
How to order food at a restaurant	Could I see the menu, please? I'd like to order the grilled salmon. Does this dish contain any nuts?	May I have a glass of water? Can I substitute fries for a salad? Are there any vegetarian options?

4.5.1 Datasets

Duolingo Relevance (DuoRD) Dataset. To evaluate our method, we collected a small scale dataset of 61 learner inputs from learners of Duolingo, a popular language learning app. For each input, we asked the learner to rate 15 exercises as relevant or irrelevant to their input, resulting in 915 total exercises rated. Exercises were sourced a pool of approximately 40,000 sentence pairs across four distinct courses. To ensure that the dataset was not skewed too heavily towards relevant or irrelevant responses, we utilize a sampling approach. Using mHyER, we retrieved the top 555 exercises in terms of similarity. To form the set of 15 exercises for the learner to rate, we select the top 5 scoring exercises deterministically (Tier 1). From the next 50 highest scoring exercises, we randomly select 5 exercises uniformly at random without replacement (Tier 2). We repeat this sampling again, randomly drawing 5 exercises from the remaining 500 exercises (Tier 3). We observe that 64% of exercises from Tier 1 were rated as relevant, 50% from Tier 2, and 34% from Tier 3, resulting in 49% of all exercises rated as relevant.

Tatoeba Tags dataset. To test our method on a larger scale, we construct a retrieval dataset from Tatoeba, a public database of sentences and their translations. We begin by noting that when sentences are uploaded to Tatoeba, they are often tagged by grammatical concepts, language specific concepts, or topics. For example, the sentence “The brown bear is an omnivore” is tagged with “animals” and the sentence “That way I kill two birds with one stone” is tagged with “idiomatic expression”. We treat each of these tags as a learner input, and deem an exercise relevant if it has been tagged accordingly. While per sentence tags are not necessarily exhaustive, they provide useful signal for evaluating retrieval approaches with typical retrieval metrics as well as binary classification metrics, as we discuss in the Section 4.5.2. We form 3 benchmarks for evaluation, collectively referred to as the Tatoeba Tags dataset:

- English benchmark: only English sentences with 139 tags and 89,392 sentences.
- Spanish from English benchmark: Spanish-English sentence pairs with 114 tags in Spanish and 49,258 pairs.
- English from Spanish benchmark: Spanish-English sentence pairs with 108 tags in Spanish and 46,837 pairs.

To form the benchmarks, we collect all tags corresponding to the benchmark, filter out tags and sentences containing profanity, merge similar tags together, and then perform benchmark specific language and content processing. We then keep only the tags with more than 20 sentences and download the corresponding sentences. The benchmark specific processing is done to better align the benchmark with how learners would interact with real-world language learning courses. Specifically, we perform both language and content processing. For language processing, we translate all tags (which appear in a variety of languages) to the L1. This is done to emulate the learning process: we use tags as a stand-in for learner inputs, which are likely to be the learner's L1. For content processing, we remove tags that do not make sense in the context of a particular learning direction. For example, a Spanish speaker learning English would not input "voseo" (a Spanish grammatical concept), nor would an English speaker learning Spanish input "British English".

4.5.2 Evaluation procedure and metrics

For the DuoRD dataset, we treat the 915 exercises that have been rated for some learner input as the exercise set. Because each of the 915 exercises was not assigned a relevance rating for every learner inputs, we cannot use typical information retrieval metrics such as Recall or Precision. As a result, we treat evaluation on this dataset as a binary classification problem, where the goal is to predict whether an exercise is relevant or irrelevant. To evaluate methods, we use area under the receiver operating characteristic curve (AUC) and accuracy. To compute AUC, for each retrieved exercise, we compute a *score* equal to the similarity measure between the retrieval candidate and all exercises. We then aggregate relevance labels and scores across all learner inputs to define the ROC curve. To compute accuracy, we compute the scores as in AUC, and set a threshold such that any exercise above the threshold is deemed relevant and vice versa. Because the similarity score ranges between -1 and 1, we set the threshold by sweeping over $[-1, 1)$ in increments of 0.1. We then report the highest accuracy among all thresholds in the sweep.

For the Tatoeba Tags dataset, because we have a notion of relevance, as indicated by the presence of a tag, we utilize Precision@K, which is a common metric in information retrieval that reports the fraction of the K retrieved exercises that are relevant. To compute Precision@K, we retrieve K sentences per learner input (i.e., tag) and record the fraction of the K retrieved sentences tagged with the learner input tag. Because the tagging of Tatoeba sentences is not exhaustive, the absolute values of reported Precision@K may be low, but relative performance still indicates how methods would perform if tagging was comprehensive. In light of this, we again follow the evaluation approach of the DuoRD dataset and report AUC.

When performing evaluation in both datasets, we can retrieve exercises in two distinct ways. We can synthesize retrieval candidates in the L1 and perform similarity search on the L1 exercise texts. Alternatively, we can synthesize retrieval candidates in the L2 and perform similarity search on the L2 exercise texts (example translations). As a result, we report AUC, accuracy, and precision@K in both the L1 and L2 setting.

4.5.3 Baselines

For both the DuoRD dataset and Tatoeba tags dataset, we evaluate mHyER against direct similarity search using BERT and mBERT [148], as well as the following BERT-based models: Contriever, mContriever [142], and SimCSE [152]. In particular, we use the BERT $_{\texttt{base}}$ (110 million parameters) variant of each of the above methods. These methods achieve strong unsupervised performance in a variety of retrieval and semantic text similarity tasks. BERT and mBERT were trained in a self-supervised manner by using masked language modeling and next sentence prediction objectives, with the only difference being the training data (only English for BERT and a multilingual corpus for mBERT).

Contriever and mContriever propose two new approaches in contrastively tuning BERT: (1) utilizing an inverse-cloze task and independent cropping as means of forming positive pairs and (2) utilizing a Momentum encoder as described in [142] to ensure better representation of negative items. Contriever is initialized with BERT and trained on English CCNet [164] and Wikipedia data, whereas mContriever was initialized with mBERT and trained on multiple languages in CCNet. We also consider supervised variants of Contriever and mContriever, which are finetuned on the MS MARCO [126], a large scale retrieval dataset. SimCSE uses dropout to create synthetic positive pairs for the contrastive loss by passing the same sentence through the encoder with different random dropout parameters. Starting with BERT, SimCSE is trained on Wikipedia data.

Table 4.2: Examples of exercises retrieved with direct similarity search and mHyER for the same input on the Tatoeba Tags English Benchmark. Direct similarity search is not capable of bridging the fundamental referential similarity gap between learner inputs and exercise content, as illustrated by "Subject verb agreement", "Second person", and "Colloquial" inputs. In settings where learners ask about specific topics, direct similarity search returns exercises that most literally match the learner input, as shown with the "Preference", "Cooking" and "Sports" inputs. On the other hand, mHyER retrieves exercises well aligned with the learner input.

Input	Direct similarity search	mHyER
Subject verb agreement	Correct the underlined words That's a transitive verb It's a transitive verb	The dogs are in the garden They grow flowers in the garden The children are playing in the garden
Second person	It's secondhand It is secondhand Next person please	Are you sure you want me to help you with your homework? I'm assuming you could speed through your testimony... Will you please check to see if my order has been dealt with?
Colloquial	Be punctual Speaking Talk is cheap	You drive me round the bend You're laying it on a bit thick You're joshing me
Preference	Make your choice Compromise Make a choice	Which do you like better, Mexican food or Chinese food? Which sweet do you prefer? Which do you better, pizza or tacos?
Cooking	My hobby is cooking Eat and drink Do the laundry	Pour melted butter over the popcorn Add the chives and season the salad Will you warm up the soup?
Sports	I like playing sports I love sports I like sports	One must practice every day in order to become a worldclass athlete Which do you like better skating or skiing? Which do you like better cycling or jogging?

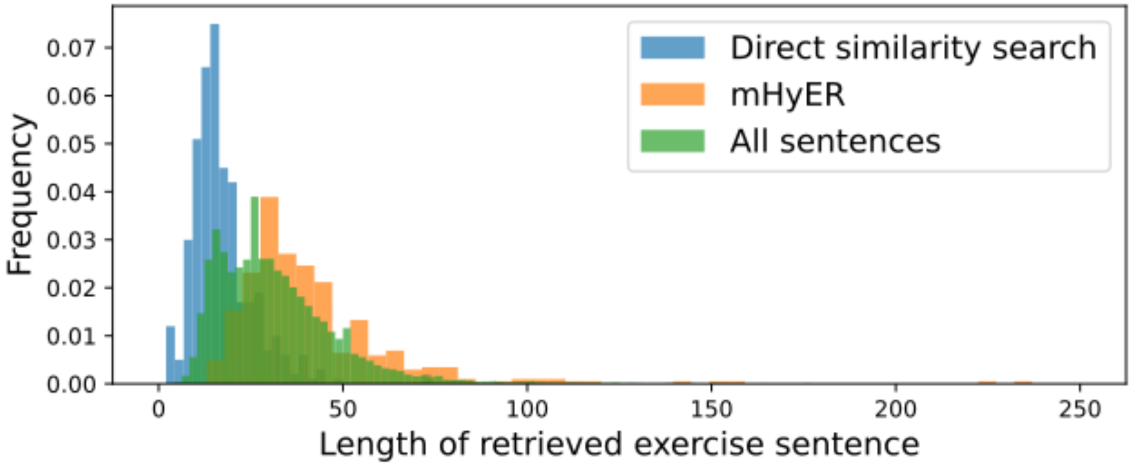


Figure 4.4: Length of the top 3 retrieved exercise sentences, measured in number of characters, for direct similarity search and mHyER. Exercises retrieved via direct similarity search are inherently *biased* in length, with a majority of exercises being relatively short. Using mHyER results in exercises of more varied length. This variation in length aligns well with the global distribution of exercises, showing that mHyER effectively translates learner inputs to the in-distribution exercises.

4.5.4 Direct similarity search vs. mHyER: A qualitative case study

Before we present our full experimental results, we first present examples of inputs and retrieved sentences on the English benchmark of the Tatoeba Tags dataset. To qualitatively gauge the difference between direct similarity search and mHyER, we provide examples of retrieved exercises for a small number of inputs in Table 4.2. We present the top three retrieved exercises measured in terms of similarity score for both direct similarity search and mHyER, using mBERT finetuned on Tatoeba data as our similarity space. The input “Subject verb agreement” highlights the “language about language” phenomena: Instead of retrieving exercises containing correct subject verb agreement, direct similarity retrieves exercises in the “language about language” part of the similarity space. These exercises contain words such as “words” and “verb”. On the other hand, mHyER is capable of bridging the gap between input and exercises, retrieving exercises that focus on ensuring sentences with plural objects have the right verb form. The “Preference” input illustrates an example of a nebulous input, as the learner wants exercises that have to do with expressing

Table 4.3: Evaluation results on the DuoRD dataset. mHyER$_{\texttt{[model]}}$ indicates that contrastive finetuning was employed with [`model`] as the initial checkpoint. +DuoRD dataset denotes that the DuoRD dataset was used for contrastive finetuning. In all cases, mHyER outperforms relevant baselines dramatically.

		AUC	AUC	Accuracy at threshold	Accuracy at threshold
		`L1`	`L2`	`L1`	`L2`
Unsupervised pretraining	BERT	0.458	0.421	0.491 @ -1.0	0.507 @ 0.9
	mBERT	0.485	0.471	0.491 @ -1.0	0.491 @ -1.0
	Contriever	0.565	0.586	0.557 @ 0.4	0.540 @ 0.3
	mContriever	0.497	0.499	0.548 @ 0.6	0.523 @ 0.5
	SimCSE	0.579	0.536	0.564 @ 0.5	0.523 @ 0.4
	mHyER$_{\text{mBERT}}$ `+DuoRD dataset`	0.679	0.669	0.631 @ 0.5	0.605 @ 0.5
	mHyER$_{\text{mContriever}}$ `+DuoRD dataset`	**0.680**	**0.678**	**0.635 @ 0.3**	**0.624 @ 0.3**
Supervised pretraining	Contriever-`sup`	0.609	0.605	0.579 @ 0.3	0.581 @ 0.2
	mContriever-`sup`	0.558	0.543	0.541 @ 0.3	0.529 @ 0.3
	mHyER$_{\text{Contriever}-\texttt{sup}}$ `+DuoRD dataset`	0.674	0.578	0.632 @ 0.4	0.562 @ 0.4
	mHyER$_{\text{mContriever}-\texttt{sup}}$ `+DuoRD dataset`	**0.684**	**0.678**	**0.637 @ 0.3**	**0.612 @ 0.4**

preferences. However, direct similarity search returns exercises explicitly about “choice”, whereas mHyER retrieves exercises that have learners practice expressing preference in more natural settings. The last two inputs, “Cooking” and “Sports”, illustrate instances where direct similarity search yields exercises that too literally match the input.

Aside from retrieval quality, we observe that retrieved results from direct similarity search also suffer from *sentence length bias*. In contrastively learned similarity spaces, it has been empirically observed that the length of a sentence is implicitly encoded in the representation of a sentence, meaning sentences of a similar length are more likely to be considered similar [154]. The retrieved exercises from direct similarity search shown in Table 4.2 clearly exhibit this bias whereas those retrieved via mHyER exhibit a higher

Table 4.4: Evaluation results on the Tatoeba Tags dataset. mHyER$_{\texttt{[model]}}$ indicates that contrastive finetuning was used with [model] as the initial checkpoint. +[dataset] denotes that [dataset] data was used for contrastive finetuning. In all cases, mHyER outperforms relevant baselines dramatically, with large gains coming from finetuning on *out-of-distribution* data.

		English		English (L2) from Spanish (L1)				Spanish (L2) from English (L1)			
		AUC	P@15	AUC	AUC	P@15	P@15	AUC	AUC	P@15	P@15
				L1	L2	L1	L2	L1	L2	L1	L2
Unsupervised pretraining	BERT	0.495	0.032	0.481	0.428	0.019	0.020	0.492	0.505	0.044	0.020
	mBERT	0.468	0.037	0.446	0.487	0.038	0.040	0.469	0.442	0.039	0.019
	Contriever	0.536	0.161	0.542	0.523	0.112	0.073	0.529	0.549	0.165	0.087
	mContriever	0.571	0.064	0.438	0.503	0.051	0.063	0.559	0.564	0.061	0.027
	SimCSE	0.646	0.115	0.535	0.559	0.069	0.054	0.635	0.610	0.127	0.068
	mHyER$_{\text{mBERT}}$ +en-from-es	0.722	0.225	0.686	0.701	0.227	0.208	0.710	0.696	0.243	0.242
	mHyER$_{\text{mBERT}}$ +es-from-en	0.717	0.223	0.697	0.693	0.219	0.211	0.702	0.706	0.237	0.244
	mHyER$_{\text{mBERT}}$ +DuoRD dataset	0.752	0.211	0.734	0.738	0.215	0.206	0.739	**0.757**	0.225	0.242
	mHyER$_{\text{Contriever}}$ +DuoRD dataset	**0.768**	0.239	0.644	**0.780**	0.106	0.232	**0.749**	0.659	0.265	0.099
	mHyER$_{\text{mContriever}}$ +DuoRD dataset	0.729	**0.258**	**0.748**	0.723	**0.267**	**0.264**	0.713	0.744	**0.271**	**0.294**
Supervised pretraining	Contriever-sup	0.541	0.164	0.491	0.492	0.120	0.086	0.530	0.492	0.180	0.105
	mContriever-sup	0.575	0.104	0.548	0.510	0.126	0.108	0.560	0.581	0.112	0.101
	mHyER$_{\text{Contriever}-\texttt{sup}}$ +DuoRD dataset	**0.775**	0.246	0.668	**0.797**	0.102	0.240	**0.760**	0.692	**0.268**	0.108
	mHyER$_{\text{mContriever}-\texttt{sup}}$ +DuoRD dataset	0.738	**0.255**	**0.761**	0.734	**0.260**	**0.264**	0.722	**0.752**	0.255	**0.280**

variation in length. We confirm that this phenomena holds for all inputs in the Tatoeba Tags English benchmark by retrieving the top 3 exercises across all 139 tags with both direct similarity search and mHyER. For each exercise, we record its length (measured in number of characters). As shown in Figure 4.4, exercises retrieved with mHyER are longer on average, aligning remarkably well with the global sentence length distribution. On the other hand, direct similarity search yields sentences that are notably shorter on average. This empirical observation highlights that generating in-distribution retrieval candidates allows us to retrieve sentences of varied length that track well with our set of exercises.

4.5.5 Experimental results

In this section, we present our experimental results on the DuoRD dataset and Tatoeba Tags dataset. For both datasets, we consider two starting points for fine-tuning the BERT embedding model: *Unsupervised pretraining*, where we contrastively train a BERT checkpoint that has been pretrained in an unsupervised manner, and *supervised pretraining*, where we start with a BERT checkpoint that has been pretrained on MS MARCO [126], a large scale retrieval dataset that covers different tasks, such as passage ranking and keyphrase extraction. In all settings, mHyER$_{\texttt{[model]}}$ denotes mHyER with starting with [`model`] as its initial checkpoint for contrastive training. [`model`]-`sup` indicates [`model`] was trained in a supervised manner. **We emphasize that at no point in training mHyER is labeled training data for exercise retrieval used**; `sup` only indicates MS MARCO was used to train the initial BERT checkpoint. For all experiments, we take the `[CLS]` representation as the sentence representation, except when working with Contriever and mContriever, where we use their custom mean pooling[4]. For all experiments with mHyER, we adopt the training setup of [156], which is adapted from [152], including all default hyperparameters. For retrieval, we synthesize $K_h = 10$ hypothetical retrieval candidates from GPT-4 and perform nearest neighbors search with the averaged embedding of all K_h candidates.

DuoRD dataset. The evaluation results of baselines and mHyER on the DuoRD dataset are presented in Table 4.3. For both unsupervised and supervised settings, we contrastively finetune BERT checkpoints on the full 40K exercises in the DuoRD dataset. In the unsupervised pretraining setting, we start our contrastive finetuning with two multilingual checkpoints: mBERT and mContriever. In this setting, mHyER outperforms all relevant baselines in both AUC and accuracy, with mHyER $_{\text{mContriever}}$ achieving the best performance among all methods. **mHyER $_{\text{mContriever}}$ results in 36.8% and 40.2% AUC gains over**

[4] See https://huggingface.co/facebook/contriever for further details.

mContriever and mBERT, respectively. It is notable that direct similarity search generally fails to perform well, highlighting that the gap between learner inputs and relevant exercises: BERT, mBERT, and mContriever baselines fail to even achieve an AUC of 0.5 corresponding to random guessing, reinforcing the fact that **direct similarity search cannot overcome the fundamental mismatch between how learners describe what they want to learn and exercise content.** In the supervised pretraining setting, we start our finetuning from the Contriever-`sup` and mContriever-`sup` checkpoints, which were finetuned on labeled MS MARCO data. mHyER once again outperforms all baselines, with mHyER $_{\text{mContriever}-\texttt{sup}}$ as the best performing method. Here, supervised pretraining modestly improves the performance of direct similarity search, suggesting that supervised pretraining can lessen the referential similarity gap in a limited manner. The improvement due to supervised pretraining is less pronounced when utilizing mHyER, with even one instance of decreased accuracy. This suggests synthesized retrieval candidates bridge the gap to the point where further improvement is difficult.

Tatoeba Tags dataset. The evaluation results of baselines and mHyER on the Tatoeba Tags dataset are presented in Table 4.4. On this dataset, we experiment with contrastive finetuning on *out-of-distribution data*. This experiment was inspired by empirical observations from finetuning mBERT. In particular, we contrastively finetuned mBERT on the Spanish from English benchmark (denoted `es-from-en`) and the English from Spanish benchmark (denoted `en-from-es`), as well as the 40K *out-of-distribution* sentence pairs from the DuoRD dataset (which contains English-Spanish pairs). We observe that finetuning on the DuoRD dataset outperforms finetuning on in-distribution data. This surprising observation leads us to finetune Contriever and mContriever checkpoints with the DuoRD dataset in both the unsupervised and supervised settings. In the unsupervised setting, we once again observe poor performance from direct similarity search baselines and sizable increases in

Table 4.5: Ablation results on the Tatoeba Tags dataset. We experiment by removing either the contrastive finetuning step or the retrieval candidate synthesis step. +GPT indicates that retrieval candidates were used with no contrastive finetuning, whereas +DuoRD indicates that direct similarity search was used after contrastively finetuning on the DuoRD dataset. In a vast majority of cases, contrastive finetuning and retrieval candidate synthesis boost performance, with retrieval candidates generally contributing more.

		English		English (L2) from Spanish (L1)				Spanish (L2) from English (L1)			
		AUC	P@15	AUC L1	AUC L2	P@15 L1	P@15 L2	AUC L1	AUC L2	P@15 L1	P@15 L2
Unsupervised pretraining	mContriever	0.571	0.064	0.438	0.503	0.051	0.063	0.559	0.564	0.061	0.027
	mContriever +GPT	0.676	0.237	0.613	0.663	0.213	0.213	0.643	0.602	0.245	0.217
	mContriever +DuoRD	0.665	0.096	0.670	0.665	0.119	0.106	0.656	0.657	0.090	0.077
	mHyER$_{\text{mContriever}}$ +DuoRD	**0.729**	**0.258**	**0.748**	**0.723**	**0.267**	**0.264**	**0.713**	**0.744**	**0.271**	**0.294**
Supervised pretraining	mContriever-sup	0.575	0.104	0.548	0.510	0.126	0.108	0.560	0.581	0.112	0.101
	mContriever-sup +GPT	0.731	0.250	0.642	0.724	0.238	0.243	0.706	0.636	**0.263**	0.258
	mContriever-sup +DuoRD	0.672	0.106	0.678	0.677	0.128	0.120	0.662	0.661	0.113	0.091
	mHyER$_{\text{mContriever-sup}}$ +DuoRD	**0.738**	**0.255**	**0.761**	**0.734**	**0.260**	**0.264**	**0.722**	**0.752**	0.255	**0.280**

performance when using mHyER: **Up to 39% increases in AUC and more than doubling the performance of precision@15 between the best mHyER method and best direct similarity.** We observe similar gains in the supervised pretraining setting. Methods that use Contriever (pretrained only on English data) typically perform better when retrieving in English, whereas methods using mContriever typically perform better in multilingual settings.

4.5.6 Ablation study

The two key steps in mHyER are multilingual contrastive pretraining and synthesizing retrieval candidates. To characterize the relative contributions of each step, we create variants of mHyER performing direct similarity search after contrastive pretraining or retrieving with GPT-synthesized retrieval candidates with a non-finetuned encoder (i.e., HyDE [147]). As shown in Table 4.5, the combination of both stages yields the best performance in the vast majority of cases. Utilizing only synthesized retrieval candidates results in the larger increases in precision compared to contrastive finetuning, while the opposite is true for AUC. This suggests that the two steps drive performance increases in complementary ways: Contrastive finetuning changes the similarity space such that relevant exercises are closer to learner inputs at a *global* level, resulting in increases in AUC (which measures a global ranking of predicitions). However, direct similarity search still cannot overcome referential similarity gaps, and hence, increases in precision@15 are low relatively. Meanwhile, synthesizing retrieval candidates directly improves retrieval quality, resulting in higher retrieval quality, but does not change representations, resulting in relatively lower increases in AUC.

4.6 Conclusion

In this chapter, we introduce the problem of exercise retrieval for learner directed language learning and highlight an important challenge in this setting: how learners express what they want to learn and exercise content are fundamentally semantically different. The effects of this referential similarity gap are especially pronounced when attempting to retrieve exercises via direct similarity search: even models supervised on MS MARCO, a large scale retrieval dataset, struggle to bridge this referential similarity gap. As a result, we propose mHyER, a zero-shot retrieval approach that leverages the generative capabilities of pretrained LLMs to synthesize relevant in-distribution sentences which are then used to retrieve exercises. We form two novel benchmark datasets by collecting human responses and processing publicly available data. mHyER outperforms several strong baselines, including ones trained in a supervised fashion.

Future work. mHyER lays the methodological foundation for self-directed online language learning. Many interesting directions of future work exist, ranging from investigating different learning areas to methodological extensions that accommodate labeled relevance information. We discuss several of these directions below.

mHyER provides concrete methodology that can enable future investigations into the effects of self-directed learning on long-term learning outcomes and curriculum design at scale. Self-directed learning also play a role in improving other components of personalized education systems. For example, a learner input into a self-directed learning system can be viewed as an indicator of a self-perceived weakness, which would provide a powerful form of supervision for estimating user skill levels. Studying how inputs and outputs of complementary parts of a unified personalized education system is an important direction of future work.

Another interesting avenue of future work is investigating if analogous “language about language” phenomena appear in settings other than language learning. We hypothesize that such phenomena exist in one form or another across all learning settings. For example, how learners describe what they want to review in math (e.g., “right angles”) exhibits a similar fundamental mismatch with exercise text (e.g., “Compute the length of the hypotenuse of this triangle”). If such gaps exist, methods capable of bridging the referential similarity gap, like mHyER, will be required across different learning settings. Characterizing the degree to which such gaps appear, as well as how such gaps differ, in different learning settings remains important and open work.

From a system design and learner experience perspective, developing machine learning methods to retrieve relevant exercises based on learner inputs is a foundational piece of any self-directed language learning system. However, serving a set of exercises that maximizes relevance may not lead to the best learner experience. Instead, the objective of exercise retrieval can be made more flexible: Instead of retrieving K exercises that maximize relevance, we retrieve all exercises with scores that exceed some pre-determined threshold. Then, this set of relevant exercises can be re-ranked based on additional criteria, such as difficulty level (with information from Knowledge Tracing-based parts of the system) or diversity (in terms of difficulty or length). Regardless of precise objective (top K vs. all relevant exercises), the referential similarity gap persists, making mHyER especially suitable for this initial retrieval step.

Methodologically, mHyER was designed explicitly with the goal of zero-shot retrieval. However, opportunities to collect learner relevance feedback grow as self-directed learning systems get implemented. Such feedback can then be used to train retrieval methods. Investigating how to effectively use limited learner feedback to help retrieval methods bridge the referential similarity gap remains an open question. Additionally, extensions of mHyER to learning settings with multi-modal exercises is direction of future work. Using newly developed multi-modal models to measure similarity in different domains, such as images or audio, can unlock a richer learning experience for learners.

CHAPTER 5
REMOVING HUMANS FROM THE LOOP: LABELED DATASET GENERATION WITH NO ADDITIONAL HUMAN ANNOTATIONS

In this chapter[1], we introduce the HandsOff framework, a technique capable of producing an unlimited number of synthetic images and corresponding labels after being trained on less than 50 pre-existing labeled images. Recent work leverages the expressive power of generative adversarial networks (GANs) to generate labeled synthetic datasets. These dataset generation methods often require new annotations of synthetic images, which forces practitioners to seek out annotators, curate a set of synthetic images, and ensure the quality of generated labels. Our framework avoids the practical drawbacks of prior work by unifying the field of GAN inversion with dataset generation. We generate datasets with rich pixel-wise labels in multiple challenging domains such as faces, cars, full-body human poses, and urban driving scenes. Our method achieves state-of-the-art performance in semantic segmentation, keypoint detection, and depth estimation compared to prior dataset generation approaches and transfer learning baselines. We additionally showcase its ability to address broad challenges in model development which stem from fixed, hand-annotated datasets, such as the long-tail problem in semantic segmentation.

5.1 Introduction

The strong empirical performance of machine learning (ML) models has been enabled, in large part, by vast quantities of labeled data. The traditional machine learning paradigm, where models are trained with large amounts of *human labeled* data, is typically bottlenecked by the significant monetary, time, and infrastructure investments needed to obtain said labels. This problem is further exacerbated when the data itself is difficult to collect. For example,

[1] The work in this chapter appears in [4]

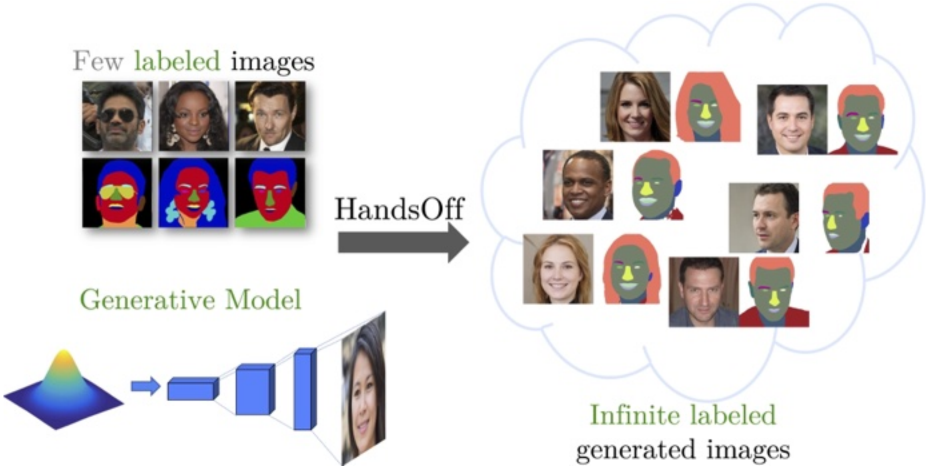


Figure 5.1: The HandsOff framework uses a small number of existing labeled images and a generative model to produce **infinitely** many labeled images.

collecting images of urban driving scenes requires physical car infrastructure, human drivers, and compliance with relevant government regulations.

Finally, collecting real labeled data can often lead to imbalanced datasets that are unrepresentative of the overall data distribution. For example, in *long-tail settings*, the data used to train a model often does not contain rare, yet crucial edge cases [165].

These limitations make collecting ever increasing amounts of hand labeled data unsustainable. We advocate for a shift away from the standard paradigm towards a world where training data comes from an *infinite collection* of automatically generated labeled images. Such a dataset generation approach can allow ML practitioners to *synthesize* datasets in a *controlled* manner, unlocking new model development paradigms such as controlling the quality of generated labels and mitigating the long-tail problem.

In this work, we propose HandsOff, a generative adversarial network (GAN) based dataset generation framework. HandsOff is trained on a small number of *existing* labeled images and capable of producing an infinite set of synthetic images with corresponding labels (Figure 5.1).

To do so, we unify concepts from two disparate fields: dataset generation and GAN inversion. While the former channels the expressive power of GANs to dream new ideas in the form of images, the latter connects those dreams to the knowledge captured in annotations. In this way, our work brings together what it means to dream and what it means to know. Concretely, this chapter makes the following contributions:

1. We propose a novel dataset generating framework, called HandsOff, which unifies the fields of dataset generation and GAN inversion. While prior methods for dataset generation [166] require new human annotations on synthetically generated images, HandsOff uses GAN inversion to train on existing labeled datasets, eliminating the need for human annotations. With ≤ 50 real labeled images, HandsOff is capable of producing high quality image-label pairs (Section 5.3).

2. We demonstrate the HandsOff framework's ability to generate semantic segmentation masks, keypoint heatmaps, and depth maps across several challenging domains (faces, cars, full body fashion poses, and urban driving scenes) by evaluating performance of a downstream task trained on our synthetic data (Section 5.4.2, Section 5.4.3, and Section 5.4.4).

3. We show that HandsOff is capable of mitigating the effects of the long-tail in semantic segmentation tasks. By modifying the distribution of the training data, HandsOff is capable of producing datasets that, when used to train a downstream task, dramatically improve performance in detecting long-tail parts (Section 5.4.5).

5.2 Related work

Our work is built on GANs [43], which consist of a generator that synthesizes new images, and a discriminator that discerns between real and generated images. Recent advances in GANs [51, 46, 47, 48, 167, 49] have demonstrated an ability to generate highly realistic images in numerous domains. We utilize the popular StyleGAN2 architecture [48], which synthesizes images by passing randomly sampled inputs through a series of *style blocks*. Remarkably, StyleGAN2's $\mathcal{W}$ and $\mathcal{W}+$ latent spaces form rich representations of images in

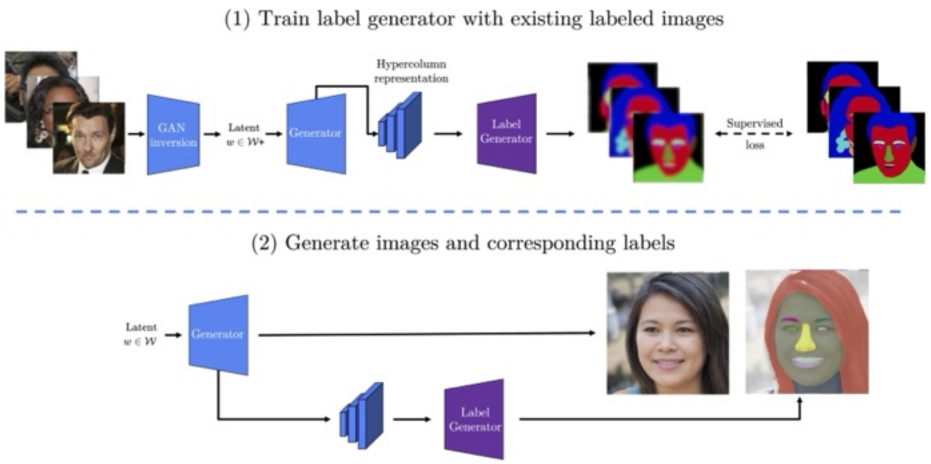

(1) Train label generator with existing labeled images
GAN inversion
Latent $w \in \mathcal{W}+$
Generator
Hypercolumn representation
Label Generator
Supervised loss
(2) Generate images and corresponding labels
Latent $w \in \mathcal{W}$
Generator
Label Generator

Figure 5.2: The HandsOff framework. (Top) GAN inversion is used to obtain training image latent codes, which are then used to form hypercolumn representations. The label generator is then trained with the hypercolumn representations and original labels. (Bottom) To generate datasets, the trained label generator is used in conjunction with a StyleGAN2 generator to produce image-label pairs.

a disentangled manner [50, 168, 169, 170], which can be utilized to edit complex semantic attributes in generated images [168, 171, 101, 172, 173, 174]. The ability to identify semantically meaningful parts of generated images in the latent representation suggests that it could be used to generate pixel-level labels. This capability, coupled with GANs' ability to generate troves of high quality images, serves as the basis for generating synthetic image *datasets* [166, 175, 176, 177].

We build upon DatasetGAN [166], which trains a label generator using representations of an image formed from the GAN latent code. DatasetGAN requires *human annotation of GAN generated images*, which burdens a practitioner to seek out annotations for every new domain of interest. In addition to labeling, users also must actively *curate* images to label to ensure diverse semantic feature coverage and avoid GAN created artifacts. Furthermore, should the labeling scheme change and render the original labels obsolete, then additional annotations are again required. Acquiring additional labels is especially contrived when a large of number of quality human annotated images already exist. A framework that leverages these *real* preexisting labeled images would circumvent all of these drawbacks. EditGAN [172], a follow-on contribution to DatasetGAN, utilizes encoder-based reconstructions to perform image editing. BigDatasetGAN[176] exploits the pre-trained encoder of VQGAN[178] to utilize existing labeled *synthetic* images. In contrast, our approach links latents of labeled *real* images to their labels by employing GAN inversion, the process of mapping a real image to the latent space of a GAN.

The myriad of inversion techniques range from encoder-based approaches [179, 173, 180, 181], which utilize trained encoders to map images directly to the latent space, to optimization-based approaches [182, 169, 170], which directly optimize a similarity loss (e.g., LPIPS [183]) to obtain latents. Some methods modify generator weights to increase image reconstruction quality [184, 174, 170]. Our work exclusively uses inversion methods that do not modify the generator, since the generator must remain unperturbed to generate new images from the original data distribution. We invert images to the $\mathcal{W}+$ space, which is more expressive than the $\mathcal{W}$ space and leads to higher quality reconstructions [50].

5.3 The HandsOff framework

The HandsOff framework, shown in Figure 5.2, consists of three main components: (1) a generator (realized as a GAN), which maps a latent code $w \in \mathcal{W}$ to an image X, (2) an inverter, which maps an image X to a latent code w, and (3) a label generator, which maps a latent code w to a pixel-wise *label* Y, such as a semantic segmentation mask. HandsOff exploits the fact that the generator's latent space forms a rich, disentangled representation of images. Since these latent spaces already encode semantically meaningful concepts from images [168, 169, 170], we aim to train a 'label generator' that maps latents in this space to *labels*.

Unfortunately, training this label generator requires paired data of latents w with labels Y. One approach, espoused by prior work [166], could be to map the latent w to an image X, and ask annotators to manually label the image. However, in many applications, paired data of (X, Y) is readily available, thanks to the careful efforts of dataset collectors. Our key insight is that *existing labeled image datasets can be used to train a label generator on*

GAN latent spaces, using techniques from the GAN inversion literature. Below, we describe our specific approach for GAN inversion (Section 5.3.1), our representation of the GAN's latent space (Section 5.3.2), and finally, our label generator (Section 5.3.3).

5.3.1 GAN inversion

The key step in the HandsOff framework is to connect advances in GAN inversion to dataset generation. GAN inversion allows us to use a small number of pre-existing labeled images to create a dataset of labeled *latents*. Our use of pre-existing labels allows practitioners to re-purpose existing labeled datasets, avoiding the cost of acquiring labels, including the prerequisite of maintaining annotation workstreams in their machine learning pipelines.

Our GAN inversion is inspired by popular approaches in the image-editing community [172, 185]. Given a pre-trained generator G, we first train an encoder to predict a latent $w^{(e)}$ from an input image X. In practice, this feed-forward encoder results in a good initial inversion of an image to a latent input. To refine this initial estimate further, we solve the following regularized optimization problem:

$$\min_{w:\|w-w^{(e)}\|_2^2\leq c_{reg}} \mathcal{L}_{LPIPS}(X, G(w))+\lambda_{\ell_2}\|X - G(w)\|_2^2$$

where $\mathcal{L}_{LPIPS}$ is the Learned Perceptual Image Patch Similarity (LPIPS) loss [183]. Although this problem is highly non-convex, in practice we find that using a fixed number of gradient descent iterations significantly refines the latent code. This refinement step requires additional inference time, but this additional cost is incurred only once on a small number of training images. In our experiments, we utilize ReStyle [173] as the encoder, but we emphasize that our framework is amenable to *any* GAN inversion procedure that does not modify the generator weights. Note that common approaches for GAN inversion fine-tune the *generator* in order to achieve a better *inversion* for a specific image [184, 174, 170]. To ensure our generator can produce *new* images from the task domain, we keep the generator parameters frozen throughout the inversion process.

5.3.2 Hypercolumn representation

GAN inversion allows us to map images X to latent codes w. We could use these latent codes directly to train a label generator that maps latent codes w to labels Y. However, this discards the rich representations encoded by the intermediate layers within the generator. Rather than training on w directly, we construct a hypercolumn representation $S^{\uparrow}$ from the generator's intermediate layers. Specifically, we use a StyleGAN2 generator, where the latent code w is used to modulate convolution weights in intermediate style blocks, which progressively grow an input to the final output image. For a 1024 $\times$ 1024 resolution image, there are $L = 18$ style blocks. We utilize the approach of [166] and take the intermediate output of these style blocks, upsample them channel-wise to the resolution of the full image, then concatentate each upsampled intermediate output channel-wise to obtain pixel-wise hypercolumns. Our final hypercolumn representation is denoted by $S^{\uparrow}$, with each pixel j now having a hypercolumn $S^{\uparrow}[j]$ of dimension C. Due to the high dimensionality of the hypercolumns ($C = 6080$ for 1024×1024 images), we cap the generated image resolution to 512 $\times$ 512, and downsample intermediate outputs from higher resolutions.

5.3.3 Label generator

The label generator exploits the semantically rich latent space of the generator to efficiently produce high quality labels for generated images. Because the latent codes already map to semantically meaningful parts of generated images, simple, efficient models suffice for generating labels. Specifically, like in [166], we utilize an ensemble of M multilayer perceptrons (MLPs). The MLPs operate on a pixel-level, mapping a pixel's hypercolumn to a label. To generate a label for a synthetic image, we pass the hypercolumn formed by latent code w through the M MLPs, and aggregate the outputs (via majority vote or averaging) to produce a label. The M MLPs are trained using a small number ($\sim$50) of

Table 5.1: Downstream task performance for semantic segmentation tasks across various domains, reported in mIOU (↑). HandsOff outperforms all baselines across all domains with both 16 and 50 labeled training images. × indicates a method that could not be run for a particular domain due to methodological shortcomings, such as requiring additional hand-labeled data.

	# labeled images	CelebAMask-HQ 8 classes	Car-Parts 10 train	DeepFashion-MM 8 classes	DeepFashion-MM 10 classes	Cityscapes 8 classes
DatasetGAN	16	0.7013	×	×	×	×
EditGAN	16	0.7244	0.6023	×	×	×
Transfer Learning	16	0.4575	0.3232	0.5192	0.4564	0.4954
HandsOff (Ours)	16	**0.7814**	**0.6222**	**0.6094**	**0.4989**	**0.5510**
Transfer Learning	50	0.6197	0.4802	0.6213	0.5559	0.5745
HandsOff (Ours)	50	**0.7859**	**0.6679**	**0.6840**	**0.5565**	**0.6047**

Table 5.2: Downstream task performance for keypoint detection and depth estimation. HandsOff outperforms all other methods when trained on 16 or 50 labeled images, demonstrating an impressive ability in generating *continuous*-valued keypoint heatmaps and depth maps.

	# labeled	CelebAMask-HQ			DeepFashion-MM			Cityscapes-Depth		
	images	PCK-0.1 ↑	PCK-0.05 ↑	PCK-0.02 ↑	PCK-0.1 ↑	PCK-0.05 ↑	PCK-0.02 ↑	mNMSE ↓	RMSE ↓	RMSE-log ↓
Transfer Learning	16	78.96	42.06	7.32	91.24	83.52	48.21	0.4022	18.12	2.75
HandsOff (Ours)	16	**97.19**	**76.36**	**17.44**	**94.19**	**88.48**	**70.22**	**0.2553**	**14.52**	**1.64**
Transfer Learning	50	90.88	61.75	12.30	91.24	83.52	48.20	0.2525	15.07	3.01
HandsOff (Ours)	50	**97.71**	**79.99**	**19.10**	**95.41**	**90.89**	**74.02**	**0.1967**	**13.01**	**1.58**

pre-existing labeled images with a cross-entropy loss for generating discrete labels (e.g., segmentation masks) and mean-squared error loss for generating continuous labels (e.g., keypoint heatmaps).

Our use of an ensemble of MLPs naturally provides a way to filter out potentially poor labels by using the prediction uncertainty as a proxy for label quality. For discrete labels, we can utilize Jensen-Shannon divergence [186, 187, 188, 166] across the M MLPs to produce pixel-wise uncertainty maps. For predicting continuous labels, we compute the pixel-wise variance across the MLP outputs. In both cases, the overall image uncertainty is computed by summing across all pixels.

5.4 Experimental results

We extensively evaluate HandsOff in generating both discrete (segmentation masks) and continuous (keypoint heatmaps and depth) labels across four challenging domains: Faces, Cars, Full-Body Human Poses, and Urban Driving Scenes. We utilize various pre-trained StyleGAN2 generators [48, 189, 190] and ReStyle inverters [173]. To train the label generator, we utilize existing labels from CelebAMask-HQ [191], Car-Parts [192], DeepFashion-MultiModal [193, 194], and Cityscapes [195]. The key assumption of HandsOff is that GAN inverted image reconstructions align well with the original labels. We present visualizations of reconstructed image alignment in Section C.3.1. Our label generator architecture across all domains and tasks is an $M = 10$ ensemble of 2-hidden layer MLPs. This simple architecture is a distinct strength of the HandsOff framework: intensive parameter and architecture finetuning are not necessary to achieve state-of-the-art empirical performance. For the label generator, we provide training details in Section C.1.3, architecture details in Section C.1.5, and ablations in Section C.2.

5.4.1 Experimental set-up

Downstream network. In all domains and tasks, we utilize DeepLabV3 with a ResNet151 backbone as our downstream network. We generate 10,000 synthetic images and labels, filter out the top 10% most uncertain images (see Section 5.3.3), and train our downstream network for 20 epochs with the 9,000 remaining images. For segmentation, we have DeepLabV3 output a probability distribution over all of the parts for each pixel, whereas for keypoints or depth, we have DeepLabV3 output continuous values. Due to the dynamic nature of elements in the Cityscapes dataset, slight imperfections in the reconstructions uniquely affect segmentation mask alignment. To mitigate this, we perform an extra fine tuning step with the original 16 or 50 labeled examples used to train the label generator while training for semantic segmentation. Training details for the downstream network can be found in Section C.1.3 and ablations can be found in Section C.2.

Baselines. We compare HandsOff against three baselines: DatasetGAN, EditGAN, and Transfer Learning. We are only able to evaluate DatasetGAN in the face domain, as

DatasetGAN is unable to accommodate the change in labeling scheme from their custom labeled car dataset to the larger Car-Parts-Segmentation dataset, thus highlighting another drawback of requiring GAN labeled images. For EditGAN, we adopt the image editing framework to synthesize labels for images. However, we are unable to test in the full-body human poses and urban driving scene domains, as EditGAN has only released checkpoints for the face and car domains. For the Transfer Learning baseline, we initialize DeepLabV3 with pretrained weights on ImageNet, then finetune the classification head of the model on the 16 or 50 labeled images used to train HandsOff until convergence. This baseline is used to benchmark our method, which is trained on up to 50 labeled images, against a model that is trained on 100,000+ *labeled* out-of-domain images in addition to the 16 or 50 labeled in-domain images.

Datasets. For faces, we split CelebAMask-HQ into a set of 50 training, 450 validation, and 29,500 testing images. We collapse the 19 original segmentation classes into 8 and scale the keypoint locations in the low resolution version of images found in CelebA to the full resolution images. For cars, we retain the original 400 image train set, split the test set into a set of 20 images for validation and 80 images for testing, and collapse the 19 original classes into 10. For full-body human poses, we split DeepFashion-MultiModal into a set of 200 training, 500 validation, and 12,000 testing images. We collapse the 24 original segmentation classes into 8 and 10 classes and retain the original 21 labeled keypoint locations. For Cityscapes, because the ground truth test labels are not released, we split 300 and 1275 images from the original train set for validation and test, respectively. We utilize the eight groups (e.g., human, vehicle, etc) as our class labels. Note that while our train sets may contain more than 50 images, we use *at most* 50 labeled images from the train sets to train HandsOff in each domain. Details on class collapse can be found in Section C.1.2.

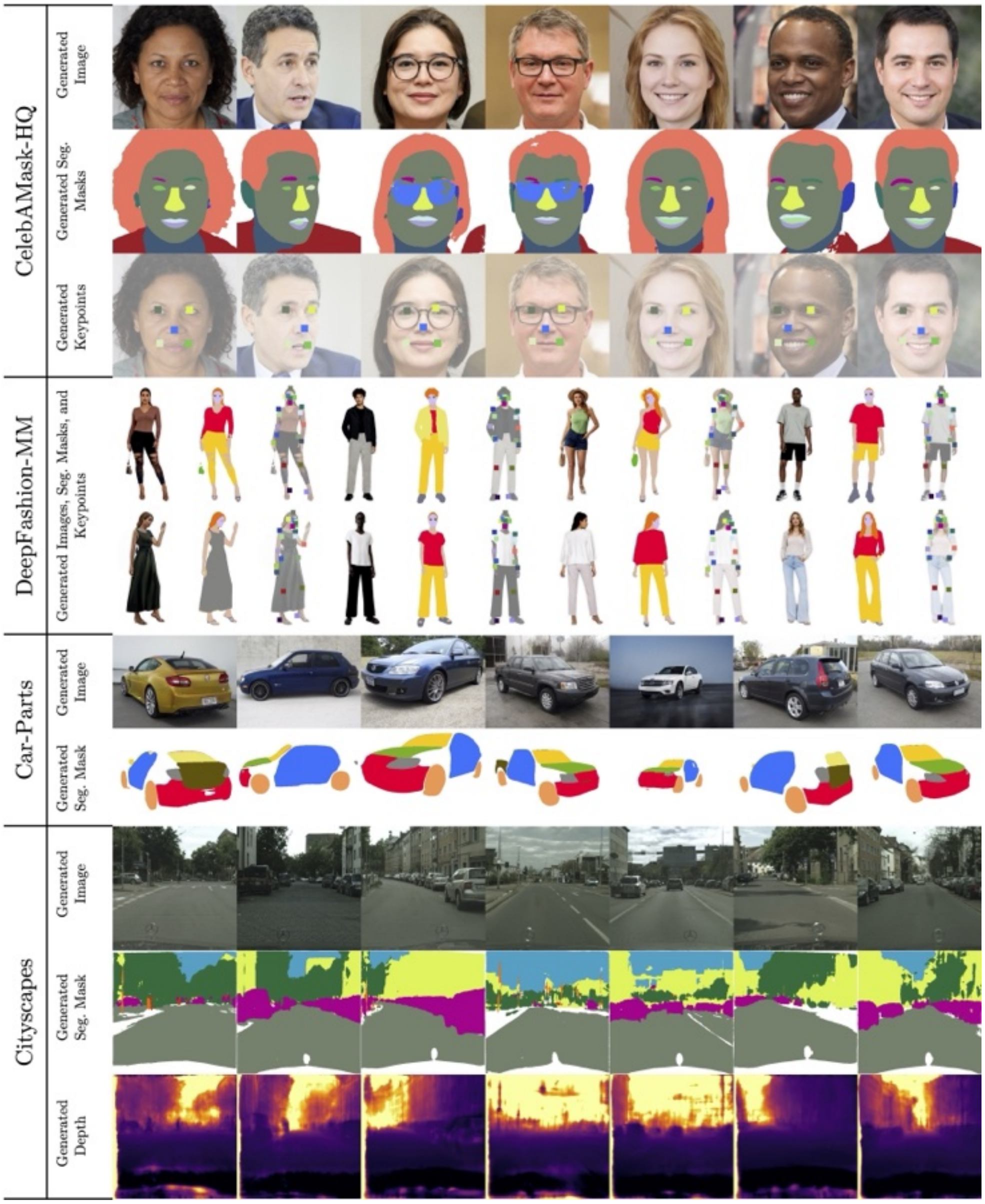


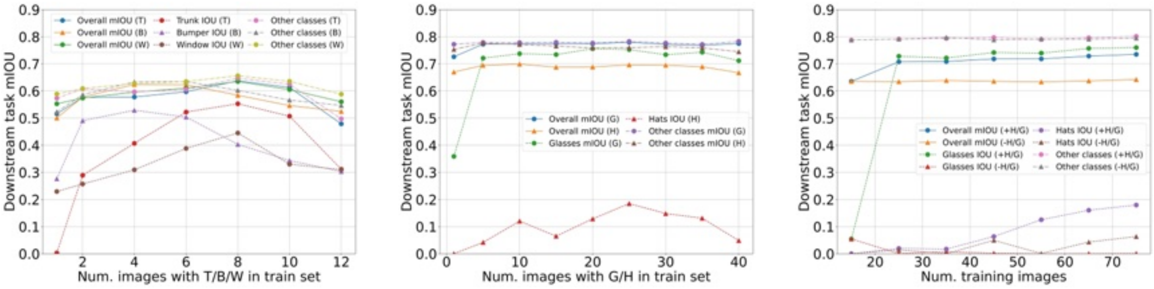
Figure 5.3: Examples of HandsOff generated labels (segmentation masks, keypoints, and depth) across four different domains. Generated labels capture fine details across various object orientations (CelebAMask-HQ, Car-Parts), object poses (DeepFashion-MM), and lighting conditions (Cityscapes). Note that HandsOff correctly assigns the label “skin” to the visible parts of the leg in the ripped areas of jeans (DeepFashion-MM, first row, first human) and correctly assigns the labels “jacket” and “shirt”, despite the fact that the jacket and shirt are almost indistinguishable color-wise (DeepFashion-MM, first row, second human). Furthermore, generated keypoints are accurate despite partial occlusion, such as eyes behind glasses (CelebAMask-HQ, third and fourth image) or feet covered by long pants (DeepFashion-MM, second row, last human). HandsOff is also capable of identifying spatially small objects, such as street signs (Cityscapes, first, third, and fourth image).

5.4.2 HandsOff generated datasets

We visualize the generated image-label pairs from HandsOff in Figure 5.3. HandsOff is capable of generating very high quality labels across all domains. In the face domain, HandsOff is capable of producing segmentation masks that can correctly distinguish left/right features like eyes or ears and identify rare occurring classes such as glasses. Furthermore, it produces extremely accurate keypoint locations even when such locations may be partially occluded. Within the full-body human pose domain, HandsOff produces finely detailed segmentation masks, best illustrated by the segmentation mask for the first human in the top row of Figure 5.3, who is wearing a pair of ripped jeans and the second human in the top row who is wearing the same colored jacket and shirt (see caption for more details). Generated labels are consistently high quality across a diverse array of object orientations, as seen in the various face rotations, human poses, or car orientations of Figure 5.3. Finally, in extremely complex scenes, such as Cityscapes, HandsOff produces labels for visually minuscule classes, such as street lamps or traffic signs. Additional examples of generated labels can be found in Section C.3.3.

5.4.3 Segmentation results

As seen in Table 5.1, we achieve **state-of-the-art performance** on synthetic data trained semantic segmentation in all four domains, as measured in mean Intersection-over-Union (mIOU). Specifically, HandsOff outperforms DatasetGAN by 11.4% and EditGAN by 7.9% in the face domain when trained with *the same number of labeled images*. Increasing the number of labeled training images for HandsOff results in further performance gains, with 12.1% and 8.5% improvements over DatasetGAN and EditGAN, respectively. Unlike DatasetGAN, we are able to increase the number of labeled training images without incurring the associated costs of collecting new human annotated images. We emphasize again that with new domains, such as full-body human poses or urban driving scenes, it is not possible to train DatasetGAN-based frameworks as they rely on manual labels for GAN generated



(a) (b) (c)

Figure 5.4: Substitution experiments for various long-tail parts; (a) in cars - trunk (T), back bumper (B), back window (W); (b) in faces - glasses (G), hats (H). As the proportion of images containing the long-tail part increases in the training set, the performance of the long-tail class improves until it enters the *overfitting* regime. Non-long-tail mIOU tracks closely with overall IOU, implying dramatic gains in long-tail IOU do not come at the expense of other parts. (c) Addition experiments for face long-tail parts. (+H/G) indicates that images containing hats and images containing glasses are added to a base set, while (-H/G) indicates images containing neither hats nor glasses are added. The long-tail IOU of both parts *simultaneously* increase as images containing hats and images containing glasses are added to the base training set, with no negative impact on the performance of other classes.

images. Therefore, we benchmark against the transfer learning baseline in these domains. Notably, HandsOff outperforms the transfer learning baseline by 17.4% (full-body human poses) and 11.2% (urban driving scenes) when both methods are trained on 16 labeled images; and 10.1% (full-body human poses) and 5.3% (urban driving scenes) when trained on 50 images.

5.4.4 Keypoint and depth results

We utilize HandsOff to generate *continuous* valued labels for keypoints and depth tasks. As seen in Table 5.2, we demonstrate strong empirical performance in generating both keypoints and depth maps. To synthesize keypoints, we utilize the keypoint heatmap regression frramework, where our label generator is asked to output a continuous-valued spatial heatmap for each keypoint. See Section C.1.6 for a detailed explanation of keypoint regression. For downstream task performance, we report the Percentage of Correct Keypoints (PCK) for different threshold values α, denoted PCK-α. For a keypoint to be predicted correctly, the estimate must be no further from the true keypoint than $\alpha \cdot \max\{h, w\}$, where h and w are the height and width of the minimum size bounding box that contains all of the keypoints. We note that even for small α (i.e., $\alpha = 0.02$), HandsOff is able to correctly predict 2.4× and $1.5\times$ more keypoints than the transfer learning baseline in the face and full-body human pose domains, respectively. This implies that HandsOff is able to predict keypoints up to an extremely tight radius of the original keypoint location compared to other methods.

For depth, we report masked normalized mean-squared error (mNMSE), root mean-squared error (RMSE), and root mean-squared error of the log-depth values (RMSE-log). Because Cityscapes depth maps contain corrupted depth values, we train HandsOff only non-corrupted pixels. Furthermore, to compute mNMSE, we compute the normalized mean-squared error only on the non-corrupted pixels. That is, let $\widehat{\boldsymbol{y}}$ and $\boldsymbol{y}$ are the predicted and true depth maps, respectively, and $\boldsymbol{M}$ be a mask indicating the non-corrupted elements of $\boldsymbol{y}$. mNMSE is computed as $\frac{\|\widehat{\boldsymbol{y}}_{\boldsymbol{M}} - \boldsymbol{y}_{\boldsymbol{M}}\|_2^2}{\|\boldsymbol{y}_{\boldsymbol{M}}\|_2^2}$, where $\boldsymbol{a}_{\boldsymbol{M}}$ denotes the depth map $\boldsymbol{a}$ at non-corrupted locations. When reporting RMSE and RMSE-log, we adopt the standard practice [196, 197] in depth estimation of cropping the middle 50% of the image and clamping predicted depth values to be within 0.001 and 80 before computing RMSE and RMSE-log values. As shown in Table 5.2, HandsOff is able to achieve a sizable advantage in all three metrics, outperforming transfer learning, resulting in 36.5%, 19.9%, and 40.27% decreases in mNMSE, RMSE, and RMSE-log when trained on 16 labeled images and 22.1%, 13.6%, and 47.6% decreases when trained on 50 labeled images.

5.4.5 Long-tail semantic segmentation

The HandsOff framework's ability to generate high quality synthetic datasets unlocks new degrees of freedom for model development previously unachievable with fixed, hand-annotated datasets. We now explore one example: mitigating the effects of the long-tail common in semantic segmentation datasets. For CelebAMask-HQ, images with hats and

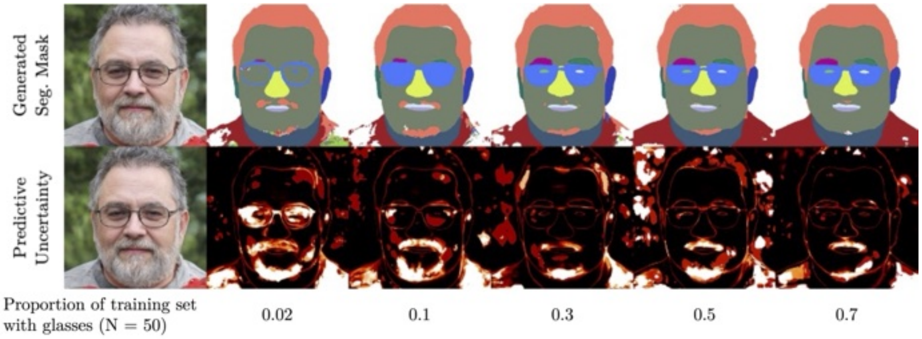


Figure 5.5: Visualization of generated segmentation mask (top row) and pixel-wise label generator uncertainty (bottom row) as the proportion of the training set containing the glasses increases. Not only do we see qualitative improvement in the generated label for glasses, we also see that the classifier is less *uncertain* when generating the correct label.

glasses make up less than 5% of the 30,000 labeled images, and a similar situation exists with trunks, back bumpers, and back windows in the Car-Parts dataset. These examples form the long-tail classes of their respective datasets, and their rare occurrence during training results in poor model performance at evaluation time.

The HandsOff framework altogether sidesteps this limitation of traditional datasets: by generating labeled synthetic images, we can control the occurrence of rare classes in our training data and significantly mitigate the effects of the long-tail. Because training the label generator requires less than 50 annotated images, we only require 5-10 occurrences of long-tail classes in order to generate an unlimited number of those occurrences in our synthetic dataset. Our experiments precisely quantify the small number of annotated examples of rare classes required to significantly improve downstream task performance on those classes. They fall into two categories: **Substitution** experiments, that fix a total number of training images and vary the proportion of rare class occurrences, and **Addition** experiments, that grow the size of the training set by adding images with rare classes. The substitution experiments ensure that any gains in the performance of identifying the long-tail class are not a by-product of increasing training set size. We perform substitution experiments considering only one long-tail part at a time. On the other hand, the addition setting is indicative of how a practitioner would deploy HandsOff: starting with a base set of labeled training images and further augmenting it with images containing rare classes deemed crucial to identify. To mirror what often happens in practice, we perform addition experiments by adding images containing multiple long-tail classes at a time.

Substitution. We begin with an initial set of 16 (cars) or 50 (faces) labeled images containing one image of the rare part, and then vary the proportion of the rare part. As seen in Figure 5.4a and Figure 5.4b, a small proportion of rare classes results in poor class identification performance, but as the proportion of images with long-tail classes increases, the long-tail part IOU increases by as much as 0.55 for car trunks and 0.40 for face glasses before eventually plateauing. We note that hats are a particularly challenging part to generate labels for due to the diversity of their size, shape, color, and orientation. Nevertheless, we still see a sizable increase of 0.2 IOU. We additionally plot the overall mIOU and the mIOU of non-long-tail parts to demonstrate that modifying the composition of the training set does not hurt performance on non-long-tail parts. In other words, shifting the training set part distribution to an extent has negligible impacts on the performance of non-long-tail parts, while resulting in large gains in long-tail class detection. Beyond proportions of $\sim$0.7, further increasing the proportion of the training set eventually causes drops in both long-tail part IOU and the mIOU of non-long-tail parts, owing to the label generator hallucinating long-tail classes where they do not belong. The impacts of substituting images with long-tail classes are best illustrated in Figure 5.5. As the proportion of images with glasses grows, the generated mask captures glasses with increasing accuracy, eventually even distinguishing eyes that are visible through the glasses. Underneath the segmentation masks, we showcase the pixel-wise label generator uncertainty measured by Jensen-Shannon divergence (See Section 5.3.3). Not only does the generated label improve qualitatively, the label generator is *less uncertain* about the region of the image corresponding to the glasses. Additional visual examples of both segmentation mask and label generator uncertainty can be found in Section C.3.4.

Addition. We augment a small training set of 15 images with additional images containing hats or glasses. Figure 5.4c demonstrates significant IOU increases (+0.71) in long-tail classes. The figure further highlights that these increases are not simply due to additional examples: targeted additions outperform the scenario where we add the same number of images, but the added images do **not** contain hats or glasses. These improvements in long-tail classes do not come at the expense of performance in other classes, as demonstrated by the overall mIOU and mIOU of non-long-tail classes. Unlike the substitution experiments, these performance improvements do not eventually drop, since the number of training examples continues to increase.

Our experiments showcase the power of the HandsOff framework to mitigate the long-tail problem. By explicitly including images with the long-tail class in our label generator training data, we are able to bridge the gap between performance in rare and common classes. The number of images with long-tail classes necessary to generate high quality labels of the long-tail is even smaller than the already small number of images needed to train HandsOff, meaning that the gains in long-tail class performance essentially come for free. If the long-tail class has been deemed crucial to identify, then it is likely that a practitioner has access to $\sim$20 labeled images containing the long-tail class. The performance gains in long-tail performance achieved by HandsOff are not practically replicable in DatasetGAN, where human supervision is needed to both identify generated images containing the long-tail class and provide precise pixel-level annotations.

5.5 Conclusion

We present the HandsOff framework, which produces high quality labeled synthetic datasets without requiring further annotation of images for a multitude of tasks across various challenging domains. HandsOff achieves state-of-the-art performance over several recent baselines when training a downstream network with our synthetically generated data. Furthermore, HandsOff enables user control of the training data composition, leading to dramatic performance gains in long-tail semantic segmentation. This suggests that HandsOff can play a vital role in curtailing the effects of the long-tail. While synthetic datasets have the potential to supplant human annotations, they can also complement them. We leave as future work to investigate the collaborative power of having a human-in-the-loop refine synthetically generated annotations, and bring about the best of both worlds.

CHAPTER 6
CONCLUSION AND FUTURE WORK

Machine learning models are improving at a breakneck pace, driven by new novel architectures, training procedures, and a wealth of available training data. This thesis examines the crucial role humans play in the development of these intelligent machine counterparts. In the development of large models, human feedback has proven to be an invaluable source in guiding models towards more desirable outputs. As such, collecting human feedback on model outputs is a large-scale ongoing process, necessitating heavy investment in terms of time, money, and infrastructure. Part I of this thesis investigates two methods of easing this data collection burden, from stretching the effectiveness of responses we may already have (Chapter 2) to designing new ways of eliciting expressive responses (Chapter 3).

The advent and availability of powerful models opens an exciting orthogonal avenue of research: How we can leverage existing tools to *avoid* the burden of human feedback collection. Part II of this thesis examines two specific application settings where collecting human feedback can be avoided with contemporary pretrained models. In Chapter 4, we consider an information retrieval setting, and show that the generative of ability of language models can generate relevant content for user queries, greatly improving retrieval performance in the cold-start regime. In Chapter 5, we successfully utilize image editting models to remove humans annotators from a popular synthetic dataset generation framework. In all, the work in this thesis results in several exciting future directions of research.

The work in Chapter 2 has already yielded a line of follow-up working investigating crowd-based preference and metric learning from paired comparisons [100, 198], which proves rigorous statistical guarantees. Mahalanobis metrics, while more expressive than the standard Euclidean metric, allow only for first order feature interactions. Whether this adequately captures the intricacies of human preference judgements remains an open question. Indeed, recent work has begun investigating learning more complex models of human preference, such as distributions [199, 200].

The work in Chapter 3 demonstrates the expressive power of PAQs in a controlled empirical setting. Applying such queries to real-world perception settings is a natural direction of future work. Additionally, the structure of a PAQ makes the image domain a natural setting for application. Adapting the PAQ to more discrete settings, such as natural language, is an open line of work. Statistically, the inverted measurement paradigm presented in this chapter introduces several open statistical problems. Whether or not the derived cube-root estimation rate is fundamental (i.e., minimax optimal) is an interesting avenue of future work. If not, design of improved statistical estimators under this measurement model is an exciting future direction.

The work in Chapter 4 examines retrieval in the specific context of online language learning, a modality that is largely text. A promising line of future work is extending the developed retrieval approaches to multi-modal application areas, such as geometry or natural sciences, by utilizing multi-modal generative models. More generally, because retrieval itself has been integrated into natural language generation, a process known as retrieval-augmented generation (RAG) approaches, it would interesting to explore how generative models can help themselves in retrieval. Instead of utilizing direct search, could language models conditionally generate content based on user prompts, then utilize such content to search for external sources to correct or augment model responses?

The work in Chapter 5 develops new methodology for utilizing real, existing images and annotations for GAN-based synthetic dataset generation. Extensions of HandsOff have already been implemented with diffusion models [201, 202], enabling exciting avenues of future exploration, such as open-vocabulary dataset generation. These methods operate similarly to HandsOff, where a pretrained image generation model is fixed, and a label generator is learned on top of existing representations. An interesting line of future exploration is the training of generative models and label generators entirely from scratch. Could teach a model to jointly synthesize images and labels lead to better learned representations, and therefore higher quality images?

The partnership between humans and machines is ever-evolving, with much emphasis placed on the development of machines. Partnerships, however, require collaboration, meaning humans still have a crucial role to play.

Appendices

APPENDIX A
SUPPLEMENTARY MATERIAL FOR CHAPTER 2

A.1 Proof of Proposition 1

Proof. Let $\boldsymbol{w} \in \mathbb{R}^d$ be arbitrary. Note that for any point $\boldsymbol{x} \in \mathbb{R}^d$, one can easily show that

$$\|\boldsymbol{x} - \boldsymbol{u}\|_{\boldsymbol{\Sigma}^\star}^2 = \|\boldsymbol{x} - \boldsymbol{w}\|_{\boldsymbol{\Sigma}^\star}^2 \tag{A.1}$$

if and only if

$$\langle 2\boldsymbol{x} - \boldsymbol{u} - \boldsymbol{w}, \boldsymbol{\Sigma}^\star(\boldsymbol{u} - \boldsymbol{w})\rangle = 0. \tag{A.2}$$

This follows simply by expanding the expressions on both sides of (Equation A.1) and rearranging the terms to obtain (Equation A.2).

We now show that if $\boldsymbol{u}$ is identifiable then $\boldsymbol{\Sigma}^\star$ is strictly positive definite. Suppose for the sake of a contradiction that $\boldsymbol{\Sigma}^\star$ is not strictly positive definite, i.e., that there exists a non-zero $\boldsymbol{v} \in \mathbb{R}^d$ such that $\boldsymbol{\Sigma}^\star \boldsymbol{v} = \boldsymbol{0}$. Let $\boldsymbol{w} = \boldsymbol{u} - \boldsymbol{v}$. Then, by (Equation A.2)

$$\langle 2\boldsymbol{x} - \boldsymbol{u} - \boldsymbol{w}, \boldsymbol{\Sigma}^\star(\boldsymbol{u} - (\boldsymbol{u} - \boldsymbol{v}))\rangle = \langle 2\boldsymbol{x} - \boldsymbol{u} - \boldsymbol{w}, \boldsymbol{\Sigma}^\star \boldsymbol{v}\rangle = 0.$$

From this we can show that, $\|\boldsymbol{x} - \boldsymbol{u}\|_{\boldsymbol{\Sigma}^\star}^2 = \|\boldsymbol{x} - (\boldsymbol{u} - \boldsymbol{v})\|_{\boldsymbol{\Sigma}^\star}^2$. This is a contradiction since $\boldsymbol{u}$ cannot be identifiable as $\boldsymbol{w} = \boldsymbol{u} - \boldsymbol{v} \neq \boldsymbol{u}$ would yield identical observations.

We now show that if $\boldsymbol{\Sigma}^\star$ is positive definite then $\boldsymbol{u}$ is identifiable. Suppose that $\boldsymbol{w} \in \mathbb{R}^d$ satisfies $\|\boldsymbol{x} - \boldsymbol{u}\|_{\boldsymbol{\Sigma}^\star}^2 = \|\boldsymbol{x} - \boldsymbol{w}\|_{\boldsymbol{\Sigma}^\star}^2$ for all $\boldsymbol{x} \in \mathbb{R}^d$. From (Equation A.2) we have that because $\langle 2\boldsymbol{x} - \boldsymbol{u} - \boldsymbol{w}, \boldsymbol{\Sigma}^\star(\boldsymbol{u} - \boldsymbol{w})\rangle = 0 \; \forall \boldsymbol{x} \in \mathbb{R}^d$, it must be the case that $\boldsymbol{\Sigma}^\star(\boldsymbol{u} - \boldsymbol{w}) = \boldsymbol{0}$. If $\boldsymbol{\Sigma}^\star$ is positive definite, then it must be the case that $\boldsymbol{u} - \boldsymbol{w} = \boldsymbol{0}$, and hence $\boldsymbol{w} = \boldsymbol{u}$. □ □

A.2 Additional Synthetic Simulation Results

Additional results for single-step estimation For the single-step estimation experiment found in Section 2.4.1, we also quantify algorithm performance via the normalized Kendall's Tau distance and the fraction of top 5 and 20 items correctly identified. The median (or interpolated median) and 25% and 75% quantiles are reported in Figure A.1. While the normalized Kendall's Tau distance decreases for $d = 2, 5$, and 10, it does so rather slowly. This is due to the fact that many items are very similar to each other in terms of their distance from $\boldsymbol{u}$, and hence getting the exact ordering of *all* items correct is rather difficult. However, the performance in identifying the top $5, 10$, and 20 items is strong, which indicates that the algorithm is in fact learning which items are important.

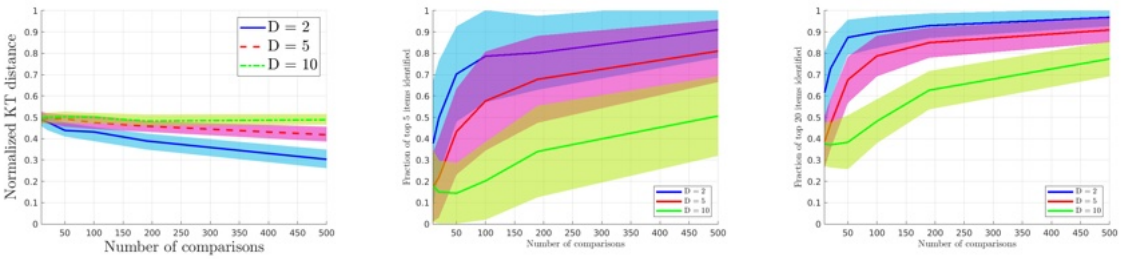


Figure A.1: Median normalized Kendall's Tau distance and interpolated median fraction of top 5 and 20 items identified over 100 trials, plotted with 25% and 75% quantiles. Regularization parameters: $\beta_1 = 2, \beta_2 = 0.002, \beta_3 = 0.001, \alpha = 1$.

Single-step estimation when $\boldsymbol{\Sigma}^\star = \boldsymbol{I}$ We demonstrate the effectiveness of our algorithm when $\boldsymbol{\Sigma}^\star = \boldsymbol{I}$ and compare performance with **Euclidean Algorithm 1** and **Euclidean Algorithm 2** as defined in Section 2.4.1. We sweep the performance for all three algorithms for $D = 2$ over different numbers of comparisons between 10 and 500. For a fixed number of comparisons, we perform 100 trials and report the median (or interpolated median) and 25% and 75% quantile for UR error, normalized Kendall's Tau distance, and the fraction of top 5, 10, and 20 items identified. For each trial, we generate a new metric and ideal point and $N = 100$ new items. As seen in Figure A.2, there is no significant loss in performance

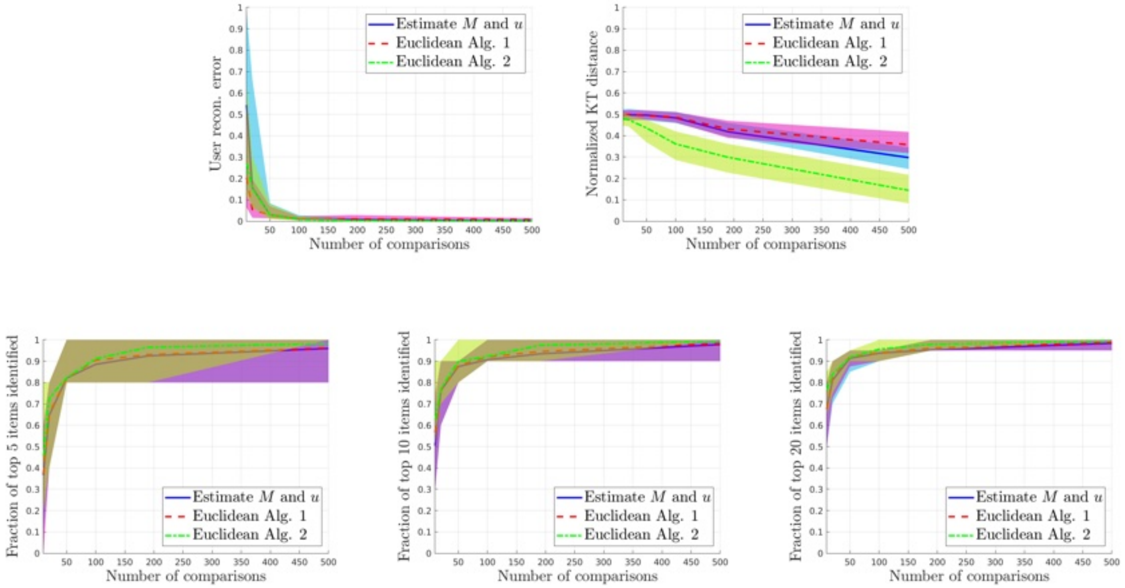


Figure A.2: Comparison of singe-step estimation against Euclidean Algorithms 1 and 2 when the true distance metric is $\boldsymbol{I}$. Regularization parameters: $\beta_1 = 2, \beta_2 = 0.002, \beta_3 = 0.001, \alpha = 1$.

when using our algorithm, especially as the number of comparisons increases. Thus, adding the additional flexibility to allow for $\boldsymbol{\Sigma}^\star \neq \boldsymbol{I}$ does not seem to result in any significant penalties, even when $\boldsymbol{\Sigma}^\star$ is in fact $\boldsymbol{I}$.

Additional results for alternating estimate For the alternating estimation experiment found in Section 2.4.1, we also quantify algorithm performance via the WER error, normalized Kendall's Tau distance, and fraction of top $5, 10$ and 20 items correctly identified. The median (or interpolated median) and 25% and 75% quantiles are reported in Figure A.3. In the intermediate regime (between 40 and 200 comparisons), the alternating estimate generally improves the WER error and fraction of top K items identified. The normalized Kendall's Tau distance remains relatively the same for all comparisons, but the improvement in the fraction of top K items indicates that the algorithm improves in identifying the which items are close to the ideal point.

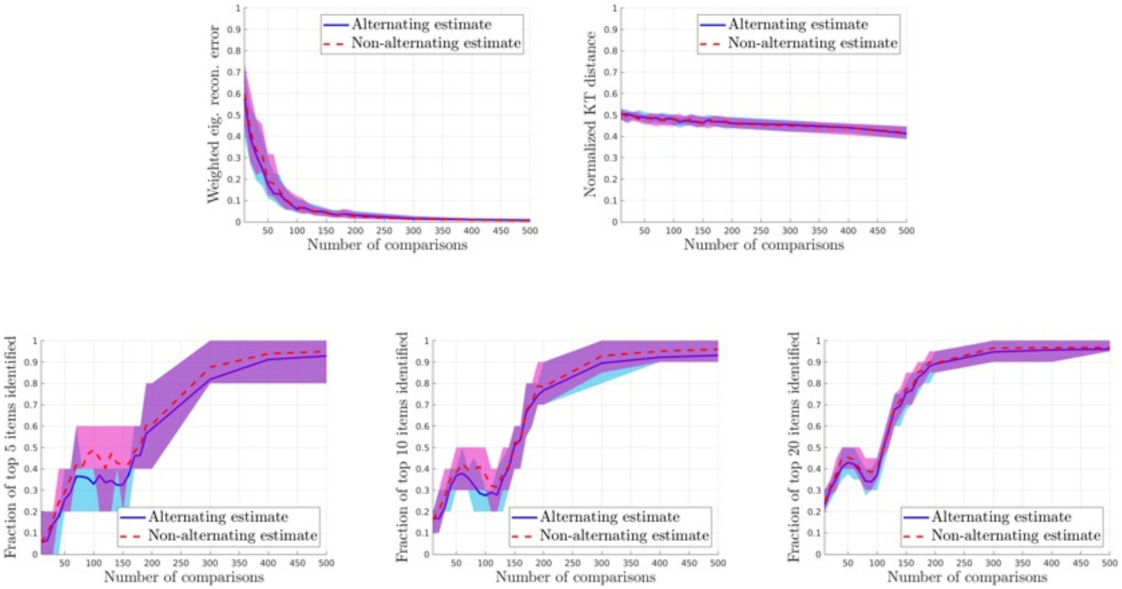


Figure A.3: Median WER error, normalized Kendall's Tau distance, and interpolated median for top $5, 10$, and 20 items for single-step and alternating estimation. Regularization parameters: $\beta_1^{(0)} = 2, \beta_2^{(0)} = 0.002, \beta_3^{(0)} = 0.0001, \alpha^{(0)} = 1$; $\beta_1^{(k)} = \frac{2}{3}, \beta_2^{(k)} = \frac{1}{15}, \beta_3^{(k)} = \frac{7}{1500}, \alpha^{(k)} = \frac{1}{2}$ for $k \geq 1$.

A.3 Data Pre-processing

***Unranked Candidates* dataset pre-processing** The *Unranked Candidates* dataset is originally comprised of $3,789$ total applicants, with 191 admitted with fellowship, 530 admitted without fellowship, and 3068 denied candidates. Ten raw features are associated with each candidate (Self-reported GRE analytical writing, self-reported GRE verbal, self-reported GRE quantitative, official GRE analytical writing, official GRE verbal, official GRE quantitative, GPA, and up to three scored letters of recommendation). Some candidates have missing entries for some of the ten raw features. Depending on which features are used to generate input data for the algorithm, we remove candidates with relevant missing data. If GRE scores are used, for each candidate, we take the official GRE scores to be the true GRE scores. If the official GRE scores are missing, then we take the self-reported scores. The raw GPA scores are already normalized on a 0 to 4 scale, but the normalization resulted in some unusable entries. If the GPA feature is used, we only keep candidates with GPAs between 1 and 4. The LoR score is computed as described in Section 2.4.2. In all, there are 3305 candidates with no missing entries (176 admitted with fellowship, 455 admitted candidates, and 2674 denied candidates).

***Ranked Candidates* dataset pre-processing** The *Ranked Candidates* dataset originally contains 89 candidates with four raw features (GRE analytical writing, GRE verbal, GRE quantitative, and GPA). For this dataset, there is only one GRE score available to us, so there is pre-processing needed to discern between self-reported and offiical. There is one candidate with missing raw features who is discarded, leaving us with 88 usable candidates.

A.4 Additional Experimental Results

Additional results for *Unranked Candidates* dataset As reported in Section 2.4.2, the ideal point and metric is learned using a set of 100 candidates (n_F = 33, $n_A = 33$, and $n_D = 34$) and all possible comparisons (3333). The significant feature interactions are reported in Table A.1, along with the corresponding eigenvalues. The weighted difference and sum of GPA and GRE writing score are the top two feature interactions and are almost equally important, followed by the LoR score and the weighted difference between GRE quantitative and verbal scores. The most insignificant feature interaction is the weighted sum of the quantitative and verbal scores.

Using the same number of candidates and comparisons, we also learn feature interactions and ideal points for pairs of features. For all pairs of features aside from GRE verbal vs. GRE quantitative (presented in Section 2.4.2), we display the level sets for the learned metric in Figure A.5. We again note that learning the ideal point with inherently restrictive features leads to unexpected behavior. In many cases, the ideal point value falls well outside of the allowed range for many of the features. For example in the GRE quantitative vs. GPA pair, the ideal GPA is 35, which is much larger than 4. In these cases, the fact that

Table A.1: Feature interactions and corresponding eigenvalues for the *Unranked Candidates* dataset for $n_F = 33, n_A = 33, n_D = 34$ and 3333 comparisons. Regularization parameters: $\beta_1 = \frac{1}{650}, \beta_2 = \frac{1}{6500}, \beta_3 = \frac{2}{65} \cdot 10^{-6}, \alpha = 1$.

Feature interactions in $\widehat{\mathbf{\Sigma}}^{\star}$.	
$\lambda_1 = 1991$	0.909 GRE writing $- 0.392$ GPA
$\lambda_2 = 1971$	0.919 GPA $+ 0.393$ GRE writing
$\lambda_3 = 1178$	0.982 LoR
$\lambda_4 = 861$	0.942 GRE quant $- 0.310$ GRE verbal
$\lambda_5 = 286$	0.942 GRE verbal $+ 0.319$ GRE quant

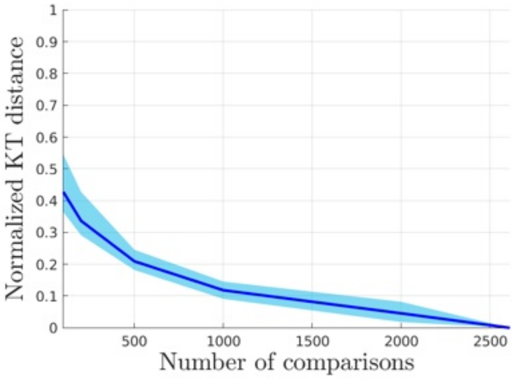


Figure A.4: Normalized Kendall's Tau distance for top 11 ranked candidates identified. Regularization parameters: $\beta_1 = \frac{7}{6002}, \beta_2 = \frac{1}{6002}, \beta_3 = \frac{2}{6002} \cdot 10^{-4}, \alpha = 1$.

the ideal value is higher than the maximum allowed values indicates that the larger the score, the better. This is consistent with our expectation that the optimal set of features should be the maximum value for all possible features. Many pairs of features do not have meaningful learned interactions, but pairs of features such as GRE writing vs. GPA do have some meaningful interaction.

Additional results for *Ranked Candidates* dataset For the *Ranked Candidates* dataset, we also record the the normalized Kendall's Tau distance for the top 11 candidates. We choose to evaluate the ranking of the top 11 candidates because these candidates are the ones most likely to be admitted. The median normalized Kendall's Tau distance and 25% and 75% quantiles can be found in Figure A.4. As the number of comparisons increases, we are able to extremely accurately predict the exact ranking of the top 11 candidates.

The learned metric using all 2610 comparisons does not exhibit any meaningful feature interactions. GPA and GRE writing are the top two features with roughly equal eigenvalues, followed by GRE quantitative. The GRE verbal score is the least significant feature. This is consistent with our expected order of significance of features for candidates.

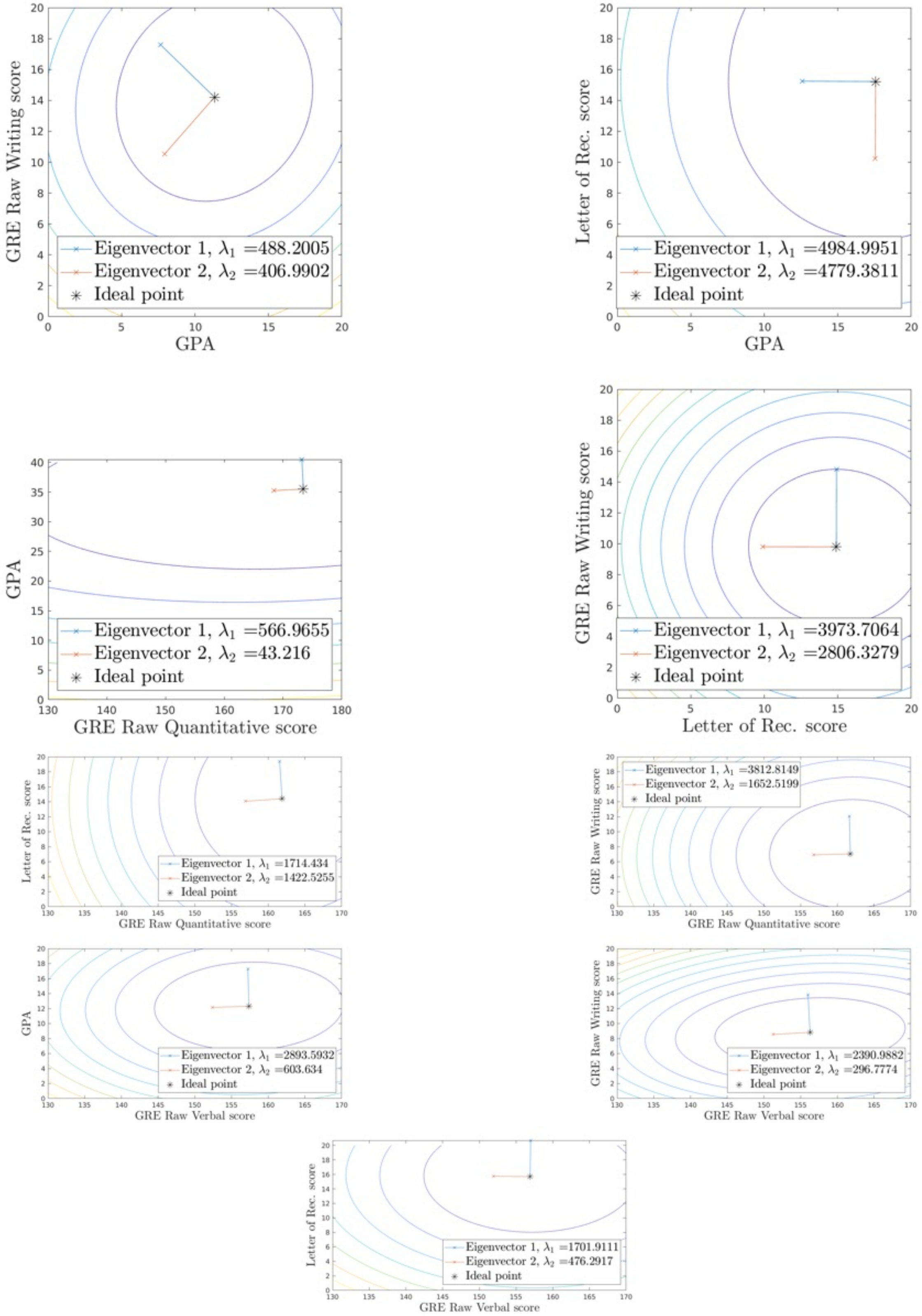


Figure A.5: Level sets for pairs of features for *Unranked Candidates* dataset.

APPENDIX B
SUPPLEMENTARY MATERIAL FOR CHAPTER 3

B.1 Simulation details

In this section, we provide details for the simulation results presented in Figure 3.2. For our experiments, we adopt a normalized version of the setup of [106] and form the metric $\mathbf{\Sigma}^\star$ by $\mathbf{\Sigma}^\star = \boldsymbol{L}\boldsymbol{L}^\top / \|\boldsymbol{L}\boldsymbol{L}^\top\|_F$, where $\boldsymbol{L}$ is a 50×10 matrix with i.i.d. Gaussian entries. We sweep the number of query responses N, estimate the metric with $\widehat{\mathbf{\Sigma}}$, and report the normalized estimation error $\|\mathbf{\Sigma}^\star - \widehat{\mathbf{\Sigma}}\|_F / \|\mathbf{\Sigma}^\star\|_F$ averaged over 10 independent trials. For each query response, items are drawn i.i.d. from a standard multivariate normal distribution, similar to [203].

Pairwise comparison setup. For pairwise comparisons, we use value of $y = 10$ to denote the squared distance at which items become dissimilar, following our distance-based model for human perception (see Section 3.3.1). For the i-th pairwise comparisons, we draw two items $\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}$ i.i.d. from a standard multivariate normal distribution. We record the pairwise comparison outcome $\epsilon_i \in \{-1, +1\}$ as $\epsilon_i = \mathrm{sign}(\|\boldsymbol{x}_1^{(i)} - \boldsymbol{x}_2^{(i)}\|_{\mathbf{\Sigma}^\star}^2 - y)$. To estimate the metric from pairwise comparisons, we utilize a nuclear-norm regularized hinge loss and solve the following optimization problem:

$$\widehat{\mathbf{\Sigma}}_{\text{PC}} \in \underset{\mathbf{\Sigma} \succeq \mathbf{0}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \max\{0, y - \epsilon_i \|\boldsymbol{x}_1^{(i)} - \boldsymbol{x}_2^{(i)}\|_{\mathbf{\Sigma}}^2\} + \lambda_{\text{PC}} \|\mathbf{\Sigma}\|_*.$$

Triplet setup. For the i-th triplet, we draw three items $\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}, \boldsymbol{x}_3^{(i)}$ i.i.d. from a standard multivariate normal distribution and record the outcome $\epsilon_i \in \{-1, +1\}$ as $\epsilon_i = \mathrm{sign}(\|\boldsymbol{x}_1^{(i)} - \boldsymbol{x}_2^{(i)}\|_{\mathbf{\Sigma}^\star}^2 - \|\boldsymbol{x}_1^{(i)} - \boldsymbol{x}_3^{(i)}\|_{\mathbf{\Sigma}^\star}^2)$. To estimate the metric from triplet responses, we follow [25] and utilize a nuclear-norm regularized hinge loss and solve the following optimization problem:

$$\widehat{\boldsymbol{\Sigma}}_{\text{T}} \in \operatorname*{arg\,min}_{\boldsymbol{\Sigma} \succeq \mathbf{0}} \frac{1}{N} \sum_{i=1}^{N} \max \left\{ 0, 1 - \epsilon_i \left(\| \boldsymbol{x}_1^{(i)} - \boldsymbol{x}_2^{(i)} \|_{\boldsymbol{\Sigma}}^2 - \| \boldsymbol{x}_1^{(i)} - \boldsymbol{x}_3^{(i)} \|_{\boldsymbol{\Sigma}}^2 \right) \right\} + \lambda_{\text{T}} \| \boldsymbol{\Sigma} \|_*.$$

Ranking-k query setup. For the i-th ranking query with a reference item $\boldsymbol{x}_0$ and k items $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ to be ranked, we draw all items i.i.d. from a standard multivariate normal distribution. For each item $\boldsymbol{x}_k$, we compute the squared distance $\|\boldsymbol{x}_0 - \boldsymbol{x}_k\|_{\boldsymbol{\Sigma}^\star}^2$. To determine the ranking of items, we sort the items based on this squared distance. To estimate the metric, we follow the approach of [28] and decompose the full ranking into its constituent triplets. A ranking consisting of k items can equivalently be decomposed into $k(k-1)/2$ triplet responses. To estimate the metric, we decompose each ranking query and use the triplet estimator presented above with regularization parameter λ_{R} to obtain estimate $\widehat{\boldsymbol{\Sigma}}_{\text{R-}k}$.

PAQ setup. For the i-th PAQ response, we draw the reference item $\boldsymbol{x}_i$ and query vector $\boldsymbol{a}_i$ i.i.d. from the standard multivariate normal distribution. We then receive a scaling γ_i^2 satisfying $\gamma_i^2 = y / \boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i$, with $y = 10$. To perform estimation, we leverage our method presented in Section 3.4. Our theoretical results indicate that the averaging parameter m should be set to 1 in the noiseless setting. Furthermore, the truncation threshold τ is large relative to our responses γ_i^2, meaning no truncation is employed. As a result, we solve the nuclear-norm regularized trace regression problem

$$\widehat{\boldsymbol{\Sigma}}_{\text{PAQ}} \in \operatorname*{arg\,min}_{\boldsymbol{\Sigma} \succeq \mathbf{0}} \frac{1}{N} \sum_{i=1}^{N} \left(\langle \boldsymbol{a}_i \boldsymbol{a}_i^\top, \boldsymbol{\Sigma} \rangle - \frac{y}{\gamma_i^2} \right)^2 + \lambda_{\text{PAQ}} \| \boldsymbol{\Sigma} \|_*.$$

In all cases above, we solve all optimization problems with `cvxpy` and normalize the estimated metric $\widehat{\boldsymbol{\Sigma}}_{\{\text{PC, T, R-}k\text{, PAQ}\}}$ to be unit Frobenius norm to ensure consistent scaling when compared against the true metric $\boldsymbol{\Sigma}^\star$. We use a value of 0.05 for all regularization parameters $\lambda_{\{\text{PC, T, R-}k\text{, PAQ}\}}$ and observe similar performance trends for other choices of regularization parameter.

B.2 Scale equivariance

In this section, we verify that the scale-equivariance of our derived theoretical bounds (Equation 3.15) and (Equation 3.16). Specifically, we denote by $\mathbf{\Sigma}^\star$ and $\widehat{\mathbf{\Sigma}}$ the ground-truth and the estimated matrices corresponding to value y. We denote by $\mathbf{\Sigma}_c^\star$ and $\widehat{\mathbf{\Sigma}}_c$ the ground-truth and estimated matrices corresponding to value $c_{\mathsf{scale}}y$ for any $c_{\mathsf{scale}} > 0$. By definition, we have $\mathbf{\Sigma}_c^\star = c_{\mathsf{scale}}\mathbf{\Sigma}^\star$, and it can be verified that solving the optimization program (Equation 3.10) yields $\widehat{\mathbf{\Sigma}}_c = c_{\mathsf{scale}}\widehat{\mathbf{\Sigma}}$. Hence, one expects the error bound to scale as c_{scale}. To verify this linear scaling in c_{scale}, we confirm that the noise η scales as c_{scale}.

Under the ground-truth metric $\mathbf{\Sigma}^\star$, if the user responds with an item that is a distance $y + \eta$ away from the reference item, then that same item is a distance $c_{\mathsf{scale}}(y + \eta)$ away from the reference under the scaled setting. As a result, the noise scales as a result of the choice of y. Therefore, the following values in the upper bound (Equation 3.15) can be written as scaled versions of their corresponding "ground-truth" values.

Noise	$\eta = c_{\mathsf{scale}}\ \eta_\star$	Noise median	$\mu_y = c_{\mathsf{scale}}\ \mu_y^\star$
Noise upper bound	$\eta^\uparrow = c_{\mathsf{scale}}\ \eta_\star^\uparrow$	Boundary upper bound	$y^\uparrow = c_{\mathsf{scale}}\ (y_\star + \eta_\star^\uparrow)$
Noise variance	$\nu_\eta^2 = c_{\mathsf{scale}}^2\ \nu_{\eta,\star}^2$	Singular values	$\sigma_k = c_{\mathsf{scale}}\ \sigma_k^\star$

Substituting these scaled expressions into the upper bounds (Equation 3.15) and (Equation 3.16), we have

$$\|\widehat{\mathbf{\Sigma}}_c - \mathbf{\Sigma}_c^\star\|_F \leq c_{\mathsf{scale}}\ C'\ \frac{(\sigma_1^\star)^2}{\sigma_r^\star} \frac{(y_\star^\uparrow)^{4/3}(\nu_{\eta,\star}^2)^{1/3}}{(\mu_y^\star)^2}\ r^{3/2} \left(\frac{d}{N}\right)^{1/3}$$

in the high-noise regime and

$$\|\widehat{\mathbf{\Sigma}}_c - \mathbf{\Sigma}_c^\star\|_F \leq c_{\mathsf{scale}}\ C'\ \frac{(\sigma_1^\star)^2}{\sigma_r^\star} \left(\frac{y_\star^\uparrow}{\mu_y^\star}\right)^2 r^{3/2} \left(\frac{d}{N}\right)^{1/2}$$

in the low-noise regime. Note that the constant C' is independent of c_{scale}.

B.3 Background and preliminary results

In this section, we provide an overview of the key tools that are utilized in our proofs.

B.3.1 Inverted measurement sensing matrices result in estimation bias

Recall from (Equation 3.5) that the random sensing matrix $\boldsymbol{A}^{\mathsf{inv}}$ takes the form

$$\boldsymbol{A}^{\mathsf{inv}} = \frac{y + \eta}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \boldsymbol{a}\boldsymbol{a}^\top.$$

Standard trace regression analysis assumes that for some sensing matrix $\boldsymbol{A}$ and measurement noise η, $\mathbb{E}\left[\eta \boldsymbol{A}\right] = \mathbf{0}$. Specifically, it is often typically assumed that η is zero-mean conditioned on the sensing matrix $\boldsymbol{A}$. The following lemma shows that for the inverted measurements, we have $\mathbb{E}[\eta \boldsymbol{A}^{\mathsf{inv}}] \neq 0$, resulting in bias in estimation.

Lemma 1. *Let $\boldsymbol{A}^{\mathsf{inv}}$ be the random matrix defined in* (Equation 3.5) *and η be the measurement noise. Then*

$$\mathbb{E}\left[\eta \boldsymbol{A}^{\mathsf{inv}}\right] \neq \mathbf{0}.$$

The proof of Lemma 1 is provided in Section B.3.6. Hence, utilizing established low-rank matrix estimators for inverted measurements result in biased estimation.

B.3.2 Sub-exponential random variables

Our analysis utilizes properties of sub-exponential random variables, a class of random variables with heavier tails than the Gaussian distribution.

Lemma 2 (Moment bounds for sub-exponential random variables [6, Proposition 2.7.1(b)])**.** *If X is a sub-exponential random variable, then there exists some constant c (only dependent* *on the distribution of the random variable X) such that for all integers $p \geq 1$,*

$$(\mathbb{E}|X|^p)^{1/p} \leq cp.$$

B.3.3 Bernstein's inequality

In our proofs, we use Bernstein's inequality to bound the sums of independent sub-exponential random variables.

Lemma 3 (Bernstein's inequality, adapted from [204, Theorem 2.10])**.** *Let $X_1, \ldots, X_n$ be independent real-valued random variables. Assume there exist positive numbers u_1 and u_2 such that*

$$\mathbb{E}\left[X_i^2\right] \leq u_1 \quad \textit{and} \quad \mathbb{E}\left[|X_i|^p\right] \leq \frac{p!}{2} u_1 u_2^{p-2} \textit{ for all integers } p \geq 2,$$

Then for all $t > 0$,

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}\left[X_i\right])\right| \geq \sqrt{\frac{2u_1 t}{n}} + \frac{u_2 t}{n}\right) \leq 2\exp(-t).$$

B.3.4 Moments of the ratios of quadratic forms

The quadratic term $\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}$ appears in the denominator of our sensing matrices, so we use the following result to quantify the moments of the ratios of quadratic forms.

Lemma 4. *There exists an absolute constant $c > 0$ such that the following is true. Let $\boldsymbol{a} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, $\boldsymbol{\Sigma}^\star \in \mathbb{R}^{d\times d}$ be any PSD matrix with rank r, and $\boldsymbol{U} \in \mathbb{R}^{d\times d}$ be an arbitrary symmetric matrix.*

(a) Suppose that $r > 8$. Then we have

$$\mathbb{E}\left(\frac{1}{\boldsymbol{a}^T \boldsymbol{\Sigma}^\star \boldsymbol{a}}\right)^4 \leq \frac{c}{\sigma_r^4 r^4}.$$

(b) Suppose that $r > 2$. Then we have

$$\mathbb{E}\left(\frac{\boldsymbol{a}^\top \boldsymbol{U}\boldsymbol{a}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}\right) \leq \frac{c}{\sigma_r r}\|\boldsymbol{U}\|_*.$$

The proof of Lemma 4 is presented in Section B.3.6.

B.3.5 A fourth moment bound for $\bar{\gamma}^2$

Recall from (Equation 3.7) that the averaged measurement $\bar{\gamma}^2$ takes the form

$$\bar{\gamma}_i^2 = \frac{1}{m}\sum_{j=1}^{m} \frac{y + \eta_i^{(j)}}{\boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i} = \frac{y + \bar{\eta}_i}{\boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i}.$$

Throughout our analysis, we utilize the fact that $\bar{\gamma}^2$ has a bounded fourth moment, as characterized in the following lemma.

Lemma 5. *Assume $r > 8$. Then there exists a universal constant $c > 0$, such that*

$$\mathbb{E}\left(\bar{\gamma}^2\right)^4 \leq c\left(\frac{y + \eta^\uparrow}{\sigma_r r}\right)^4,$$

where σ_r is the smallest non-zero singular value of $\boldsymbol{\Sigma}^\star$.

The proof of Lemma 5 is presented in Section B.3.6. For notational simplicity of the proofs, we denote $M = c\left(\frac{y+\eta^\uparrow}{\sigma_r r}\right)^4$.

B.3.6 Proofs of preliminary lemmas

In this section, we present proofs for preliminary lemmas from Section B.3.1, Section B.3.4, and Section B.3.5.

Proof of Lemma 1

Using the independence of the noise η and the sensing vector $\boldsymbol{a}$, and the assumption that η is zero mean, we have

$$\begin{aligned}\mathbb{E}\left[\eta \boldsymbol{A}^{\mathsf{inv}}\right] &= \mathbb{E}\left[\frac{\eta(y+\eta)}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \boldsymbol{a}\boldsymbol{a}^\top\right] \\ &= \mathbb{E}\left[\eta(y+\eta)\right] \cdot \mathbb{E}\left[\frac{1}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \boldsymbol{a}\boldsymbol{a}^\top\right] \\ &= \nu_\eta^2 \, \mathbb{E}\left[\frac{1}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \boldsymbol{a}\boldsymbol{a}^\top\right]. \qquad \text{(B.1)}\end{aligned}$$

The expectation in (Equation B.1) is non-zero, because the random matrix $\frac{1}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \boldsymbol{a}\boldsymbol{a}^\top$ is symmetric positive definite almost surely. Therefore, we have $\mathbb{E}\left[\eta \boldsymbol{A}^{\mathsf{inv}}\right] \neq \boldsymbol{0}$, as desired.

Proof of Lemma 4

Since $\boldsymbol{\Sigma}^\star$ is symmetric positive semidefinite, it be decomposed as $\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^\top$, where $\boldsymbol{Q}$ is a square orthonormal matrix and $\boldsymbol{\Sigma}$ is a diagonal matrix with non-negative entries. Multiplying $\boldsymbol{a}$ by any square orthonormal matrix does not change its distribution. Therefore, without loss of generality, we assume that $\boldsymbol{\Sigma}^\star$ is diagonal with all non-negative diagonal entries. We first note that the moments of the ratios in both parts of Lemma 4 exist, because by [205, Proposition 1], for non-negative integers p and q, the quantity $\mathbb{E}\frac{\left(\boldsymbol{a}^\top \boldsymbol{U}\boldsymbol{a}\right)^p}{\left(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}\right)^q}$ exists if $\frac{r}{2} > q$. Furthermore, we use the following expression from [205, Proposition 2]:

$$\mathbb{E}\frac{\left(\boldsymbol{a}^\top \boldsymbol{U}\boldsymbol{a}\right)^p}{\left(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}\right)^q} = \frac{1}{\Gamma(q)} \int_0^\infty t^{q-1} \cdot |\boldsymbol{\Delta}_t| \cdot \mathbb{E}\left(\boldsymbol{a}^\top \boldsymbol{\Delta}_t \boldsymbol{U} \boldsymbol{\Delta}_t \boldsymbol{a}\right)^p \mathrm{d}t, \qquad \text{(B.2)}$$

where $\boldsymbol{\Delta}_t = (\boldsymbol{I}_d + 2t\boldsymbol{\Sigma}^\star)^{-1/2}$ and $|\boldsymbol{\Delta}_t|$ is the determinant of $\boldsymbol{\Delta}_t$. To characterize the determinant $|\boldsymbol{\Delta}_t|$, we note that $\boldsymbol{\Delta}_t$ is a diagonal matrix whose d diagonal entries are

$$\frac{1}{(1+2t\sigma_1)^{1/2}}, \; \ldots, \; \frac{1}{(1+2t\sigma_r)^{1/2}}, \; 1, \; \ldots, \; 1.$$

Hence, the determinant is the product $|\boldsymbol{\Delta}_t| = \prod_{i=1}^{r} \frac{1}{(1+2t\sigma_i)^{1/2}}$. Furthermore, this product can be bounded as:

$$|\boldsymbol{\Delta}_t| \leq \frac{1}{(1+2t\sigma_r)^{r/2}}. \tag{B.3}$$

We now prove parts (a) and (b) separately.

Part (a). Using the integral expression (Equation B.2) with $p = 0$ and $q = 4$, and the upper bound (Equation B.3) on the determinant, we have

$$\begin{aligned}\mathbb{E}\left(\frac{1}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}\right)^4 &= \frac{1}{\Gamma(4)} \int_0^\infty t^3 \cdot |\boldsymbol{\Delta}_t| \,\mathrm{d}t \\ &\leq \frac{1}{\Gamma(4)} \int_0^\infty t^3 \frac{1}{(1+2t\sigma_r)^{r/2}} \,\mathrm{d}t.\end{aligned}$$

Denoting $s := 1 + 2t\sigma_r$, we have

$$\begin{aligned}\mathbb{E}\left(\frac{1}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}\right)^4 &\leq \frac{1}{2\Gamma(4)\sigma_r} \int_1^\infty \left(\frac{s-1}{2\sigma_r}\right)^3 \frac{1}{s^{r/2}} \,\mathrm{d}s \\ &\lesssim \frac{1}{\sigma_r^4} \int_1^\infty \frac{(s-1)^3}{s^{r/2}} \,\mathrm{d}s \\ &= \frac{1}{\sigma_r^4} \int_1^\infty \left(\frac{s^3}{s^{r/2}} - 3\frac{s^2}{s^{r/2}} + 3\frac{s}{s^{r/2}} - \frac{1}{s^{r/2}}\right) \mathrm{d}s \\ &= \frac{1}{\sigma_r^4} \left(\frac{2}{r-8} - \frac{6}{r-6} + \frac{6}{r-4} - \frac{2}{r-2}\right) \\ &\leq \frac{c}{\sigma_r^4 r^4},\end{aligned}$$

as desired.

Part (b). Using the integral expression (Equation B.2) $p = q = 1$ and the upper bound (Equation B.3) on the determinant, we have

$$\begin{aligned}\mathbb{E}\left(\frac{\boldsymbol{a}^\top \boldsymbol{U}\boldsymbol{a}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}\right) &= \frac{1}{\Gamma(1)}\int\limits_0^\infty |\boldsymbol{\Delta}_t| \cdot \mathbb{E}\left[\boldsymbol{a}^\top \boldsymbol{\Delta}_t \boldsymbol{U}\boldsymbol{\Delta}_t \boldsymbol{a}\right] \mathrm{d}t \\ &\leq \frac{1}{\Gamma(1)}\int\limits_0^\infty \frac{1}{(1+2t\sigma_r)^{r/2}}\mathbb{E}\left[\boldsymbol{a}^\top \boldsymbol{\Delta}_t \boldsymbol{U}\boldsymbol{\Delta}_t \boldsymbol{a}\right] \mathrm{d}t. \end{aligned} \tag{B.4}$$

We now bound the expectation term in (Equation B.4). Note that for $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$, we have $\mathbb{E}\left[\boldsymbol{a}^\top \boldsymbol{B}\boldsymbol{a}\right] = \operatorname{tr}(\boldsymbol{B})$ for any symmetric matrix $\boldsymbol{B}$. Therefore, we have

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{a}^\top \boldsymbol{\Delta}_t \boldsymbol{U}\boldsymbol{\Delta}_t \boldsymbol{a}\right] &= \operatorname{tr}(\boldsymbol{\Delta}_t \boldsymbol{U}\boldsymbol{\Delta}_t) \\ &\overset{\text{(i)}}{\leq} \|\boldsymbol{\Delta}_t \boldsymbol{U}\boldsymbol{\Delta}_t\|_* \\ &\overset{\text{(ii)}}{\leq} \|\boldsymbol{U}\|_*, \end{aligned} \tag{B.5}$$

where (i) the fact that $\operatorname{tr}(\boldsymbol{B}) \leq \|\boldsymbol{B}\|_*$ for any symmetric matrix $\boldsymbol{B}$. Furthermore, (ii) follows from Hölder's inequality for Schatten-p norms, where we have that $\|\boldsymbol{\Delta}_t \boldsymbol{U}\boldsymbol{\Delta}_t\|_* \leq \|\boldsymbol{\Delta}_t\|_{\text{op}}^2 \cdot \|\boldsymbol{U}\|_*$. Because $\boldsymbol{\Delta}_t$ is diagonal and the entries of $\boldsymbol{\Delta}_t$ are bounded between 0 and 1, we bound the operator norm as $\|\boldsymbol{\Delta}_t\|_{\text{op}} \leq 1$. Substituting (Equation B.5) to (Equation B.4), we obtain

$$\begin{aligned}\mathbb{E}\left(\frac{\boldsymbol{a}^\top \boldsymbol{U}\boldsymbol{a}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}\right) &\leq \|\boldsymbol{U}\|_* \cdot \int\limits_0^\infty \frac{1}{(1+2t\sigma_r)^{r/2}} \mathrm{d}t \\ &\lesssim \frac{1}{\sigma_r r} \cdot \|\boldsymbol{U}\|_*, \end{aligned}$$

as desired.

Proof of Lemma 5

By the assumption that the noise is upper bounded by $\eta^{\uparrow}$, we have $y+\bar{\eta} \leq y+\eta^{\uparrow}$. Therefore, we have

$$\begin{aligned}
\mathbb{E}\left(\bar{\gamma}^2\right)^4 &= \mathbb{E}\left(\frac{y+\bar{\eta}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}\right)^4 \\
&\leq (y+\eta^{\uparrow})^4 \cdot \mathbb{E}\left(\frac{1}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}\right)^4 \\
&\overset{\text{(i)}}{\lesssim} \left(\frac{1}{\sigma_r r}\right)^4,
\end{aligned}$$

where step (i) applies Item (a) of Lemma 4.

B.4 Proof of Proposition 3

In the proof, we decompose the operator norm $\left\|\frac{1}{n}\sum_{i=1}^{n} y\widetilde{\boldsymbol{A}}_i - \frac{1}{n}\sum_{i=1}^{n}\langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}_i\right\|_{\text{op}}$ from (Equation 3.20) into individual terms and bound them separately. We define random matrices

$$\bar{\boldsymbol{A}} = \bar{\gamma}^2 \boldsymbol{a}\boldsymbol{a}^\top = \frac{y+\bar{\eta}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \boldsymbol{a}\boldsymbol{a}^\top \tag{B.6}$$

and

$$\widetilde{\boldsymbol{A}} = \widetilde{\gamma}^2 \boldsymbol{a}\boldsymbol{a}^\top = \left(\frac{y+\bar{\eta}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \wedge \tau\right) \boldsymbol{a}\boldsymbol{a}^\top \tag{B.7}$$

as the sensing matrix formed with the m-averaged responses $\bar{\gamma}$ and truncated responses $\widetilde{\gamma}$, respectively.

Step 1: decompose the error into five terms. We begin by adding and subtracting multiple quantities as follows:

$$
\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} y\widetilde{\boldsymbol{A}}_i - \frac{1}{n}\sum_{i=1}^{n}\langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}_i &= \frac{1}{n}\sum_{i=1}^{n} y\widetilde{\boldsymbol{A}}_i - \mathbb{E}\left[y\widetilde{\boldsymbol{A}}\right] + \mathbb{E}\left[y\widetilde{\boldsymbol{A}}\right] - \mathbb{E}\left[y\bar{\boldsymbol{A}}\right] \\
&\quad + \mathbb{E}\left[y\bar{\boldsymbol{A}}\right] - \mathbb{E}\left[\langle \widetilde{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}\right] + \mathbb{E}\left[\langle \widetilde{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}\right] \qquad \text{(B.8)} \\
&\quad - \frac{1}{n}\sum_{i=1}^{n}\langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}_i \\
&\overset{(i)}{=} \frac{1}{n}\sum_{i=1}^{n} y\widetilde{\boldsymbol{A}}_i - \mathbb{E}\left[y\widetilde{\boldsymbol{A}}\right] + \mathbb{E}\left[y\widetilde{\boldsymbol{A}}\right] - \mathbb{E}\left[y\bar{\boldsymbol{A}}\right] \\
&\quad + \mathbb{E}\left[\langle \bar{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star\rangle \bar{\boldsymbol{A}}\right] - \mathbb{E}\left[\langle \widetilde{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}\right] - \mathbb{E}\left[\bar{\eta}\bar{\boldsymbol{A}}\right] \\
&\quad + \mathbb{E}\left[\langle \widetilde{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}\right] - \frac{1}{n}\sum_{i=1}^{n}\langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}_i, \qquad \text{(B.9)}
\end{aligned}
$$

where step (i) is true by substituting $y = \langle \bar{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star\rangle - \bar{\eta}$ to the term of $\mathbb{E}\left[y\bar{\boldsymbol{A}}\right]$, and the fact that the noise term $\bar{\eta}$ is zero-mean. By triangle inequality, we group the terms in (Equation B.9) and bound the operator norm by

$$
\begin{aligned}
\left\|\frac{1}{n}\sum_{i=1}^{n} y\widetilde{\boldsymbol{A}}_i - \frac{1}{n}\sum_{i=1}^{n}\langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}_i\right\|_{\text{op}} &\le \underbrace{y\left\|\frac{1}{n}\sum_{i=1}^{n} \widetilde{\boldsymbol{A}}_i - \mathbb{E}\left[\widetilde{\boldsymbol{A}}\right]\right\|_{\text{op}}}_{\text{Term 1}} \\
&\quad + \underbrace{y\left\|\mathbb{E}\left[\widetilde{\boldsymbol{A}}\right] - \mathbb{E}\left[\bar{\boldsymbol{A}}\right]\right\|_{\text{op}}}_{\text{Term 2}} \\
&\quad + \underbrace{\left\|\mathbb{E}\left[\langle \bar{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star\rangle \bar{\boldsymbol{A}}\right] - \mathbb{E}\left[\langle \widetilde{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}\right]\right\|_{\text{op}}}_{\text{Term 3}} \\
&\quad + \underbrace{\left\|\mathbb{E}\left[\langle \widetilde{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}\right] - \frac{1}{n}\sum_{i=1}^{n}\langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}_i\right\|_{\text{op}}}_{\text{Term 4}} \\
&\quad + \underbrace{\left\|\mathbb{E}\left[\bar{\eta}\bar{\boldsymbol{A}}\right]\right\|_{\text{op}}}_{\text{Term 5}}. \qquad \text{(B.10)}
\end{aligned}
$$

In the remaining proof, we bound the five terms in (Equation B.10) individually. We first discuss the nature of these five terms.

- **Terms 1 and 4:** These two terms characterize the difference between the empirical mean of quantities involving $\widetilde{\boldsymbol{A}}$ and their true expectation (see Lemma 6 and Lemma 9). In the proof, we show that the empirical mean concentrates around the expectation with high probability, as a function of the number of sensing vectors n.

- **Terms 2 and 3:** These two terms characterize the difference in expectation introduced by truncating $\bar{\boldsymbol{A}}$ to $\widetilde{\boldsymbol{A}}$ (see Lemma 7 and Lemma 8). Hence, these two terms characterize biases that arise from truncation. They diminish as $\tau \to \infty$, because setting τ to ∞ is equivalent to no thresholding, and hence $\widetilde{\boldsymbol{A}}$ becomes identical to $\bar{\boldsymbol{A}}$. Since expectations are considered, these two terms depend on the threshold τ, but not the number of sensing vectors n or the averaging parameter m.

- **Term 5:** Term 5 is a bias term that arises from the fact that the mean of the noise η conditioned on sensing matrix $\bar{\boldsymbol{A}}$ is non-zero. We show that this bias scales like $\frac{1}{m}$ (see Lemma 10) in terms of the averaging parameter m.

Putting these terms together, Terms 1 and 4 depend on n, Terms 2 and 3 depend on τ, and Term 5 depends on m. In Corollary 1, we set the values of τ, n and m to balance these terms.

Step 2: bound the five terms individually. In what follows, we provide five lemmas to bound each of the five terms individually. In the proofs of the five lemmas, we rely on an upper bound on the fourth moment of the m-sample averaged measurements $\bar{\gamma}^2$. As shown in Lemma 5 in Section B.3.5, for some absolute constant c, this fourth moment can be upper bounded by a quantity that we denote M:

$$\mathbb{E}[(\bar{\gamma}^2)^4] \leq M = c\left(\frac{y^{\uparrow}}{\sigma_r r}\right)^4. \tag{B.11}$$

We also rely heavily on the following truncation properties relating the m-sample averaged measurements $\bar{\gamma}^2$ and truncated measurements $\tilde{\gamma}^2$:

$$\tilde{\gamma}_i^2 \leq \tau \tag{TP1}$$

$$\tilde{\gamma}_i^2 \leq \bar{\gamma}_i^2 \tag{TP2}$$

$$\bar{\gamma}_i^2 - \tilde{\gamma}_i^2 = (\bar{\gamma}_i^2 - \tilde{\gamma}_i^2) \cdot \mathbf{1}\{\bar{\gamma}_i^2 \geq \tau\}. \tag{TP3}$$

The following lemma provides a bound for Term 1.

Lemma 6. *Let $\widetilde{\boldsymbol{A}}_1, \ldots, \widetilde{\boldsymbol{A}}_n$ be i.i.d copies of a random matrix $\widetilde{\boldsymbol{A}}$ as defined in* (Equation B.7). *There exists an absolute constant $c > 0$ such that for any $t > 0$, we have*

$$\left\| \frac{1}{n} \sum_{i=1}^{n} \widetilde{\boldsymbol{A}}_i - \mathbb{E}\left[\widetilde{\boldsymbol{A}}_i\right] \right\|_{\mathrm{op}} \leq c \left(\sqrt{\frac{M^{1/2} t}{n}} + \frac{\tau t}{n} \right)$$

with probability at least $1 - 2 \cdot 9^d \cdot \exp(-t)$.

The proof of Lemma 6 is provided in Section B.4. The next lemma provides an upper bound for Term 2.

Lemma 7. *Let $\bar{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{A}}$ be the random matrices defined in* (Equation B.6) *and* (Equation B.7), *respectively. Then there exists an absolute constant $c > 0$ such that*

$$\left\| \mathbb{E}\left[\widetilde{\boldsymbol{A}}\right] - \mathbb{E}\left[\bar{\boldsymbol{A}}\right] \right\|_{\mathrm{op}} \leq \frac{c M^{1/2}}{\tau}.$$

The proof of Lemma 7 is provided in Section B.4. The following lemma provides an upper bound for Term 3. Recall that the quantity $y^{\uparrow}$ denotes $y + \eta^{\uparrow}$.

Lemma 8. *Let $\bar{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{A}}$ be the random matrices defined in* (Equation B.6) *and* (Equation B.7), *respectively. Then there exists an absolute constant $c > 0$ such that*

$$\left\| \mathbb{E}\left[\langle \bar{\boldsymbol{A}}, \boldsymbol{\Sigma}^{\star} \rangle \bar{\boldsymbol{A}}\right] - \mathbb{E}\left[\langle \widetilde{\boldsymbol{A}}, \boldsymbol{\Sigma}^{\star} \rangle \widetilde{\boldsymbol{A}}\right] \right\|_{\mathrm{op}} \leq \frac{c\, y^{\uparrow} M^{1/2}}{\tau}.$$

The proof of Lemma 8 is provided in Section B.4. The following lemma provides an upper bound for Term 4.

Lemma 9. *Let $\widetilde{\boldsymbol{A}}_1, \ldots, \widetilde{\boldsymbol{A}}_n$ be i.i.d copies of a random matrix $\widetilde{\boldsymbol{A}}$ defined in* (Equation B.7). *There exists an absolute constant $c > 0$ such that for any $t > 0$, we have*

$$\left\| \mathbb{E}\left[\langle \widetilde{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star \rangle \widetilde{\boldsymbol{A}} \right] - \frac{1}{n} \sum_{i=1}^{n} \langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{\Sigma}^\star \rangle \widetilde{\boldsymbol{A}}_i \right\|_{\mathrm{op}} \leq c\, y^\uparrow \left(\sqrt{\frac{M^{1/2} t}{n}} + \frac{\tau t}{n} \right)$$

with probability at least $1 - 2 \cdot 9^d \cdot \exp\left(-t\right)$.

The proof of Lemma 9 is provided in Section B.4. We note that Terms 2 and 3 are bias that result from shrinkage, but crucially are inversely dependent on the shrinkage threshold τ. This fact allows us to set τ so that the order of Terms 2 and 3 match the order of Terms 1 and 4.

The final lemma bounds Term 5, which is a bias that arises from the dependence of the sensing matrix $\bar{\boldsymbol{A}}$ on the noise η.

Lemma 10. *Let $\bar{\boldsymbol{A}}$ be the random matrix defined in Equation* (Equation B.6). *Suppose that $\boldsymbol{\Sigma}^\star$ has rank r with $r > 2$. Then there exists an absolute constant $c > 0$ such that*

$$\mathbb{E}\left[\left\| \bar{\eta} \bar{\boldsymbol{A}} \right\|_{\mathrm{op}} \right] \leq \frac{c}{\sigma_r r} \frac{\nu_\eta^2}{m}.$$

The proof of Lemma 10 is provided in Section B.4. We note that the bias scales with the variance of the m-sample averaged noise $\bar{\eta}$, which scales inversely with m.

Step 3: combine the five terms. We set $t = (\log 9 + 1)d$. Substituting the bounds from Lemma 6– Lemma 10 back to (Equation B.10) and taking a union bound, we have that with probability at least $1 - 4\exp(-d)$,

$$\begin{aligned}\left\|\frac{1}{n}\sum_{i=1}^{n} y\widetilde{\boldsymbol{A}}_i - \frac{1}{n}\sum_{i=1}^{n}\langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{\Sigma}^\star\rangle \widetilde{\boldsymbol{A}}_i\right\|_{\mathrm{op}} &\lesssim \left(y^\uparrow + 1\right)\left(\sqrt{\frac{M^{1/2}d}{n}} + \frac{d}{n}\tau + \frac{M^{1/2}}{\tau}\right) + \frac{1}{\sigma_r r}\frac{\nu_\eta^2}{m} \\ &\overset{(i)}{\lesssim} y^\uparrow\left(\frac{y^\uparrow}{\sigma_r r}\sqrt{\frac{d}{n}} + \frac{d}{n}\tau + \left(\frac{y^\uparrow}{\sigma_r r}\right)^2\frac{1}{\tau}\right) + \frac{1}{\sigma_r r}\frac{\nu_\eta^2}{m},\end{aligned}$$

where step (i) is true by substituting in the expression (Equation B.11) for M.

Proof of Lemma 6.

Let $\mathcal{A}_{\frac{1}{4}} \subseteq \mathcal{S}^{d-1}$ be a $\frac{1}{4}$-covering of the d-dimensional unit sphere $\mathcal{S}^{d-1} := \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$. By a covering argument [6, Exercise 4.4.3], for any symmetric matrix $\boldsymbol{U} \in \mathbb{S}^{d\times d}$, its operator norm is bounded by $\|\boldsymbol{U}\|_{\mathrm{op}} \leq 2\sup_{\boldsymbol{v}\in\mathcal{A}_{\frac{1}{4}}}\left|\boldsymbol{v}^\top \boldsymbol{U}\boldsymbol{v}\right|$. Hence, we have

$$\begin{aligned}\left\|\frac{1}{n}\sum_{i=1}^{n}\widetilde{\boldsymbol{A}}_i - \mathbb{E}\left[\widetilde{\boldsymbol{A}}\right]\right\|_{\mathrm{op}} &\leq 2\sup_{\boldsymbol{v}\in\mathcal{A}_{\frac{1}{4}}}\left|\boldsymbol{v}^\top\left(\frac{1}{n}\sum_{i=1}^{n}\widetilde{\boldsymbol{A}}_i - \mathbb{E}\left[\widetilde{\boldsymbol{A}}\right]\right)\boldsymbol{v}\right| \\ &= 2\sup_{\boldsymbol{v}\in\mathcal{A}_{\frac{1}{4}}}\left|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}^\top\widetilde{\boldsymbol{A}}_i\boldsymbol{v} - \mathbb{E}\left[\boldsymbol{v}^\top\widetilde{\boldsymbol{A}}\boldsymbol{v}\right]\right|. \qquad \text{(B.12)}\end{aligned}$$

We invoke Bernstein's inequality. We first show that the Bernstein condition holds. Namely, we show that for each integer $p \geq 2$, we have that for any unit vector $\boldsymbol{v} \in \mathbb{R}^d$,

$$\mathbb{E}\left|\boldsymbol{v}^\top\widetilde{\boldsymbol{A}}\boldsymbol{v}\right|^p \leq \frac{p!}{2}u_1 u_2^{p-2}, \qquad \text{(B.13)}$$

where $u_1 = c_1 M^{\frac{1}{2}}$ and $u_2 = c_2\tau$ for some universal positive constants c_1 and c_2. Given the Bernstein condition (Equation B.13), we then apply Bernstein's inequality to bound (Equation B.12).

Proving the Bernstein condition (Equation B.13)**.** We fix any unit vector $\boldsymbol{v} \in \mathbb{R}^d$. Since $\widetilde{\boldsymbol{A}} = \widetilde{\gamma}^2\boldsymbol{a}\boldsymbol{a}^\top$, we have $\boldsymbol{v}^\top\widetilde{\boldsymbol{A}}\boldsymbol{v} = \widetilde{\gamma}^2(\boldsymbol{v}^\top\boldsymbol{a})^2$. Recall that the random vector $\boldsymbol{a}$ is distributed as $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$. Since $\boldsymbol{v}$ is a unit vector, it follows that $\boldsymbol{v}^\top\boldsymbol{a} \sim \mathcal{N}(0, 1)$. Denote by

$G \sim \mathcal{N}(0,1)$ a standard normal random variable. For any integer $p \geq 2$, we have

$$\begin{aligned}
\mathbb{E}\left|\boldsymbol{v}^\top \widetilde{\boldsymbol{A}} \boldsymbol{v}\right|^p = \mathbb{E}\,\left(\widetilde{\gamma}^2 G^2\right)^p &\overset{\text{(i)}}{\leq} \tau^{p-2}\mathbb{E}\left[\left(\widetilde{\gamma}^2\right)^2 G^{2p}\right] \\
&\overset{\text{(ii)}}{\leq} \tau^{p-2} \cdot \mathbb{E}\left[\left(\bar{\gamma}^2\right)^2 G^{2p}\right] \\
&\overset{\text{(iii)}}{\leq} \tau^{p-2}\left(\mathbb{E}\left[\left(\bar{\gamma}^2\right)^4\right] \cdot \mathbb{E}\left[G^{4p}\right]\right)^{1/2} \\
&\overset{\text{(iv)}}{\leq} \tau^{p-2}\left(M \cdot \mathbb{E}\left[G^{4p}\right]\right)^{1/2}, \qquad \text{(B.14)}
\end{aligned}$$

where steps (i) and (ii) follow from (Equation TP1) and (Equation TP2), respectively; step (iii) follows from Cauchy–Schwarz inequality; and step (iv) follows upper bounding the fourth moment of $\bar{\gamma}^2$ with the quantity M from (Equation B.11).

Note that since G is standard normal, by definition G^2 follows a Chi-Square distribution with 1 degree of freedom, and hence sub-exponential. By Lemma 2 in Section B.3.2, there exists some constant $c > 0$ such that we have $\left(\mathbb{E}\left[(G^2)^p\right]\right)^{1/p} \leq cp$ for all $p \geq 1$. Hence, we have $\left(\mathbb{E}\left[G^{4p}\right]\right)^{1/2p} \leq 2cp$ and

$$\begin{aligned}
\left(\mathbb{E}\left[G^{4p}\right]\right)^{1/2} \leq (2cp)^p &= \left(\frac{p}{e}\right)^p \cdot (2ec)^p \\
&\overset{\text{(i)}}{<} p! \cdot (2ec)^p \qquad \text{(B.15)}
\end{aligned}$$

where step (i) is true by Stirling's inequality that for all $p \geq 1$,

$$p! > \sqrt{2\pi p}\left(\frac{p}{e}\right)^p e^{\frac{1}{12p+1}} > \left(\frac{p}{e}\right)^p.$$

Substituting (Equation B.15) back to (Equation B.14) and rearranging terms completes the proof of the Bernstein condition (Equation B.13).

Applying Bernstein's inequality to bound (Equation B.12). By Bernstein's inequality (see Lemma 3), given condition (Equation B.13), we have that for any unit vector $\boldsymbol{v} \in \mathbb{R}^d$

and any $t > 0$,

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}^\top \widetilde{\boldsymbol{A}}_i\boldsymbol{v} - \mathbb{E}\left[\boldsymbol{v}^\top \widetilde{\boldsymbol{A}}\boldsymbol{v}\right]\right| \geq 2\left(\sqrt{\frac{c_1 M^{1/2} t}{n}} + \frac{c_2 \tau t}{n}\right)\right) \leq 2\exp\left(-t\right). \quad \text{(B.16)}$$

By [6, Corollary 4.2.13], the cardinality of the covering set $\mathcal{A}_{\frac{1}{4}}$ is bounded above by 9^d. Therefore, taking a union bound on (Equation B.16), we have

$$\mathbb{P}\left(\sup_{\boldsymbol{v}\in\mathcal{A}_{\frac{1}{4}}}\left|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}^\top \widetilde{\boldsymbol{A}}_i\boldsymbol{v} - \mathbb{E}\left[\boldsymbol{v}^\top \widetilde{\boldsymbol{A}}\boldsymbol{v}\right]\right| \geq 2\left(\sqrt{\frac{c_1 M^{1/2} t}{n}} + \frac{c_2 \tau t}{n}\right)\right) \leq 2\cdot 9^d \cdot \exp\left(-t\right). \quad \text{(B.17)}$$

Substituting in (Equation B.12) to (Equation B.17), for any $t > 0$, we have

$$\mathbb{P}\left(\left\|\frac{1}{n}\sum_{i=1}^{n}\widetilde{\boldsymbol{A}}_i - \mathbb{E}\left[\widetilde{\boldsymbol{A}}\right]\right\|_{\mathrm{op}} \lesssim \sqrt{\frac{M^{1/2} t}{n}} + \frac{\tau t}{n}\right) \geq 1 - 2\cdot 9^d \cdot \exp(-t),$$

as desired.

Proof of Lemma 7

By definition of the operator norm, we have

$$\left\|\mathbb{E}\left[\widetilde{\boldsymbol{A}}\right] - \mathbb{E}\left[\bar{\boldsymbol{A}}\right]\right\|_{\mathrm{op}} = \sup_{\boldsymbol{v}\in\mathcal{S}^{d-1}}\left|\boldsymbol{v}^\top\left(\mathbb{E}\left[\bar{\boldsymbol{A}}\right] - \mathbb{E}\left[\widetilde{\boldsymbol{A}}\right]\right)\boldsymbol{v}\right|.$$

We fix any $\boldsymbol{v} \in \mathcal{S}^{d-1}$, and bound $\boldsymbol{v}^\top\left(\mathbb{E}\left[\bar{\boldsymbol{A}}\right] - \mathbb{E}\left[\widetilde{\boldsymbol{A}}\right]\right)\boldsymbol{v}$. Similar to the proof of Lemma 6, we note that $\boldsymbol{v}^\top\boldsymbol{a} \sim \mathcal{N}(0,1)$ and denote the random variable $G \sim \mathcal{N}(0,1)$. Substituting in the expression for sensing matrices $\bar{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{A}}$, we have

$$\begin{aligned}\left|\boldsymbol{v}^\top \left(\mathbb{E}\left[\bar{\boldsymbol{A}}\right] - \mathbb{E}\left[\widetilde{\boldsymbol{A}}\right]\right)\boldsymbol{v}\right| &= \left|\boldsymbol{v}^\top \mathbb{E}\left[\bar{\gamma}^2 \boldsymbol{a}\boldsymbol{a}^\top - \widetilde{\gamma}^2 \boldsymbol{a}\boldsymbol{a}^\top\right]\boldsymbol{v}\right| \\ &\overset{\text{(i)}}{=} \mathbb{E}\left[\left(\bar{\gamma}^2 - \widetilde{\gamma}^2\right) G^2\right] \\ &\overset{\text{(ii)}}{=} \mathbb{E}\left[\left(\bar{\gamma}^2 - \widetilde{\gamma}^2\right) G^2 \cdot \mathbb{1}\{\bar{\gamma}^2 \geq \tau\}\right] \\ &\leq \mathbb{E}\left[\bar{\gamma}^2 G^2 \cdot \mathbb{1}\{\bar{\gamma}^2 \geq \tau\}\right] \\ &\overset{\text{(iii)}}{\leq} \left(\mathbb{E}\left[(\bar{\gamma}^2 G^2)^2\right] \cdot \mathbb{E}\left[\mathbb{1}\{\bar{\gamma}^2 \geq \tau\}\right]\right)^{1/2} \\ &\overset{\text{(iv)}}{\leq} \left(\mathbb{E}\left[|\bar{\gamma}^2|^4\right] \cdot \mathbb{E}\left[|G^2|^4\right]\right)^{1/4} \left(\mathbb{P}\left(\bar{\gamma}^2 \geq \tau\right)\right)^{1/2}, \end{aligned} \tag{B.18}$$

where where step (i) is true because $\bar{\gamma}^2 \geq \widetilde{\gamma}^2$ from to (Equation TP2), step (ii) is true due to (Equation TP3), and steps (iii) and (iv) follow from Cauchy–Schwarz inequality. We proceed by bounding each of the terms in (Equation B.18) separately. First, we can upper bound the fourth moment $\mathbb{E}\left[|\bar{\gamma}^2|^4\right]$ by the quantity M from (Equation B.11). Second, G^2 is a sub-exponential random variable. By Lemma 2 in Section B.3.2, we have that $\mathbb{E}\left[|G^2|^4\right]^{1/4} \leq c$ for some constant c. It remains to bound the term $\left(\mathbb{P}\left(\bar{\gamma}^2 \geq \tau\right)\right)^{1/2}$. We have

$$\begin{aligned}\mathbb{P}\left(\bar{\gamma}^2 \geq \tau\right) &\overset{\text{(i)}}{\leq} \frac{\mathbb{E}\,|\bar{\gamma}^2|^2}{\tau^2} \\ &\overset{\text{(ii)}}{\leq} \frac{\left(\mathbb{E}\,|\bar{\gamma}^2|^4\right)^{1/2}}{\tau^2} \\ &\overset{\text{(iii)}}{\leq} \frac{M^{1/2}}{\tau^2},\end{aligned}$$

where step (i) follows from Markov's inequality, step (ii) follows from Cauchy–Schwarz inequality, and step (iii) follows from the fourth moment bound on the averaged scaling $\bar{\gamma}^2$. Putting everything together back to (Equation B.18), we have

$$\left|\boldsymbol{v}^\top \left(\mathbb{E}\left[\bar{\boldsymbol{A}}\right] - \mathbb{E}\left[\widetilde{\boldsymbol{A}}\right]\right)\boldsymbol{v}\right| \lesssim \frac{M^{1/2}}{\tau}$$

for any vector $\boldsymbol{v} \in \mathcal{S}^{d-1}$. Therefore,

$$\left\|\mathbb{E}\left[\widetilde{\boldsymbol{A}}\right] - \mathbb{E}\left[\bar{\boldsymbol{A}}\right]\right\|_{\text{op}} \lesssim \frac{M^{1/2}}{\tau},$$

as desired.

Proof of Lemma 8

Substituting in the definitions $\bar{\boldsymbol{A}} = \bar{\gamma}^2 \boldsymbol{a}\boldsymbol{a}^\top$ and $\widetilde{\boldsymbol{A}} = \widetilde{\gamma}^2 \boldsymbol{a}\boldsymbol{a}^\top$, we have

$$\langle \bar{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star \rangle \bar{\boldsymbol{A}} - \langle \widetilde{\boldsymbol{A}}, \boldsymbol{\Sigma}^\star \rangle \widetilde{\boldsymbol{A}} = \left(\bar{\gamma}^4 - \widetilde{\gamma}^4\right) \left(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}\right) \boldsymbol{a}\boldsymbol{a}^\top.$$

Therefore, our goal is to bound the operator norm

$$\left\|\left(\bar{\gamma}^4 - \widetilde{\gamma}^4\right) \left(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}\right) \boldsymbol{a}\boldsymbol{a}^\top\right\|_{\text{op}} = \sup_{\boldsymbol{v} \in \mathcal{S}^{d-1}} \left|\boldsymbol{v}^T \left(\bar{\gamma}^4 - \widetilde{\gamma}^4\right) \left(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}\right) \boldsymbol{a}\boldsymbol{a}^\top \boldsymbol{v}\right|.$$

Similar to the proof of Lemma 7, we fix any vector $\boldsymbol{v} \in \mathcal{S}^{d-1}$. Again, note that $\boldsymbol{v}^\top \boldsymbol{a} \sim \mathcal{N}(0,1)$ and denote $G \sim \mathcal{N}(0,1)$. We have

$$\begin{aligned}
\left|\boldsymbol{v}^\top \mathbb{E}\left[\left(\bar{\gamma}^4 - \widetilde{\gamma}^4\right) \left(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}\right) \boldsymbol{a}\boldsymbol{a}^\top\right] \boldsymbol{v}\right| &\overset{\text{(i)}}{=} \mathbb{E}\left[\left(\bar{\gamma}^4 - \widetilde{\gamma}^4\right) \left(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}\right) G^2\right] \\
&= \mathbb{E}\left[\left(\bar{\gamma}^2 + \widetilde{\gamma}^2\right) \left(\bar{\gamma}^2 - \widetilde{\gamma}^2\right) \left(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}\right) G^2\right] \\
&\overset{\text{(ii)}}{\leq} \mathbb{E}\left[2\bar{\gamma}^2 \left(\bar{\gamma}^2 - \widetilde{\gamma}^2\right) \left(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}\right) G^2\right] \\
&\overset{\text{(iii)}}{=} 2\mathbb{E}\left[(y + \bar{\eta}) \left(\bar{\gamma}^2 - \widetilde{\gamma}^2\right) G^2\right] \\
&\overset{\text{(iv)}}{\leq} 2(y + \eta^\uparrow) \mathbb{E}\left[\left(\bar{\gamma}^2 - \widetilde{\gamma}^2\right) G^2 \mathbf{1}\{\gamma^2 \geq \tau\}\right]
\end{aligned}$$

where steps (i) and (ii) are true because $\bar{\gamma}^2 \geq \widetilde{\gamma}^2$ from (Equation TP2), step (iii) follows from the definition $\bar{\gamma}^2 = \frac{y+\bar{\eta}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}$, and step (iv) follows from (Equation TP3) and the definition of $\eta^\uparrow$ as the upper bound on the noise η.

The rest of the proof follows the exact steps of the proof of Lemma 7 in Section B.4.

Therefore, we have the bound

$$\left\|\mathbb{E}\left[\left(\bar{\gamma}^4-\widetilde{\gamma}^4\right)\left(\boldsymbol{a}^\top\boldsymbol{\Sigma}^\star\boldsymbol{a}\right)\boldsymbol{a}\boldsymbol{a}^\top\right]\right\|_{\mathrm{op}}\lesssim\frac{y^\uparrow M^{1/2}}{\tau},$$

as desired.

Proof of Lemma 9

The proof follows the steps as in the proof of Lemma 6, and we describe the difference of the two proofs. We again apply Bernstein's inequality.

Proving a Bernstein condition. We prove a Bernstein condition with $u_1 = c_1(y+\eta^\uparrow)^2$ and $u_2 = c_2(y+\eta^\uparrow)\tau$. Namely, for every integer $p \geq 2$, we have (cf. (Equation B.13) in Lemma 6)

$$\mathbb{E}\left[\left|\boldsymbol{v}^\top\langle\widetilde{\boldsymbol{A}},\boldsymbol{\Sigma}^\star\rangle\widetilde{\boldsymbol{A}}\boldsymbol{v}\right|^p\right]\leq\frac{p!}{2}u_1u_2^{p-2}. \tag{B.19}$$

To show (Equation B.19), we plug in $\widetilde{\boldsymbol{A}} = \widetilde{\gamma}^2\boldsymbol{a}\boldsymbol{a}^\top$ and have

$$\begin{aligned}\mathbb{E}\ \left|\boldsymbol{v}^\top\langle\widetilde{\boldsymbol{A}},\boldsymbol{\Sigma}^\star\rangle\widetilde{\boldsymbol{A}}\boldsymbol{v}\right|^p &= \mathbb{E}\ \left(\widetilde{\gamma}^2\boldsymbol{a}^\top\boldsymbol{\Sigma}^\star\boldsymbol{a}\right)^p\cdot\left|\boldsymbol{v}^\top\widetilde{\boldsymbol{A}}\boldsymbol{v}\right|^p\\ &\overset{\text{(i)}}{\leq}\mathbb{E}\ \left(\bar{\gamma}^2\boldsymbol{a}^\top\boldsymbol{\Sigma}^\star\boldsymbol{a}\right)^p\cdot\left|\boldsymbol{v}^\top\widetilde{\boldsymbol{A}}\boldsymbol{v}\right|^p\\ &\overset{\text{(ii)}}{=}\mathbb{E}\ (y+\bar{\eta})^p\cdot\left|\boldsymbol{v}^\top\widetilde{\boldsymbol{A}}\boldsymbol{v}\right|^p\\ &\overset{\text{(iii)}}{\leq}(y+\eta^\uparrow)^p\cdot\mathbb{E}\ \left|\boldsymbol{v}^\top\widetilde{\boldsymbol{A}}\boldsymbol{v}\right|^p,\end{aligned} \tag{B.20}$$

where step (i) follows from (Equation TP2), step (ii) follows from the definition $\bar{\gamma}^2 = \frac{y+\bar{\eta}}{\boldsymbol{a}^\top\boldsymbol{\Sigma}^\star\boldsymbol{a}}$, and step (iii) follows from the definition of $\eta^\uparrow$ as the upper bound on the noise η. Substituting in (Equation B.13) from Lemma 6 to bound the term $\mathbb{E}\ \left|\boldsymbol{v}^\top\widetilde{\boldsymbol{A}}\boldsymbol{v}\right|^p$ in (Equation B.20) completes the proof of the Bernstein condition (Equation B.19).

Applying Bernstein's inequality. The rest of the proof follows in the same manner as the proof of Lemma 6 in Section B.4, with an additional factor of $(y + \eta^{\uparrow})$. We have

$$\left\| \mathbb{E}\left[\langle \widetilde{\boldsymbol{A}}, \boldsymbol{\Sigma}^{\star} \rangle \widetilde{\boldsymbol{A}} \right] - \frac{1}{n} \sum_{i=1}^{n} \langle \widetilde{\boldsymbol{A}}_i, \boldsymbol{\Sigma}^{\star} \rangle \widetilde{\boldsymbol{A}}_i \right\|_{\mathrm{op}} \lesssim y^{\uparrow} \left(\sqrt{\frac{M^{1/2} t}{n}} + \frac{\tau t}{n} \right)$$

with probability at least $1 - 2 \cdot 9^d \cdot \exp(-t)$, as desired.

Proof of Lemma 10

Recall that by definition $\bar{\boldsymbol{A}} = \bar{\gamma}^2 \boldsymbol{a}\boldsymbol{a}^\top = \frac{y + \bar{\eta}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \boldsymbol{a}\boldsymbol{a}^\top$. We have

$$\begin{aligned} \left\| \mathbb{E}\left[\bar{\eta} \bar{\boldsymbol{A}} \right] \right\|_{\mathrm{op}} &= \left\| \mathbb{E}\left[\bar{\eta}(y + \bar{\eta}) \frac{\boldsymbol{a}\boldsymbol{a}^\top}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \right] \right\|_{\mathrm{op}} \\ &= \left\| \mathbb{E}\left[\bar{\eta}(y + \bar{\eta}) \right] \cdot \mathbb{E}\left[\frac{\boldsymbol{a}\boldsymbol{a}^\top}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \right] \right\|_{\mathrm{op}} \\ &= \frac{\sigma_\eta^2}{m} \cdot \left\| \mathbb{E}\left[\frac{\boldsymbol{a}\boldsymbol{a}^\top}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \right] \right\|_{\mathrm{op}}. \end{aligned} \tag{B.21}$$

To bound the operator norm term in (Equation B.21), we apply Item (b) of Lemma 4 in Section B.3.4. For any matrix $\boldsymbol{U}$, we have

$$\mathbb{E}\left[\frac{\boldsymbol{a}^\top \boldsymbol{U} \boldsymbol{a}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \right] \lesssim \frac{1}{\sigma_r r} \|\boldsymbol{U}\|_*. \tag{B.22}$$

Note that $\frac{\boldsymbol{a}\boldsymbol{a}^\top}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}}$ is symmetric positive semidefinite, so we have

$$\begin{aligned} \left\| \mathbb{E}\left[\frac{\boldsymbol{a}\boldsymbol{a}^\top}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \right] \right\|_{\mathrm{op}} &= \sup_{\boldsymbol{v} \in \mathcal{S}^{d-1}} \left| \boldsymbol{v}^\top \mathbb{E}\left[\frac{\boldsymbol{a}\boldsymbol{a}^\top}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \right] \boldsymbol{v} \right| \\ &= \sup_{\boldsymbol{v} \in \mathcal{S}^{d-1}} \mathbb{E}\left[\frac{\boldsymbol{a}^\top (\boldsymbol{v}\boldsymbol{v}^\top) \boldsymbol{a}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \right] \\ &\overset{\text{(i)}}{\lesssim} \frac{1}{\sigma_r r} \sup_{\boldsymbol{v} \in \mathcal{S}^{d-1}} \|\boldsymbol{v}\boldsymbol{v}^\top\|_* \\ &\overset{\text{(ii)}}{=} \frac{1}{\sigma_r r}, \end{aligned} \tag{B.23}$$

where step (i) is true by substituting in (Equation B.22) with $\boldsymbol{U} = \boldsymbol{v}\boldsymbol{v}^T$, and step (ii) is true because $\boldsymbol{v}$ is unit norm, and hence $\|\boldsymbol{v}\boldsymbol{v}^\top\|_* = 1$. Substituting (Equation B.23) back to (Equation B.21), we have

$$\left\|\mathbb{E}\left[\bar{\eta}\bar{\boldsymbol{A}}\right]\right\|_{\mathrm{op}} \lesssim \frac{1}{\sigma_r r} \cdot \frac{\nu_\eta^2}{m},$$

as desired.

B.5 Proof of Proposition 4

We analyze the term $\frac{1}{n}\sum\limits_{i=1}^{n}\langle\widetilde{\boldsymbol{A}}_i, \boldsymbol{U}\rangle^2$ from (Equation 3.21). Recall from the definition of $\widetilde{\boldsymbol{A}}$ that for any $i = 1, \ldots, n$,

$$\widetilde{\boldsymbol{A}}_i = \widetilde{\gamma}_i^2 \boldsymbol{a}_i \boldsymbol{a}_i^\top = \left(\frac{y + \bar{\eta}_i}{\boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i} \wedge \tau\right) \boldsymbol{a}_i \boldsymbol{a}_i^\top,$$

so we have

$$\langle\widetilde{\boldsymbol{A}}_i, \boldsymbol{U}\rangle^2 = \left(\frac{y + \bar{\eta}_i}{\boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i} \wedge \tau\right)^2 \left(\boldsymbol{a}_i^\top \boldsymbol{U} \boldsymbol{a}_i\right)^2. \tag{B.24}$$

From (Equation B.24), we have that for any matrix $\boldsymbol{U}$, the term $\sum\limits_{i=1}^{n}\langle\widetilde{\boldsymbol{A}}_i, \boldsymbol{U}\rangle^2$ is nondecreasing in τ when $\tau > 0$. Defining a random matrix

$$\widetilde{\boldsymbol{A}}^{\tau'} := \left(\frac{y + \bar{\eta}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \wedge \tau'\right) \boldsymbol{a}\boldsymbol{a}^\top, \tag{B.25}$$

for any $\tau' \in (0, \tau]$, we have

$$\frac{1}{n}\sum_{i=1}^{n}\langle\widetilde{\boldsymbol{A}}_i, \boldsymbol{U}\rangle^2 \geq \frac{1}{n}\sum_{i=1}^{n}\langle\widetilde{\boldsymbol{A}}_i^{\tau'}, \boldsymbol{U}\rangle^2, \tag{B.26}$$

where for every $i = 1, \dots, n$, matrix $\widetilde{\boldsymbol{A}}_i^{\tau'}$ is formed with the same realizations of random quantities $\boldsymbol{a}_i$ and $\bar{\eta}_i$ as $\widetilde{\boldsymbol{A}}_i$. The two matrices only differ in choice of truncation threshold: τ' instead of τ. As a result, for the rest of the proof, we lower bound $\frac{1}{n}\sum_{i=1}^{n}\langle \widetilde{\boldsymbol{A}}_i^{\tau'}, \boldsymbol{U}\rangle^2$ for an appropriate choice of τ' to be specified later. To proceed, we use a small-ball argument [206, 207] based on the following lemma.

Lemma 11 ([207, Proposition 5.1], adapted to our notation)**.** *Let $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n \in \mathbb{R}^{d\times d}$ be i.i.d. copies of a random matrix $\boldsymbol{X} \in \mathbb{R}^{d\times d}$. Let $E \subset \mathbb{R}^{d\times d}$ be a subset of matrices. Let $\xi > 0$ and $Q > 0$ be real values such that for every matrix $\boldsymbol{U} \in E$, the marginal tail condition holds:*

$$\mathbb{P}\left(|\langle \boldsymbol{X}, \boldsymbol{U}\rangle| \geq 2\xi\right) \geq Q. \tag{B.27}$$

Define the Rademacher width as

$$W := \mathbb{E}\left[\sup_{\boldsymbol{U}\in E} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \langle \boldsymbol{X}_i, \boldsymbol{U}\rangle\right],$$

where $\varepsilon_1, \dots, \varepsilon_n$ are i.i.d. Rademacher random variables independent of $\{\boldsymbol{X}_i\}_{i\in[n]}$. Then for any $t > 0$, we have

$$\inf_{\boldsymbol{U}\in E} \left(\frac{1}{n}\sum_{i=1}^{n}\langle \boldsymbol{X}_i, \boldsymbol{U}\rangle^2\right)^{1/2} \geq \xi(Q - t) - 2W.$$

with probability at least $1 - \exp\left(-\frac{nt^2}{2}\right)$.

Recall the error set $\mathcal{E}$ defined in (Equation 3.17). Because the claim (Equation 3.21) is invariant to scaling, it suffices to prove it for $\|\boldsymbol{U}\|_F = 1$. Correspondingly, we define the set

E as

$$\begin{aligned} E &= \mathcal{E} \cap \{\boldsymbol{U} \in \mathbb{R}^{d\times d} : \|\boldsymbol{U}\|_F = 1\} \\ &= \{\boldsymbol{U} \in \mathbb{S}^{d\times d} : \|\boldsymbol{U}\|_F = 1, \|\boldsymbol{U}\|_* \leq 4\sqrt{2r}\}. \end{aligned} \tag{B.28}$$

We invoke Lemma 11 with set E defined above, $\boldsymbol{X}_i = \widetilde{\boldsymbol{A}}_i^{\tau'}$, $\xi = \frac{c_1}{2}\left(\frac{\mu_y}{\operatorname{tr}(\boldsymbol{\Sigma}^\star)} \wedge \tau'\right)$, and $Q = c_2$, where μ_y is the median of $\bar{\eta}$ and c_1 and c_2, are constants to be specified later. The rest of the proof is comprised of two steps. We first verify that our choices for ξ and Q are valid for establishing the marginal tail condition (Equation B.27). We then bound the Rademacher width W above. The following lemma verifies our choices for ξ and Q.

Lemma 12. *Consider any $\tau' \in (0, \tau]$. There exist absolute constants $c_1, c_2 > 0$ such that for every $\boldsymbol{U} \in E$, we have*

$$\mathbb{P}\left(\left|\langle \widetilde{\boldsymbol{A}}^{\tau'}, \boldsymbol{U}\rangle\right| \geq c_1\left(\frac{\mu_y}{\operatorname{tr}(\boldsymbol{\Sigma}^\star)} \wedge \tau'\right)\right) \geq c_2.$$

The proof of Lemma 12 is presented in Section B.5. We now turn to the second step of the proof, which is bounding the Rademacher width W. The next lemma characterizes this width.

Lemma 13. *Consider any $\tau' \in (0, \tau]$. Let $\widetilde{\boldsymbol{A}}_1^{\tau'}, \ldots, \widetilde{\boldsymbol{A}}_n^{\tau'} \in \mathbb{R}^{d\times d}$ be i.i.d. copies of the random matrix $\widetilde{\boldsymbol{A}}^{\tau'} \in \mathbb{R}^{d\times d}$ defined in* (Equation B.25). *Let E be the set defined in* (Equation B.28). *Then, there exists some absolute constants c_1 and c_2 such that if $n \geq c_1 d$, then we have*

$$\mathbb{E}\left[\sup_{\boldsymbol{U}\in E} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \langle \widetilde{\boldsymbol{A}}_i^{\tau'}, \boldsymbol{U}\rangle\right] \leq c_2 \tau' \sqrt{\frac{rd}{n}}.$$

The proof of Lemma 13 is presented in Section B.5. Lemma 12 establishes the marginal tail condition for Lemma 11, and Lemma 13 upper bounds the Rademacher width. We now invoke Lemma 11 and substitute in the upper bound for the Rademacher width W. For some constant c_4, if $n \geq c_4 d$, we have that with probability at least $1 - \exp\left(-\frac{nt^2}{2}\right)$,

$$\inf_{\boldsymbol{U}\in E}\left(\frac{1}{n}\sum_{i=1}^{n}\langle\widetilde{\boldsymbol{A}}_i,\boldsymbol{U}\rangle^2\right)^{1/2} \overset{\text{(i)}}{\geq} \inf_{\boldsymbol{U}\in E}\left(\frac{1}{n}\sum_{i=1}^{n}\langle\widetilde{\boldsymbol{A}}_i^{\tau'},\boldsymbol{U}\rangle^2\right)^{1/2}$$
$$\geq \frac{c_1}{2}\left(\frac{\mu_y}{\operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)}\wedge\tau'\right)(c_2 - t) - c_3\tau'\sqrt{\frac{rd}{n}},$$

where step (i) is true due to the monotonicity property (Equation B.26). We set $\tau' = \frac{\mu_y}{\operatorname{tr}(\boldsymbol{\Sigma}^\star)}$, where recall that μ_y is the median of the random quantity $y + \bar{\eta}$. By the assumption $\tau \geq \frac{\mu_y}{\operatorname{tr}(\boldsymbol{\Sigma}^\star)}$, this choice of τ' satisfies $\tau' \leq \tau$. Setting $t = \frac{c_2}{2}$, we have that with probability at least $1 - \exp\left(-\frac{c_2^2 n}{8}\right)$,

$$\inf_{\boldsymbol{U}\in E}\frac{1}{n}\left(\sum_{i=1}^{n}\langle\widetilde{\boldsymbol{A}}_i^{\tau'},\boldsymbol{U}\rangle^2\right)^{1/2} \geq \frac{c_1c_2}{4}\frac{\mu_y}{\operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)} - c_3\frac{\mu_y}{\operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)}\sqrt{\frac{rd}{n}}.$$

Recall from the definition of E (Equation B.28) that $\|\boldsymbol{U}\|_F = 1$. As a result, if $n \geq \max\left\{\left(\frac{4c_3}{c_1c_2}\right)^2, c_4\right\} rd$, we have

$$\inf_{\boldsymbol{U}\in\mathcal{E}}\frac{1}{n}\sum_{i=1}^{n}\langle\widetilde{\boldsymbol{A}}_i,\boldsymbol{U}\rangle^2 \geq \left(\frac{c_1c_2}{4}\frac{\mu_y}{\operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)}\right)^2\|\boldsymbol{U}\|_F^2$$

with probability at least $1 - \exp\left(-\frac{c_2^2 n}{8}\right)$. We conclude by setting $\kappa_{\mathcal{L}} = \left(\frac{c_1c_2}{4}\right)^2$, $c = \frac{c_2^2}{8}$, and $C = \max\left\{\left(\frac{4c_3}{c_1c_2}\right)^2, c_4\right\}$ in Proposition 4.

Proof of Lemma 12

We fix any $\boldsymbol{U} \in E$. Recall that μ_y denotes the median of $y + \bar{\eta}$. Let $\mathcal{G}$ be the event that $y + \bar{\eta} \geq \mu_y$, which occurs with probability $\frac{1}{2}$. For any $\xi > 0$, because the averaged noise $\bar{\eta}$

and sensing vector $\boldsymbol{a}$ are independent, we have

$$\begin{aligned}
\mathbb{P}\left(\left|\langle \widetilde{\boldsymbol{A}}^{\tau'}, \boldsymbol{U}\rangle\right| \geq \xi\right) &\overset{\text{(i)}}{=} \mathbb{P}\left(\left(\frac{y+\bar{\eta}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \wedge \tau'\right) \cdot \left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right| \geq \xi\right) \\
&= \mathbb{P}\left(\left(\frac{y+\bar{\eta}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \wedge \tau'\right) \cdot \left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right| \geq \xi \,\middle|\, \mathcal{G}\right) \cdot \mathbb{P}\left(\mathcal{G}\right) \\
&= \frac{1}{2}\mathbb{P}\left(\left(\frac{y+\bar{\eta}}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \wedge \tau'\right) \cdot \left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right| \geq \xi \,\middle|\, \mathcal{G}\right) \\
&\overset{\text{(ii)}}{\geq} \frac{1}{2}\mathbb{P}\left(\left(\frac{\mu_y}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \wedge \tau'\right) \cdot \left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right| \geq \xi\right),
\end{aligned} \tag{B.29}$$

where step (i) is true by plugging in the definition of $\widetilde{\boldsymbol{A}}^{\tau'}$, and step (ii) is true by the definition of the event $\mathcal{G}$. We proceed by bounding the terms in (Equation B.29) separately.

Lower bound on $\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right|$. We use the approach from [115, Section 4.1]. By Paley-Zygmund inequality,

$$\mathbb{P}\left(\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right|^2 \geq \frac{1}{2}\mathbb{E}\left[\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right|^2\right]\right) \geq \frac{1}{4}\frac{\left(\mathbb{E}\left[\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right|^2\right]\right)^2}{\mathbb{E}\left[\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right|^4\right]} \tag{B.30}$$

We now analyze the terms in (Equation B.30). As noted in [115, Section 4.1], there exists some constant $c_1 > 0$ such that for any matrix $\boldsymbol{U}$ with $\|\boldsymbol{U}\|_F = 1$,

$$\mathbb{E}\left[\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right|^2\right] \geq 1 \quad \text{and} \quad \mathbb{E}\left[\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right|^4\right] \leq c_1\left(\mathbb{E}\left[\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right|^2\right]\right)^2. \tag{B.31}$$

Note that by the definition of the set E, every matrix $\boldsymbol{U} \in E$ satisfies $\|\boldsymbol{U}\|_F = 1$. Utilizing inequalities (Equation B.30) and (Equation B.31), there exists positive constant $c_2 > 0$ such that

$$\mathbb{P}\left(\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right| \geq \frac{1}{2}\right) \geq c_2. \tag{B.32}$$

Upper bound on $\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}$. By Hanson-Wright inequality [208, Theorem 1.1], there exist some positive absolute constants c_3 and c_4 such that for any $t > 0$, we have

$$\mathbb{P}\left(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a} \leq c_3 \left(\operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right) + \|\boldsymbol{\Sigma}^\star\|_F \sqrt{t} + \|\boldsymbol{\Sigma}^\star\|_{\text{op}}\, t\right)\right) \geq 1 - 2\exp\left(-c_4 t\right).$$

We set $t = -\frac{1}{c_4}\log(\frac{c_2}{4})$ so that $2\exp\left(-c_4 t\right) = \frac{c_2}{2}$. Since $\boldsymbol{\Sigma}^\star$ is symmetric positive semidefinite, we have

$$\|\boldsymbol{\Sigma}^\star\|_F \leq \operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right) \quad \text{and} \quad \|\boldsymbol{\Sigma}^\star\|_{\text{op}} \leq \operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)$$

As a result, we have that there exists some constant $c_5 > 0$ such that

$$\mathbb{P}\Big(\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a} \leq c_5 \operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)\Big) \geq 1 - \frac{c_2}{2}. \tag{B.33}$$

Substituting the two bounds back to (Equation B.29). By a union bound of (Equation B.32) and (Equation B.33), we have

$$\begin{aligned} &\mathbb{P}\left(\left(\frac{\mu_y}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \wedge \tau'\right) \cdot \left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right| \geq \frac{1}{2}\left(\frac{\mu_y}{c_5 \operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)} \wedge \tau'\right)\right) \\ &\quad \geq \mathbb{P}\left(\frac{\mu_y}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \wedge \tau' \geq \frac{\mu_y}{c_5 \operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)} \wedge \tau'\right) + \mathbb{P}\left(\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right| \geq \frac{1}{2}\right) - 1 \\ &\quad \geq \mathbb{P}\left(\frac{\mu_y}{\boldsymbol{a}^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}} \geq \frac{\mu_y}{c_5 \operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)}\right) + \mathbb{P}\left(\left|\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{U}\rangle\right| \geq \frac{1}{2}\right) - 1 \geq \frac{c_2}{2} \end{aligned} \tag{B.34}$$

Combining (Equation B.34) and (Equation B.29), and redefining constant c_2 appropriately, we have

$$\mathbb{P}\left(\left|\langle \widetilde{\boldsymbol{A}}^{\tau'}, \boldsymbol{U}\rangle\right| \geq \frac{1}{2}\left(\frac{\mu_y}{\operatorname{tr}\left(\boldsymbol{\Sigma}^\star\right)} \wedge \tau'\right)\right) \geq c_2,$$

as desired.

Proof of Lemma 13

We begin by noting that for any matrix $\boldsymbol{U} \in E$,

$$\begin{aligned}\mathbb{E}\left[\sup_{\boldsymbol{U}\in E} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \langle \widetilde{\boldsymbol{A}}_i^{\tau'}, \boldsymbol{U}\rangle\right] &\overset{\text{(i)}}{\leq} \mathbb{E}\left[\sup_{\boldsymbol{U}\in E} \left\|\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \widetilde{\boldsymbol{A}}_i^{\tau'}\right\|_{\text{op}} \cdot \|\boldsymbol{U}\|_*\right] \\ &\overset{\text{(ii)}}{\leq} 4\sqrt{2r}\cdot\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \widetilde{\boldsymbol{A}}_i^{\tau'}\right\|_{\text{op}}, \end{aligned} \tag{B.35}$$

where step (i) follows from Hölder's inequality, and step (ii) follows from the fact that $\|\boldsymbol{U}\|_* \leq 4\sqrt{2r}$ from the definition of the set E. It remains to bound the expectation of the operator norm in (Equation B.35). We follow the standard covering arguments in [209, Section 5.4.1], [207, Section 8.6], [115, Section 4.1], with a slight modification to accommodate the bounded term $\left(\frac{y+\bar{\eta}_i}{\boldsymbol{a}_i^\top \boldsymbol{\Sigma}^\star \boldsymbol{a}_i} \wedge \tau'\right)$ that appears in each of the matrices $\widetilde{\boldsymbol{A}}_i^{\tau'}$. As a result, there exist universal constants c_1 and c_2 such that if n satisfies $n \geq c_1 d$, then we have

$$\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \widetilde{\boldsymbol{A}}_i^{\tau'}\right\|_{\text{op}}\right] \leq c_2\tau'\sqrt{\frac{d}{n}}.$$

We conclude by re-defining c_2 appropriately.

APPENDIX C
SUPPLEMENTARY MATERIAL FOR CHAPTER 5

C.1 Experimental setup

C.1.1 Dataset details

In our experiments, we utilize the following datasets. We report the licenses for all datasets that publicly list them.

- CelebAMask-HQ [191]. License: non-commercial research and educational purposes.
- Car-Parts [192].
- DeepFashion-MultiModal [194, 193]. License: non-commercial research purposes.
- SHHQ [189]. License: CC0 and free for research use.
- Cityscapes [195]. License: on-commercial research and educational purposes.

We also utilize pre-trained StyleGAN2 and ReStyle models. In the face and car domain, these models were trained on the following datasets:

- FHHQ [47]. License: Creative Commons BY-NC-SA 4.0 license by NVIDIA Corporation.
- LSUN [210].
- Stanford Cars [211]. License: non-commercial research and educational purposes.

To make the DeepFashion-MultiModal segmentation masks compatible with StyleGAN-Human, we first used the segmentation mask to determine the background for each image and set the background to white. We then re-sized each image to the same size SHHQ images.

C.1.2 Segmentation mask class collapse

Consistent with prior works [166], we collapse the original labels in each dataset into a smaller number of labeled parts. For CelebAMask-HQ dataset, we remove any distinction between left/right in a number of parts (e.g., ears, eyes, eyebrows). Furthermore, we form one mouth part consisting of upper/lower lips and mouth. Finally, we collapse all accessories and clothing into background. See Table C.1a for exact class collapse mapping. In long-tail experiments, we un-collapse the relevant long-tail classes (glasses and hats) and consider them separate classes.

For the Car-Parts dataset, we remove any distinction between left/right and front/back for parts such as doors, lights, bumpers, and mirrors. We also merge trunks and tailgates to be the same class. See Table C.1b for exact class collapse mapping.

For DeepFashion-MultiModal, we consider two degrees of class collapse. In the first, we consider the following ten classes, with original classes included in parentheses: tops (tops and ties), outerwear, dresses (dresses, skirts, rompers), bottoms (pants, leggings, belts), face (face, glasses, earrings), skin (skin, neckwear, rings, wrist accessories, gloves, necklaces), footwear (shoes and socks), bags, and hair (hair and headwear). In the second, we further collapse the classes by including outerwear in tops and bags as background. See Table C.1c and Table C.1d for exact class collapse mappings.

For Cityscapes, we utilize the eight groups listed on the Cityscapes official website as our classes, with slight modifications. We consider parts labeled sidewalk, parking, and rail track as a part of the void class. See Table C.1f for exact class collapse mapping.

C.1.3 Training setup

All experiments were run on V100 GPUs using Amazon Web Services (AWS) P3dn.24xlarge instances. Each MLP in the label generator ensemble was trained with the same parameters for all domains and tasks. Each MLP was trained for ~ 4 epochs via the Adam optimizer [212] with learning rate 0.001 and batch size 64. For all results presented in Table 5.1 and

Collapsed label (8)	CelebAMask-HQ original labels
Background	Background (0), hat (14), earring (15), necklace (16), neck (17), clothes (18)
Skin	Skin (1)
Nose	Nose (2)
Eyes	Left eye (3), right eye (4), glasses (5)
Eyebrows	Left eyebrow (6), right eyebrow (7)
Ears	Left ear (8), right ear (9)
Mouth	Mouth (10), upper lip (11), lower lip (12)
Hair	Hair (13)

(a)

Collapsed label (10)	Car-Parts original labels
Background	Background(0)
Bumper	Back bumper (1), front bumper (7)
Back window	Back glass (3)
Doors	Back left door (3), back right door (5), front left door (9), front right door (11)
Lights	Back left light (4), back right light (6), front left light (10), front right light (12)
Windshield	Front glass (8)
Hood	Hood (13)
Mirror	Left mirror (14), right mirror (15)
Trunk	Tailgate (16), trunk (17)
Wheel	Wheel (18)

(b)

Collapsed label (10)	DeepFashion-MM original labels
Background	Background(0)
Top	Top (1), tie (23)
Outerwear	Outerwear (2)
Dress	Skirt (3), dress (4), romper (21)
Bottoms	Pants (5), leggings (6), belt (10)
Face	Glasses (8), face (14), earring (22)
Skin	Neckwear (9), skin (15), ring (16), Wrist accessories (17), gloves (19), necklace (20)
Footwear	Footwear (11), socks (18)
Bags	Bags (12)
Hair	Headwear (7), hair (13)

(c)

Collapsed label (8)	DeepFashion-MM original labels
Background	Background(0), bags(12)
Top	Top (1), tie (23), outerwear (2)
Dress	Skirt (3), dress (4), romper (21)
Bottoms	Pants (5), leggings (6), belt (10)
Face	Glasses (8), face (14), earring (22)
Skin	Neckwear (9), skin (15), ring (16), Wrist accessories (17), gloves (19), necklace (20)
Footwear	Footwear (11), socks (18)
Hair	Headwear (7), hair (13)

(d)

(e) Mapping from collapsed class label to original class label in faces (a), cars (b), full-body human poses (c), (d), and urban driving scenes (e) domains. Original class numbers provided for each original class label name in parentheses.

Collapsed label (8)	Cityscapes (Fine annotations) original labels
Void	Unlabeled (0), ego vehicle (1), rectification border (2), out of ROI (3), static (4), dynamic (5), ground (6), sidewalk (8), parking (9), rail track (10)
Road	Road (7)
Construction	Building (11), wall (12), fence (13), guard rail (14), bridge (15), tunnel (16)
Object	pole (17), polegroup (18), traffic light (19), traffic sign (20)
Nature	Vegetation (21), terrain (22)
Sky	Sky (23)
Human	Person (24), rider (25)
Vehicle	UCar (26), truck (27), bus (28), caravan (29), trailer (30), train (31), motorcycle (32), bicycle (33), license plate (-1)

(f)

Table 5.2, the labeled images used to train the label generator were chosen at random. For long-tail experiments (Section 5.4.5), images with the long-tail part were identified. Then, the labeled training images were selected at random from the identified images.

Prior to training the downstream network, we filter out the top 10% most uncertain synthetically generated images, except for the long-tail experiments. No filtering is performed for long-tail experiments to ensure that images with long-tail parts, which are more likely to be "uncertain", are included in the training set for the downstream network. To train the downstream network, we again utilize the Adam optimizer [212] with learning rate 0.001 and batch size 64. We train ReStyle [173] on the set of labeled training images randomly selected from SHHQ [194, 213] and Cityscapes [195] for the full-body human poses and urban driving scene domains, respectively. We use default settings found in the ReStyle repository.

C.1.4 GAN inversion setup

For the full-body human poses and urban driving scenes domains, we train ReStyle with the candidate training examples. Our framework only uses GAN inversion to obtain latent codes for training the label generator. Training on the candidate training examples thus ensures that ReStyle optimally reconstructs these latent codes. For faces and cars, this procedure is not necessary because ReStyle optimally reconstructs the latent codes of training examples without training. For the optimization-based finetuning, we utilize $c_{reg} = 0.5$ and $\lambda_{\ell_2} = 0.1$ for all domains. We run 300 optimization steps for the car domain, 500 iterations for the face and urban driving scenes domains, and 2,000 iterations for the human full-body poses domain. See Section C.2 for ablations on GAN inversion optimization steps.

C.1.5 Label generator architecture

For all experiments, we utilize an ensemble of two layer MLPs with ReLU activations and batch normalizations for our label generator. We sweep the combination of layer widths and report the performance associated with the best performing combination for each domain and number of labeled training images. See Section C.2 for ablations on layer widths. Below, we report the combination of label generator sizes that produced the best performance. (x, y) indicates that a network with first hidden layer of width x and second hidden layer of width y was used.

Faces For segmentation, we utilize layer sizes of (256, 32) for 50 training images and (512, 64) for 16 training images. For keypoints, we utilize (512, 32) for PCK-0.1, PCK-0.05, and PCK-0.02 with 50 training images. For 16 training images, we utilize (512, 64) for PCK-0.1 and (512, 32) for PCK-0.05 and PCK-0.02.

Cars For segmentation, we utilize (512, 256) for both 50 training images and 16 training images.

Full-body human poses For segmentation, we utilize (1024, 32) and (2048, 64) for 50 training images in the 8 class and 10 class settings and (2048, 64) and (2048, 128) for 16 training images in the 8 class and 10 class settings. For keypoints, we utilize (512, 128), (256, 128), and (128, 64) for PCK-0.1, PCK-0.05, and PCK-0.02 with 50 training images. For 16 training images, we utilize (512, 256) for all three PCK thresholds.

Urban driving scenes For segmentation, we utilize (512, 64) for both 50 and 16 training images. For depth maps, we utilize (512, 256) for both 50 and 16 training images.

C.1.6 Keypoint heatmap regression

For keypoint detection experiments, we utilize a heatmap regression setup. Given an image (of size $H \times W$) and a corresponding list of K keypoints, we form a corresponding pixel-wise label for the image as follows. For each of the K keypoints, we create a $H \times W$ sized heatmap. The values of the heatmap are the values of the density of a standard two-dimensional Gaussian centered at the location of the keypoint with variance σ. We further scale the values of the heatmap by 10, so that the maximum value of the heatmap is 10. We find through hyperamater tuning that $\sigma = 25$ works well for full body while $\sigma = 5$ works well for faces. With faces, we use $\sigma = 5$ for the original sized CelebA images and then resize the mask to be of CelebAMask-HQ resolution.

The label generator and downstream task are tasked with predicting a vector of K values for each pixel. At test time, after predicting K heatmaps corresponding to the K keypoints, we take the location of the maximum element of each heatmap as the location of the keypoint. When computing the PCK metric, we only compute if a keypoint was correctly detected for visible keypoints. Information on if a particular keypoint is visible or not is provided in DeepFashion-MM, but not for CelebA.

C.2 Ablation studies

In this section, we present ablation studies that shed insights on various hyperparameters.

Hypercolumn dimension We experiment with keeping only a subset of the channels from the style block intermediate outputs from the lower resolution layers. In the StyleGAN2 generator, the first 10 style block outputs (which range from 4×4 to 128×128 resolutions) each contain 512 channels, comprising 5120 of the 6080 total channels. We quantify the effect of keeping zero or the first 64, 128, and 256 channels on the downstream task performance in the face domain. As shown in Figure C.1a, in the face domain, while utilizing only higher resolution layers degrades performance considerably, we can remove 256 of the 512 channels for the first 10 style blocks with very minimal loss in performance. This results in a hypercolumn dimension 3520, which is a 42% reduction compared to the original dimension of 6080. In our experiments, we utilize the full hypercolumn dimension, but note that due to memory considerations, utilizing a subset of the dimensions is feasible from a performance trade-off perspective.

Number of MLPs in label generator ensemble We experiment with the number of MLPs in the ensemble. We train 1, 3, 5, 7, and 10 MLPs to generate labels. As seen in Figure C.1b, in the face domain, using only 1 network results in a performance drop, but using anywhere from 3 to 7 MLPs results in performance meeting or even exceeding the performance of using all 10 MLPs. In our experiments, we utilize 10 networks to provide for more robustness in more difficult domains, such as full-body humans and urban driving scenes.

Size of MLPs in label generator ensemble We investigate whether network layer widths impact downstream performance. The original DatasetGAN framework utilizes 3-layer MLPs with intermediate dimensions of 128 and 32. We explore 7 additional combinations of layer widths: (256, 32), (256, 64), (256, 128), (512, 32), (512, 64), (512, 128), and (512, 256). As seen in Table C.3, in the face domain, for the face domain, downstream performance does not necessarily increase with increasing network widths, but remains relatively stable.

Number of labeled training images We characterize the effects of the number of labeled training images has on downstream task performance in the car domain. As emphasized throughout Chapter 5, a notable benefit HandsOff has over comparable frameworks is the ability for practitioners to increase the number of labeled training images without incurring costs of manual annotations. As observed in Figure C.1c, in the car domain, the downstream performance generally increases as the number of training images is increased, but this increase is not non-decreasing. One explanation for why is that the *composition* of the training data may have a larger impact on downstream performance than simply the number of images. This fact is explored in the long-tail experiments presented in Chapter 5. In our experiments, we report the performance with 16 labeled training images, which is the same number of training images in comparable baselines. We also report the performance of 50 labeled training images to highlight our framework's ability to accommodate more than a

$3\times$ increase in training data.

Reconstruction quality We examine the effects of GAN inversion reconstruction quality on downstream performance. Specifically, we vary the number of optimization refinement steps on the ReStyle-produced latent code. To quantitatively assess reconstruction quality, we use the value of the loss in the refinement step. As seen in Table C.4, in the car domain, as the number of optimization iterations increases, the downstream performance generally increases. However, this increase does not scale directly with reconstruction loss.

Size of generated dataset We characterize the effects of the size of the generated dataset on downstream performance. For each generated dataset size, we filter out the top 10% uncertain images. As seen in Figure C.1d, in the car domain, as the size of the dataset grows, the downstream performance generally increases. However, the performance improvement has diminishing returns, as performance improvement is most notable moving from 5,000 to 10,000 generated image-label pairs. As a result, in our experiments, we utilize dataset sizes of 10,000 to strike a balance between performance and time and computation needed to generate larger datasets.

Percent of generated dataset filtered We experiment with the percent of the dataset that is filtered out. To do so, we generate a dataset of size $10,000$ and then filter out varying percentages. As seen in Figure C.1e, in the car domain, employing filtering results in relatively similar performances. Therefore, in our experiments, we utilize a filtering percentage of 10% to strike a balance between removing highly uncertain labels and the number of image-label pairs that are used to train the downstream model.

Cityscapes downstream network finetuning. We report the effects of finetuning the trained downstream model with the original 16 or 50 labeled images used to train the label generator. As seen in Table C.5, finetuning results in increases in performance, indicating that finetuning overcomes the difficulty in producing high quality in-distribution images with a GAN.

Transfer learning pretrain dataset choice. We report the performance of the transfer learning baseline in the face and car domain when pretrained on ImageNet versus pretrained on ImageNet and COCO. As seen in Table C.6, pretraining on COCO in addition to ImageNet results in mild performance gains.

C.3 Additional results

C.3.1 Reconstructed image alignment

An underlying assumption of the HandsOff framework is that the reconstructed images resulting from GAN inversion align well semantically with the original labels. In this section, we present visual examples of reconstructed image alignment with original labels.

In the face domain, we utilize ReStyle for the encoder initialization and use 500 steps of optimization to refine the images. As seen in Figure C.2a, the reconstructions align very well with the semantic segmentation masks from CelebAMask-HQ.

In the car domain, we utilize ReStyle for the encoder initialization and use 300 steps of optimization to refine the images. As seen in Figure C.2b, the output of the ReStyle captures the overall scene very well, but struggles in preserving fine details, as shown in red circles. By utilizing the optimization based refinement step, we are able to correct for these small details. These refined images align much better with the original segmentation masks, as shown in Figure C.2b.

C.3.2 Face domain few-shot segmentation results

In this section, we compare the downstream few-shot segmentation performance of HandsOff against self-supervised approaches and diffusion-model based approaches. Namely, we compare against DDPM-Segment [214], DatasetDDPM[214], MAE[215], and SwAV[216].

DatasetDDPM and DDPM-Segment both utilize denoising diffusion probabilistic models (DDPMs). DDPM-Segment extracts intermediate network outputs from various time steps of the denoising process to form pixel-level image representations, akin to the hypercolumn representations formed from StyleGAN2 in HandsOff. Then, an ensemble of linear classifiers is trained to output a pixel-level label. DDPM-Segment is different from HandsOff in that it does not generate synthetic datasets. Instead, at inference time, the ensemble of linear classifiers is applied to the pixel-level representation of an image. DatasetDDPM simply replaces the GAN in DatasetGAN with a DDPM, forming pixel-level representations in the same manner as DDPM-Segment. For MAE and SwAV, we utilize the approach of [214] and extract intermediate layer outputs to form image representations of real images. We then train a segmenter to map from these representations to label outputs.

Table C.2: Segmentation task performance in face domain, reported in mIOU (↑). Top half: experiments performed on our splits with 8 classes. Bottom half: experiments performed on [214] splits with 19 classes. Results for DDPM-Segment, MAE, and SwAV are those as reported in Table 2 in [214].

	# labeled images	CelebAMask-HQ 8 classes
DDPM-Segment	16	0.772
DatasetDDPM	20	0.739
MAE	16	0.772
SwAV	16	0.725
HandsOff	16	**0.781**
	# labeled images	**CelebAMask-HQ 19 classes**
DDPM-Segment	20	**0.599**
MAE	20	0.578
SwAV	20	0.524
HandsOff	20	0.583

In Table C.2, we report the performance on our train/test splits with 8 classes and the train/test splits found in [214] with 19 classes. With our splits and 8 segmentation classes, HandsOff outperforms all baselines, including diffusion model-based approaches DDPM-Segment and DatasetDDPM. This is likely due to two reasons: 1. DDPM-Segment does not leverage the inherent ability of generative models to produce more samples whereas HandsOff produces a large dataset on which the downstream segmenter is trained. The volume of downstream training data compensates for the advantage that diffusion models have over GANs. 2. Unlike DatasetDDPM, HandsOff trains on annotations of real images and avoids hand annotating synthetic images, which as found by [214], when used in training, generally result in poorer performance. With the train/test splits found in [214] and 19 classes, DDPM-Segment performs slightly worse than DDPM-Segment, but outperforms the strongest self-supervised baselines (MAE [215] and SwAV [216]), as reported in [214]. We utilize the implementation of [214] to train DDPM-Segment end-to-end on our train/test splits. Furthermore, we utilize the publicly released synthetically generated datasets from DatasetDDPM to train a downstream network and evaluate on our train/test splits, as the labeled DDPM-generated images used to train DatasetDDPM were not publicly available.

C.3.3 Additional examples of generated labels

In this section, we present additional visual examples of generated images and their labels as well as examples of segmentation mask improvements in the long-tail segmentation setting.

1. In Figure C.3, we present examples in the face domain. We include examples of the predicted aggregated keypoint heatmaps used to generate the predicted keypoints. To produce the aggregated heatmap, we sum across all of the individual keypoint heatmaps.

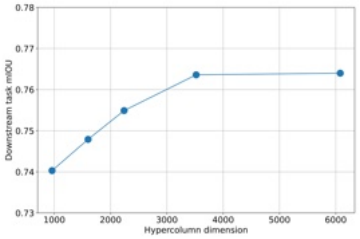
2. In Figure C.4, we present examples in the car domain.

3. In Figure C.5, we present examples in the full-body human pose domain. We again include examples of aggregated predicted heatmaps used to generate the predicted keypoints. To produce the aggregated heatmap, we sum across all of the individual keypoint heatmaps.

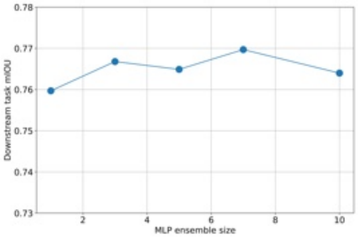
4. In Figure C.6, we present examples in the urban driving scene domain.

C.3.4 Additional examples of long-tail visualizations

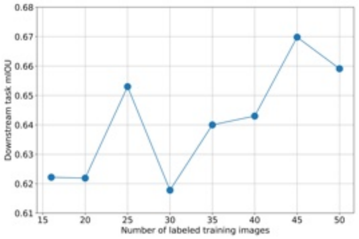
In Figure C.7a and Figure C.7b, we present examples of long-tail segmentation mask progressions and pixel-wise uncertainty measurements with glasses and hats, respectively. Uncertainty is measured by Jensen-Shannon divergence (See Section 5.3.3).



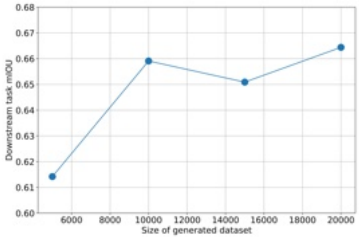
(a) Ablation for hypercolumn dimension in the face domain.



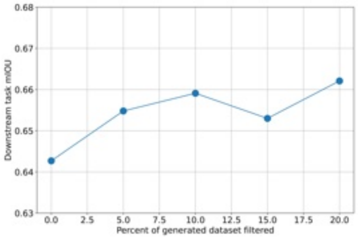
(b) Ablation for ensemble size in the face domain.



(c) Ablation for number of labeled training images in the car domain.



(d) Ablation for the size of generated dataset in the car domain.



(e) Ablation for the percent of generated dataset that is filtered in the car domain.

Table C.3: Ablation for MLP hidden layer widths in the face domain

MLP layer widths	(128, 32)	(256, 32)	(256, 64)	(256, 128)	(512, 32)	(512, 64)	(512, 128)	(512, 256)
mIOU	0.7740	**0.7859**	0.7813	0.7807	0.7828	0.7818	0.7817	0.7850

Table C.4: Ablation for GAN inversion quality in the car domain.

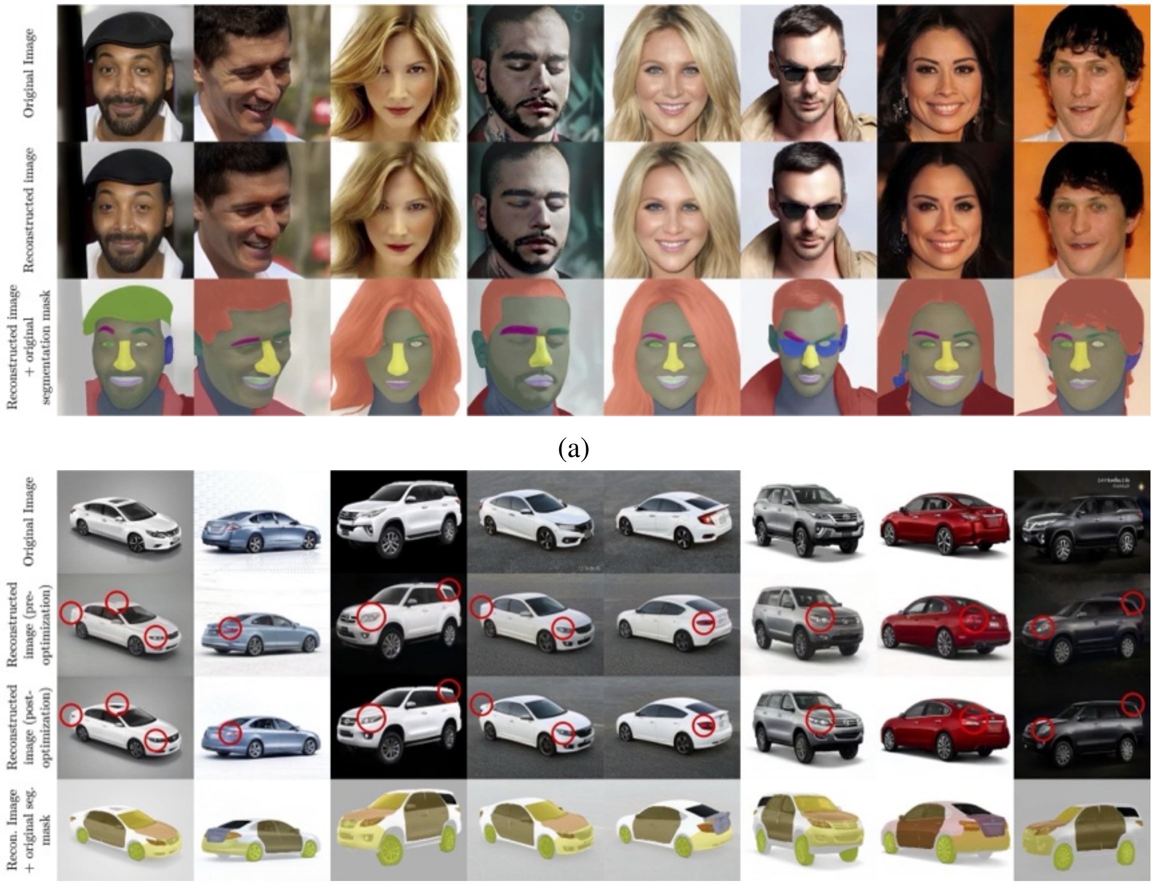
Optimization loss	3.333	2.292	2.185	2.140	2.108	2.089
Optimization iterations	0	100	200	300	400	500
mIOU	0.5735	0.6278	0.6301	**0.6679**	0.6426	0.6591

Table C.5: Ablation for Cityscapes downstream network finetuning.

# labeled images	No finetuning	Finetuning
16	0.5206	0.5510
50	0.5492	0.6047

Table C.6: Ablation for choice of pretraining dataset for transfer learning baseline.

Domain	# labeled images	ImageNet pretrain	COCO + ImageNet pretrain
Faces	16	0.4575	0.4896
Faces	50	0.6197	0.6295
Cars	16	0.3232	0.3313
Cars	50	0.4802	0.5026


Original Image
Reconstructed image
Reconstructed image + original segmentation mask
Original Image
Reconstructed image (pre-optimization)
Reconstructed image (post-optimization)
Recon. Image + original seg. mask

(a)

(b)

Figure C.2: (a) Alignment of reconstructed images with original segmentation masks in the face domain. Semantic features align almost perfectly with segmentation masks. (b) Visualization of fine detail improvement after optimization refinement in car domain. Areas of vast improvement circled in red.

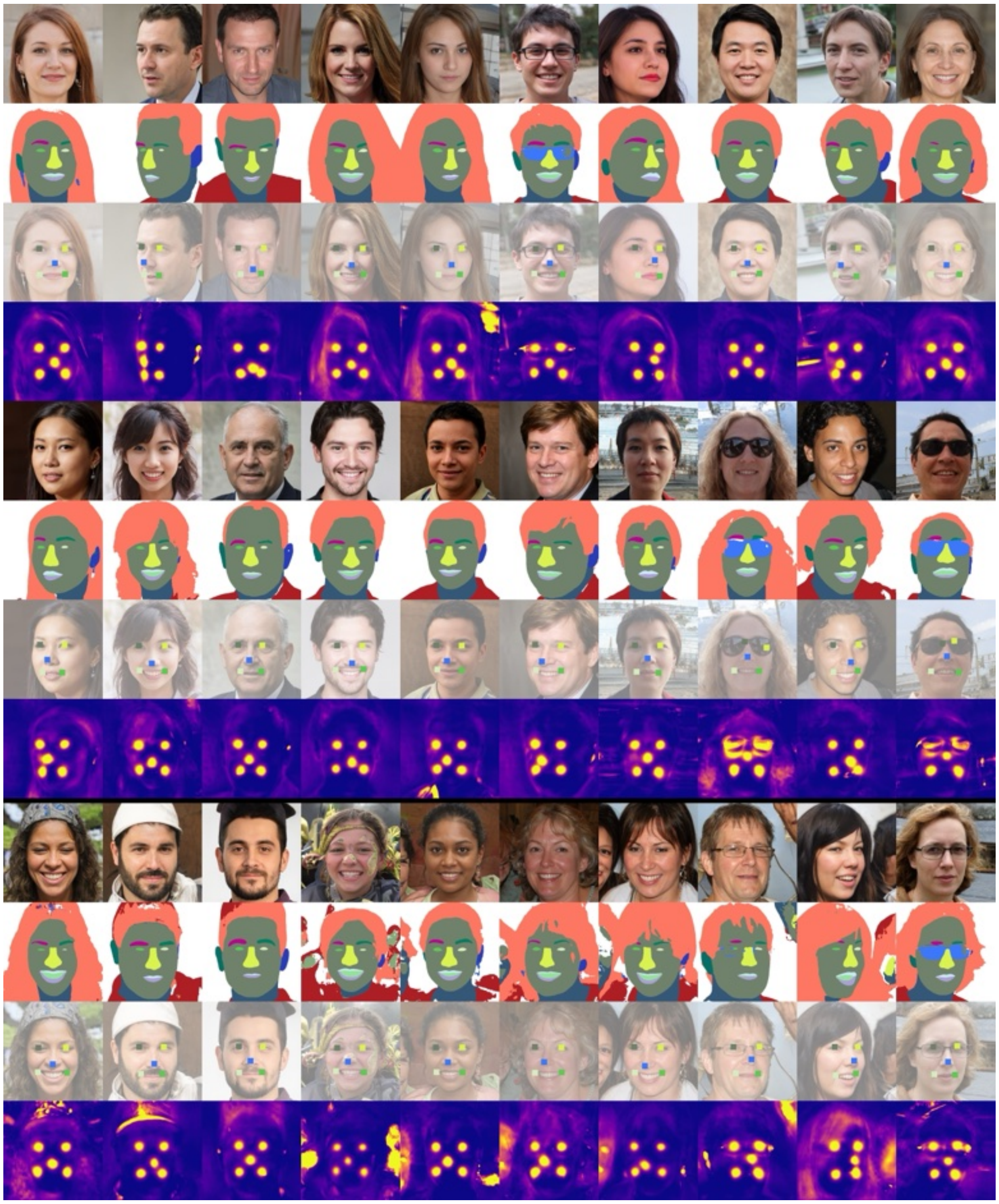


Figure C.3: Examples of HandsOff generated labels (segmentation masks, keypoint heatmaps, and keypoints) in the face domain. Last row of examples represent typical failure cases. Hats, a rare class, are occasionally mis-classified as hair or clothing. Additionally, when the image includes GAN generated artifacts, segmentation mask quality is typically lower, while keypoint locations remain accurate.

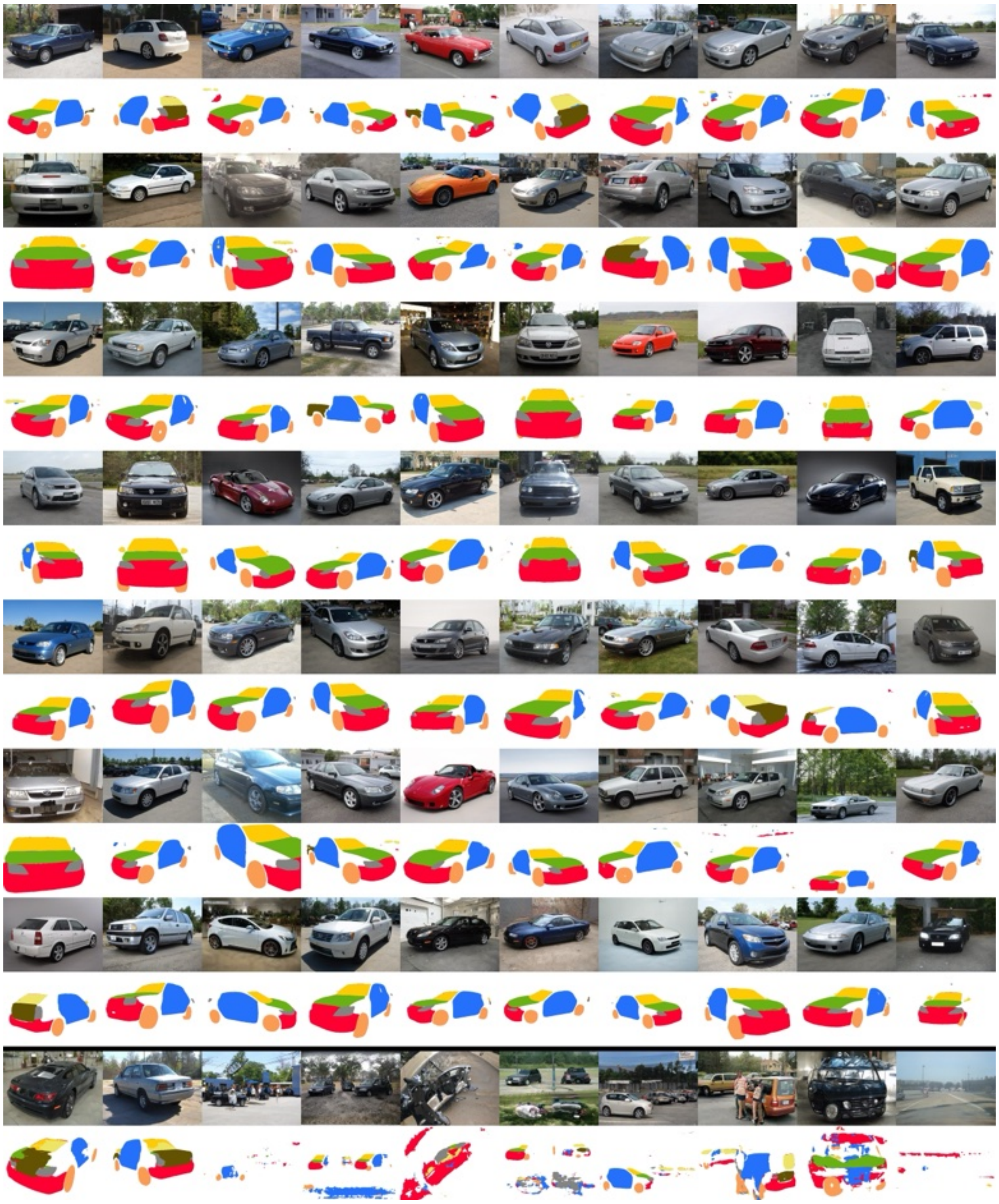


Figure C.4: Examples of HandsOff generated segmentation masks in the car domain. Last row of examples represent typical failure cases. Similar classes, such as back trunk and front hood or front glass and back glass are confounded. Additionally, segmentation performance is typically lower when GAN generated images are out of domain or incoherent.

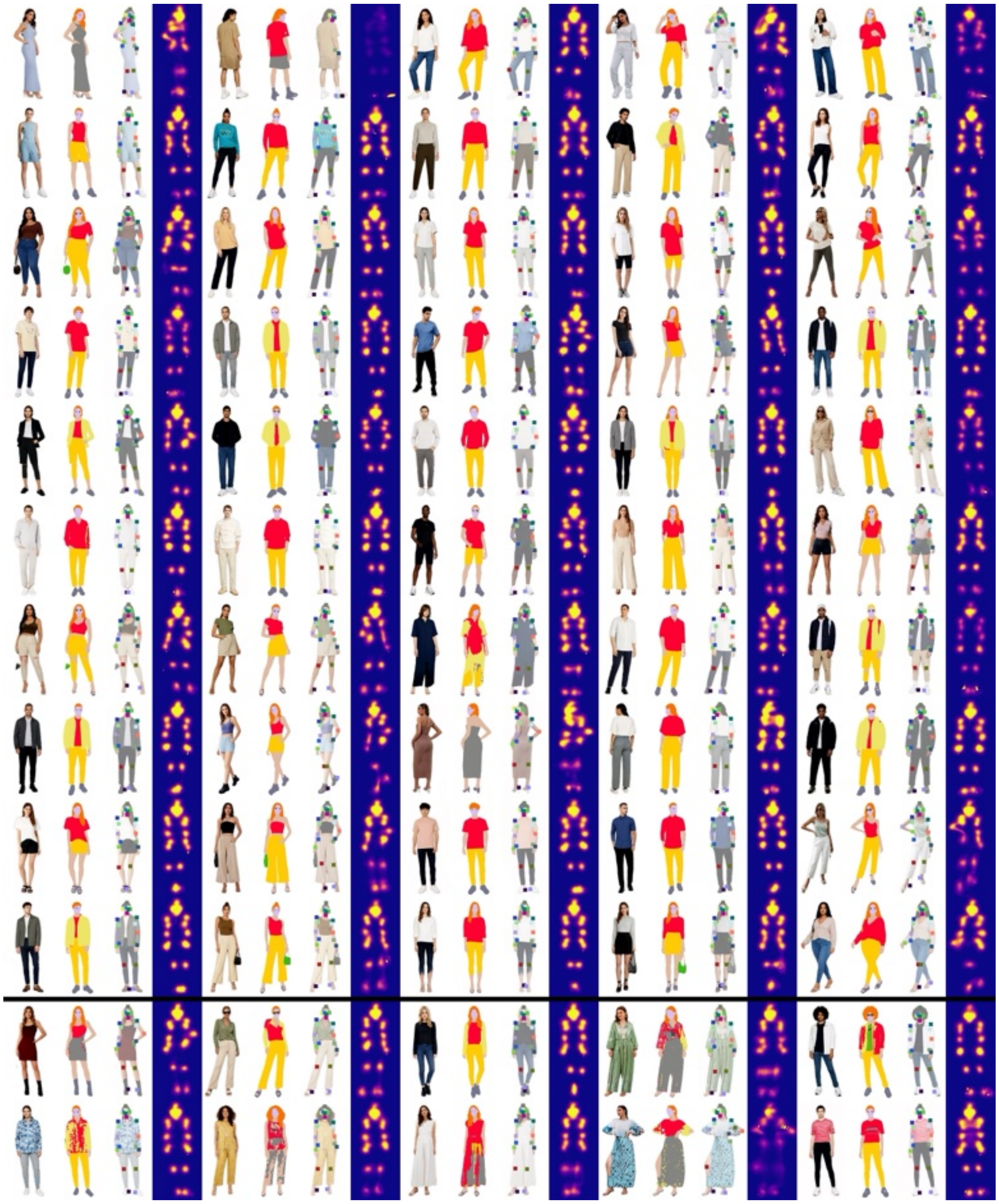


Figure C.5: Examples of HandsOff generated labels (segmentation masks, keypoint heatmaps, and keypoints) in the full-body human poses domain. Last row of examples represent typical failure cases. Similar classes, tops, outerwear, and dresses are confounded. Furthermore, patterned pieces of clothing seem to result in mixed segmentation performance. Keypoint locations remain accurate even when segmentation masks are of lower quality.

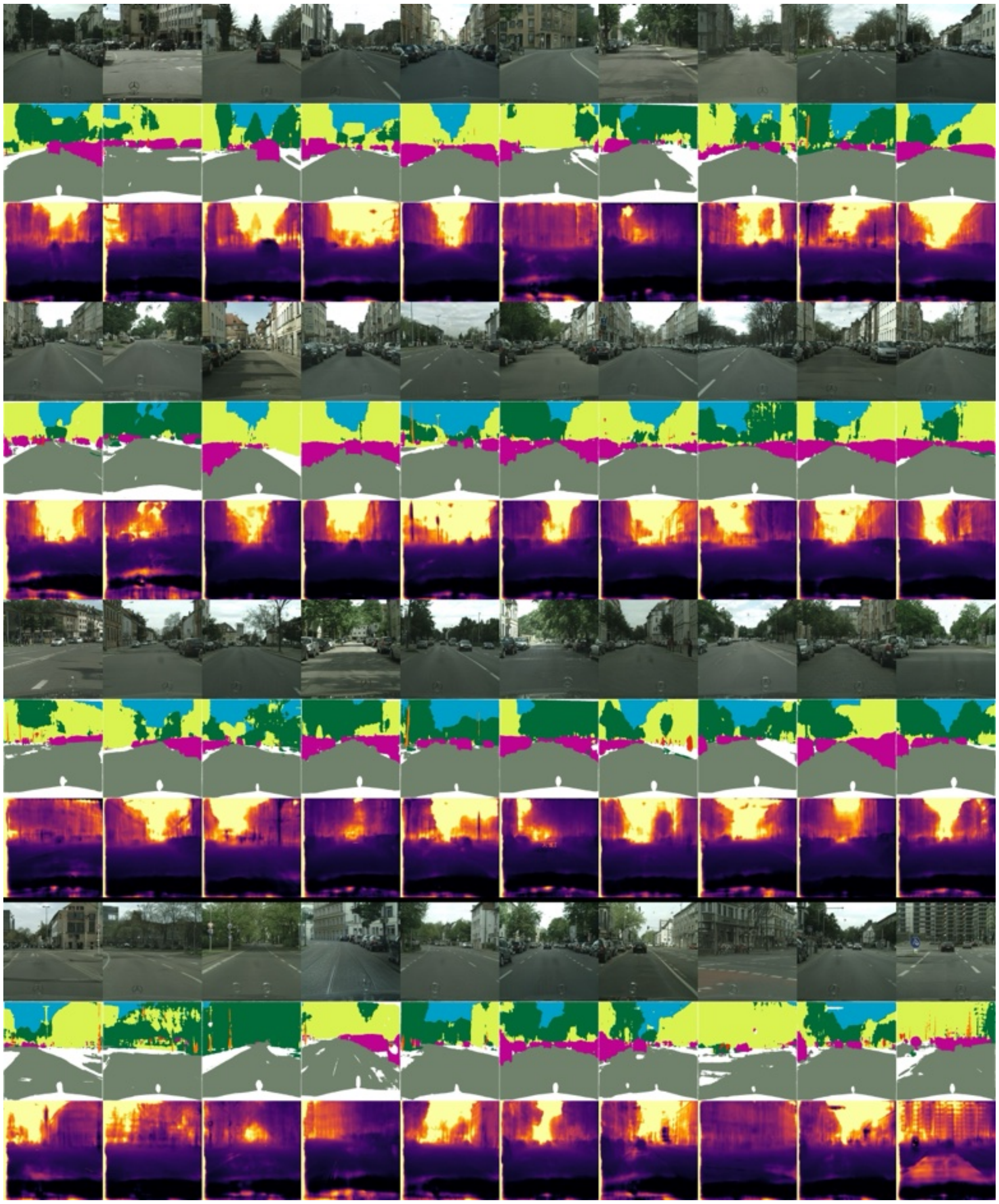
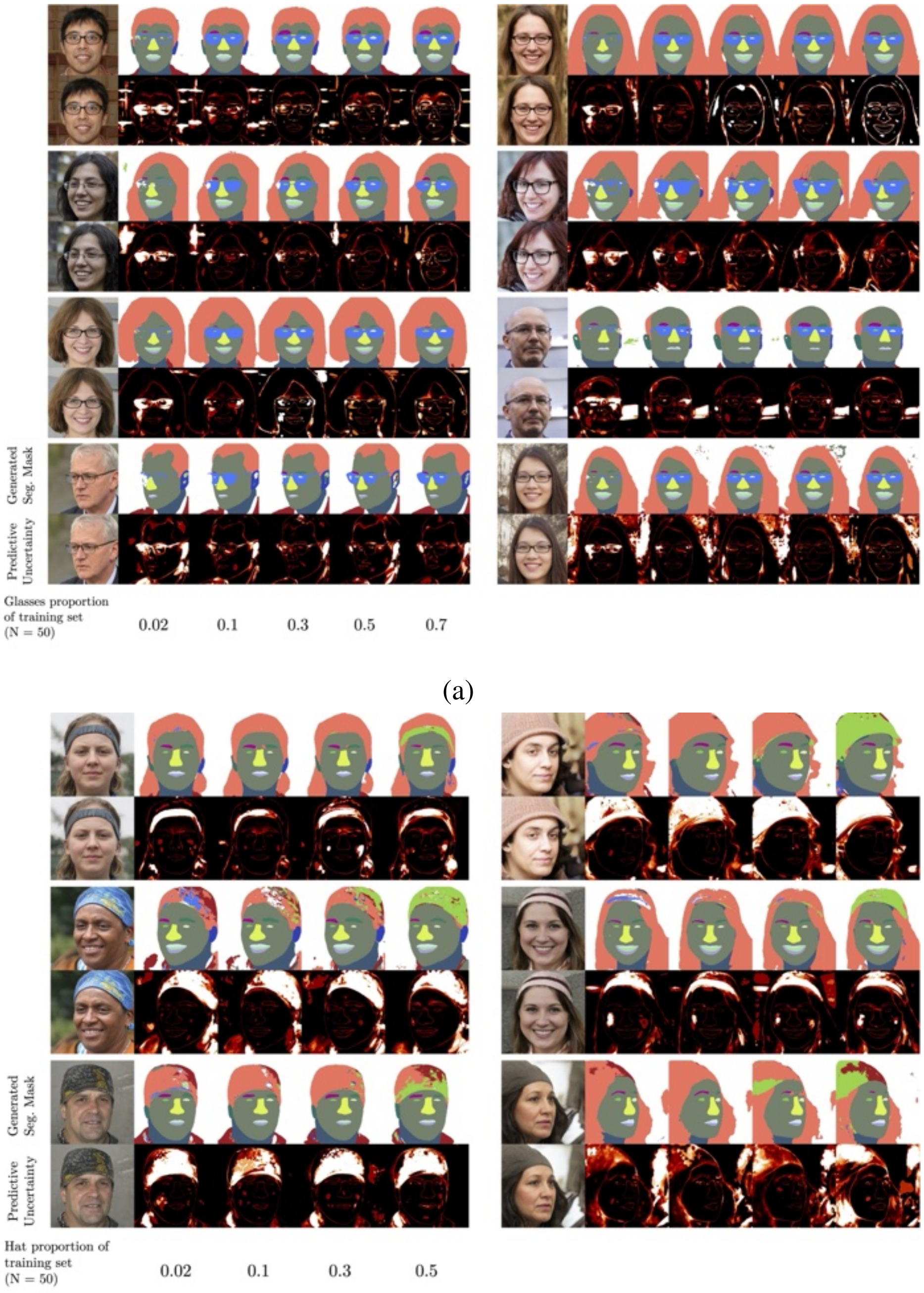


Figure C.6: Examples of HandsOff generated labels (segmentation masks and depth maps) in the urban driving scenes domain. Last row of examples represent typical failure cases. Visually small objects such as light poles and street signs are often confounded as background classes or not labeled. In cases of background buildings with many vertical lines, such lines can be mistaken as street sign poles (last image in last row). Depth maps remain relatively accurate even when segmentation masks are of lower quality.



(a)

(b)

Figure C.7: Visualization of generated segmentation mask and pixel-wise label generator uncertainty. (a) Not only do we see qualitative improvement in the generated label for glasses, we also see that the classifier is less uncertain when generating the correct label. (b) Hats are a particularly challenging class to characterize, so while the quality of the masks improves drastically, the classifier uncertainty remains relatively high. The last row of examples shows typical failure cases, where the hat is classified as semantically similar classes, such as hair or clothing.

REFERENCES

[1] A. Xu and M. Davenport, “Simultaneous preference and metric learning from paired comparisons,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[2] A. Xu, A. McRae, J. Wang, M. A. Davenport, and A. Pananjady, “Perceptual adjustment queries and an inverted measurement paradigm for low-rank metric learning,” in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[3] A. Xu, W. Monroe, and K. Bicknell, “Large language model augmented exercise retrieval for personalized language learning,” *arXiv preprint arXiv:2402.16877*, 2024.

[4] A. Xu, M. I. Vasileva, A. Dave, and A. Seshadri, “Handsoff: Labeled dataset generation with no additional human annotations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7991–8000.

[5] M. J. Wainwright, *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019, vol. 48.

[6] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018, vol. 47.

[7] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[8] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[9] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok, “Introduction to compressed sensing,” in *Compressed Sensing: Theory and Applications*, Cambridge University Press, 2012, pp. 1–64.

[10] S. Negahban and M. J. Wainwright, “Estimation of (near) low-rank matrices with noise and high-dimensional scaling,” *The Annals of Statistics*, vol. 39, no. 2, pp. 1069–1097, 2011.

[11] S. Negahban and M. J. Wainwright, “Restricted strong convexity and weighted matrix completion: Optimal bounds with noise,” *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1665–1697, 2012.

[12] H. A. David, *The method of paired comparisons*. London, 1963, vol. 12.

[13] N. B. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. J. Wainwright, “Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence,” *Journal of Machine Learning Research*, vol. 17, no. 58, pp. 1–47, 2016.

[14] J. Wang and N. B. Shah, “Your 2 is my 1, your 3 is my 9: Handling arbitrary miscalibrations in ratings,” in *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, 2019.

[15] D. Griffin and L. Brenner, “Perspectives on probability judgment calibration,” in *Blackwell Handbook of Judgment and Decision Making*. Wiley-Blackwell, 2008, ch. 9, ISBN: 9780470752937.

[16] P. Harik, B. Clauser, I. Grabovsky, R. Nungester, D. Swanson, and R. Nandakumar, “An examination of rater drift within a generalizability theory framework,” *Journal of Educational Measurement*, vol. 46, pp. 43–58, 2009.

[17] C. M. Myford and E. W. Wolfe, “Monitoring rater performance over time: A framework for detecting differential accuracy and differential scale category use,” *Journal of Educational Measurement*, vol. 46, no. 4, pp. 371–389, 2009.

[18] G. Miller, “The magical number seven, plus or minus two: Some limits on our capacity for processing information.,” *Psych. Rev.*, vol. 63, no. 2, p. 81, 1956.

[19] T. Joachims, “Optimizing search engines using clickthrough data,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 133–142.

[20] C. C. Aggarwal *et al.*, *Recommender systems*. Springer, 2016, vol. 1.

[21] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, “Deep reinforcement learning from human preferences,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[22] L. Ouyang *et al.*, “Training language models to follow instructions with human feedback,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[23] Y. Bai *et al.*, “Constitutional AI: Harmlessness from AI feedback,” *arXiv preprint arXiv:2212.08073*, 2022.

[24] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, “Direct preference optimization: Your language model is secretly a reward model,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[25] B. Mason, L. Jain, and R. Nowak, “Learning low-dimensional metrics,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[26] E. Hoffer and N. Ailon, “Deep metric learning using triplet network,” in *Proc. Int. Work. on Similarity-Based Pattern Recognition (SIMBAD)*, Copenhagen, Denmark, 2015.

[27] N. Nadagouda, A. Xu, and M. A. Davenport, “Active metric learning and classification using similarity queries,” in *Uncertainty in Artificial Intelligence*, PMLR, 2023, pp. 1478–1488.

[28] G. Canal, S. Fenu, and C. Rozell, “Active ordinal querying for tuplewise similarity learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020.

[29] R. A. Bradley and M. E. Terry, “Rank analysis of incomplete block designs: I. the method of paired comparisons,” *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[30] S. Negahban, S. Oh, and D. Shah, “Rank centrality: Ranking from pair-wise comparisons,” *Operations Research*, vol. 65, 2012.

[31] S. Negahban, S. Oh, and D. Shah, “Iterative ranking from pair-wise comparisons,” in *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Lake Tahoe, California, 2012.

[32] A. Rajkumar and S. Agarwal, “A statistical convergence perspective of algorithms for rank aggregation from pairwise data,” in *International conference on machine learning*, PMLR, 2014, pp. 118–126.

[33] H. Bong and A. Rinaldo, “Generalized results for the existence and consistency of the mle in the bradley-terry-luce model,” in *International Conference on Machine Learning*, PMLR, 2022, pp. 2160–2177.

[34] S. Chen and T. Joachims, “Predicting matchups and preferences in context,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 775–784.

[35] R. Makhijani and J. Ugander, “Parametric models for intransitivity in pairwise rankings,” in *The World Wide Web Conference*, 2019, pp. 3056–3062.

[36] A. Seshadri, A. Peysakhovich, and J. Ugander, “Discovering context effects from raw choice data,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 5660–5669.

[37] A. Seshadri, S. Ragain, and J. Ugander, “Learning rich rankings,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9435–9446, 2020.

[38] A. Bower and L. Balzano, “Preference modeling with context-dependent salient features,” in *International Conference on Machine Learning*, PMLR, 2020, pp. 1067–1077.

[39] OpenAI, *Gpt-4 technical report*, 2023. arXiv: 2303.08774 `[cs.CL]`.

[40] H. Touvron *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.

[41] H. Touvron *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.

[42] A. Vaswani *et al.*, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.

[43] I. Goodfellow *et al.*, “Generative adversarial nets,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, 2014.

[44] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International Conference on Machine :earning*, PMLR, 2017, pp. 214–223.

[45] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[46] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” in *International Conference on Learning Representations (ICLR)*, 2018.

[47] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[48] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8110–8119.

[49] T. Karras *et al.*, “Alias-Free Generative Adversarial Networks,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[50] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, “Gan inversion: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[51] A. Brock, J. Donahue, and K. Simonyan, “Large Scale GAN Training for High Fidelity Natural Image Synthesis,” in *International Conference on Learning Representations (ICLR)*, 2018.

[52] M. Kang *et al.*, “Scaling up gans for text-to-image synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 124–10 134.

[53] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[54] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *International Conference on Machine Learning*, PMLR, 2015, pp. 2256–2265.

[55] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[56] A. Q. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” in *International Conference on Machine Learning*, PMLR, 2021, pp. 8162–8171.

[57] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[58] C. H. Coombs, “Psychological scaling without a unit of measurement.,” *Psychological review*, vol. 57, no. 3, p. 145, 1950.

[59] B. Dubois, “Ideal point versus attribute models of brand preference: A comparison of predictive validity,” *ACR North American Advances*, 1975.

[60] A. Maydeu-Olivares and U. Böckenholt, “Modeling preference data,” in *The SAGE Handbook of Quantitative Methods in Psychology*, R. Millsap and A. Maydeu-Olivares, Eds., London, UK: SAGE Publications Ltd., 2009, ch. 12, pp. 264–282.

[61] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais, “Here or there: Preference judgments for relevance,” in *Proc. European Conf. on Inf. Retrieval (ECIR)*, Glasgow, Scotland, 2008.

[62] E. Hullermeier, J. Fürnkranz, W. Cheng, and K. Brinker, “Label ranking by learning pairwise preferences,” *Artif. Intell.*, vol. 172, no. 16-17, pp. 1897–1916, 2008.

[63] N. Ailon, “Active learning ranking from pairwise preferences with almost optimal query complexity,” in *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Grenada, Spain, 2011.

[64] K. G. Jamieson and R. Nowak, “Active ranking using pairwise comparisons,” *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[65] F. Wauthier, M. Jordan, and N. Jojic, “Efficient ranking from pairwise comparisons,” in *International Conference on Machine Learning*, 2013, pp. 109–117.

[66] B. Eriksson, “Learning to top-k search using pairwise comparisons,” in *Artificial Intelligence and Statistics*, 2013, pp. 265–273.

[67] Y. Chen and C. Suh, “Spectral mle: Top-k rank aggregation from pairwise comparisons,” *ArXiv*, vol. abs/1504.07218, 2015.

[68] N. B. Shah *et al.*, “Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence,” *Journal of Machine Learning Research*, vol. 17, no. 58, pp. 1–47, 2016.

[69] N. B. Shah and M. J. Wainwright, “Simple, robust and optimal ranking from pairwise comparisons,” *Journal of Machine Learning Research*, vol. 18, no. 199, pp. 1–38, 2018.

[70] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. Kriegman, and S. Belongie, “Generalized non-metric multidimensional scaling,” in *Proc. Int. Conf. Art. Intell. Stat. (AIStats)*, San Juan, Puerto Rico, 2007.

[71] K. Jamieson and R. Nowak, “Low-dimensional embedding using adaptively selected ordinal data,” in *Proc. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, 2011.

[72] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. Kalai, “Adaptively learning the crowd kernel,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, Bellevue, Washington, 2011.

[73] L. Van Der Maaten and K. Weinberger, “Stochastic triplet embedding,” in *Proc. IEEE Int. Work. Machine Learning for Signal Processing (MLSP)*, Santander, Spain, 2012.

[74] M. Davenport, “Lost without a compass: Nonmetric triangulation and landmark multidimensional scaling,” in *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2013, pp. 13–16.

[75] M. Kleindessner and U. Luxburg, “Uniqueness of ordinal embedding,” in *Conf. Learning Theory (COLT)*, Princeton, New Jersey, 2014.

[76] L. Jain, K. Jamieson, and R. Nowak, “Finite sample prediction and recovery bounds for ordinal embedding,” in *Proc. Conf. Neural Inf. Proc. Sys. (NeurIPS)*, Barcelona, Spain, 2016.

[77] E. Arias-Castro, “Some theory for ordinal embedding,” *Bernoulli*, vol. 23, no. 3, pp. 1663–1693, 2017.

[78] A. Massimino and M. Davenport, “As you like it: Localization via paired comparisons,” *submitted to J. Mach. Learn. Res.*, 2018.

[79] M. Schultz and T. Joachims, “Learning a distance metric from relative comparisons,” in *Advances in neural information processing systems*, 2004, pp. 41–48.

[80] E. Y. Liu, Z. Guo, X. Zhang, V. Jojic, and W. Wang, “Metric learning from relative comparisons by minimizing squared residual,” in *2012 IEEE 12th International Conference on Data Mining*, 2012, pp. 978–983.

[81] M. T. Law, N. Thome, and M. Cord, “Learning a distance metric from relative comparisons between quadruplets of images,” *International Journal of Computer Vision*, vol. 121, no. 1, pp. 65–94, 2017.

[82] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[83] B. McFee and G. R. Lanckriet, “Metric learning to rank,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 775–782.

[84] D. Lim and G. Lanckriet, “Efficient learning of mahalanobis metrics for ranking,” in *International conference on machine learning*, 2014, pp. 1980–1988.

[85] C. Jose and F. Fleuret, “Scalable metric learning via weighted approximate rank component analysis,” in *European conference on computer vision*, Springer, 2016, pp. 875–890.

[86] T. Zhao, J. McAuley, and I. King, “Improving latent factor models via personalized feature projection for one class recommendation,” in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ser. CIKM ’15, Melbourne, Australia: Association for Computing Machinery, 2015, pp. 821–830, ISBN: 9781450337946.

[87] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, “Collaborative metric learning,” in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 193–201.

[88] M. R. O'Shaughnessy and M. A. Davenport, "Localizing users and items from paired comparisons," in *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2016, pp. 1–6.

[89] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming, version 2.1*, http://cvxr.com/cvx, 2014.

[90] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds., http://stanford.edu/~boyd/graph_dcp.html, Springer-Verlag Limited, 2008, pp. 95–110.

[91] G. Canal, A. Massimino, M. Davenport, and C. Rozell, "Active embedding search via noisy paired comparisons," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, California, 2019.

[92] W. L. Rankin and J. W. Grube, "A comparison of ranking and rating procedures for value system measurement," *European Journal of Social Psychology*, vol. 10, no. 3, pp. 233–246, 1980.

[93] A.-W. Harzing *et al.*, "Rating versus ranking: What is the best way to reduce response and language bias in cross-national research?" *International Business Review*, vol. 18, no. 4, pp. 417–432, 2009.

[94] G. N. Yannakakis and J. Hallam, "Ranking vs. preference: A comparative study of self-reporting," in *Affective Computing and Intelligent Interaction*, S. D'Mello, A. Graesser, B. Schuller, and J.-C. Martin, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 437–446, ISBN: 978-3-642-24600-5.

[95] N. B. Shah, J. K. Bradley, A. Parekh, M. Wainwright, and K. Ramchandran, "A case for ordinal peer-evaluation in MOOCs," in *NIPS Workshop on Data Driven Education*, 2013.

[96] K. Raman and T. Joachims, "Methods for ordinal peer grading," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 20, 2014.

[97] R. D. Goffin and J. M. Olson, "Is it all relative? Comparative judgments and the possible improvement of self-ratings and ratings of others," *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 48–60, 2011.

[98] W. Barnett, "The modern theory of consumer behavior: Ordinal or cardinal?" *The Quarterly Journal of Austrian Economics*, vol. 6, no. 1, pp. 41–65, 2003.

[99] R. Batley, “On ordinal utility, cardinal utility and random utility,” *Theory and Decision*, vol. 64, pp. 37–63, 2008.

[100] G. Canal, B. Mason, R. Korlakai Vinayak, and R. Nowak, “One for all: Simultaneous metric and preference learning over multiple users,” in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[101] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, “GANSpace: Discovering interpretable GAN controls,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[102] Y. Ying, K. Huang, and C. Campbell, “Sparse metric learning via smooth optimization,” *Advances in Neural Information Processing Systems*, vol. 22, 2009.

[103] W. Bian and D. Tao, “Constrained empirical risk minimization framework for distance metric learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1194–1205, 2012.

[104] Z.-C. Guo and Y. Ying, “Guaranteed classification via regularized similarity learning,” *Neural Computation*, vol. 26, no. 3, pp. 497–522, 2014.

[105] A. Bellet and A. Habrard, “Robustness and generalization for metric learning,” *Neurocomputing*, vol. 151, pp. 259–267, 2015.

[106] Y. Chen, Y. Chi, and A. J. Goldsmith, “Exact and stable covariance estimation from quadratic sampling via convex programming,” *IEEE Transactions on Information Theory*, vol. 61, no. 7, pp. 4034–4059, 2015.

[107] A. Bellet, A. Habrard, and M. Sebban, *Metric Learning*. Springer Nature, 2022.

[108] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[109] A. Tsybakov and A. Rohde, “Estimation of high-dimensional low-rank matrices,” *The Annals of Statistics*, vol. 39, no. 2, pp. 887–930, 2011.

[110] E. J. Candes and Y. Plan, “Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements,” *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2342–2359, 2011.

[111] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, “A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers,” *Statistical Science*, vol. 27, no. 4, pp. 538–557, 2012.

[112] T. T. Cai and A. Zhang, “Sparse representation of a polytope and recovery of sparse signals and low-rank matrices,” *IEEE Transactions on Information Theory*, vol. 60, no. 1, pp. 122–132, 2013.

[113] M. A. Davenport and J. Romberg, “An overview of low-rank matrix recovery from incomplete observations,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 608–622, 2016.

[114] T. T. Cai and A. Zhang, “Rop: Matrix recovery via rank-one projections,” *The Annals of Statistics*, vol. 43, no. 1, pp. 102–138, 2015.

[115] R. Kueng, H. Rauhut, and U. Terstiege, “Low rank matrix recovery from rank one measurements,” *Applied and Computational Harmonic Analysis*, vol. 42, no. 1, pp. 88–116, 2017.

[116] A. D. McRae, J. Romberg, and M. A. Davenport, “Optimal convex lifted sparse phase retrieval and PCA with an atomic matrix norm regularizer,” *IEEE Transactions on Information Theory*, vol. 69, no. 3, pp. 1866–1882, 2022.

[117] K. A. Chandrasekher, M. Lou, and A. Pananjady, “Alternating minimization for generalized rank one matrix sensing: Sharp predictions from a random initialization,” *arXiv preprint arXiv:2207.09660*, 2022.

[118] P.-L. Loh, “Statistical consistency and asymptotic normality for high-dimensional robust M-estimators,” *The Annals of Statistics*, vol. 45, no. 2, pp. 866–896, 2017.

[119] J. Fan, Q. Li, and Y. Wang, “Estimation of high dimensional mean regression in the absence of symmetry and light tail assumptions,” *Journal of the Royal Statistical Society Series B, Statistical methodology*, vol. 79, no. 1, pp. 247–265, 2016.

[120] A. S. Nemirovskij and D. B. Yudin, *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

[121] S. Minsker, “Geometric median and robust estimation in Banach spaces,” *Bernoulli*, vol. 21, no. 4, pp. 2308–2335, 2015.

[122] D. Hsu and S. Sabato, “Loss minimization and parameter estimation with heavy tails,” *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 543–582, 2016.

[123] J. Fan, W. Wang, and Z. Zhu, “A shrinkage principle for heavy-tailed data: High-dimensional robust low-rank matrix recovery,” *The Annals of Statistics*, vol. 49, no. 3, pp. 1239–1266, 2021.

[124] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[125] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[126] P. Bajaj *et al.*, "MS MARCO: A Human Generated MAchine Reading COmprehension Dataset," *arXiv preprint arXiv:1611.09268*, 2016.

[127] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User modeling and user-adapted interaction*, vol. 4, pp. 253–278, 1994.

[128] Z. Wu, M. Li, Y. Tang, and Q. Liang, "Exercise recommendation based on knowledge concept prediction," *Knowledge-Based Systems*, vol. 210, p. 106 481, 2020.

[129] S. Huang, Q. Liu, J. Chen, X. Hu, Z. Liu, and W. Luo, "A design of a simple yet effective exercise recommendation system in k-12 online learning," in *International Conference on Artificial Intelligence in Education*, Springer, 2022, pp. 208–212.

[130] P. Cui and M. Sachan, "Adaptive and personalized exercise generation for online language learning," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023, pp. 10 184–10 198.

[131] B. Mandasari and A. Y. Wahyudin, "Flipped classroom learning model: Implementation and its impact on efl learners' satisfaction on grammar class," *Ethical Lingua: Journal of Language Teaching and Literature*, vol. 8, no. 1, pp. 150–158, 2021.

[132] S. M. Lucieer, L. Jonker, C. Visscher, R. M. Rikers, and A. P. Themmen, "Self-regulated learning and academic performance in medical education," *Medical teacher*, vol. 38, no. 6, pp. 585–593, 2016.

[133] K.-J. Kim and H. W. Jang, "Changes in medical students' motivation and self-regulated learning: A preliminary study," *International journal of medical education*, vol. 6, p. 213, 2015.

[134] S. Krashen, "Free voluntary reading: New research, applications, and controversies," *Anthology series-Seameo regional language centre*, vol. 46, no. 1, 2005.

[135] E. A. Patall, H. Cooper, and J. C. Robinson, "The effects of choice on intrinsic motivation and related outcomes: A meta-analysis of research findings.," *Psychological bulletin*, vol. 134, no. 2, p. 270, 2008.

[136] S. Geng, K. M. Law, and B. Niu, “Investigating self-directed learning and technology readiness in blending learning environment,” *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, pp. 1–22, 2019.

[137] K. Lee, M.-W. Chang, and K. Toutanova, “Latent retrieval for weakly supervised open domain question answering,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 6086–6096.

[138] V. Karpukhin *et al.*, “Dense passage retrieval for open-domain question answering,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781.

[139] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, “Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models,” in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[140] K. Wang, N. Thakur, N. Reimers, and I. Gurevych, “Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval,” in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 2345–2360.

[141] Y. Yu, C. Xiong, S. Sun, C. Zhang, and A. Overwijk, “Coco-dr: Combating the distribution shift in zero-shot dense retrieval with contrastive and distributionally robust learning,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 1462–1479.

[142] G. Izacard *et al.*, “Unsupervised Dense Information Retrieval with Contrastive Learning,” *Transactions on Machine Learning Research*, 2022.

[143] L. Bonifacio, H. Abonizio, M. Fadaee, and R. Nogueira, “Inpars: Unsupervised dataset generation for information retrieval,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 2387–2392.

[144] Z. Dai *et al.*, “Promptagator: Few-shot dense retrieval from 8 examples,” in *The Eleventh International Conference on Learning Representations*, 2022.

[145] D. Sachan *et al.*, “Improving passage retrieval with zero-shot question generation,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 3781–3797.

[146] W. Yu *et al.*, “Generate rather than retrieve: Large language models are strong context generators,” in *The Eleventh International Conference on Learning Representations*, 2022.

[147] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023, pp. 1762–1777.

[148] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[149] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint arXiv:1907.11692*, 2019.

[150] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, vol. 2, 2006, pp. 1735–1742.

[151] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*, PMLR, 2020, pp. 1597–1607.

[152] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple Contrastive Learning of Sentence Embeddings," in *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, 2021, pp. 6894–6910.

[153] Y.-S. Chuang *et al.*, "Diffcse: Difference-based contrastive learning for sentence embeddings," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 4207–4218.

[154] X. Wu, C. Gao, L. Zang, J. Han, Z. Wang, and S. Hu, "ESimCSE: Enhanced Sample Building Method for Contrastive Learning of Unsupervised Sentence Embedding," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 3898–3907.

[155] Q. Cheng, X. Yang, T. Sun, L. Li, and X. Qiu, "Improving contrastive learning of sentence embeddings from ai feedback," *arXiv preprint arXiv:2305.01918*, 2023.

[156] Y. Wang, A. Wu, and G. Neubig, "English contrastive learning can learn universal cross-lingual sentence embeddings," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 9122–9133.

[157] C. Piech *et al.*, "Deep knowledge tracing," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[158] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing.," *International Educational Data Mining Society*, 2019.

[159] D. Shin, Y. Shim, H. Yu, S. Lee, B. Kim, and Y. Choi, “Saint+: Integrating temporal features for ednet correctness prediction,” in *LAK21: 11th International Learning Analytics and Knowledge Conference*, 2021, pp. 490–496.

[160] G. Abdelrahman and Q. Wang, “Knowledge tracing with sequential key-value memory networks,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2019.

[161] S. Tong *et al.*, “Structure-based knowledge tracing: An influence propagation view,” in *2020 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2020, pp. 541–550.

[162] L. Xu and M. A. Davenport, “Dynamic knowledge embedding and tracing.,” *International Educational Data Mining Society*, 2020.

[163] S. Robertson, H. Zaragoza, *et al.*, “The probabilistic relevance framework: Bm25 and beyond,” *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[164] G. Wenzek *et al.*, “Ccnet: Extracting high quality monolingual datasets from web crawl data,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 4003–4012.

[165] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, “Deep Long-Tailed Learning: A Survey,” *arXiv preprint arXiv:2110.04596*, 2021.

[166] Y. Zhang *et al.*, “DatasetGAN: Efficient Labeled Data Factory with Minimal Human Effort,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[167] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training Generative Adversarial Networks with Limited Data,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[168] Y. Shen, J. Gu, X. Tang, and B. Zhou, “Interpreting the Latent Space of GANs for Semantic Face Editing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[169] R. Abdal, Y. Qin, and P. Wonka, “Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[170] R. Abdal, Y. Qin, and P. Wonka, “Image2StyleGAN++: How to Edit the Embedded Images?” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[171] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[172] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler, "EditGAN: High-Precision Semantic Image Editing," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[173] Y. Alaluf, O. Patashnik, and D. Cohen-Or, "Restyle: A Residual-Based StyleGAN Encoder via Iterative Refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[174] Y. Alaluf, O. Tov, R. Mokady, R. Gal, and A. Bermano, "HyperStyle: StyleGAN Inversion with HyperNetworks for Real Image Editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[175] D. Li, J. Yang, K. Kreis, A. Torralba, and S. Fidler, "Semantic Segmentation with Generative Models: Semi-Supervised Learning and Strong Out-of-Domain Generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[176] D. Li, H. Ling, S. W. Kim, K. Kreis, S. Fidler, and A. Torralba, "BigDatasetGAN: Synthesizing ImageNet with Pixel-wise Annotations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[177] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka, "Labels4Free: Unsupervised segmentation using StyleGAN," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[178] P. Esser, R. Rombach, and B. Ommer, "Taming Transformers for High-Resolution Image Synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[179] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, "Designing an Encoder for StyleGAN Image Manipulation," *ACM Transactions on Graphics (TOG)*, 2021.

[180] E. Richardson *et al.*, "Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[181] T. Wang, Y. Zhang, Y. Fan, J. Wang, and Q. Chen, "High-Fidelity GAN Inversion for Image Attribute Editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[182] E. Collins, R. Bala, B. Price, and S. Susstrunk, "Editing in Style: Uncovering the Local Semantics of GANs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[183] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[184] D. Roich, R. Mokady, A. H. Bermano, and D. Cohen-Or, "Pivotal Tuning for Latent-based Editing of Real Images," *ACM Transactions on Graphics (TOG)*, 2022.

[185] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative Visual Manipulation on the Natural Image Manifold," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[186] P. Melville, S. M. Yang, M. Saar-Tsechansky, and R. Mooney, "Active Learning for Probability Estimation Using Jensen-Shannon Divergence," in *Proceedings of the European Conference on Machine Learning*, 2005.

[187] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler, "The Power of Ensembles for Active Learning in Image Classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[188] W. Kuo, C. Häne, E. Yuh, P. Mukherjee, and J. Malik, "Cost-Sensitive Active Learning for Intracranial Hemorrhage Detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018.

[189] J. Fu *et al.*, "StyleGAN-Human: A Data-Centric Odyssey of Human Generation," 2022.

[190] R. Gadde, Q. Feng, and A. M. Martinez, "Detail Me More: Improving GAN's photo-realism of complex scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[191] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[192] K. Pasupa, P. Kittiworapanya, N. Hongngern, and K. Woraratpanya, "Evaluation of deep learning algorithms for semantic segmentation of car parts," *Complex & Intelligent Systems*, 2021.

[193] Y. Jiang, S. Yang, H. Qiu, W. Wu, C. C. Loy, and Z. Liu, "Text2Human: Text-Driven Controllable Human Image Generation," *ACM Transactions on Graphics (TOG)*, 2022.

[194] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, “DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[195] M. Cordts *et al.*, “The Cityscapes Dataset for Semantic Urban Scene Understanding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[196] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, “Unsupervised Monocular Depth and Ego-motion Learning with Structure and Semantics,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[197] J. Watson, O. Mac Aodha, V. Prisacariu, G. Brostow, and M. Firman, “The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[198] Z. Wang, G. So, and R. K. Vinayak, “Metric learning from limited pairwise preference comparisons,” *arXiv preprint arXiv:2403.19629*, 2024.

[199] G. Tatli, R. Nowak, and R. K. Vinayak, “Learning preference distributions from distance measurements,” in *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2022, pp. 1–8.

[200] G. Tatli, Y. Chen, and R. K. Vinayak, “Learning populations of preferences via pairwise comparison queries,” in *ICML 2023 Workshop The Many Facets of Preference-Based Learning*, 2023.

[201] W. Wu *et al.*, “Datasetdm: Synthesizing data with perception annotations using diffusion models,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[202] Y. Zhang, D. Zhou, B. Hooi, K. Wang, and J. Feng, “Expanding small-scale datasets with guided imagination,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[203] A. K. Massimino and M. A. Davenport, “As you like it: Localization via paired comparisons,” *Journal of Machine Learning Research*, vol. 22, no. 1, pp. 8357–8395, 2021.

[204] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013, ISBN: 9780199535255.

[205] Y. Bao and R. Kan, “On the moments of ratios of quadratic forms in normal random variables,” *Journal of Multivariate Analysis*, vol. 117, pp. 229–245, 2013.

[206] S. Mendelson, "Learning without concentration," *Journal of the ACM (JACM)*, vol. 62, no. 3, pp. 1–25, 2015.

[207] J. A. Tropp, "Convex recovery of a structured signal from independent random linear measurements," *Sampling Theory, a Renaissance: Compressive Sensing and Other Developments*, pp. 67–101, 2015.

[208] M. Rudelson and R. Vershynin, "Hanson–Wright inequality and sub-gaussian concentration," *Electronic Communications in Probability*, vol. 18, pp. 1–9, 2013.

[209] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *arXiv preprint arXiv:1011.3027*, 2010.

[210] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a Large-Scale Image Dataset using Deep Learning with Humans in the Loop," *arXiv preprint arXiv:1506.03365*, 2015.

[211] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D Object Representations for Fine-Grained Categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013.

[212] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[213] G. Hacheme and N. Sayouti, "Neural Fashion Image Captioning: Accounting for Data Diversity," in *arXiv preprint arXiv:2106.12154*, 2021.

[214] D. Baranchuk, A. Voynov, I. Rubachev, V. Khrulkov, and A. Babenko, "Label-Efficient Semantic Segmentation with Diffusion Models," in *International Conference on Learning Representations (ICLR)*, 2022.

[215] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[216] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.