Understanding Representations of Humans in Generative Image Modeling Through Discrete Counterfactual Prompt Optimization

Joshua Nathaniel Williams

CMU-CS-25-141

September 2025

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
J Zico Kolter (Chair)
Hoda Heidari
Aditi Raghunathan
Sarah Laszlo (Visa)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science.

This work was supported in part by the National Academies of Science, Engineering, and Medicine, Ford Foundation 2019 Predoctoral Fellowship.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity

Keywords: Generative Image Modeling, Prompt Optimization, Explainability, Discrete Optimization

For my family who has tirelessly supported me.

Abstract

Text-to-image (T2I) models are a common, publicly accessible class of generative model. Due to their widespread use, it is crucial to develop tools and methods that allow us to better understand how these models decide to represent their subjects, particularly human subjects. By comparing generated images across sets of carefully constructed prompts, we may uncover patterns in how these models represent various groups of people. These analyses often show specific prompts that elicit representational asymmetries, such as the prompt: "A person with glasses." being more likely to generate a male-presenting person than female-presenting.

While many such patterns are innocuous, some harmful representational biases emerge that require an intervention by developers. These approaches that rely on predefined prompt templates or fixed identity categories are effective for benchmarking known issues, yet they may unintentionally create blind spots shaped by the researchers' own background and experience. While one person's life experiences may lead them to expect (and therefore design experiments to evaluate) specific representations by the model, another person may expect a completely different set of representations and harms that the former would not consider – these differences in experience result in a wide range of potential blindspots in safety evaluations.

This thesis develops a variety of approaches, grounded in counterfactual and contrastive analyses, that act as general tools for surfacing new hypotheses related to representational asymmetries and harms in generative modeling that address these blindspots and complement existing evaluations. We first demonstrate that effective explanations for simple classifiers requires incorporating knowledge of the underlying ground-truth data distribution, without which, explanations and discoveries are prone to spurious insights. We posit a simple change to the implicit graphical model that underlies counterfactual explainability and propose a new metric that explicitly incorporates this distributional awareness.

The insights from this method then guides our approach to counterfactual explainability methods in the T2I setting. By reviewing a variety of discrete prompt optimization methods, we show how to define and encode this distributional awareness of captioned data in the optimization process. We support these methods by introducing an approach for multiobjective optimization across multiple language models, each with discrete tokenizers and text embeddings. Using the insights and methods developed throughout this thesis, we conclude by presenting an unsupervised strategy for discovering candidate prompts that encode representational asymmetries, many of which have not yet been discussed in the broader literature. Understanding and relating the learned speech and writing patterns of generative models to their outputs, allows to better understand why models represent people the way that they do and improves our ability to target specific behaviors as we train and evaluate generative models.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Classical Counterfactual Explanations . . . 3
- 1.3 Counterfactuals of Prompts . . . 4
- 1.4 Dissertation Summary . . . 7

2 Counterfactual Distributions **9**
- 2.1 Understanding Counterfactual Distributions . . . 11
  - 2.1.1 Regularized Counterfactuals . . . 12
- 2.2 Ensuring Representative Counterfactuals . . . 14
- 2.3 Extending the Proposed Framework to Complex Models . . . 15
  - 2.3.1 Extending the Proposed Framework to Complex Data . . . 16
- 2.4 Domain Knowledge in the Prior . . . 17
  - 2.4.1 Accounting for Actionability Constraints . . . 17
- 2.5 Revisiting Counterfactual Optimization . . . 18
- 2.6 Evaluation . . . 19
  - 2.6.1 Quantitative Evaluations . . . 19
  - 2.6.2 Qualitative Evaluation . . . 22
  - 2.6.3 Survey Evaluation . . . 25
- 2.7 Discussion and Future Directions . . . 27

3 (Completed Work) Discrete Optimization **29**
- 3.1 Selected Algorithms . . . 31
  - 3.1.1 PEZ . . . 32
  - 3.1.2 Greedy Coordinate Gradients . . . 32
  - 3.1.3 AutoDAN . . . 33
  - 3.1.4 Random Search . . . 33
  - 3.1.5 PRISM . . . 34
  - 3.1.6 Captioning . . . 34
- 3.2 Evaluation . . . 34
- 3.3 Empirical Results . . . 35
  - 3.3.1 Quantitatively Ranking Methods . . . 35
  - 3.3.2 Qualitatively Assessing Inverted Prompts . . . 38
- 3.4 Discussion . . . 40

3.4.1 Open Questions and Future Directions . . . 40

4 (Completed Work) FUSE **43**
4.1 Introduction . . . 43
4.2 Preliminaries . . . 44
4.2.1 Language Model Embeddings . . . 44
4.3 Methodology . . . 45
4.3.1 Shared Tokenizers . . . 45
4.3.2 Different Tokenizers . . . 46
4.4 Experiments . . . 49
4.4.1 Datasets . . . 49
4.4.2 Implementation Details . . . 49
4.5 Results . . . 52
4.5.1 Image Captioning . . . 52
4.5.2 Captioning with Sentiment . . . 52
4.6 Conclusion . . . 53

5 DrawL **55**
5.1 Background . . . 56
5.2 Methodology . . . 57
5.2.1 Dialect Application . . . 57
5.2.2 Prompt Set Construction . . . 58
5.2.3 Skin Tone Annotation . . . 60
5.3 Results . . . 60
5.3.1 Intersection Effects of Dialect on Gender and skin Tone . . . 62
5.4 Discussion and Conclusion . . . 64

6 Counterfactual Prompt Discovery **67**
6.1 Introduction . . . 67
6.2 Methodology . . . 69
6.2.1 Naive Objective: CLIP + Euclidean Distance . . . 69
6.2.2 Necessary Properties of T2I Prompts . . . 70
6.2.3 Contrastive Counterfactual Prompt Optimization . . . 72
6.2.4 Discrete Prompt Search . . . 73
6.3 Results and Analysis . . . 73
6.3.1 Exploratory Token Analysis . . . 76
6.3.2 Examining Nuanced Token Behaviors . . . 76
6.3.3 Reviewing Weak Alignment Patterns in Token Behaviors . . . 78
6.4 Discussion . . . 80

7 Conclusion **81**
7.1 Technical and Methodological Improvements . . . 82
7.1.1 Concept Bottlenecks . . . 83
7.1.2 Embedding Comparisons . . . 83

7.1.3 Text-Only Prompt Analyses . . . 84
7.1.4 Counterfactuals from Multimodal Models . . . 84
7.2 Future Work . . . 84

A Practical Consideration of the Laplace Approximation **99**

B Tensor Products **101**

C Pervasiveness of Syntax Features Across Dialects **103**
C.0.1 Null Copula is commonly used among: . . . 103
C.0.2 Double Modal is commonly used among: . . . 104
C.0.3 Habitual Be is commonly used among: . . . 104
C.0.4 Invariant Don't is commonly used among: . . . 104
C.0.5 Negative Concord is commonly used among: . . . 105
C.0.6 Completive Done is commonly used among: . . . 105
C.0.7 Quotative all is commonly used among: . . . 106
C.0.8 Ain't as the negated form of be is commonly used among: . . . 106
C.0.9 New quasi-modals with aspectual meanings (Including 'Finna') is commonly used among: . . . 107
C.1 Dataset Details . . . 107
C.2 Precomputing the Gradient $V_i^+ V_j$ . . . 108

D Soft Edit Distance **111**
D.1 Additional Examples of Discovered Representational Asymmetries . . . 112

List of Figures

1.1 Images generated using the prompt: "A dog playing". A 3 token prompt still generates diverse images including different breeds, different backgrounds, and different objects. While the text input space may be interpretable, the major details of the image are governed by a difficult to interpret, initial noise distribution. 5

2.1 Counterfactual explanations for a loan classifier. Left: examples rejected by the model. Right: minimal changes (highlighted) that result in acceptance. 10

2.2 Comparison of approaches to counterfactual generation; counterfactuals with the proposed prior never leave the data distribution. (Black Dot) Reference, $\mathbf{x}$. (Green) Counterfactual Distribution. (Black Line) Desired predicted output, $y' = A\mathbf{x}' + b$. In all figures, L is the precision of residuals, γ is the weight on l_2 distance, and α controls similarity/distance in our approach. 11

2.3 Comparison of approaches to counterfactual generation; counterfactuals with the proposed prior never leave the data distribution. (Black Dot) Reference, $\mathbf{x}$. (Green) Counterfactual Distribution. (Black Line) Desired predicted output, $y' = A\mathbf{x}' + b$. In all figures, L is the precision of residuals, γ is the weight on l_2 distance, and α controls similarity/distance in our approach. 13

2.4 Comparison of approaches to counterfactual generation; counterfactuals with the proposed prior never leave the data distribution. (Black Dot) Reference, $\mathbf{x}$. (Green) Counterfactual Distribution. (Black Line) Desired predicted output, $y' = A\mathbf{x}' + b$. In all figures, L is the precision of residuals, γ is the weight on l_2 distance, and α controls similarity/distance in our approach. 14

2.5 All images have 99% certainty for the desired class based on the trained classifier. Our proposed approach produces counterfactual images that, while further from the reference images than those generated using the L2 distance, exhibit more semantically meaningful features associated with each class. Additionally, our approach avoids the class mixing observed when traversing the VAE's latent space. 24

2.6 Preference matrices for survey responses on each dataset. Each cell shows how often a respondent preferred the row method to the column method–darker colors imply a greater preference. Each method seems to excel on different types of data. 25

3.1 Images of three dogs generated by Gemini 2.5 Flash using the following conversation: [0.25cm] **User:** "can you generate me a picture of a happy dog?" **Assistant:** "Sure, here 30

3.2 Comparison between images generated by inverted prompts and images generated by the original prompts. 36

3.3 CLIP Similarity between the inverted prompt and images generated by the original prompt. This CLIP Similarity is the objective that each optimizer is maximizing. 37

3.4 Cosine Similarity between text embeddings for the original and inverted prompts. Based on the metric used by [99] 38

4.1 The FUSE adapter connecting two transformer models for parallel inference. Inputs from Model 1 flow through the adapter by converting to text, retokenizing with Model 2's tokenizer, and embedding into Model 2's input space. The backward pass receives the gradient from Model 2, and multiplies it by the precomputed $\tilde{V}_1^+ * \tilde{V}_2$. 44

4.2 An $\mathbb{R}^{9\times d\times 2}$ tensor vocabulary over words: "the quick brown fox jumps over the lazy dog". Each plain-text word represents its corresponding $\mathbb{R}^d$ embedding, and each $\boldsymbol{\emptyset}$ is a 0 vector. We approximate the gradient for a mapping from model $\mathcal{M}_i$'s embeddings to $\mathcal{M}_j$'s embeddings by computing the t-product $\tilde{V}_i^+ * \tilde{V}_j$, where $\tilde{V}_i^+$. 47

4.3 Example Captions that using a FUSE Adapter to minimize the sum of GPT2-Medium, CLIP-VIT-B/32, and a Bert-based Sentiment Classifier via AutoDAN [157]. This combination of models controls through synonyms that indicate tone or through creating additional context for each image to denote tone. Note that AutoDAN does not have a clear stopping condition, a caption may stop in the middle of a sentence. 51

5.1 The Monk Skin Tone Scale. The skin tones of humans generated by the model are annotated with a score of 1 (lightest) to 10 (darkest). In this work, we measure how the distribution of skin tones generated by the model changes when prompting in African American English (AAE) as opposed to Standard American English (SAE) 56

5.2 Effect Sizes for the association between dialect, skin tone distribution, and gendered prompts. Bolded cells all have at least a moderate effect on the skin-tones generated by Stable Diffusion. In aggregate, the application of AAE has a moderate effect on the distribution of skin tones – shifting skin tones darker. 60

5.3 Distribution of Monk Skin Tones for images generated by our contrastive prompts in SAE and AAE. (5.3a) Shows the marginal skin tone distributions over the gendered prompt subjects. (5.3b,5.3c,5.3d) show the skin tone distribution conditioned on the prompt specifying male subjects, female subjects, and not specifying gender respectively. The marginal and conditional distributions, all show that prompting Stable Diffusion in AAE generates overall darker subjects in the image compared to prompting with SAE. 61

5.4 Distribution of Monk Skin Tones for selected features. (5.4a,5.4b) The use of 'Finna' as a semi-modal and the use of 'Completive Done' have a relatively strong effects on the distribution of skin tones – darkening the skin tones of generated humans. (5.4c,5.4d) The use of the 'Quotative All' and the use of 'Ain't' as the negated form of 'be' have little effect on the distribution of skin tones. 63

6.1 Overview of our approach. 68
6.2 Images generated by: 'A {group} drinking coffee in the morning'. Each image is captioned by a solution to Eq. (6.1) and Eq. (6.6). Solutions based on text-image alignment and Euclidean distance include spurious tokens and lack semantic coherence. Our approach allows not only for solutions that align with the image and are coherent, but also may incorporate additional cultural associations. 70
6.3 Frequency distributions of the 25 most common tokens that have a greater prevalence in female-specified prompts (top) and male-specified prompts (bottom). As discussed in Section 6.3.2, some symmetric prompts, such as "riding" and "driving" are distinctly female and male-associated. 75
6.4 Images of prompts with notable effects on gender representations derived from discovered token frequency differences. We find several potential asymmetries, whose use may guide more targeted explorations of gender across unexpected or underexplored bias axes. 77
6.5 Images of prompts with ambiguous effects on gender representations derived from the token frequency differences in Figure 6.3. Many prompts that do not induce a pragmatic shift in gender representation, may still surface nuances in the way language influences the discovery process. 79

D.1 Images of prompts with notable effects on gender representations derived from discovered token frequency differences. We find several potential asymmetries, whose use may guide more targeted explorations of gender across unexpected or underexplored bias axes. 113
D.2 Images of prompts with notable effects on gender representations derived from discovered token frequency differences. We find several potential asymmetries, whose use may guide more targeted explorations of gender across unexpected or underexplored bias axes. 114

List of Tables

2.1 Benchmarking table comparing our proposed counterfactual distance with an l2 distance metric across 4 datasets, showing that while our approach increases runtime, it generates counterfactuals significantly closer to the underlying data distribution as measured by the number of nearest neighbors who share the desired label (YNN), without a significant decrease in performance across any other metric. 21

3.1 Example images and corresponding 20-token prompts. Each image is generated by the *original prompt* and we show examples of the inversion result from each method. Other AutoDAN with the language prior applied, no discrete optimizer produces more human-readable prompts than another despite the quantitative differences their performance. 39

4.1 Comparison of SOA image captioning methods. 52
4.2 Comparison of SOA sentiment-based image captioning methods. 53

5.1 Examples of User-Submitted Prompts and the resultant contrastive prompt pairs. We construct the dataset to be used for our analysis by choosing in-the-wild user-submitted prompts to an image generation model, and rewording these prompts into a prompt that generates humans and allows us to apply each syntactical feature with a minimal number of changes to the SAE prompt. 59

Chapter 1

Introduction

1.1 Motivation

Asking "What if" questions is a major way that we discover the causal relationships embedded within our environment. Even as children, we use the tools at our disposal in order to understand domain-specific causal relations [107, 132]. As we navigate new and unknown settings, we ensure our understanding of the systems around us through the knowledge that an action X causes the effect Y, and by intervening on X, we can change Y-hence we can change our environment[88]. It is through this lens that we have long determined the safety or harms inherent to the systems with which we interact.

Understanding causality in real-world systems is challenging, as these systems are often black boxes. We rarely find ourselves in scenarios where we have a full understanding of the features that are taken into account and how they impact a final decision. As a result, a significant amount of effort is focused on abstracting these systems into forms that we can understand. Counterfactual analysis—examining cases where minimal changes to input features lead to different outcomes—has emerged as a powerful tool to shed light on hidden decision-making processes. By studying these changes, we can probe the behavior of a variety of processes, uncover hidden patterns, and approximate their causal maps in order to build systems that best align with our values and expectations.

These approaches apply equally to human and machine systems. Counterfactual reasoning explores the implicit rules and biases that influence a wide variety of outcomes, and have a large body of work and methods that support these investigations. One such method is the use of proxy models [101], which address the question, "What if we learn an inherently interpretable model that makes the same decisions as a complex model?". As many interpretable models, such as logistic regressions, allow us to explicitly see how a feature influences our output, we expect that understanding how the interpretable model behaves with respect to each feature will serve as an explanation for the complex model by proxy.

However, proxy models often limit model expressivity. To explain more flexible models, there has been some success with Saliency Maps [67, 113, 116, 120] that provide a score for each of the input features (commonly using information about the gradients at some point in the network). This approach presents the user with a numerical value for how each feature relates

to the output. Additional work [1, 4, 124] has also provided sanity checks for such methods in order to guide researchers in deciding when and which method best meets the needs of their task at hand.

Further development has led to researchers incorporating a causal understanding into this explanation process [12]. For example, Zhao and Hastie [154] had the insight that a commonly used visualization of black-box models, Partial Dependence Plots (PDP) [33], are effectively equivalent to Pearl's Backdoor Criterion [90]. Thus, PDPs not only provide information on the relationship between the target output and a feature, but also their *causal* relationship. In a similar vein, further work [110] considers the case of "Granger Causality" in which a signal, X is said to *cause*, Y, if there exist no features outside of X that provide additional predictive performance. The change in predictive performance with/without each feature and can then be scored to see how much each feature can be said to cause the target variable.

As we transition from classification to generative modeling, the parallels between human and machine systems become particularly salient. While we have the capability to generate text, images, and videos from simple prompts, the interplay between text prompts and generated outputs remains opaque. While classification has a single expected output, there are innumerable ways that a model could depict an image coming from the prompt: "A dog playing". Defining the expected output alone is a challenge, let alone how the input space relates. Both work in explainability and interpretability are still grappling with new challenges introduced by adapting prior work to this new setting.

In our work, we focus on this challenge through the lens of counterfactual reasoning. We extend counterfactual reasoning to the domain of prompt engineering, treating prompts as the primary input feature of importance. By analyzing minimal prompt modifications that induce specific semantic changes in generated outputs, we aim to uncover interpretable relationships between text tokens and their high-dimensional manifestations. We thus focus on adapting counterfactual explainability methods to the generative modeling space. Counterfactual Explanations [50, 71, 131] seek to provide a user with a set of points from the input space that are similar to an initial feature vector or *reference*, but receive a different prediction by the decision-making model. By basing our explanations within counterfactual explainability strategies, we may be able to discover new behaviors in models in largely unsupervised ways. In this way, we scope our work with the following thesis statement:

Thesis Statement: This thesis establishes a formal, mathematical structure for adapting counterfactual explainability techniques in classifiers to generative models. We propose a framework for distinguishing counterfactual explanations from adversarial examples. By leveraging this framework, we introduce novel distance metrics and optimization strategies to identify counterfactual prompts—minimal prompt modifications that induce specific semantic changes in generated outputs. These prompts are then used to generate contrastive prompts that reveal interpretable relationships between prompt tokens and their corresponding image patterns.

1.2 Classical Counterfactual Explanations

Counterfactual analysis methods for machine learning models emerged as legal frameworks set new standards and requirements around the use of automated decision processes. Reflecting the European Union's General Data Protection Regulation (GDPR) and its codification of the right to explanation, Wachter et al. [131] propose three primary goals for an explanation:

(1) To inform and help the subject understand why a particular decision was reached.
(2) To provide grounds to contest adverse decisions.
(3) To understand what could be changed to receive a desired result in the future, based on the current decision-making model.

Consider a person applying for a job. Their application packet shows previous work experience, education, and skills. An automated screener removes their packet from consideration before it gets seen by a human. How could a person understand why they did not move forward?

Generally, an applicant could look at the applications of those who were moved forward. What kinds of roles did they have? What education level did they have? What companies did they work for in the past? All of these would then be compared to the original applicants packet in order to understand the decision from the automated screener. In this way the applicant could understand why they were not selected and what to change. If the applicant observed problematic behavior, the decisions could then allow the applicant to contest the process.

Counterfactual explanations seek to mimic this process by probing the decision boundaries of a classifier (the screener in the above case). By generating synthetic data that is similar to the original applicant's packet, but received a different decision, we could provide a more targeted explanation that gives even more insight about the behavior of the original classifier. Formally we can define a counterfactual as:

Definition 1 (Counterfactual Explanations)**.** *For some input space, $\mathcal{X}$, consider a model $f : \mathcal{X} \to \mathcal{Y}$, a reference point, $\mathbf{x} \in \mathcal{X}$, and a desired predicted label y'. Let $\epsilon, \delta \in \mathbb{R}^+$ be two given parameters. The set of counterfactual explanations, with parameters ϵ, δ for the predicted label, $f(\mathbf{x}) \neq y'$, is defined as follows:*

$$\begin{aligned} \mathrm{cf}(\mathbf{x}, y'; \epsilon, \delta) := \{\mathbf{x}' \in \mathcal{X} : \\ \mathrm{dist}_1(y', f(\mathbf{x}')) \leq \delta, \quad \mathrm{dist}_2(\mathbf{x}, \mathbf{x}') \leq \epsilon\}, \end{aligned} \tag{1.1}$$

where $\mathrm{dist}_i : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ *are distance functions.*

The distance between the desired predicted outcome, y', and the label of the counterfactual, $f(\mathbf{x}')$, is often described as some convex loss function (eg. squared error), and the distance between the reference and counterfactual as some applicable distance metric, such as the l_1 norm scaled by the Median Absolute Deviation (MAD), Edit Distances [35] or the Eu-

clidean/Mahalanobis distance [50, 71, 81, 131, 142]:

$$\begin{aligned} \text{dist}_1(y', f(\mathbf{x}')) &:= ||y' - f(\mathbf{x}')||_2^2, && \text{(prediction accuracy constraint)} \\ \text{dist}_2(\mathbf{x}, \mathbf{x}') &:= ||\mathbf{x} - \mathbf{x}'||_2^2 && \text{(proximity constraint)} \end{aligned}$$

This immediately gives rise to the most common method of solving Eq (1.1); minimize the sum of dist_1 and dist_2, which as pointed out in [30], is akin to an adversarial attack on the classifier,

$$\mathbf{x}' = \arg\min_{\tilde{\mathbf{x}}} ||y' - f(\tilde{\mathbf{x}})||_2^2 + \gamma ||\mathbf{x} - \tilde{\mathbf{x}}||_2^2, \tag{1.2}$$

where $\gamma \in \mathbb{R}^+$ scales the influence on distance. While this form underlies a large portion of work, authors often apply additional regularization or pre/post-processing to create desirable properties. For example, Mothilal et al. [81] introduce a diversity regularizer to encourage subsequent explanations to be distinct from one another. Kang et al. [50] solve Eq. (1.2) via coordinate descent in order to minimize hamming distance, and other work [71, 115] introduces regularizers that encourage low-cost solutions in terms of fairness/causality respectively. A detailed review of other such methods can be found in [129].

Generating a counterfactual image thus requires us to define a tractable form of $\text{dist}_1(y', f(\mathbf{x}'))$ and $\text{dist}_2(\mathbf{x}', \mathbf{x})$. While $\text{dist}_2(\mathbf{x}', \mathbf{x})$ may be simply treated as the euclidean distance between embedding spaces or as the hamming distance between string tokenizations, we will show that this will not be sufficient in order to align with human intuitions on minimal distance counterfactuals. But for now, treat dist_2 as such and we can focus our attention on dist_1, the difference between a ground truth image and an image generated by the prompt $\mathbf{x}'$.

1.3 Counterfactuals of Prompts

Generative models have difficulty fitting into the above approach. In the text space, if a text-to-text model gives an unexpected response, we may want to understand how to change our initial input in order to get the expected response. This approach could give valuable information about a large model's learned patterns and boundaries. Yet, actually finding those minimal changes is a particularly difficult task. The space of possible text sequences grows exponentially in the number of tokens; as we encourage larger and larger contexts in language modeling, an exhaustive search is impractical.

In this thesis, we want to develop approaches that can make such approaches more practical and efficient. Here, we use image generation as a sandbox to develop such approaches that allow us to search the prompt space before bringing them to the text-to-text setting. Not only is the token input space generally much smaller than the text space, expected and unexpected responses are easier to articulate. Rather than trying to understand why an AI assistant made some mathematical error when asked to verify a mathematical proof, the problems can be simplified to something akin to "why did you generate an image of a golden retriever instead of a doberman?"

We have alluded to difficulties in applying counterfactual strategies to image generation above, but to make this issue more clear, consider the process that the generative model follows when generating an image. Modern image generation uses a tokenizer to map a given string to a set of tokens. A language model then maps each token to a unique, vector that is then passed

Figure 1.1: Images generated using the prompt: “A dog playing”. A 3 token prompt still generates diverse images including different breeds, different backgrounds, and different objects. While the text input space may be interpretable, the major details of the image are governed by a difficult to interpret, initial noise distribution.

to the full generative model which then maps generates an image using a variety of approaches including diffusion [44], GANs [27], recently proposed Autoregressive strategies [123], among others. Regardless of the input text, the language model standardizes the text embeddings by mapping all sequences to vectors of the same dimensionality.

Alongside the text embedding process, the generative models sample from some initial noise distribution that uses the text embeddings to guide the generation process. As the text embeddings are used to guide the initial noise, extremely small text input spaces correlate with a wide variety of output images. Thus, the input space for the generative model is broken into an interpretable input space of tokens and a difficult to interpret noise input space. For example, Figure 1.1 shows several images generated with the prompt “A dog playing”. The text space governs the broad aspects of an image, but the specifics of the breed, the location, whether the dog is playing with a toy or not, whether the dog is dirty while playing, etc, are governed by the initial noise sample.

Investigating counterfactual “what if’s” through methods such as saliency maps or causal discovery inherently prioritize the difficult to interpret noise distribution as it encodes the vast majority of the detail in the image. When applied to the text space, these methods don’t improve over natural human intuition. Without any additional work, humans understand that the image shows a dog, because the prompt used the token “dog”. We understand that it shows a ball or a dog running by using the token “playing”. We understand that it shows a single dog because the prompt used the article “A”. We have a reasonable understanding of the causal map from text to image without relying on external techniques.

Thus the problem of explainability in text-to-image settings hinges on either finding semantic meaning in a high-dimensional noise distribution or encouraging a greater reliance on the text space, ideally relying so strongly on the text space that the image generated is effectively independent of the initial noise sample. Prompt engineering has emerged as a useful tool for doing the latter. For example, if we generate an image using the following text: “A golden-retriever playing alone with a red ball on a grassy field, while running toward the camera”, we have removed all flexibility of the model to choose breed, what “playing” means, background, how the

dog is presented, etc. Across multiple generated images, all images will look approximately the same. By designing explanation methods for the interpretable prompt space, we can limit the influence of randomness in the generative process and gain a consistent understanding for why an image is depicted in a certain way.

The process by which we explore the prompt space is generally described in the literature as *Prompt Inversion* [28]. These methods have emerged as a useful way of defining functions that can express the difference between a ground truth image and the prompt that could potentially generate the image. While the goal is clear, this problem leads to a wide variety of approaches. For example, given an initial prompt estimate, Sohn et al. [117] have found that the image generation process can be effectively controlled by strategically refining the text-processor to better represent specific content styles. In order to avoid finetuning a given model, Wen et al. [137] have leveraged CLIP's [95] embedding space to directly optimize natural language inputs to be close to target images instead of exact recreations. Later, Mahajan et al. [72] proposed an inversion technique that backpropagates through intermediate steps of the diffusion process instead of relying on CLIP embeddings. Although, in practice this approach remains costly. It requires careful control of the timesteps that we backpropagate through, and often struggles with latent diffusion models. Despite not being ideal, it has been found that training a captioner on a dataset of prompt-image pairs effectively learns the prompt distribution well enough to act as an inverter [99].

Focusing on the exact discrete solution is particularly necessary for counterfactual generation. While, most work in prompt inversion focuses on 'soft embeddings', in which we optimize within the continuous space of embeddings, some work has shown that Khashabi et al. [55] have shown that interpreting such embeddings may lead to spurious interpretations. The authors found that if we operate in the continuous space of input embeddings, we can generate an image with any content that we want, while ensuring that we have not strayed far from a token in the discrete token space. This not only removes the human interpretability that we seek for counterfactuals, but degenerates the problem to one in which operating in the continuous space of embeddings allow us to generate arbitrary images with embeddings of arbitrary distance to each other. We therefore believe that explanations must rely on the discrete embedding space, despite the comparative difficulty of discrete optimization methods.

While soft embeddings have been more popular, discrete methods have found success through a variety of startegies include projected gradient methods [137] and MCMC-style sampling using external multimodal models [40]. These methods are often complementary to similar work in adversarial attacks on language models [152]. As in the T2I setting, they face the challenge of optimizing over a discrete space where tools like branch-and-bound or convex relaxations have limited applicability. Strategies for discrete optimization often rely on heuristics to constrain the search space for fine-grained exploration [159].

Our work bridges these technical approaches to prompt optimization with practical bias discovery in T2I systems. Unlike previous methods that either start with predefined demographic categories [13, 24], we ultimately propose a contrastive analysis to surface natural variations in linguistic representation.

1.4 Dissertation Summary

Our contributions in this thesis span several released papers that provide and expand upon existing knowledge in the counterfactual explainability space, while building toward our ultimate goal of an automated process for discovery of representational asymmetries. This dissertation is laid out in the following order:

[142] Rethinking Distance Metrics for Counterfactual Explainability

We introduce a formal mathematical framework to define counterfactuals as distinct from adversarial examples. We introduce significant background information and present several ideas that will be picked up on in later work. Through our framing, we introducing a new distance metric tailored for the counterfactual setting and explore the effects of different assumptions on how counterfactual points are generated and defined.

[144] Prompt Recovery for Image Generation Models: A Comparative Study of Discrete Optimizers

We introduce the problem of prompt recovery and discuss how we can search the input space of a language model in order to have some desired output. We argue that a discrete optimization process may be necessary to faithfully explore the prompt space, otherwise one may find suboptimal solutions, even if the outputs *seem* better than another. We follow this discussion by comparing several discrete optimization strategies and their converegence behaviors.

[141] FUSE-ing Language Models: Zero-Shot Adapter Discovery for Prompt Optimization Across Tokenizers

In [142], we argue that knowledge of the underlying distributions from which ground truth data is generated may be a necessary component of meaningful counterfactual examples. We thus define the underlying prompt distribution for images as a human-readable prompt that aligns with the content in an image. As the discrete optimizations discussed in [144] often require some degree of information about the gradient of a loss function with respect to a given prompt, we introduce a method for preserving gradient information across discrete tokenization/embedding spaces. This method allows us to compose multiple models and solve multiobjective optimization problems across models and applications.

[140] DrawL: Understanding the Effects of Non-Mainstream Dialects in Prompted Image Generation

Here we discuss the value of counterfactual and contrastive prompt analyses to explore implicit biases of models. Through the lens of dialect, we show that biases in image generation models are not limited to explicit behaviors. We find that image generation models have a surprising sensitivity to the user's dialect, effectively conditioning the distribution of representations in

generated images according to the user's dialect. Semantically equivalent prompts in mainstream American English typically generates lighter skinned subjects, while African American English typically generates darker skinned subjects.

[139] Counterfactual Prompt Discovery: Revealing Hidden Representations in Text-to-Image Models

We introduce our formal method for discovering potential bias axes in image generation models through an unsupervised, contrastive search of prompts. Guided by the insights in the experimental design of [140], given two images, we jointly solve for the prompt that aligns with both, while also ensuring a minimal number of token differences between the prompts. Through a small pilot study, we find that a variety of unexpected representation asymmetries including platform-specific biases (e.g., 'Flickr' and 'Shutterstock' being female and male associated) and sensitivity to passive and active phrasing ('riding' being female associated, while 'driving' being male associated).

By building on recent work that examines representational harms in multimodal systems [2, 103], but with a focus on discovering new bias dimensions rather than measuring known ones. Throughout this thesis, we introduce several methods to scaffold the proposed discovery methods. As will be discussed in Chapter 6, this approach is by no means a replacement for standard auditing techniques. We propose a complementary approach to standard auditing techniques that expands the behaviors that we typically test for.

Chapter 2

Rethinking Distance Metrics for Counterfactual Explainability

Counterfactual examples are a powerful way of providing explanations for the underlying behavior of black boxes. By comparing inputs that give different results, the deltas between those inputs exactly tell what can be changed in order to get some different or desired result.

In order to better situate the reader, we show two examples of counterfactuals in Figure 2.1. On the left side we show a toy examples of rejected applicants for a loan, and on the right we show counterfactuals – minimal changes to their original application that would have led to a loan approval. The first applicant applied for a loan with an income of $30,000 per year, a high school education, and heavy debt. In this case the counterfactual tells the applicant that if they go to college and paid off some of their debt, then they would have been approved for the loan. Similarly, for the second applicant, as they have some college, if the focused on increasing their income they would have been approved. These counterfactuals provide users with a form of *recourse*, that gives them direction for getting a more desirable result.

However, there are other benefits of this approach to explainability. Consider the bottom example. Here, the model provided an explanation to the user that they reapply in 20 years. While this is an explanation, the user cannot act on it, and instead of highlighting how the user should change themselves, it highlights an issue in the model's behavior and how it weights different features. This provides direction for model designers to address such issues in the decision boundaries. For users, counterfactual explanations are valuable [127] if they are *plausible*, wherein the explanation is not self-contradictory and points to a viable real-world profile of attributes; and *actionable*, wherein explanations recommend modifications that one could act on (e.g., not recommending that a person reduces their age, or get a doctorate, when they only have high-school education) [71].

Original Applicant: Rejected

Feature	Value
Income	$30,000
Education	High School
Debt	$20,000
Age	25
Lines of Credit	1

Counterfactual: Accepted

Feature	Value
Income	$30,000
Education	Bachelor's
Debt	$5,000
Age	25
Lines of Credit	1

Original Applicant: Rejected

Feature	Value
Income	$32,000
Education	Some College
Debt	$18,000
Age	25
Lines of Credit	1

Counterfactual: Accepted

Feature	Value
Income	$55,000
Education	High School
Debt	$20,000
Age	25
Lines of Credit	1

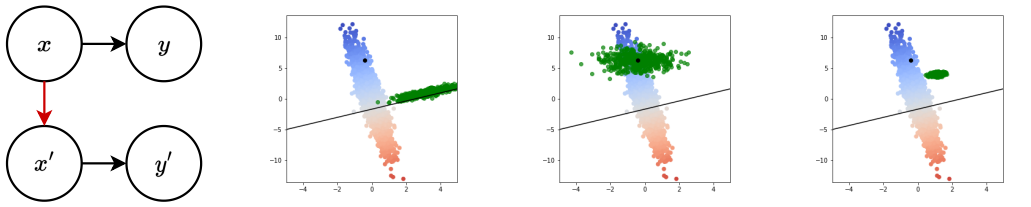
Original Applicant: Rejected

Feature	Value
Income	$45,000
Education	Bachelor's
Debt	$10,000
Age	20
Lines of Credit	2

Counterfactual: Accepted

Feature	Value
Income	$45,000
Education	Bachelor's
Debt	$10,000
Age	40
Lines of Credit	2

Figure 2.1: Counterfactual explanations for a loan classifier. Left: examples rejected by the model. Right: minimal changes (highlighted) that result in acceptance.



(a) Standard PGM (b) $L = I, \gamma = 1$ (c) $L = 0, \gamma = 1$ (d) $L = I, \gamma = 0.05$

Figure 2.2: Comparison of approaches to counterfactual generation; counterfactuals with the proposed prior never leave the data distribution. (Black Dot) Reference, $\mathbf{x}$. (Green) Counterfactual Distribution. (Black Line) Desired predicted output, $y' = A\mathbf{x}' + b$. In all figures, L is the precision of residuals, γ is the weight on l_2 distance, and α controls similarity/distance in our approach.

This highlights one of the primary challenges in generating counterfactual prompts. How do we ensure that all inputs, particularly prompts to generative models, are comparable to reasonable inputs in the wild? In order to better set up the discrete prompt optimization setting that we focus on in later chapters, we first focus on counterfactuals in a continuous space using simpler classifiers. As pointed out in [11], counterfactual explanations have distinct challenges, including: 1) emphasizing the features that are easiest to change may conceal the fact that decisions still rely on immutable characteristics; 2) explanations may react to underlying information that is invisible to the model; 3) 'The Framing Trap' as described in [112], pointing to the failure of the model to capture the entire social system from which the data is generated.

In this chapter, we investigate how to define similarity between counterfactuals in a way that avoids these issues. We focus on the relationship between a known data point, its counterfactuals, and the underlying data distribution. In Section 2.1, we show that this the implicit decisions made on this relationship have strong implications for the resultant counterfactuals, and explicate the distinction between counterfactuals and adversarial examples. While there exists a significant body of work that studies how to generate counterfactuals that respect the underlying data distribution (For example, [51] show that even under imperfect knowledge of an underlying causal model, we can craft approaches that encourage meaningful forms of recourse and [86] show that the latent space of a variational autoencoder holds a depth of knowledge that allows us to find counterfactuals), we find that our framing of the relationship between counterfactual and reference is enough to encourage semantically meaningful counterfactuals, even under comparatively weak assumptions on the structure of the underlying data.

2.1 Understanding Counterfactual Distributions

As described above and in Chapter 1, counterfactual explanations seek to provide a user with a set of points from the input space that are similar to the initial feature vector or *reference*, but receive a different prediction by the decision-making model. Prior work [11, 62] has expressed

concern about generating counterfactuals via a variation of Eq. (1.2) – restated here for clarity:

$$\mathbf{x}' = \arg\min_{\tilde{\mathbf{x}}} ||y' - f(\tilde{\mathbf{x}})||_2^2 + \gamma||\mathbf{x} - \tilde{\mathbf{x}}||_2^2. \tag{1.2}$$

This concerns stems from the inability to guarantee actionability for explainees, i.e., distance metrics, such as Euclidean Distance place equal weight on each input and no restriction on the direction that the feature delta can take. This allows for changes that may be impossible to act on, such as recommending that a person become younger in order to be approved for a loan. In order to present a different perspective on why such methods lead to these issues and to motivate our approach, consider the simple case of a Linear Regression Model, $y = A\mathbf{x} + b + \epsilon$. Our labels, y, are samples from, $\mathcal{N}(A\mathbf{x} + b, L^{-1})$, where $\mathbf{x} \sim \mathcal{N}(\mu, \Lambda^{-1})$, $\mu \in \mathbb{R}^n$, and Λ^{-1}, L are PSD matrices in $\mathbb{R}^{n\times n}$ and $\mathbb{R}^{|y|\times|y|}$ respectively. By re-framing Eq. (1.2) as an equivalent quadratic,

$$\begin{aligned}\mathbf{x}' = \arg\min_{\tilde{\mathbf{x}}} &(y' - A\tilde{\mathbf{x}} - b)^T L(y' - A\tilde{\mathbf{x}} - b) \\ &+ (\mathbf{x} - \tilde{\mathbf{x}})(\gamma I)(\mathbf{x} - \tilde{\mathbf{x}}).\end{aligned} \tag{2.1}$$

The earlier objective becomes the negative log probability of some known Gaussian distribution. Counterfactual generation methods, in the linear setting, can be framed as an instance of sampling from this Gaussian distribution. Moreover the solution to Eq. (1.2) is the mode of that entailed distribution.

Underlying this process is the generative model expressed in Fig. 2.2a. This graph is a representation of the counterfactual posterior for reference, $\mathbf{x}$, and desired predicted outcome, y',

$$p(\mathbf{x}'|\mathbf{x}, y') \propto p(y'|\mathbf{x}')p(\mathbf{x}'|\mathbf{x})p(\mathbf{x}). \tag{2.2}$$

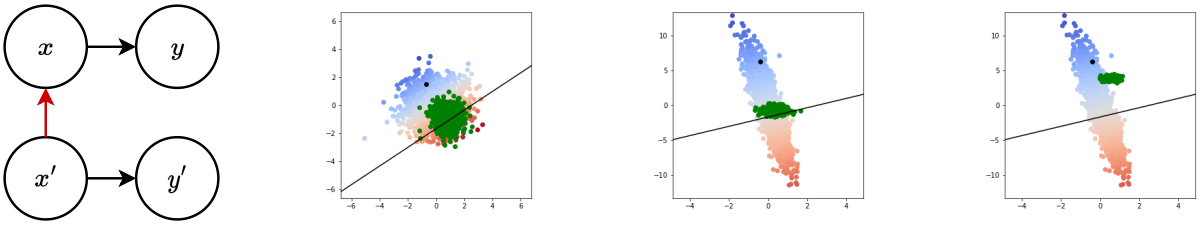
As the reference, $\mathbf{x}$, is fixed a priori, $p(\mathbf{x})$ can be pushed into the proportionality constant and our prior over the explanations effectively becomes $p(\mathbf{x}'|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \gamma I)$. Such a prior, assumes that counterfactual explanations do not come from the true data distribution. Instead, this states that such explanations *only* exist in relation to the reference point.

Note that this simple generative model depicts, not the *data* generation process, but the assumptions inherent within the *counterfactual* generation process. Advancements from prior work that focus on the data generation process, are parallel to our investigation of the assumptions on the counterfactual generation process. We emphasize that by not associating the explanation generation process with the underlying data distribution, it gives rise to the potential for the generative model 2.2a to produce explanations outside of the data distribution.

We show several visualizations of this effect in Fig. 2.2. Fig. 2.2b shows that under common conditions (euclidean distance and variance of residuals is 1), the distribution of counterfactuals can sit entirely in a regions of the space that have near-zero probability wrt. the distribution of data. Fig. 2.2c shows that under the case where we place no emphasis on accuracy for the desired counterfactual, the distribution of counterfactuals centers around the reference, yet it still has tails that lie in these near-zero probability regions.

2.1.1 Regularized Counterfactuals

As the underlying data is Gaussian, one may suspect that this lack of representation can be corrected by applying a Gaussian regularizer that encourages the distribution to be representative



(a) Regularized PGM (b) $L = I, \gamma = 1$ (c) $L = I, \gamma = 1$ (d) $L = 1, \gamma = 0.5$

Figure 2.3: Comparison of approaches to counterfactual generation; counterfactuals with the proposed prior never leave the data distribution. (Black Dot) Reference, $\mathbf{x}$. (Green) Counterfactual Distribution. (Black Line) Desired predicted output, $y' = A\mathbf{x}' + b$. In all figures, L is the precision of residuals, γ is the weight on l_2 distance, and α controls similarity/distance in our approach.

of the underlying data. In other words, by assuming that both $\mathbf{x}, \mathbf{x}' \sim \mathcal{N}(\mu, \Lambda^{-1})$, it may be possible to better control the behavior that causes explanations to sit in extremely low probability regions of the data distribution:

$$\mathbf{x}' = \arg\min_{\tilde{x}} ||y' - f(\tilde{x})||_2^2 + \gamma_1 ||x - \tilde{x}||_2^2 + (\tilde{x} - \mu)^T (\gamma_2 I \odot \Lambda)(\tilde{x} - \mu). \tag{2.3}$$

(If the underlying data distribution is standard normal, then this regularizer is $\gamma_2 ||\mathbf{x}'||_2^2$).

As done for Eq (1.2), a quadratic formulation of Eq (2.3), tells us that this objective is also expressing a known Gaussian distribution, and the optimization problem is simply finding its mode,

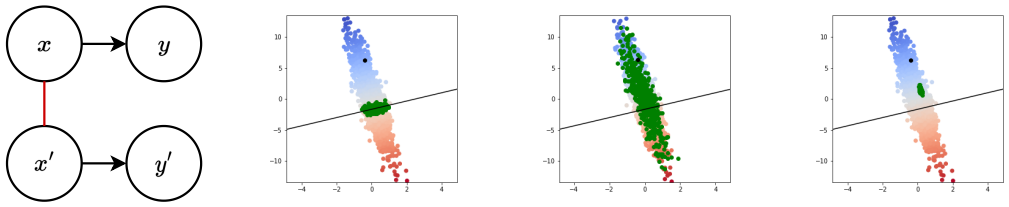
$$\begin{aligned} \mathbf{x}'|\mathbf{x}, y' &\sim \mathcal{N}(\mu_{cf}, \Lambda_{cf}^{-1}) \\ \Lambda_{cf}^{-1} &= (W + A^T L A + \Lambda)^{-1} \\ \mu_{cf} &= \Lambda_{cf}^{-1} \Big(A^T L b - A^T L y' + W x + \Lambda \mu \Big). \end{aligned} \tag{2.4}$$

Moreover, the resultant counterfactual distribution is entailed by the PGM under Figure 2.3a. One can see this by considering the factorization of the joint distribution for Figure 2.3a,

$$p(\mathbf{x}, \mathbf{x}', y', y) = p(y'|\mathbf{x}')p(y|\mathbf{x})p(\mathbf{x}|\mathbf{x}')p(\mathbf{x}')$$

In this case, $\mathbf{x}$ d-separates $\mathbf{x}'$ from y, we drop y in order to express the distribution over only the terms that are dependent on $\mathbf{x}'$. By regularizing the counterfactual optimization problem, we are reversing the dependency on the counterfactual and reference, effectively going against our intuition about what counterfactuals are, by saying that the *reference* that we observe is determined by the *counterfactual*.

We provide visualizations of this distribution in Figure 2.3. Empirically, it seems that while regularization creates a graphical model that runs counter to our intuition, it does address some



(a) Proposed PGM (b) $L = I, \alpha = 0$ (c) $L = 0, \alpha = 0$ (d) $L = I, \alpha = 0.995$

Figure 2.4: Comparison of approaches to counterfactual generation; counterfactuals with the proposed prior never leave the data distribution. (Black Dot) Reference, $\mathbf{x}$. (Green) Counterfactual Distribution. (Black Line) Desired predicted output, $y' = A\mathbf{x}' + b$. In all figures, L is the precision of residuals, γ is the weight on l_2 distance, and α controls similarity/distance in our approach.

of the issues from before. Most notably in comparing figures 2.2b and 2.3c, this form of regularization does encourage the distribution to stay toward higher probability regions of the data distribution. However, as shown in Figure 2.3d this encouragement may not be enough.

Moreover, as can be seen in Figure 2.3b, unlike in the previous case, this new distribution places lower emphasis on returning counterfactuals with the desired label. The proposed fix in Eq. (2.3) encourages a heavy trade-off between representativeness of the underlying data distribution and ensuring counterfactuals that tightly cluster around the desired label.

2.2 Ensuring Representative Counterfactuals

Our proposed framework addresses these issues by treating the counterfactual, not as a new point to generate, but instead as simply an unobserved point within the data distribution. For ease of exposition, we continue to focus on the case of explaining Linear Regression Models, $f(\mathbf{x}) = A\mathbf{x}+b$, before expanding to more complex settings, including neural networks. Although linear models often do not need explanations, such models exactly express the distribution of counterfactual explanations and serve as a clear comparison to Eq. 1.2.

Given an input, $\mathbf{x} \sim \mathcal{N}(\mu, \Lambda^{-1})$, to a decision-making model, f, with output $y = f(\mathbf{x})$, counterfactual explanations methods seek to explain why the model labeled $\mathbf{x}$ with label y, by choosing points, $\mathbf{x}'$, from the set of all possible counterfactuals (Def. 1). This set of explanations is expressed via three components: A prior on the relationship between the reference and the counterfactual, the likelihood of the desired y' given x', and a prior on the data distribution.

The key idea of our approach is that while counterfactuals are often considered to be wholly dependent on the reference, as shown by the directed edge in Fig. 2.2a, we should treat $\mathbf{x}$ and $\mathbf{x}'$ as dependent on one another. Just as we consider a reference, $\mathbf{x}$, as existing somewhere within the input space, counterfactual explanations exist a priori within this space. Their codependency is expressed in the generative model (Fig. 2.4a) via an *undirected* edge between $\mathbf{x}$ and $\mathbf{x}'$.

While a subtle distinction, the choice of joint distribution over $\mathbf{x}$ and $\mathbf{x}'$ has a significant impact on the selected counterfactuals. In this work we express the distribution over reference

and counterfactual with the form,

$$p(\mathbf{x}, \mathbf{x}') = \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \Lambda^{-1} & W \\ W^T & \Lambda^{-1} \end{bmatrix} \right) \tag{2.5}$$

The relationship between x and x' are entirely defined by a correlation matrix, W, and the marginals are defined as the observed data distribution, $\mathcal{N}(\mu, \Lambda^{-1})$. While W can be any positive semi-definite matrix, in order to express the correlation between counterfactual and reference, we suggest defining, $W = \alpha\Lambda^{-1}$, where $\alpha \in (0, 1)$. Should $\alpha = 1$, we have the degenerate case in which $\mathbf{x}$ and $\mathbf{x}'$ are perfectly correlated. This places no emphasis on having $f(\mathbf{x}') = y'$. On the other hand, $\alpha = 0$ implies that $\mathbf{x}$ and $\mathbf{x}'$ are independent draws from the same distribution, which in turn emphasizes choosing $\mathbf{x}'$ such that $f(\mathbf{x}') = y'$. Scaling α from 1 to 0 scales the similarity between reference and counterfactual.

As in the previous section, the posterior of our recommended graphical model remains Gaussian. Moreover, we can express its distribution, for a linear regression, analytically. Under this framing, we generate similar distributions to those shown in the top half of Figure 2.2. The joint prior recommended here restricts the distributions of counterfactual explanations to stay within the data distribution. The most striking example of which, Figure 2.4c well illustrates the implications of this new prior, and the semantic questions that we pose. If we ask an algorithm to generate a counterfactual which neither emphasizes the desired predicted label, y', nor the similarity to the reference, $\mathbf{x}$, the Wachter et al. [131] framing from Eq. (1.2), returns any value, $\mathbf{x}' \in \mathcal{R}^n$, however, in this same circumstance, the form introduced here is constructed to exactly match the data distribution. Without emphasis on y' nor $\mathbf{x}$, counterfactuals are simply samples from the data distribution.

2.3 Extending the Proposed Framework to Complex Models

In Section 2.2, we focused on recommending a change to the graphical model that underlies Counterfactual Explanation generation methods. We introduced a prior that allows us to express the relationship between the reference $\mathbf{x}$ and counterfactual $\mathbf{x}'$ in terms of underlying data distribution. Section 2.2 was restricted to the linear regression model; here, we show how to express this prior in more complex decision settings.

Consider a multi-class classification setting[1] in which decisions are made by a differentiable model, $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} \in \{0, 1\}^m$ is some categorical labeling. For a given outcome, we sample counterfactuals by first splitting the network into two sections; the first $N - 1$ layers being the feature representation, $r : \mathcal{X} \to \mathbb{R}^m$, and the second being a linear output layer. The full network takes the form, $f(x) = \sigma(w^T r(x))$, where $\sigma(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$ is the softmax function; the posterior distribution over the reference, $\mathbf{x}$, explanation $\mathbf{x}'$, and desired predicted

[1]There are many settings in which we would like to generate counterfactual explanations where one may not have access to the model itself (eg. a vision API) or if the decision-making model is non-differentiable (eg. decision-trees); WLOG, we can train a new model to mimic decisions and reduce the problem to the considered case

outcome, $y \sim \text{Categorical}(p)$ follows,

$$\begin{aligned} p(x'|y'x) &= p(x|x')p(y'|x';r)p(x') \\ &= \mathcal{N}(x|\mu_{x|x'}, S^{-1}_{x|x'}) \times (y'^T\sigma(w^T r(x')) \times \mathcal{N}(x'|\mu, \Lambda^{-1}), \end{aligned} \tag{2.6}$$

Unlike the linear case, by considering the learned representation of the network, $r(x)$, we are introducing another degree of uncertainty over model weights. We can introduce a prior over the networks output weights, in order to capture this uncertainty, and return a fully Bayesian model. We marginalize over the set of all possible output weights under this prior in order to average out our uncertainty.

$$\begin{aligned} p(x'|x, y'; \phi, t) = \mathcal{N}(x|\mu_{x|x'}, S^{-1}_{x|x'})\times \\ \int_w (y'^T\sigma(w^T r(x')) + t^T\sigma(w^T r(\phi)) \times \mathcal{N}(w|0, I) \times \mathcal{N}(x'|\mu, \Lambda^{-1})dw., \end{aligned} \tag{2.7}$$

where ϕ are all other points in the dataset, and t are the corresponding outputs of the decision-maker for inputs ϕ. Similarly to how one would perform a Bayesian Logistic Regression, we perform a Laplace Approximation on the integrand in order to simplify the process of marginalizing over the weights, and ensure that we have a Gaussian form for the counterfactual distribution.[2]

As we consider the outcome, y to be categorical, the integrand is effectively dependent only on x'. Thus, a Laplace Approximation,

$$g(x'|y') \sim \mathcal{N}(\mu_{\tilde{x}}, \Lambda^{-1}_{\tilde{x}}) \approx \int_w p(y'|x';r)p(w|\phi, t)p(x')dw,$$

can be considered as learning a new prior over the data distribution. Whereas $p(x')$ may cover the entire data distribution, $g(x'|y)$ covers only the region of the data distribution that corresponds to label y. Generating counterfactual explanations then amounts to sampling from the posterior,

$$g(x'|x, y') \propto\!\!\!\!\sim p(x|x')g(x'),$$

in which $p(x'|x, y')$ is Gaussian, and $\propto\!\!\!\!\sim$ is defined as 'approximately proportional to'. In Appendix A, we include a discussion on the practical considerations for incorporating the Laplace Approximation in this setting.

2.3.1 Extending the Proposed Framework to Complex Data

We often choose to use complex decision-making models, such as deep networks, due to the fact that the relationships in the data cannot be expressed through simple, linear relationships (eg. convolutional filters in images). In such cases, we cannot directly sample from the counterfactual distribution in Eq (2.6), due to the fact that we cannot express an effective prior over the

[2]This restriction to be Gaussian is not particularly necessary. As in the main text, we focus on the Gaussian case for this work in order to make our manipulation of the posterior more easily understandable and to allow for easier sampling, however, we can perform various off-the-shelf methods of sampling from a posterior distribution in order to sample arbitrary distributions.

data, ie. images cannot be reliably generated by randomly sampling pixel values. Without an effective prior on the space of counterfactuals, counterfactual explanations for complex data are functionally equivalent to adversarial perturbations, as has been pointed out in [30].

In such cases, engineers often opt to use generative models, which allow them to sample from an underlying latent space and pass this sample through a generator that maps into the input space. We follow a similar approach, by placing a prior not on the input space, but on a Gaussian latent space, and include the latent decoder, $\mathrm{d} : \mathbb{R}^k \to \mathcal{X}$, that maps from the latent space into the input space.

$$p(x, y, x', y') = p(x|l; \mathrm{d})p(y|l; r, \mathrm{d})p(l);\ x' = \mathrm{d}(l) \tag{2.8}$$

$$= \mathcal{N}(x|\mathrm{d}(l), S^{-1}) \times (y^T \sigma(A^T r(\mathrm{d}(l)) + b) \times \mathcal{N}(l|\mu, \Lambda^{-1}), \tag{2.9}$$

There are various ways that one may represent the Gaussian latent space (eg. Normalizing flows [100] or Variational Auto Encoders [57]). However, once this encoding/decoding is learned, the sampling process itself bears no further difference from Section 2.3.

Importantly, we can engineer the decoding layer to allow us to address the issues of normalizing features that have very different scales [11]. Commonly prior work on counterfactual explanations use the Median Absolute Deviation (MAD) under the L1 norm [81] in order to allow for optimizing the counterfactual objective, however through this encoding/decoding approach, we can express any feature that we have to normalize through a Gaussian latent variable and decode into the desired scale. For example, one may encode income as the exponential of a Gaussian latent variable or one may encode categorical features as the softmax of a vector of independent Gaussians, and binary features as the sigmoid of a Gaussian.

2.4 Domain Knowledge in the Prior

As stated in prior work [52, 62], the challenge of generating counterfactual explanations hinges on finding changes to the input that are plausible (ie. the explanation could potentially exist), actionable (ie. the explanation recommends changes that are possible for one to make), and give the explainee direction to change themselves. In this section, we show how the counterfactual prior, $p(\mathbf{x}, \mathbf{x}')$, and the resultant posterior, can express several forms of actionability. While, one can use any off-the-shelf method of sampling from a non-Gaussian posterior, throughout the remainder of this work, we focus on the Gaussian case in order to ensure an easy to visualization.

2.4.1 Accounting for Actionability Constraints

As described in Karimi et al. [52], the features of a actionable counterfactual explanations can be subdivided into three distinct categories: (a) **Mutable:** features for which a counterfactual explanation may change freely (Eg. bank account balance); (b) **Immutable:** Non-Actionable features for which under no circumstances we change from the reference input (eg. race); (c) **Mutable but Non-Actionable:** features that can change only as a result of other features changing (eg. credit score). Such explanations can be achieved by manipulation of the prior on the distance between the reference and counterfactual, the prior on counterfactual distribution, and the posterior, $p(\mathbf{x}'|\mathbf{x}, y')$.

Mutable. Mutable features may be freely changed and require no additional transformations.

Immutable. Recall that we express the correlation between reference $\mathbf{x}$ and counterfactual $\mathbf{x}'$ as, $W = \alpha\Lambda^{-1}$, where $\alpha \in (0, 1)$ and Λ^{-1} is positive semi-definite. If $\alpha = 1$, the reference and the counterfactual are perfectly correlated and $\mathbf{x} = \mathbf{x}'$. As such, we can express immutable features through the covariance, $cov(\mathbf{x}, \mathbf{x}') = W$. We set features as immutable through the following adjustment to W:

$$W = \sigma\sigma^T \odot (\alpha - 1)\Lambda^{-1} + \Lambda^{-1}$$
$$\sigma_i = \begin{cases} 0 & x_i \in \text{immutable} \\ 1 & o.w., \end{cases}$$

In other words, we enforce immutability by requiring a perfect correlation between immutable features of $\mathbf{x}'$ and $\mathbf{x}$.

Mutable, Non-Actionable. For such cases in which an explainee may be unable to directly influence an outcome (eg. one cannot directly affect credit score; scores change as a result of other actions), a counterfactual treats the non-actionable features as being collinear with respect to their causal ancestors, regardless of the evaluated posterior. We express these features, through a prior that encodes causal dependencies between features. First, find the distribution of counterfactual explanations $p(\mathbf{x}'|\mathbf{x}, y')$. Then consider a counterfactual as a tuple of causal ancestors and descendants, $\mathbf{x}' = \left(c', e'\right)^T$ in which e' are mutable, non-actionable features and c' are all others. We express mutable, non-actionable features by first marginalizing over e',

$$p(c'|x, y') = \int_{e'} p\big(\left(c', e'\right)^T |y', x\big)\, de' = \mathcal{N}(\mu', \Lambda_{c'}^{-1}).$$

We then find the weights of the linear model $e' = Ac' + b$, and express the mutable, non-actionable features as having come from the conditional distribution, $p(e'|c') = \mathcal{N}(e'|Ac' + b, \Lambda_{z'}^{-1})$, where $\Lambda_{z'}^{-1}$ is covariance of the residuals. The updated counterfactual distribution takes the form,

$$p(\mathbf{x}'|\mathbf{x}, y') = p(e'|c')p(c'|\mathbf{x}, y').$$

2.5 Revisiting Counterfactual Optimization

Up to this point, we have primarily focused on sampling explanations from a known probability distribution, however, it may be helpful to understand our approach in terms of optimizing an objective. While the discussed approach to incorporate domain knowledge can still be applied to derive an optimization objective, we focus here on the simple case of only mutable features in order to provide a clear comparison in our results to prior work. Recall the posterior of the counterfactual distribution from Eq. (2.2),

$$p(\mathbf{x}'|\mathbf{x}, y') \propto p(y'|\mathbf{x}')p(\mathbf{x}'|\mathbf{x})p(\mathbf{x}).$$

By minimizing the negative log-likelihood of this posterior for our chosen prior, we can express the task of generating counterfactual explanations as optimizing the following objective,

$$\begin{aligned}\mathbf{x}' = \arg\min_{\tilde{\mathbf{x}}} \; & \tilde{\mathbf{x}}^T \Lambda \tilde{\mathbf{x}} - 2\tilde{\mathbf{x}}^T \Lambda \big((1-\alpha)\mu + \alpha \mathbf{x}\big) \\ & + \gamma ||y' - f_\theta(\tilde{\mathbf{x}})||.\end{aligned} \tag{2.10}$$

The previously considered norm-ball on the distances used by prior work becomes the mahalanobis distance of samples $\mathbf{x}'$ from a set of observations with mean, $(1-\alpha)\mu - \alpha\mathbf{x}$, and covariance, Λ^{-1}. In other words, we are drawing a line from the mean of the data distribution to the reference, $\mathbf{x}$ and returning points that have the desired class by sampling $\mathbf{x}'$ from around a point on this line.

2.6 Evaluation

In this Section, we evaluate our approach through both a quantitative and qualitative lens. We first compare our proposed approach with several counterfactual generation techniques across a variety of evaluation metrics and datasets. We then investigate its efficacy for more complex image data. We show that the proposed framing encourages explanations to lie further from the decision boundary, so as to produce counterfactuals that are more representative of the ground truth data. We further perform a qualitative evaluation on whether users find explanations across methods satisfying through an Amazon Mechanical Turk Survey.

2.6.1 Quantitative Evaluations

We use the CARLA [87] counterfactual benchmarking tool in order to compare our proposal with several existing counterfactual generation methods:

- `Wachter` [131], which solves Eq. (1.2)
- `DiCE` [81], which adds a diversity regularizer to Eq. (1.2) to generate a large, diverse set of counterfactuals at once. For this evaluation, we generate 3 counterfactuals per reference point.
- `FACE` [92], which chooses counterfactuals by traversing a nearest-neighbor graph over the observed data, until reaching an instance that has the desired label.
- `Growing Spheres` [61], which iteratively samples an expanding set of points around a given reference until a sample lies across the decision boundary.
- `CCHVAE` [86], which uses a variational autoencoder (VAE) to estimate the generative process for a given instance, and returns counterfactuals by sampling within the l_p sphere around a reference in the latent space.

In order to compare against our approach, we replace the distance metric in `Wachter`, `DiCE`, `Growing Spheres` and `FACE` with ours in Eq. (2.10). We designate this choice of the distance metric with the identifier `(Ours)` in Table 2.1. Additionally, as our approach is dependent on the underlying data distribution, we include a comparison against `CCHVAE` in order to evaluate the effectiveness of a method that traverses a learned latent space, rather than staying within

the feature space. Each method's parameters were chosen independently via a grid search that sought to find the parameters that minimize the l_2 distance to the reference, while ensuring that the method generates counterfactuals of the desired class with at least 99% success rate.

We generate 3000 counterfactuals for every method across each dataset and evaluate different methods over five metrics (See [87] for more information on the specifics of how these metrics are calculated.)

- $\boldsymbol{l_2}$, the average l_2 distance between the generated counterfactuals and the reference.
- $\boldsymbol{l_\infty}$, the average l_∞ distance between the generated counterfactuals and the reference.
- **yNN**, the number of nearest neighbors with the desired label. Based on a desideratum formulated by [62], a desirable property of counterfactuals is that they lie close to observed data that has the desired label. This metric captures this property by finding the proportion of a counterfactual's K nearest neighbors in the observed data that have the desired label (here, we set $K = 5$).
- **Redundancy**, the number of features for a given counterfactual that can be changed back to the reference value without changing the counterfactual class (i.e., the number of unnecessary changes wrt. the classifier's predicted output).
- **Diversity**, the diversity of the generated counterfactuals based on the metric defined in [81].
- **T(S)**, the average number of seconds required for a method to generate a single counterfactual.

ADULT

METHOD	l_2	l_∞	YNN	REDUN.	DIV.	T(S)
WACHTER	0.009	0.081	0.058	4.910	-	0.758
WACHTER (OURS)	0.018	0.069	0.115	3.872	-	0.280
DICE	0.028	0.088	0.137	3.703	0.101	0.007
DICE (OURS)	0.039	0.123	0.191	3.870	0.121	0.650
FACE	2.201	0.891	0.612	3.789	-	22.175
FACE (OURS)	2.444	0.842	0.877	4.358	-	15.439
GROWING SPHERES	1.057	0.684	0.137	3.920	-	0.002
GROWING SPHERES (OURS)	1.553	0.694	0.159	4.400	-	0.036
CCHVAE	0.206	0.279	1.000	9.111	-	0.002

RICE

METHOD	l_2	l_∞	YNN	REDUN.	DIV.	T(S)
WACHTER	0.091	0.118	0.18	2.466	-	0.010
WACHTER (OURS)	0.122	0.121	0.415	2.878	-	0.005
DICE	0.082	0.184	0.104	3.114	0.074	0.471
DICE (OURS)	0.115	0.196	0.326	1.762	0.055	0.587
FACE	0.691	0.125	0.747	4.1	-	0.405
FACE (OURS)	0.258	0.255	1.000	6.3	-	0.425
GROWING SPHERES	0.110	0.209	0.061	2.991	-	0.005
GROWING SPHERES (OURS)	0.119	0.213	0.156	3.703	-	0.039
CCHVAE	0.192	0.241	0.502	2.332	-	0.080

HOME EQUITY LINE OF CREDIT

METHOD	l_2	l_∞	YNN	REDUN.	DIV.	T(S)
WACHTER	0.069	0.053	0.112	1.788	-	0.232
WACHTER (OURS)	0.078	0.079	0.186	6.948	-	0.048
DICE	0.059	0.082	0.161	9.869	0.104	0.477
DICE (OURS)	0.088	0.112	0.340	11.465	0.0854	0.506
FACE	1.054	0.701	0.729	13.863	-	2.873
FACE (OURS)	1.172	0.735	0.992	16.535	-	2.109
GROWING SPHERES	0.074	0.112	0.147	14.986	-	0.003
GROWING SPHERES (OURS)	0.086	0.120	0.152	15.243	-	0.061
CCHVAE	1.524	0.635	0.997		-	0.489

GIVE ME SOME CREDIT

METHOD	l_2	l_∞	YNN	REDUN.	DIV.	T(S)
WACHTER	0.006	0.018	0.289	7.558	-	0.005
WACHTER (OURS)	0.010	0.053	0.341	7.214	-	0.589
DICE	0.072	0.155	0.737	7.437	0.170	0.872
DICE (OURS)	0.092	0.148	0.772	7.879	0.127	0.533
FACE	0.625	0.530	0.993	8.154	-	1.792
FACE (OURS)	0.668	0.542	1.000	8.399	-	3.697
GROWING SPHERES	0.006	0.044	0.258	7.055	-	0.003
GROWING SPHERES (OURS)	0.013	0.071	0.403	7.036	-	0.148
CCHVAE	0.491	0.467	1.000	9.401	-	0.001

Table 2.1: Benchmarking table comparing our proposed counterfactual distance with an l2 distance metric across 4 datasets, showing that while our approach increases runtime, it generates counterfactuals significantly closer to the underlying data distribution as measured by the number of nearest neighbors who share the desired label (YNN), without a significant decrease in performance across any other metric.

In nearly all cases, using the metric in Eq. (2.10) encourages counterfactuals to sit more closely to the region of the feature space for which their neighbors have the desired predicted class (i.e. increases **yNN**). We see this effect regardless of the method used.

Moreover, we see that our method generally increases the euclidean distance to the reference. This is expected behavior as we are comparing against methods that explicitly optimize for this metric. Yet, despite our approach not improving over the alternatives for this metric, we find that our approach is not significantly worse in terms of l_2 distance. Using the objective in equation (2.10) effectively gives up a small degree of l_2 similarity in order to encourage counterfactuals that are more clear examples of the desired class.

We also see except in the case of the *Home Equity Line of Credit* dataset, generating counterfactuals according to Wachter et al. [131], we decrease the number of unnecessary features changed from the reference (i.e. REDUNDANCY). However, when adding the diversity regularizer from Mothilal et al. [81], we lose this benefit. Upon further investigation for this specific case, we find that the distribution is highly anisotropic; there is a very large difference between the largest and smallest eigenvalues, 2 orders of magnitude larger than any other considered dataset. The principal axis as defined by the eigenvalues of the covariance matrix is also not particularly informative for the classifier. Thus in order to maintain faithfulness to the original distribution, the counterfactuals change along the minor axes. This encourages changes to a large number of features, only some of which are necessary for crossing the decision boundary.

Outside of the case of Wachter et al. [131], we find that applying additional regularizers encourages our method to change a larger number of redundant features than the alternative. For similar reasons to the *Home Equity Line of Credit* dataset above, applying a diversity regularizer with our proposed approach encourages points to be distinct from one another. This puts a greater emphasis on the minor axes as defined by the eigenvalues of the covariance matrix and in turn encourages more redundant changes as the number of counterfactuals generated by DICE increases. Similarly in the case of FACE, the nearest neighbor to a point as defined by the Mahalanobis Distance in equation (2.10), will define nearby points as those with small changes along the principal axes of the data. If the principal axis is uniformative for the classifier, the method will traverse along the minor axes. As in the previous cases, this more quickly builds up small changes to a counterfactual, increasing the number of redundant features changed from the reference.

2.6.2 Qualitative Evaluation

In Figure 2.5, we show how this new objective changes counterfactuals as compared to the Wachter et al. [131] objective in Eq. (1.2) wrt. euclidean distance. We generate a variety of counterfactuals for the Fashion MNIST dataset [146] and focus solely on the implications of the change in the underlying graphical model by comparing the distance metric used in Eq. (1.2) to the metric used in Eq. (2.10). In addition, we compare to counterfactuals generated by a variational autoencoder, by finding counterfactuals by traversing the learned latent space. While a great deal of work has built on Eq. (1.2) via a variety of different approaches, these techniques and recommendations still apply under our recommended mahalanobis distance. We show how our prior changes the baseline for generating counterfactuals.

Fashion MNIST Counterfactual Explanations.

In order to generate the images in Figure 2.5, we train a simple neural network, $f_\theta : \mathcal{X} \to \{0, 1\}^{10}$ to classify articles of clothing from Fashion MNIST.

While not a dataset that one traditionally treats as Gaussian, we map Fashion MNIST into our setting by applying a logit transform, $\log\left(\frac{|x-\epsilon|}{1-|x-\epsilon|}\right)$ to the grayscaled images and express the data distribution's mean and covariance as the mean and covariance of the dataset's logits. In order to ensure that the covariance matrix is non-singular, we apply a small degree of Gaussian noise to each of the pixel logits.

Figure 2.5 shows that our approach encourages semantically meaningful changes to the reference images. For example, the *Bag* $\to$ *T-Shirt* counterfactual using l_2 distance provides a noisy sleeve outline, however, the distance function entailed by our approach introduces a clear set of sleeves. As we allow explanations to stray further from the reference and closer to the desired class ($\alpha = 0.3$), rather than finding explanations that move out of the distribution and become adversarial, we instead introduced more nuanced changes that bring us closer to the prototypical form for the desired class. For example, consider counterfactual *Shirt* $\to$ *Pullover*, pullovers generally have longer sleeves than torsos; decreasing α subtly shortens the waist.

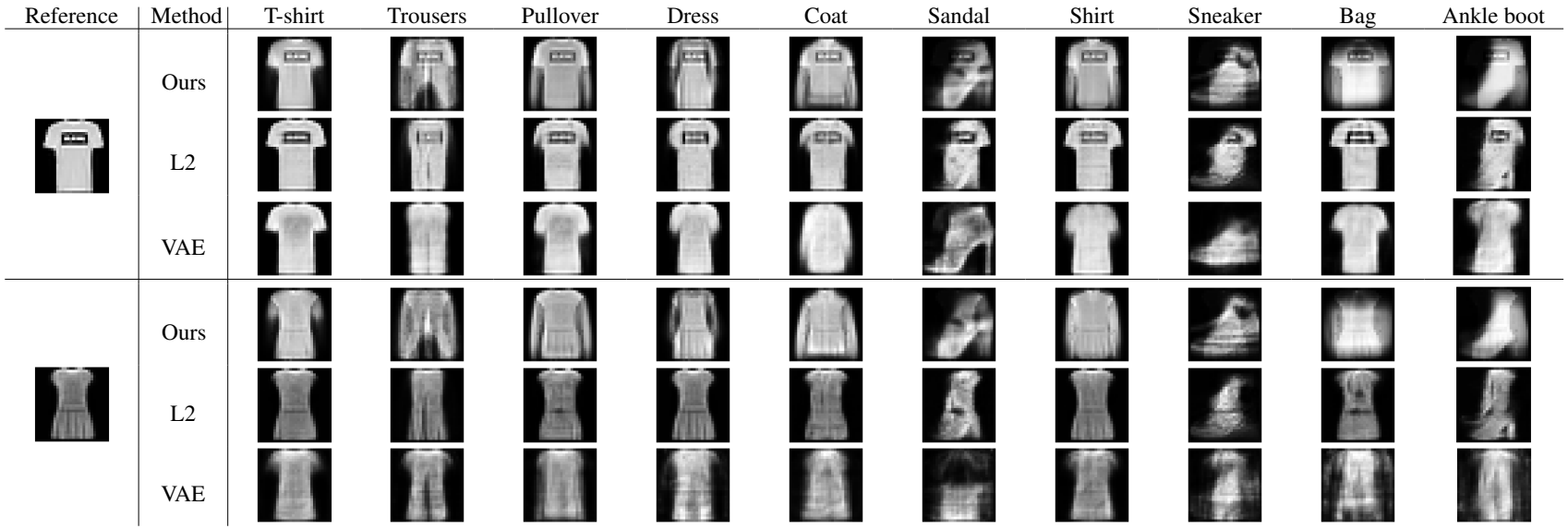
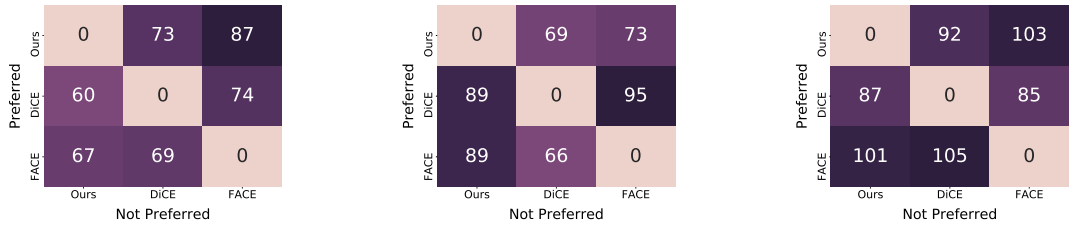


Figure 2.5: All images have 99% certainty for the desired class based on the trained classifier. Our proposed approach produces counterfactual images that, while further from the reference images than those generated using the L2 distance, exhibit more semantically meaningful features associated with each class. Additionally, our approach avoids the class mixing observed when traversing the VAE's latent space.



(a) German Credit (b) LUCAS0 (c) Adult

Figure 2.6: Preference matrices for survey responses on each dataset. Each cell shows how often a respondent preferred the row method to the column method–darker colors imply a greater preference. Each method seems to excel on different types of data.

2.6.3 Survey Evaluation

While we focused on evaluating the overall results in the previous section, we are also interested in comparing how these explanations are perceived by users. We evaluated the subjective characteristics of our approach via a human-subjects survey on Amazon Mechanical Turk (AMT). Our evaluation proceeded as follows: Each participant was first introduced to the decision-making context; before being prompted to respond to a series of 12 pairwise comparison questions, in which they were provided randomly generated counterfactuals from two different methods at a time. Respondents chose which explanation was most 'satisfying' to them, and wrote a short justification that describes the reasoning for their choice. We aggregated the preferences and performed a binomial test to determine statistically significant preferences among methodologies with respect to each dataset.

Decision-making Contexts. Each respondent was randomly assigned to one of three hypothetical decision-making contexts based on three tabular datasets: LUCAS [37], Adult [26], and German Credit [26]. As stated above, Appendix C.1 provides details on these datasets, yet at a high-level: LUCAS allows us to investigate whether respondents' knowledge of causal dependencies influences their preferred explanation; Adult allows us to investigate to what extent a respondent's background knowledge of a social system influences their preferences; and German Credit allows us to investigate how respondents' preferences change when the model utilizes a large feature set, making it difficult for respondents to understand all dependencies.

Counterfactual Generation Methods. As described in our quantitative evaluation, one way of interpreting our approach is as a sliding scale between algorithms that sample counterfactuals from a region around a reference and algorithms that sample counterfactuals as instances from the underlying dataset. Our experiments investigate whether participants have a preference for one side or another in this dichotomy. Thus, we contrast a middle ground α in our approach with two existing counterfactual explanation methods that align with these extremes: **Diverse Counterfactual Explanations (DiCE)** [81] and **Feasible and Actionable Counterfactual Examples (FACE)** [92]. We initially hypothesize that participants prefer a set of actionable changes in line with ensuring plausibility above all else. This entails that preferences for would have the ordering from least to most preferred: 'DiCE (*Implausible*)', 'Ours (*Relaxed Plausibility*)', 'FACE (*Strictly Plausible*)'

Findings

For each dataset, we analyze the participants' preferences, and review the justifications for each preference. We conclude with a discussion of the commonalities and differences among justifications.

German (N=430 comparisons). German credit shows no significant preference for one method over another, however, as shown in Fig. 2.6 our approach is slightly preferred to both FACE ($p = 0.062$) and DiCE ($p = 0.149$). Respondents seemed to prefer explanations that were more different from the reference as they perceived these cases as more detailed. For example, one respondent justified their preference with: *'Method [FACE] seems more satisfactory to me because it is more descriptive in its credit requirements.'* Another with: *[Ours] includes more data that would matter more when making a decision.'* 28 of the 430 choices explicitly listed that having more detail was the primary reason for preferring a given explanation; only 2 preferred having fewer changes.

In addition, participants gave a great deal more focus on those features for which their pre-existing beliefs align with credit worthiness: *'Id use [DiCE] because it mentions employment and his good credit score. It does not mention his other debts though. I had a hard time choosing because of that.'* Potentially due to the participants' existing intuition on the information relevant to credit worthiness, they may disregard explanations that do not fit their existing beliefs.

These factors may play into the reasons for why our proposed method was more preferred than the alternatives. While DiCE optimizes for minimal changes, explainees preferred a wider set of changes that allow for more flexibility in what sorts of changes could potentially be enacted. On the other hand, participants often listed continuous features such as the amount of credit requested or loan duration in months as a major reasons for choosing one explanation over another: *'Method [Ours] makes more sense because it provides valid reasons including credit amount and duration and employment duration...'* FACE finds explanations from within the dataset. Without a large number of samples from which to choose, the mix of features on very different scales may be giving more preference to methods such as ours or DiCE that allow for new points to be generated as explanations. Our method would thus be the preferred approach due to not being as susceptible to either case.

Adult (N=573 comparisons). Respondents on this set of data gave no statistically significant preference for any particular method, however, as shown in Fig. 2.6, there was a small degree of preference for FACE over DiCE (p=0.084). Participant preference justifications also varied significantly. As in German Credit, a common theme that emerged was that participants seemed to prefer explanations that had a greater number of changes from the reference: *'Method [FACE] is much more detailed and gives more information to make a better informed decision of the person in question. Method [Ours] has less information makes it less satisfying and harder to fully judge the person.'* At least 40 of the 573 comparisons for this dataset justified their preference by a combination of 'more details', 'more information', and 'less restrictions'. Some respondents even went so far as to choose the explanation with a greater number of changes because the alternative had too few changes: *'Method [FACE] has too few changes to get up to ¿50k a year.'* Respondents explicitly disagreed with the classifier because the changes were too subtle. In contrast, only 8 cases out of the 573 explicitly listed that they chose one explanation over another due to that explanation having fewer changes.

Outside of the number of changes, dependencies among covariates led to participants labeling potential explanations as implausible: *"Method [FACE] lists a doctorate but that degree probably isn't necessary for tech support."* Modeling dependencies between features is necessary in order to avoid such cases, however, no method excels here. Alternatively, many respondents chose a preferred explanation based on a single feature that made the most sense to them: *'contain [sic] technical level occupation'* or *'working hours is more than the other'* . In cases where an option is unreasonable, participants default to the alternative, regardless of its plausibility. For example, one explanation suggested working 99 hours per week: *'99 hours is too many hours to compare to'*

These reasons do not lend themselves to being solved by any of the considered benchmarks. While not a significant preference, the FACE algorithm does not return implausible points, however, when traversing the KNN graph, after a few steps, FACE no longer encourages making minimal changes. It seems that FACE is preferred due to its propensity of returning distant explanations, while guaranteeing plausibility.

LUCAS (N=481 **comparisons)**. As shown in Fig. 2.6, participants on the LUCAS dataset were found to have a statistically significant preference for DiCE to FACE (p=0.014), a nearly significant preference for DiCE to Ours (p=0.065), and a slight preference between Ours and FACE ($p = 0.119$).

As LUCAS is a synthetic binary dataset with causal dependencies, respondents seemed to prefer explanations that fit more closely to their understanding of these causal relationships. For example, one respondent justified their preference as: *"With lung cancer, smoking is such a strong indicator, or correlator. Anxiety provides a reason why tey [sic] are a smoker, extra evidence."* . One participant had a particularly detailed understanding of the underlying dynamics: *"The methodology of anxiety being the main factor in this prediction leads me to assume that the fact they have Yellow Fingers means they smoke, whereas Method [FACE] states they don't which is wrong..."* This would imply that participants prefer methodologies that better adhere to the true distribution of data. However, as DiCE, which does not use this information, has a statistically significant preference over the other methods, there may be another reason that supercedes faithfulness to the data distribution when determining preferences.

Some participants pointed to specific features as being less preferable to change: *"It would make the person's life much harder cause he has the peer pressure mess with him."* and *"i take yellow fingers over anxiety any day."* Rather than emphasizing plausibility, the underlying cost that a person places on each feature seems to play a greater role. DiCE may be the preferred method because making minimal changes with the greatest impact decreases the potential for changing ancillary features which people place a high cost on. By considering the conditional dependencies in our method or FACE, we are more likely to include the low-probability outcomes that correlate to these high-cost changes (e.g., facing peer pressure and anxiety without being a smoker).

2.7 Discussion and Future Directions

Here, we have provided essential background not only on definitions of counterfactual explanations, but also we have provided a new framework suited for generating them by revisiting

their underlying generative model. We have shown that that conventional optimization-based counterfactual explanation methods contain an implicit assumption: counterfactuals are sampled from a ball centered at the reference point rather than from the underlying data distribution. This assumption results in unrepresentative explanations that effectively function as adversarial perturbations, as they are generated independently of the true data distribution.

To address this limitation, we develop an approach that explicitly maintains fidelity to the data distribution while incorporating sophisticated notions of plausibility. We introduce a new distance metric for counterfactual optimization that emerges naturally from our theoretical foundations. We demonstrate how this metric can encode various definitions of actionability and feasibility through targeted modifications of the counterfactual sampling distribution.

Through empirical evaluation against existing counterfactual generation methods, we establish the practical advantages of our approach. Our method consistently produces counterfactual explanations that exhibit stronger adherence to the ground truth data distribution. Furthermore, qualitative analysis on complex datasets, such as Fashion MNIST, reveals that our approach uniquely preserves semantic coherence in generated counterfactual images, avoiding the artificial or implausible modifications common to existing methods.

In Section 2.6.3:, we further explore whether users differentiate counterfactual explainability methods and have preferences based on types of explanations. In order to evaluate conditions of usability for our approach, we benchmarked our approach against several existing counterfactual generation methods and conducted an AMT survey in which respondents perform a binary forced-choice task expressing their preferences among explanation methods. We found no universal preference for one explanation approach regardless of the extent to which they encode plausibility or actionability. While participants understand the relationships among features, they seem to rely on a subjective notion of cost for certain modifications. As Barocas et al. [11] and Selbst et al. [112] highlight, explanations are often rational only in the context of ensuring a desired outcome from a model, but not with respect to the goals that individuals have for themselves. This is consistent with our observations.

In the next chapter, we introduce the prompt inversion settings. We focus on formally defining the prompt inversion setting, while beginning to develop several ideas related to how we discuss counterfactual explanations in the generative model setting.

Chapter 3

Prompt Recovery for Image Generation Models: A Comparative Study of Discrete Optimizers

The previous chapter introduced necessary background for counterfactual explainability, highlighting the mathematical foundations of counterfactual explanations and the inherent assumptions within. Here, we turn to a discussion on counterfactual explanations in the generative model setting, focusing on how we can solve for them through discrete optimization techniques.

Most generative models today transform some written text into desired outputs – whether through chatbots that hold conversations based on written queries, or image, video, and music generation models that product content based on textual prompts. For instance when a user prompts a generative image model with "an image of a happy dog", they would expect to get back an image of a dog playing or smiling at the camera.

Figure 3.1, shows 3 images generated with *Gemini 2.5 Flash* of happy dogs (See the caption in Figure 3.1 for the exact prompts and responses). Each image aligns with the requested prompts, however these results beg the question: why are these all golden retrievers? The most common breeds of dogs among American dog owners are (in order of popularity): French Bulldogs, Labrador Retrievers, Golden Retrievers, German Shepherds, and Poodles [38]. Why is the third most commonly owned dog the archetypal "happy dog" in this model's representation? Moreover, what aspects of the prompt would need to change in order to generate different breeds?

While these questions will be addressed more in-depth in later chapters, we focus on introducing the methods by which we will answer these questions here through discrete prompt optimization.

Discrete optimization over natural language has several applications including jailbreaking LLMs [6, 157, 159] and measuring memorization [53, 111]. Whereas prompt optimization strategies in the text generation space have specific goals, such as generating targeted strings, the image generation space has struggled with tractable options for aligning prompts and generated images.

In generative image settings, CLIP serves as a proxy model for the full generative process because it enables direct text-image comparison and provides convex scores that can guide optimization. The practical challenges of directly optimizing prompts by backpropagating the diffusion process are still being addressed by researchers. Mahajan et al. [72] have attempted to

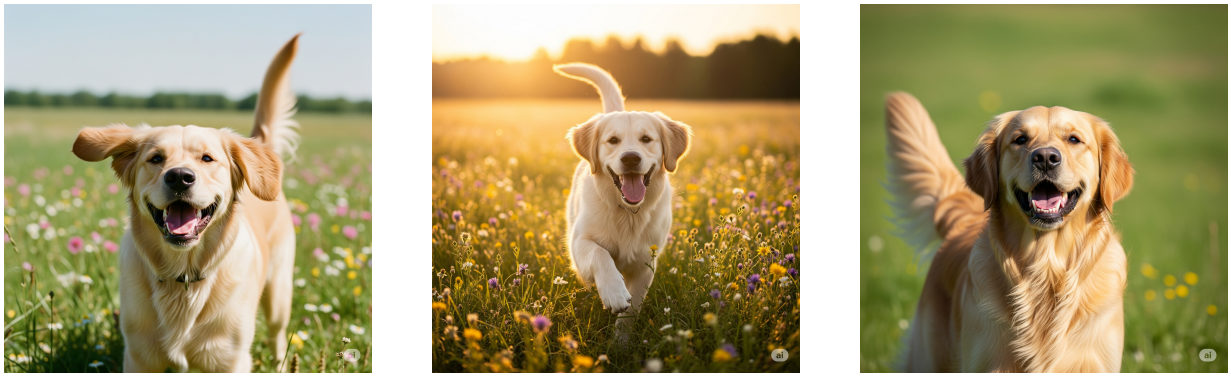


Figure 3.1: Images of three dogs generated by Gemini 2.5 Flash using the following conversation:

User: "can you generate me a picture of a happy dog?"
Assistant: "Sure, here is an image of a happy dog:"
User: "can you generate another one?"
Assistant: "Sure, here is another image of a happy dog:"
User: "one more?"
Assistant: "Here's one more happy dog for you:"

address this burden by focusing on the similarity between predicted noise residuals at specific diffusion timesteps, rather than generating full images. However, in alignment with prior work on noise inversion [78, 118] the authors find that prompts only have strong influence on the generated image during a narrow range of timesteps. At early timesteps, the image becomes largely "locked in," so even substantial changes to the prompt have little effect. At later timesteps, the stochasticity of the diffusion process leads to large variations in the final image, even when the correct prompt is used. This unpredictability makes it difficult to rely on noise residual comparisons for consistent prompt inversion.

Yet, direct discrete optimization is not the only method for finding viable prompts. Several approaches focus on using black box models to sample prompts. Both Zhang et al. [148] and He et al. [40] use pretrained language models to extract prompts for given a output across text generation and image generation tasks respectively. Moreover, as we show in this work, even a simple captioner that has not been finetuned for prompt generation often outperforms discrete optimization methods. In fact, Reade et al. [99] have found that a captioner fine-tuned on pairs of prompts and the images that they generate can effectively sample prompts that are exceptionally similar to the ground truth.

Despite this performance, we focus on the discrete case as direct discrete optimization can be beneficial for providing a human-readable understanding of the behavior of the image generation models. Just as prior work on counterfactual explanations directly optimizes inputs for desired outputs, we can use discrete prompts to probe the implicit representational decision boundaries of generative models. While generative models are not classifiers with explicit decisions and accuracy metrics, they are constantly making decisions on their representations based on the prompts. From background color to subject ethnicity, discrete optimization methods may provide a useful understanding of the relationship between prompts and images [140].

We emphasize solidifying ways of comparing discrete optimizers for image generation tasks.

Even with the rise of novel discrete optimization methods, standard prompt recovery comparisons over images are missing. In this work , we provide a holistic benchmark on not only the similarity between prompt and image, but also the similarity among images generated by the inverted prompts which to the best of our knowledge has not been standardized in this setting.

3.1 Selected Algorithms

To best situate the optimizers we study, it is critical to pose the prompt inversion problem formally. Consider a tokenizer that maps from natural language to sequences of integer valued tokens corresponding to a list of indices of tokens that the tokenizer can express $\mathbb{T} \in \{0, ..., N\}$. Let $x \in \mathbb{T}^s$ be a length s sequence of tokens and $\mathbf{E} \in \mathbb{R}^{N \times d}$ be a matrix whose rows are d-dimensional embedding vectors, one for each token in the vocabulary. To embed a sequence x, we can define $\mathbf{X} \in \{0, 1\}^{s \times N}$ s.t. $\sum_{i=1}^{T} \mathbf{X}_{j,i} = 1 \; \forall j \in \{1, ..., M\}$ to be a matrix of one-hot encoded rows for the integers in the sequence x. The product $\mathbf{XE}$ defines an $s \times d$ embedding.

Given a stochastic generative model, $\mathcal{M} : \mathbb{R}^{s \times d} \rightarrow \mathcal{Y}$, prompt inversion techniques seek to find the sequence of tokens x, or equivalently their corresponding one-hot encodings $\mathbf{X}$, that solve $\mathcal{M}^{-1}(Y); Y \in \mathcal{Y}$. Typically we express the solution as the minimizer of some loss function $\mathcal{L}$, or the solution to the following optimization problem.

$$\underset{\mathbf{X} \in \{0,1\}^{s \times N}}{\operatorname{argmin}} \quad \mathcal{L}(\mathcal{M}(\mathbf{XE}), Y) \quad \text{s.t.} \quad \sum_{i=1}^{N} X_{j,i} = \mathbf{1} \; \forall \; j \in \{1, ..., s\} \tag{3.1}$$

For the considered language models we embed text using the above process and pass the embeddings $\mathbf{XE}$ to the model $\mathcal{M}$. As $\mathcal{M}$ has an $R^{s \times d}$ input space, we can in theory give it any input from this space, even if that input is not part of the embedding matrix E and receive a valid output. For carefully tuned strategies, allowing continuous embeddings may be able to recreate desired images significantly faster than recovering the original prompts.

Yet, as discussed in Chapter 1, while there has been more success for solving this problem over soft embeddings in the continuous rather than the discrete space, Khashabi et al. [55] show that embeddings in $\mathbb{R}^d$ outside of the discrete set of the rows of $\mathbf{E}$ have little use for the recovery of discrete tokens. Formally, consider a classifier that assigns a class $c \in C$ to an image, $\mathcal{I}$. Embeddings within the row space of $\mathcal{V}$, where their nearest neighbor in $\mathcal{V}$, ϵ_i, generates an image with class c_i, can generate an image of any class, $c \in C$.

We argue that prompt inversion methods for the purpose of interpretability and/or explainability should focus on strategies for discrete optimization within the embedding table $\mathbf{E}$ as a consequence of this behavior. The goal is not simply recovery of an image, but an understanding of *why* this image was generated in the first place.

While the gradient of $\mathcal{M}$ with respect to an embedding x exists for all $x \in \mathbb{R}^{s \times d}$, continuous descent-based methods risk finding minima outside of $\mathbb{T}^s$, leaving us without hard tokens. Moreover, computing the gradient through the full generation model $\mathcal{M}$ may be too expensive (for example when $\mathcal{M}$ is a diffusion model forward passes may take multiple seconds), prior work often uses CLIP [95] to encode images and text in a shared latent space. Some of the methods we examine operate wholly within CLIP's latent space to compute the loss between the prompt and

the target image. These methods approximate Equation (3.1) by solving the following problem where $\mathcal{L}_{\text{CLIP}}$ is a similarity loss defined over CLIP embeddings.

$$\underset{\mathbf{X}\in\{0,1\}^{s\times N}}{\text{argmin}} \quad \mathcal{L}_{\text{CLIP}}(\mathbf{XE}, Y) \quad \text{s.t.} \quad \sum_{i=1}^{N} X_{j,i} = \mathbf{1}\ \forall\ j \in \{1, ..., s\} \tag{3.2}$$

Below, we discuss several algorithms for solving the problem and discuss a variety of strategies for approaching Eq. (3.2). In section 3.2 we go more into detail on the performance and convergence of each algorithm.

3.1.1 PEZ

The first approach we consider is PEZ [137], a version of projected gradient descent where descent steps are made in the continuous embedding space. The gradients of the objective in Equation (3.2) are evaluated at points in embedding space corresponding to real tokens, but the trajectory of the iterates may deviate from the discrete token set.

Let $\texttt{Proj}_{\mathbf{E}}(\cdot)$ be an operator that projects vectors (or matrices row-wise) from $\mathbb{R}^d$ to their nearest row-vector of $\mathbf{E}$, and let $\xi_i \in \mathbb{R}^{s\times d}$ be a soft prompt. As an iterative gradient-based optimizer, PEZ produces a sequence of iterates $[\xi_0, \xi_1, ..., \xi_n]$ as it solves the minimization problem in Equation (3.2). To update from ξ_i to ξ_{i+1}, PEZ computes the gradient of the loss at the hard prompt $\texttt{Proj}_{\mathbf{E}}(\xi_i)$ and takes a step in the direction of this gradient from the soft prompt ξ_i and then calls $\texttt{Proj}_{\mathbf{E}}(\xi_i)$ to project back to the space of hard prompts.

Formally, given some point, ϵ within the continuous space, $\mathbb{R}^d$, PEZ only computes gradients at the nearest neighbor of ϵ within $\mathcal{V}$,

$$\nabla_{\epsilon}\mathcal{L}(\mathcal{M}(\epsilon), \mathcal{I}) = \nabla_{\texttt{Proj}_{\mathcal{V}}(\epsilon)}\mathcal{L}(\mathcal{M}(\texttt{Proj}_{\mathcal{V}}(\epsilon)), \mathcal{I}), \tag{3.3}$$

.

Thus, PEZ gives a fast, lightweight method of discrete optimization while still using gradient-based descent to approximately solve the problem in Equation (3.1). For more information, see Algorithm 1 as described by Wen et al. [137]. For a single image, we run PEZ over the CLIP loss for 3000 steps and return the prompt the maximizes the CLIP similarity between the image embedding and the text embedding of the prompt.

3.1.2 Greedy Coordinate Gradients

Greedy Coordinate Gradients (GCG) [159] is an alternative method for optimizing over the discrete vocabulary using the gradients of the objective with respect to the matrix $\mathbf{X}$ in Equation (3.2). In particular, we compute the gradient of the loss with respect to $\mathbf{X}$, which is a matrix of the same shape that approximately ranks token swaps. As each entry in a given row of $\mathbf{X}$ corresponds to a token in the vocabulary, each row i in its gradient relays to us how influential changing the token x_i to each other token in the vocabulary might be in lowering the loss. We compute $\nabla_{\mathbf{X}}\mathcal{L}_{\text{CLIP}}(\mathcal{M}(\mathbf{XE}), Y)$, then, just as gradient descent methods takes steps in the opposite direction of the gradient, we select a random batch of candidate swaps from the top k largest entries of the *negative* gradient.

A given swap corresponds to a single token change in x and we directly compute the loss for each of these candidates and greedily accept the best one as our new iterate. As done with PEZ, we run GCG over the CLIP loss for 3000 steps, returning the best prompt as determined by CLIP similarity between the image embedding and the prompt's embedding.

3.1.3 AutoDAN

AutoDAN [157] was proposed as a method of finding human-readable adversarial attacks on aligned language models. The optimizer solves Eq. (3.1) by iteratively optimizing a single token appended to the current prompt. Given an initial prefix, e.g., "Image of a", the algorithm searches for the token that follows 'a' that minimizes the objective function. The optimizer incorporates a 'readability' objective based on the log probability of the next token given an underlying language model. Similarly to GCG, AutoDAN employs a coarse-to-fine search strategy by appending an initial token, $\hat{x}$ to the current iterate x, and scores each token in the vocabulary according to the following scoring function:

$$\text{score}(x_i) = -\left(\nabla_{\hat{x}}\mathcal{L}([x, \hat{x}]E)\right) + \log(p(x_i|x)) \tag{3.4}$$

The algorithm selects the top k scoring tokens and performs a fine-grained search by computing the exact loss over each, taking the token that minimizes the loss, $\mathcal{L}$. This minimizing token is then appended to x, giving $x_{t+1} = \begin{bmatrix} x_t & x_i^* \end{bmatrix}$.

AutoDAN was originally designed for text-to-text language models, where the log probability, $\log(p(x_i|x))$ was directly available. However, in this review, we use CLIP to determine the quality of the prompt, which does not inherently compute the log probability. We thus use FUSE [141], a recently proposed approach for solving multi-objective problems across models and tokenizers. FUSE approximates the jacobian of a mapping between the two models and uses the embeddings of a text-to-text language model, such as GPT2 to compute both the log probability, $\log(p(x_i|x))$, and the gradient, $\nabla_{x_{GPT}}\mathcal{L}_{CLIP}(f(x_{GPT}))$, where f maps from GPT's embeddings to CLIP's embeddings. While we introduce FUSE here, this algorithm is the focus of Chapter 4 and will the exact mathematical form and intuition will be detailed there. Here, FUSE allows us to apply a language prior when optimizing a prompt with CLIP. We additionally explore the scenario in which we do not use a language prior, by reverting to the standard case in which we fix $p(x_i|x) = \frac{1}{N}$. In our experiments, we run AutoDAN for 16 steps, which enforces a a maximum token length of 16 due to one-by-one generation of new tokens. We also utilize a beam search with a beam width of 5.

3.1.4 Random Search

Andriushchenko [6] suggests that such sophisticated strategies may not be critical for prompt optimization—given enough time, random searches can perform adequately in a variety of settings. Thus, we explore a variant of random search [97]. While random search traditionally selects random candidates from within a ball around the the current iterate, this approach does not directly map to hard prompting. True random samples around these high-dimensional embedding spaces are sampled from a ball of with negligible volume around the initial embedding; a nearest neighbor projection would often fail to return a new candidate.

In order to address this limitation, we randomly sample from new tokens from the l_0 ball around each element in the sequence $\mathbf{XE}$. At every iteration, we select a batch of candidatesand greedily accept the best single-token replacement as the next iterate. We compare the prompt found by Random Searching over the same number of steps as done for PEZ and GCG, determining the best prompt by CLIP similarity in the same way.

3.1.5 PRISM

PRISM, proposed by He et al. [40], highlights that text-to-image generation is not a one-to-one mapping – multiple prompts can describe the same image, and many images can correspond to the same prompt. Rather than relying on discrete token space optimization, PRISM optimizes over a distribution of prompts. Inspired by LLM jailbreaking methods [eg. 17], PRISM leverages in-context learning in vision-language models (VLMs) to iteratively refine the prompt distribution. This process incorporates the history of reference images, generated prompts, output images from an anchor text-to-image model, and evaluations from a VLM judge, using techniques similar to chain-of-thought [136] and textual gradient [93]. After K iterations across N parallel streams, the best-performing prompt is selected using the same VLM judge. In our experiments, we use GPT-4-o-mini as the VLM and Stable Diffusion XL-Turbo [105] as the anchor text-to-image model, following He et al. [40]'s setup with $N = 6$ and $K = 5$. To ensure fair comparisons, we limit the generated prompts to 20 tokens.

3.1.6 Captioning

Lastly, we use automated image captions as a proxy for the inverted prompts. Given that a prompt for an image generation model likely encodes information about the setting of the desired image, its subject, its quality, and other properties, we assume that captioning an image provides a human-readable token sequence with some or all of these same properties necessary to generate the image. Moreover, as captioners are typically autoregressive, they have the potential to return an approximate inversion much faster than other methods.

Here, we focus on a single model, BLIP-2 [65]. This model is a generic and compute-efficient vision-langauge pre-training (VLP) method. VLP techniques aim to learn multimodal foundation models on a variety of vision-language tasks. BLIP-2 leverages a trainable module, the Q-former, in order to bridge the gap between a frozen image encoder and a frozen LLM, facilitating image-text matching tasks, image-grounded text generation tasks, and image-text contrastive learning tasks. We prioritize BLIP-2's image-grounded text generation as the frozen CLIP-style encoder aligns well with the above prompt inversion methods, all of which use frozen CLIP encoders.

3.2 Evaluation

For each optimizer detailed above, we assess their performance across several criteria. Considering the stochastic nature of image generation, we measure the effectiveness of an inverted prompt by asking the following questions.

1. How similar (FID [43], KID [14]) are images generated with the inverted prompt to images generated by the original prompt?
2. How well (CLIP [42]) do the inverted prompt and original image align?
3. How well (Text Embedding Similarity [99]) does the semantic content of the inverted prompt align with the semantic content of the original prompt?

We address the stochasticity inherent to the image generation process by averaging the performance of each method across several images generated by the original prompt and the inverted prompts. We randomly sample 100 prompts from an existing dataset of prompts[1] used by Stable Diffusion [102].[2] Given each of these prompts, we generate 10 baseline images for each baseline prompt, and invert each according to all of the seven methods considered here. Once we have found an inverted prompt, we generate 2 images from these prompts, and compute our metrics across the 10 baseline prompts and images and the 20 images based on the 10 inverted prompts. In addition, we choose 75 log scaled time points within the 3000 optimization steps used for PEZ, GCG, and Random Search and repeat our full analysis on a subset of DiffusionDB prompts in order to better understand the convergence of each method.

3.3 Empirical Results

In this section we present quantitative and qualitative results comparing each method. Across several metrics, we see the quantitative rankings are consistent, but we find upon qualitative examination that these numeric rankings show only a partial picture. Examining the images and the recovered prompts themselves we see trade offs across methods.

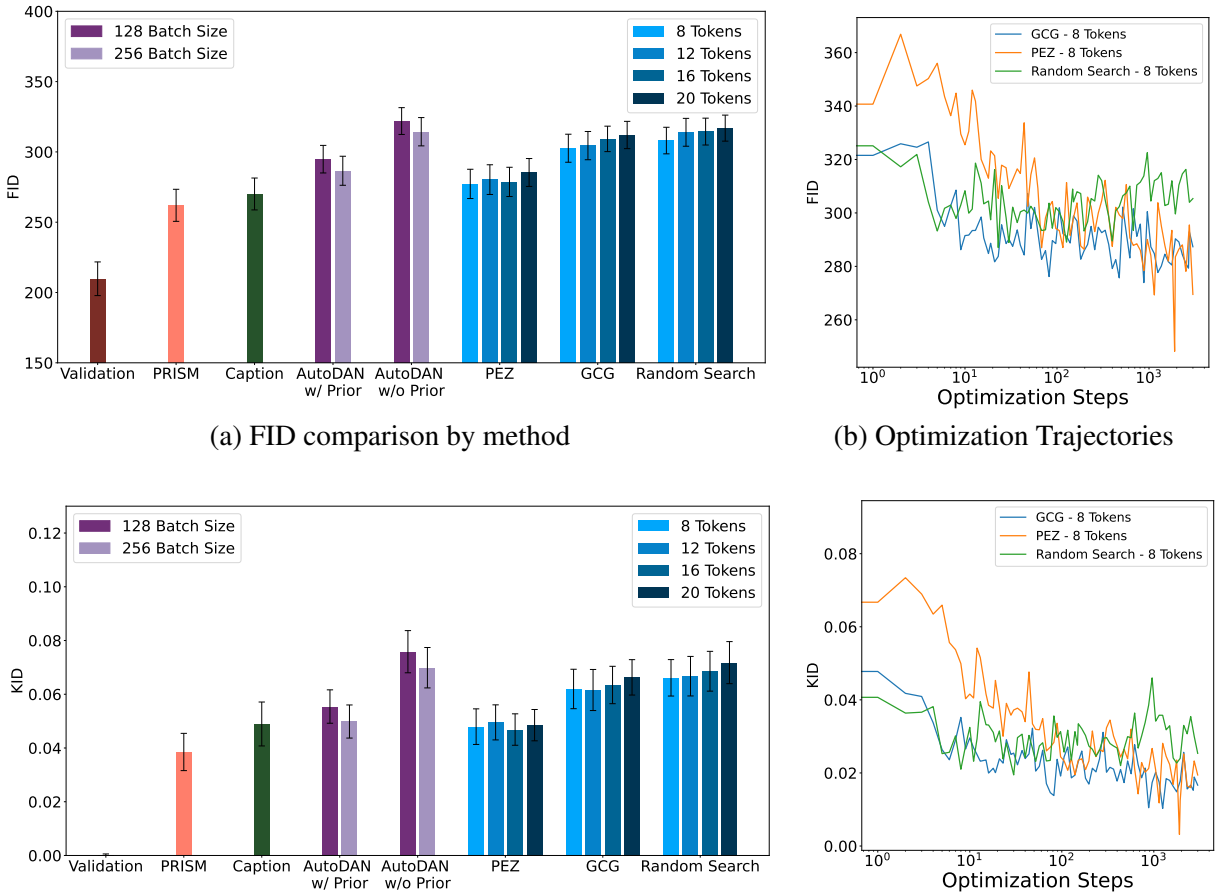
3.3.1 Quantitatively Ranking Methods

Image to Image Comparisons For image to image comparisons (Figure 3.2), we analyze images generated by the best early-stopped prompt for each method and the convergence rates across our considered image similarity metrics for each algorithm. Our validation set, which consists of the ground truth prompts has an FID of 209.78 and a KID of -0.001. The KID score in particular tells us that the closer any algorithm gets to a KID of 0, the more similar that prompt will be to the ground truth, whereas, while the ordering may be consistent with FID scores, it is possible that using FID rather than KID may incorrectly show that a method improves over the validation set.

We find that generating images from PRISM prompts provide the most similar images to those generated by the original prompt, with those images generated by BLIP-2 and PEZ as close seconds; PRISM gives average FID and KID values of 262.015 and 0.0385 respectively, while the captioner generates images with average FID and KID values of 270.085 and 0.0489. PEZ follows these with average FID and KID values of and 280.392 and 0.0482. In addition, we see a significant gap in performance between AutoDAN with a prior and AutoDAN without a

[1] Poloclub DiffusionDB dataset of prompt-image pairs [135]

[2] StableDiffusion 2-1: `stabilityai/stable-diffusion`.



(a) FID comparison by method

(b) Optimization Trajectories

(c) KID comparison by method

(d) Optimization Trajectories

Figure 3.2: Comparison between images generated by inverted prompts and images generated by the original prompts.

prior, where the former performs much more similarly to the captioners and the latter performing in line with GCG and a Random Search.

Analyzing the objective trajectory over the course of optimization reveals interesting trade-offs. We used a small validation set to determine the number of steps for all algorithms to converge for the given prompts and images used in this study. We determined that all optimizers stop receiving meaningful improvements after 3000 steps. We observe that GCG and a Random Search find a prompt comparable to their best early-stopped prompt within the first 25 steps and then struggle to descend further, analagous to applying too high of a learning rate to optimization problems. On the other hand, PEZ has a slower convergence, but it descends consistently across all steps until it finds prompts that improve over both the GCG and Random Search prompts. Moreover, as PEZ uses a single forward and backward pass, it requires much less time to run than the comparison methods. In other words, PEZ finds prompts that generate images more similar to the ground truth in much less time than all other optimizers considered here, except for the BLIP-2 Captioner.

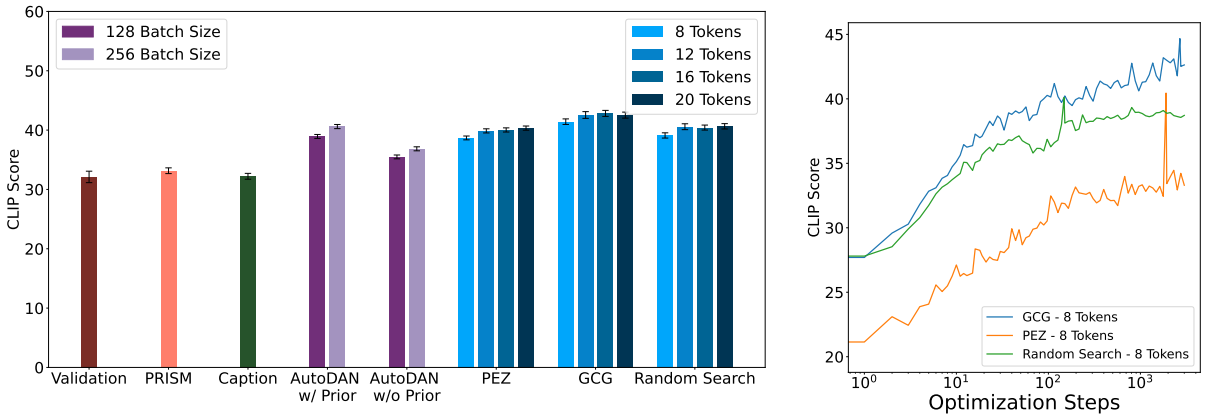


Figure 3.3: CLIP Similarity between the inverted prompt and images generated by the original prompt. This CLIP Similarity is the objective that each optimizer is maximizing.

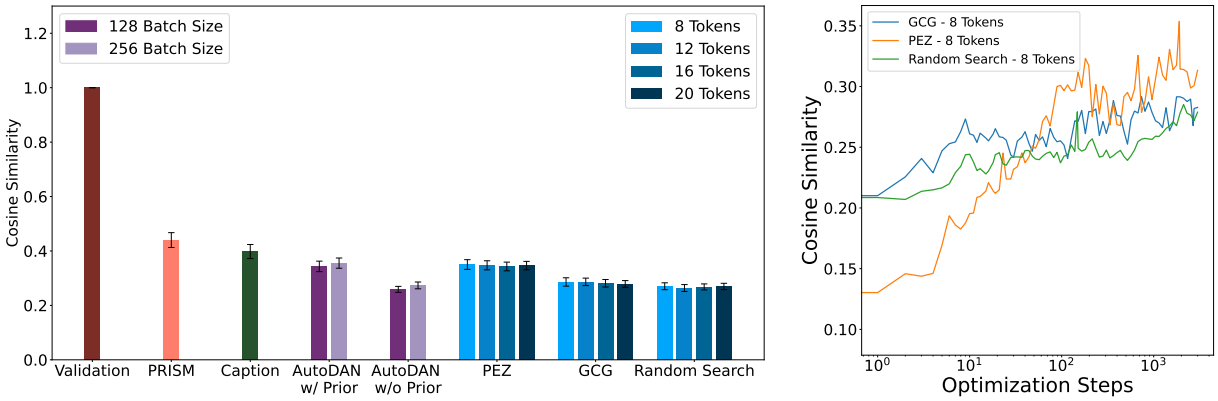
Text to Image Comparisons When we focus on the alignment between the text and images we see an interesting trend emerge. We first compare the CLIP similarity between the inverted prompts and the original image (in the top of Figure 3.3). Note that this is the optimization objective used across all optimizers.

We find that all optimizers do a good job maximizing their objective. While AutoDAN without a language prior performs the worst over all optimizers, it still does a better job of maximizing the CLIP similarity over the validation set, PRISM, and the BLIP2 Captioner. Optimizing the objective with GCG and AutoDAN with a language prior performs the best over the discrete optimizers, with PEZ coming a close third. The contrast between the performance of each optimizer on their objective and their relative lack of performance across the image-to-image and text-to-text metrics suggests that the CLIP objective is acting as a poor proxy for finding prompts for generative image models. While there may be room for improvement over the CLIP objective for this task, this comparison allows us to take a better look at the convergence rates of all optimizers on their objective. Just as in the image-to-image comparison, GCG and Random Search quickly find a good prompt (within 20 steps) and then very slowly improve from there.

Yet PEZ follows a much more gradual curve, with sharp peaks when new optima are found. As these are log scaled in their x-axes, we do not see all peaks except for the early stopped result. The average prompt found with PEZ is much lower than the comparison methods, but the peaks are in line with the other methods. Additionally, GCG and Random Search again very quickly within the first 20 steps and then very slowly update from there. This overreliance on early-stopping may be a weakness for PEZ. Rather than oscillating tightly around the optima, PEZ oscillates wildly around high quality prompts. In essence, PEZ may better explore the prompt space, while methods incorporating fine-grained search (such as GCG) are more adept at exploiting it.

Text to Text Comparisons Lastly, we compare the similarity in the text of the found prompts to the ground truth prompts. Figure 3.4 shows the cosine similarity between the text embeddings[3]

[3]Embeddings were computed using **sentence-transformers/all-MiniLM-L6-v2** to be in line with [99]



(a) Original images v. recovered images

(b) Change During Optimization

Figure 3.4: Cosine Similarity between text embeddings for the original and inverted prompts. Based on the metric used by [99]

of the found prompts and the ground truth prompts.

Just as in the image-to-image case, we find that using responses from PRISM as the inverted prompt outperforms all of our comparisons, with a cosine similarity of 0.440 to the original prompt; the BLIP-2 captioner comes in second with a cosine similarity of 0.397 to the original prompt. AutoDAN with a language prior and PEZ follow behind with respective similarities of 0.355 and 0.346 averaged across all lengths. GCG, Random Search, and AutoDAN without a language prior remain clustered together in terms of their performance. Moreover, when looking at their convergence rate, we see the same story as above. GCG and Random Search very quickly ascend, and while PEZ ascends more slowly it eventually exceeds GCG and Random Search in their performance within the first 100 optimizations steps of its allowed 3000 steps.

3.3.2 Qualitatively Assessing Inverted Prompts

In the quantitative evaluation above, we show that PRISM and the captioner return prompts that may be better across several metrics compared to searching for a prompt via discrete optimization. Here, we show a qualitative example (Table 3.1) of an image generated by one of the ground truth prompts and the different results that each method find. Other than AutoDAN with the language prior applied, no discrete optimizer produces human-readable prompts despite the quantitative similarities in their performance. We thus separate this subsection into natural language and keyword-based prompts that without a language prior.

PRISM provides prompts that are exceptionally more detailed than the comparison methods; opting for short descriptive clauses rather than the full sentences that BLIP-2 uses. As described above, when the length of a prompt is limited, the additional stop words required by full-sentence prompts reduce the number of concepts that can be included in a prompt, significantly affecting the final image. On the other hand, AutoDAN's language prior seems to finds natural language prompts that evoke the imagery described by the image, "character from Magic Treefolk..." does not describe anything from the image (to the best of our knowledge there is not media called Magic Treefolk), but if such media existed then we would not be surprised to find that something

Original Prompt	a friendly goblin with a big ((human)) nose and wide eyes, dark hairs, big ears, covered in branches and moss, portrait by daniel docciu and dave dorman and jeff easley
PRISM	Friendly green goblin face, smiling, tangled branches, soft forest background
BLIP-2	green troll in a tree with leaves and branches on it's head and a smiley face on it's face
AutoDAN w/ Prior	character from Magic Treefolk depicting Green Elf head with smile during 2015 promotional image walkthrough art group image
AutoDAN w/o Prior	animated jester troll grass wordpress goblin frightening branch artwork today)) oman ly 6 jester head. _ reid /
PEZ	newmlb mtgnflrevealed loki revealed reveal).. goofy smiling scary creepy ytree ̈! orc =)) arbormates
GCG	typically greener donny recent reported atrist..... frightening cohen substantially _: . kal ears googmirrowickedcriticalrole trees believes
Random Search	cantenzenegger oaks]m̈odo grin grassy goblin ytless...(̃! ¿·¡ sends iconforeveryimp huskdns

Table 3.1: Example images and corresponding 20-token prompts. Each image is generated by the *original prompt* and we show examples of the inversion result from each method. Other AutoDAN with the language prior applied, no discrete optimizer produces more human-readable prompts than another despite the quantitative differences their performance.

called Magic Treefolk included depictions of goblins or other forest critters.

When comparing each recovered prompt to the original prompt, there is often a significant amount of information lost during the generation process that is unrecoverable. Both Random Search and PEZ capture basic information such as "trees" or "green". These methods try to included the single tokens that encode as much information as possible. Similarly to the "Magic Treefolk" example above, GCG uses the token "criticalrole" for a similar purpose. Critical Role[76], a 'Dungeons & Dragons'-based web series, embeds a relationship between the prompt and creatures found in Dungeons & Dragons through a single token. Moreover, without the need for a language prior, it does not need to waste tokens fitting 'criticalrole' into a coherent sentence. Yet, it may cause an overreliance on these 'keyword' tokens and allow unrelated tokens such as 'goog' to be included in a prospective prompt. This comparison may shed light on why PEZ outperforms GCG and random search, as PEZ appears to stay more on topic. PEZ includes the tokens "loki", "tree", "arbor', "scary", "goofy", "orc" and "smiling", while GCG and Random Search do not provide significantly more specificity than "green", "trees", and "criticalrole"; and "oaks", 'goblin" and "grassy" respectively. At its core, PEZ is a projected gradient descent method, using common optimizers, such as SGD or Adam with a weight decay. This approach encourages some form of regularization in its optimization, that discourages the one-and-done

approach that GCG and Random Search seem to use, where they discourage repeating the same general concepts or tokens in a prompt.

3.4 Discussion

Our results have important implications for the practical applications, limitations, and future directions of prompt inversion methods. This section examines these findings and their broader significance for the field.

For practitioners seeking effective prompts from images, our work demonstrates that image captioning tools represent the most promising approach. They are fast, as the heavy lifting is done ahead of time in training these models rather than optimizing anything per image in deployment. They also best capture natural sounding language, a goal that discrete optimizers might better incorporate as these tools mature.

However, our results are limited by the fact that the diffusion and image-text embedding space is so heavily driven by only a few models. As the ecosystem of state-of-the-art text-prompted image generation models expands and diversifies, the trends we observe may not generalize. Furthermore, minor variations in the optimization strategies could have large impacts on these results. As with any empirical benchmarking study, our findings should be interpreted within the context of the current technological landscape and may require revision as the field evolves.

3.4.1 Open Questions and Future Directions

Our analysis reveals several intriguing phenomena that that warrant further investigation. Most notable, Zou et al. [159] report that GCG is effective at jailbreaking LLMs and PEZ is not. This stands in stark contrast to these two methods relative performance at prompt inversion. This raises the funamental questions about why optimization over natural language exhibits such different characteristics across these domains. While it is possible that the loss landscape is simply too dissimilar between text and image generation for PEZ's performance to directly compare to GCG across settings, we note that GCG still finds the lowest minima of the objective for all compared prompt inversion methods. The coarse to fine grained search performs well in both settings, however it is likely that there is much more to image generation than can be captured with the CLIP objective.

If one assumes that the original image and the solution to the objective sit somewhere near each other in space then instead of directly searching for the solution to Eq. (3.2), by exploring nearby we may find better solutions. As originally implemented in [137] and in this analysis, the optimizer that solves PEZ uses a strong amount of weight decay (0.1 decay) on the embeddings during the search. This strong regularization forces PEZ to oscillate over some typical space where the ground truth image sits, potentially converging to a suboptimal global minima.

Another puzzling observation concerns that similar performance of GCG and random search across the prompt space. Why does gradient information make so little difference? The intuition that the gradient signal is informative comes from observing the success of PEZ, so why is the combination of search and gradient-based optimization in GCG leave it so similar to random search alone?

A preliminary hypothesis emerges from examining the optimization trajectories: GCG and Random search are always aligned until approximately 100 steps in the optimization and then the random search slightly degrades, while GCG slightly improves. This may lead to a case where a strong algorithm is a combination of the two. The faster random search can be run for some fixed number of steps before transitioning to GCG.

Despite these advances, prompt inversion remains far from solved and represents an excellent testbed for novel discrete optimization approaches. The insights from this study directly inform our subsequent research directions. Given GCG's consistent performance on the target objective and its relevance to the optimization challenges discussed in later chapters, we adopt GCG as our primary optimizer for the remainder of this thesis.

In the following chapter, we examine FUSE—our method for incorporating language priors from AutoDAN— in greater detail and demonstrate how GCG can be applied to inversion problems in this context. This application effectively enables the construction of complex multi-objective optimization problems across diverse model architectures using various loss functions.

Chapter 4

FUSE-ing Language Models: Zero-Shot Adapter Discovery for Prompt Optimization Across Tokenizers

4.1 Introduction

As we've shown in Chapter 2, generating counterfactual examples from a given reference necessitates that the counterfactual itself be sampled from the same distribution as the reference. This connection allows us to to separate a counterfactual example from an adversarial example. When dealing with text, the sampling distribution is not well-defined, however, here we use some of the insights from Chapter 3 and consider the distribution of a prompt as the distribution of text that aligns with an image, while being human-readable, i.e., a human could have written this text during normal use. In this chapter, we show how to construct a differentiable optimizer that can both encode the image-text alignment and human readability using proxy models, even when those proxies have different tokenizers.

The primary challenge here stems from the myriad of individuals and organizations who train and fine-tune large language models (LLMs) for their own needs. So many different models and applications has led to a plethora of models with unique ways of processing, tokenizing, and embedding text, creating a challenges for knowledge transfer and interoperability across models. In turn, siloing the insights and capabilities of any single model. One popular way of enabling interoperability is through *prompting* strategies. These approaches leverage the ability for text to be passed across models, by converting tasks into formats that LLMs can solve. However, the uniqueness of different models' token and embedding spaces creates difficulties in automated methods for prompt discovery.

While prompting strategies have found success across a variety of tasks including adversarial text generation [159], text summarization [153], and prompt discovery for generative models [138], the non-differentiable nature of text remains a limitation. One way of addressing this challenge is by encouraging a standardized tokenization and embedding strategy, where every new model or architecture uses the same tokenizer and embedding space. Despite the potential for fostering cooperation across models, it is unlikely that model developers will converge on

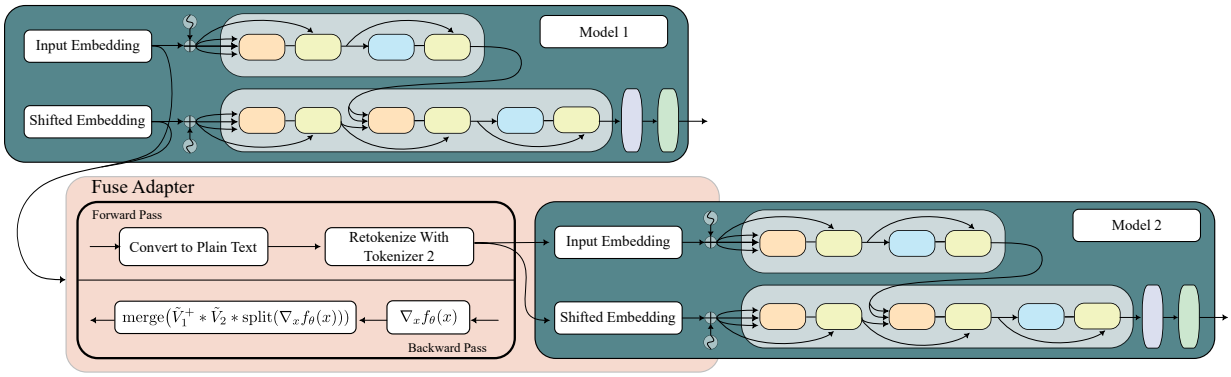


Figure 4.1: The FUSE adapter connecting two transformer models for parallel inference. Inputs from Model 1 flow through the adapter by converting to text, retokenizing with Model 2's tokenizer, and embedding into Model 2's input space. The backward pass receives the gradient from Model 2, and multiplies it by the precomputed $\tilde{V}_1^+ * \tilde{V}_2$.

a single tokenization. Yet, such a standardized representation may not be necessary if we can freely compute forward and backward passes across models, regardless of their tokenization.

Here, we propose one such method of computing gradients across different models' discrete embedding spaces, even if these spaces are defined in terms of different tokenizers. Our approach, which we call *FUSE* (Flexible Unification of Semantic Embeddings) inserts a simple module that approximates the functionality of an adapter layer that maps between the embeddings of multiple models without finetuning. We find that rather than focusing on individual tokens, if we instead focus on groups of tokens separated by whitespace, then we can track how a full word is represented in the embedding space and create a necessary equivalence among tokenizers that can be leveraged to map from one model to another. We then derive a strategy to compute one such differentiable map, and find that we can approximate the gradient of a language model's output with respect to another model's embedding space solely in terms of the first model's embedding and a precomputed tensor.

4.2 Preliminaries

4.2.1 Language Model Embeddings

Given a string, a tokenizer maps it to a set of tokens, $\mathbf{t} \in \{0, ..., |V|\}^s$, where s is the length of the tokenized string and $|V|$ is the number of unique tokens in the tokenizer. The model then applies a mapping $\mathcal{E} : \{0, ..., |V|\}^s \to \mathbb{R}^{s \times d}$ which indexes these tokens across a discrete set, mapping each to a unique d-dimensional embedding, $E \in \mathbb{R}^{s \times d}$.

Alternatively, we can represent the embedding function ($\mathcal{E}$) itself as a matrix, $V \in \mathbb{R}^{|V| x d}$, where each row corresponds to the embedding vector for a specific token in the vocabulary. By representing the tokens as one-hot encodings over the vocabulary, $X \in \{0, 1\}^{s \times |V|}$, we can express the embedding vectors with a lookup operation $E = XV$. In this framing, V is both a matrix and denotes the set of discrete embedding vectors for a model.

With this set of preliminary information in hand, we proceed to outline our approach, starting from the simple case in which models share a tokenizer, but have different embeddings (i.e., strings will always be tokenized to the same $\mathbf{t}$, but the embedding mapping, $\mathcal{E}(\mathbf{t})$ differs between models. We then build on this case and extend it to the case in which models tokenize words differently and have different embedding mappings (i.e., words may be separated arbitrarily, the model vocabularies have different lengths, and each embedding may have a different dimensionality across models).

For the latter case, understanding how to multiply tensors is crucial for our approach (for a full primer, see Appendix B). When working with tensors of order greater than 2, their multiplication has been well-defined in terms of the t-product operator, $*$ [56]. The t-product defines an associative and left/right distributive multiplication operation of $\tilde{A} \in \mathbb{R}^{m \times k \times p_1 \times \cdots \times p_n}$ and $\tilde{B} \in \mathbb{R}^{k \times n \times p_1 \times \cdots \times p_n}$, where $\tilde{A} * \tilde{B} \in \mathbb{R}^{m \times n \times p_1 \times \cdots p_n}$. We also make use of the folding and unfolding operation introduced alongside the t-product that reshapes an $\mathbb{R}^{d_1 \times d_2 \times \cdots \times d_n}$ tensor into a partitioned tensor in $\mathbb{R}^{d_1 d_n \times \cdots \times d_{n-1}}$ tensor and back,

$$\text{unfold}(\tilde{X}) = \begin{bmatrix} \tilde{X}_1 & \tilde{X}_2 & \cdots & \tilde{X}_n \end{bmatrix}^T \qquad \text{fold}(\text{unfold}(\tilde{X})) = \tilde{X}.$$

Note that Kilmer and Martin [56] require, $\tilde{A}$ and $\tilde{B}$ to have their first two dimensions of the appropriate shape for matrix multiplication and each of the remaining dimensions must be the same size, however this product can also be generalized to arbitrary tensor sizes as long as the first two dimensions are appropriate sizes for matrix multiplication.

The key idea in our work is that while current tokenizers may split the same word arbitrarily, they always respect white-space separation. We can build shared representations across embedding spaces by focusing on groups of white-space separated tokens and their embeddings, represented as third order tensors, rather than individual tokens and embeddings represented by matrices. In doing so, we find that we can approximate the gradient of a language model's output with respect to another model's embedding space solely in terms of the first model's embedding of a string and a precomputed tensor.

4.3 Methodology

4.3.1 Shared Tokenizers

Recall that the embedding of a set of tokens for model i, can be represented as, $E_i = XV_i$, where X is a one-hot encoding across the vocabulary, V_i [1]. Our goal is to solve a multi-objective optimization over K models, in which each model is solving a different task whose loss is computed with a differentiable $\mathcal{L}_i(E_i)$.

$$\arg\min_X \sum_{i=1}^{K} \mathcal{L}_i(XV_i). \tag{4.1}$$

As each model uses the same tokenizer, X is shared for each model. This problem can clearly be solved via any off-the-shelf optimizer. However, consider the pedagogical case in which we

[1]Note that V_i uses the subscript i to denote the vocabulary of a particular model, not the token index within a model's vocabulary.

want to directly optimize the embedding vectors, E_i, instead of the one-hot encodings. In other words:

$$\arg\min_{E_i \in V_i} \sum_{i=1}^{K} \mathcal{L}_i(E_i). \tag{4.2}$$

Solving equation (4.2) becomes less clear. One approach is to choose one model to be the *primary model*, and use its embeddings as input to all other models by introducing an adapter $\mathcal{T}_{i:j} : V_i \to V_j$ that maps from model i's vocabulary to model j's vocabulary. With the introduction of $\mathcal{T}_{i:j}$, we only optimize in the primary model's embedding space and our objective becomes,

$$\arg\min_{E_i} \mathcal{L}_i(E_i) + \sum_{j \neq i} \mathcal{L}_j(\mathcal{T}_{i:j}(E_i)). \tag{4.3}$$

With a differentiable representation of $T_{i:j}$, then this equation can be solved via gradient-based optimization. However, as the vocabulary matrices are not square, they are not invertible; we cannot directly map from the embedding space to back to token space. We instead approximate a linear map for $\mathcal{T}_{i:j}$ using the Moore-Penrose inverse (pseudoinverse) of the model's vocabulary, $V_i^+ = V_i^T(V_i V_i^T)^{-1}$. By using the pseudoinverse, $E_i V_i^+ \approx X$, we can substitute $E_i V_i^+$ for every instance of X in Equation (4.1) and set $\mathcal{T}_{i:j}(E_i) = E_j \approx E_i V_i^+ V_j$. The gradient of Equation (4.3), is then a simple application of the chain rule,

$$\nabla_{E_i} \mathcal{L}_j(\mathcal{T}_{i:j}(E_i)) \approx \Big(\nabla_{E_j} \mathcal{L}_j(E_j)\Big) V_i^+ V_j. \tag{4.4}$$

Pay particular attention to the fact that the approximate gradient is no longer dependent on the embedding of the model that we want to map *from*, only on the embedding that we want to map *to*. We can thus map E_i to E_j in a non-differentiable way (e.g., convert back to text and retokenize), compute the gradient of the loss for model j, with respect to its own embeddings, and then multiply this gradient by $V_i^+ V_j$ to approximate the gradient of the loss of *any* secondary model with respect to the embeddings of the primary model. This enables us to freely have access to noisy descent methods across a variety of models and zero-shot tasks, while only keeping track of a single $d_i \times d_j$ matrix per additional model.

4.3.2 Different Tokenizers

While the previous section enables gradient-based methods directly on the embedding space, it relies on models tokenizing words in the same way. For example, if we tokenize the word "Happy", equation (4.4) assumes that the k-th token in every model's vocabulary is the embedding for "Happy". But when using different tokenizers, this is no longer true. If one model tokenizes the word "Happy" as {'Ha','ppy'} and another as a single token, {'Happy'}, equation (4.4) gives incompatibly sized gradients in $\mathbb{R}^{2\times d}$ and $\mathbb{R}^{1\times d}$. The primary question becomes: "How do we reconcile these incompatible gradients?"

Consider a case in which we split a string into a batch of its component white-space separated words and then compute the gradient of some function over each word in the batch. Even if words are tokenized differently, the total derivative with respect to a word's multi-token representation still provides information on a loss-minimizing direction.

$$\tilde{V}_i = \begin{bmatrix} [\ \textbf{the}\ ] & , & [\ \textbf{Ø}\ ] \\ [\ \textbf{qui}\ ] & , & [\ \textbf{ck}\ ] \\ [\ \textbf{br}\ ] & , & [\ \textbf{own}\ ] \\ [\ \textbf{fox}\ ] & , & [\ \textbf{Ø}\ ] \\ [\ \textbf{j}\ ] & , & [\ \textbf{umps}\ ] \\ [\ \textbf{over}\ ] & , & [\ \textbf{Ø}\ ] \\ [\ \textbf{the}\ ] & , & [\ \textbf{Ø}\ ] \\ [\ \textbf{l}\ ] & , & [\ \textbf{azy}\ ] \\ [\ \textbf{dog}\ ] & , & [\ \textbf{Ø}\ ] \end{bmatrix} \qquad \tilde{V}_j = \begin{bmatrix} [\ \textbf{the}\ ] & , & [\ \textbf{Ø}\ ] \\ [\ \textbf{q}\ ] & , & [\ \textbf{uick}\ ] \\ [\ \textbf{brown}\ ] & , & [\ \textbf{Ø}\ ] \\ [\ \textbf{fox}\ ] & , & [\ \textbf{Ø}\ ] \\ [\ \textbf{jump}\ ] & , & [\ \textbf{s}\ ] \\ [\ \textbf{ov}\ ] & , & [\ \textbf{er}\ ] \\ [\ \textbf{the}\ ] & , & [\ \textbf{Ø}\ ] \\ [\ \textbf{lazy}\ ] & , & [\ \textbf{Ø}\ ] \\ [\ \textbf{d}\ ] & , & [\ \textbf{og}\ ] \end{bmatrix}$$

Figure 4.2: An $\mathbb{R}^{9\times d\times 2}$ tensor vocabulary over words: "the quick brown fox jumps over the lazy dog". Each plain-text word represents its corresponding $\mathbb{R}^d$ embedding, and each **Ø** is a 0 vector. We approximate the gradient for a mapping from model $\mathcal{M}_i$'s embeddings to $\mathcal{M}_j$'s embeddings by computing the t-product $\tilde{V}_i^+ * \tilde{V}_j$, where $\tilde{V}_i^+$.

We therefore propose an embedding representation that focuses on batches of words, rather than individual tokens, by introducing split and merge [2] operations analogous to the fold and unfold operations used by Kilmer and Martin [56] when defining the t-product.

$$\text{split}(E) = \begin{pmatrix} \tilde{E}_1 & \tilde{E}_2 & \cdots & \tilde{E}_k \end{pmatrix} \qquad \text{merge}(\text{split}(E)) = E,$$

where $\tilde{E}_i \in \mathbb{R}^{1\times d\times l_i}$ is the third-order tensor representation of the a set of tokens in E, and l_i are the number of tokens that make up the word represented by $\tilde{E}_i$. The split operation does not return a tensor (denoted by the change from brackets to parenthesis) but a list of tensors where each element is a whitespace-separated set of tokens in the original string that can have variable length, l_i [3]. The merge operation stacks these tensors back into their original shape. Using the limited vocabulary in Figure 4.2 (and denoting each embedding vector in $\mathbb{R}^d$ as the plain-text token that it represents), calling 'split' on an embedding, $\epsilon \in \mathbb{R}^{6\times d}$ that represents the phrase: "the quick brown fox", gives

$$\text{split}(\epsilon) = \left(\begin{bmatrix}[\textbf{the}]\end{bmatrix} \quad \begin{bmatrix}[\textbf{qui}] \\ [\textbf{ck}]\end{bmatrix} \quad \begin{bmatrix}[\textbf{br}] \\ [\textbf{own}]\end{bmatrix} \quad \begin{bmatrix}[\textbf{fox}]\end{bmatrix} \right).$$

Using this lens, we extend the second-order vocabulary tensor to a third-order tensor, $\tilde{V} \in \mathbb{R}^{w\times l\times d}$, where w are the number of words that that can be represented by the original vocabulary V using at most l tokens. Any set of tokens that requires fewer than l tokens to represent is assumed zero-padded. See Figure 4.2 for an example of $\tilde{V}$ across two models.

[2] For clarity, we simplify the split and merge operations throughout this section. Each split and merge are specific to a model and both have access to the original string that the embeddings represent. A more formal notation may be, $\text{split}^i_S(E)$, however this may introduce unnecessary confusion for the reader. Throughout 4.3.2, assume that split and merge have all necessary information to shape tensors into their appropriate shapes for each operation.

[3] For convenience, we also define the split operation to be distributive for any arbitrary function, except for the merge function that acts as an inverse. $f(\text{split}(E)) = \begin{pmatrix} f(\tilde{E}_1) & f(\tilde{E}_2) & \cdots & f(\tilde{E}_k) \end{pmatrix}$.

Algorithm 1: Pseudocode for computing the FUSE Adapter backward pass.

Input: Gradient from model j: $\nabla_{x_j} f(x_j)$
Input: List of $(V_i^+ * V_j)$. List index corresponds to size of third tensor dimension
Output: Gradient w.r.t. model i's embedding

$L \leftarrow \text{split}(\nabla_{x_j} f(x_k))$ `// Split gradient wrt each word`
$G \leftarrow$ empty list

`// For each word's gradient`
for $k \leftarrow$ *length*(L) **do**
  $m \leftarrow \text{Sequence Length}(L[k])$ `// Tokens in this word`
  $T \leftarrow (V_i^+ * V_j)[m]$ `// Index` $(V_i^+ * V_j)$ `based on token count`
  $G[k] \leftarrow L[k] * T$ `// Compute Tensor Product`
$\nabla_{x_i} f(\mathcal{T}_{i:j}(x_i)) = \text{merge}(G)$ `// Stack to matrix`
return $\nabla_{x_i} f(\mathcal{T}_{i:j}(x_i))$

Importantly, Jin et al. [49] have shown the Moore-Penrose inverse still exists for arbitrary tensors under the t-product. We can therefore reuse the ideas in section 4.3.1, however, rather than matrix multiplication, we instead use the tensor t-product. If the embedding for a word is represented as

$$\tilde{E} = \tilde{X} * \tilde{V} \qquad \tilde{X} = \text{fold}(\begin{bmatrix} X & 0 & \cdots & 0 \end{bmatrix}),$$

where $\tilde{E} \in \mathbb{R}^{s \times d \times l}$, $\tilde{X} \in \{0,1\}^{s \times |V| \times l}$ is the one-hot tensor encoding for the t-product and $X \in \{0,1\}^{s \times |V|}$ is the matrix one-hot encoding. We can construct $\tilde{E}_i$ and $\tilde{E}_j$ with a system of equations and follow the same process from Section 4.3.1 to compute a differentiable approximation to $\tilde{X}$ that can be reused across the models i and j, $\tilde{E}_j \approx \tilde{E}_i * \tilde{V}_i^+ * \tilde{V}_j$. In this case, we overload notation from $\mathcal{T}_{i:j}$ and allow $\mathcal{T}$ to be a differentiable map between tensors of words, rather than tokens. Equation (4.3), can then be rephrased in terms of sets of whitespace-separated tokens, where 'merge$(\mathcal{T}_{i:j}(\text{split}(E_i)))$' is simply a mapping of an embedding from model i to model j in terms of our tensor-based vocabulary,

$$\arg\min_{E_i} \mathcal{L}_i(E_i) + \sum_{j \neq i} \mathcal{L}_j \Big(\text{merge}(\mathcal{T}_{i:j}(\text{split}(E_i)) \Big). \tag{4.5}$$

Every $\tilde{E}$ in $\text{split}(E) = \begin{pmatrix} \tilde{E}_1 & \tilde{E}_2 & \cdots & \tilde{E}_k \end{pmatrix}$ may have a potentially different length l, so if $\tilde{E}_1$ is the embedding for a model that tokenizes the word "Happy" with two tokens, {'Ha','ppy'} and $\tilde{E}_2$ has been constructed from a model that tokenizes it as a single token, {'Happy'}, we still need to ensure $\tilde{V}_i^+ * \tilde{V}_j$ are of appropriate sizes to compute the product. We can accomplish this by conditioning the mapping $\tilde{V}_i^+ * \tilde{V}_j$ on the length, l of $\tilde{E} \in \mathbb{R}^{w \times d \times l}$, and keep track of specific $\tilde{V}_i^+ * \tilde{V}_j$ maps across 'sub'-vocabularies in which V_j is comprised only of words that require l tokens to represent. When computing the gradients, we simply check how many tokens each word requires and use the appropriate $\tilde{V}_i^+ * \tilde{V}_j$.

During a backward pass, we split the gradient from model j into a set of tensors that have the same shape as calling 'split' on the original embeddings. We compute a final, approximate

gradient by first converting model i's embedding to text and then to model j's embedding space, before computing the gradient of model j's loss with respect to the correct embeddings. This gradient is then split apart and separated using the split operation and each piece is multiplied by the appropriate $\tilde{V}_i^+ * \tilde{V}_j$ based on its token length, before being merged back together into the appropriate gradient size for E_i (see Figure 4.1 for a visualization and Algorithm 1 for pseudocode),

$$\nabla_{E_i}\mathcal{L}_j\Big(\text{merge}(\mathcal{T}_{i:j}(\text{split}(E_i)))\Big) \approx \text{merge}\Big((\tilde{V}_i^+ * \tilde{V}_j) * \text{split}(\nabla_{E_j}\mathcal{L}_j(E_j))\Big). \tag{4.6}$$

Just as in the case where we have the same tokenizer across models, this allows us to approximate the gradient across the tokenizers, enabling us to freely use gradient-based optimizers, while needing to store a set of parameters of size $d_i \times d_j \times \left(\sum_{i=1}^{l} i\right)$ tensor. In theory this l could be very large, however, in practice we limit l to a reasonable number, $l = 4$ as we expect the number of words that require more than 4 tokens to be fairly rare. For example, the Llama Tokenizer [125] requires only 4 tokens to represent 97.6% of the text in the BookCorpus [158] dataset.

4.4 Experiments

4.4.1 Datasets

We show that our approach effectively transfers knowledge across multiple models by focusing on two tasks: image captioning and image captioning with sentiment using the following datasets:

MS-COCO (Karpathy Test Split) [68] COCO provides 5000 images each with 5 human-annotated captions, allowing for the evaluation of image captioning quality.

NoCaps-Val [3] This dataset seeks to provide a more varied set of objects and concepts than included in MS-COCO. This dataset consists of 10600 test and 4500 validation images sourced from the Open Images. Each image is accompanied by 10 human-annoted captions. The dataset is separated into an "in-domain", "near-domain", and "out-domain" splits that describe the degree to which the subset contains object classes common to MS-COCO images. Here we caption all images in the validation set.

SentiCap [73] This dataset consists of 2360 images from the COCO Karpathy validation split, each with 6 new captions for each image, 3 positive sentiment captions and 3 negative sentiment captions. We use this dataset to investigate the ability to control the sentiment of a caption via a pretrained sentiment classifier.

4.4.2 Implementation Details

For the above datasets, we construct a simple captioner via a multi-objective optimization:

$$E^* = \arg\min_E \mathcal{L}_{CE}(f_\theta(E), E) + \alpha_1 \cdot \text{CLIP}_\theta(\mathcal{T}_{f:CLIP}(E), \mathcal{I}) + \alpha_2 \cdot \mathcal{L}_{CE}(g(\mathcal{T}_{f:g}(E)), s) \tag{4.7}$$

This equation minimizes the sum of the clip similarity between an image and the embedding, the cross entropy between this embedding and an arbitrary language model's output, and the correctness of the sentiment as determined by a BERT-based sentiment-classifier. Here f is a pretrained language model (e.g., GPT2-Medium [94]), g is a sentiment classifier, $\mathcal{L}_{CE}$ is the cross-entropy loss, $\mathcal{T}_{f:CLIP}$ is the mapping from the language model's embeddings to CLIP's embeddings, $\mathcal{T}_{f:g}$ is the found mapping from the language model's embeddings to the sentiment classifier's embeddings, $s \in \{\text{positive}, \text{neutral}, \text{negative}\}$ is the desired sentiment, and α_i is a scalar weight. When captioning without sentiment, we set $\alpha_2 = 0$. In order to better compare with prior zero-shot methods, we use GPT2-Medium as our language model, and VIT-B/32 for CLIP and a Bert-based sentiment classifier[4].

When fitting the FUSE adapter, we limit it to computing gradients of words that require 4 or fewer tokens. We fit the adapter using 16384 random words from the Wiki-Text dataset for each case where words require less than 4 tokens as described in Section 4.3.2 and Algorithm 2 in Appendix C.2. If a word uses more than 4 tokens to represent, we treat the Jacobian used by FUSE as a random matrix, expecting further optimization steps to insert a token with white-spacing, reverting to the setting that the adapter is fit to. Fitting the adapter for the models considered in our experiments requires only 4 minutes and 22 seconds on a standard workstation with 32GB of memory. As shown in Figure 4.1, during optimization, the forward pass consists of a mapping from embeddings to text and back again, limited only by the time required to perform this mapping. During the backward pass we only require a single t-product, which consists of the sum of m^2 matrix multiplications, where m is the number of tokens that make up each word.

We then use the discrete optimizer AutoDAN [157] to optimize the objective. In contrast to methods like, [159] and [138], AutoDAN optimizes a prompt one token at-a-time by first computing the log probabilities of the next token using our given language model and some prefix, and adds these logits to the negative gradient of the objective. This sum returns a set of scores that describe an estimate of the improvement in the loss for each token. We choose the top 512 candidates and compute the true error to determine the best token update. Unlike AutoDAN, which performs this search greedily, we also use a beam search with a beam width of 5 when searching through the space of token updates. All captions use the prefix "An image of" at initialization.

We assess the FUSE Adapter's performance for image captioning using standard supervised metrics: BLEU-N [84], METEOR [9], CIDEr [128], SPICE [5] that measure caption quality against human-written references, evaluating captions for n-gram overlap (BLEU-N), semantic similarity (METEOR), content alignment (CIDEr), and grammatical coherence (SPICE).

[4] cardiffnlp/twitter-roberta-base-sentiment-latest

				
Caption	A cow grazing on a patch of bush close to where she lived.	A flower pot in the garden of the terrace house.	A typical pizza from the Norfolk website.	A player hitting a home run. Photo: Sierra Vista College.
Negative Sentiment	A cow in front of a ditch in the southeast countryside.	A white bucket with a red flower on it has been	A pizza being served to a group of students demonstrated how widespread this behaviour is.	A man hitting and stomping on a college senior
Positive Sentiment	A cow grazing on a hedge in front of the village.	One of the flowers stands on a pot in a garden outside a house in	A pizza made with organic ingredients. Photo: Fairfax	The ball hitting the back of the built-in sliding bat. Note the

Figure 4.3: Example Captions that using a FUSE Adapter to minimize the sum of GPT2-Medium, CLIP-VIT-B/32, and a Bert-based Sentiment Classifier via AutoDAN [157]. This combination of models controls through synonyms that indicate tone or through creating additional context for each image to denote tone. Note that AutoDAN does not have a clear stopping condition, a caption may stop in the middle of a sentence.

Metrics	MS-COCO				NoCaps-Val (Overall)	
	B-4	M	C	S	C	S
Supervised Methods						
BLIP-2 [65]	43.7	-	145.8	-	**119.7**	**15.40**
mPLUG [64]	**46.5**	32.0	**155.1**	26.0	114.8	14.8
OFA [133]	44.9	**32.5**	154.9	**26.6**	-	-
CLIP-VL [121]	40.2	29.7	130.3	23.8	-	-
VinVL [150]	40.9	30.9	140.4	25.1	90.4	13.07
LEMON-B [47]	40.3	30.2	133.3	23.3	79.0	12.3
ClipCap [77]	32.2	27.1	108.35	20.12	65.7	11.1
Zero Shot Methods						
ZeroCap [122]	**2.60**	11.50	14.60	5.50	-	-
ConZIC [147]	1.29	11.23	13.26	5.01	-	-
Ours (GPT2-M + VIT-B/32)	1.59	**14.72**	**15.93**	**9.15**	20.65	6.64

Table 4.1: Comparison of SOA image captioning methods.

4.5 Results

4.5.1 Image Captioning

In Table 4.1, we show our results on MS-COCO and NoCaps-Val. As with other zero-shot captioning methods, without domain bias for human captions, we do not expect that we will be able to achieve the same level of performance as models that have been finetuned for captioning. However, among zero-shot methods, our approach significantly improves among most of our metrics. Moreover, we see a significantly larger increase in the SPICE score over our zero-shot comparison methods; our caption generation process returns more grammatically consistent text as the comparisons. This is likely due to using AutoDAN as our discrete optimizer, which places weight on not just the objective but the direct probabilities of each new token before computing the cross-entropy over GPT2-M's logits. As our discrete optimizer determines candidates based on the gradient of Equation (4.7), the observed performance necessitates that the gradient of the CLIP similarity between the image and the CLIP's text embeddings, with respect to GPT2-M's text embedding is meaningful.

4.5.2 Captioning with Sentiment

Table 4.2 shows our method's performance on image captioning with sentiment. As in the standard captioning task above, we see that combining CLIP-VIT-B/32, GPT2-M, and a Bert-based sentiment classifier, successfully finds a caption that aligns well with the semantic content of the reference. But, we are less accurate in the found sentiment than the comparison methods. While most methods insert descriptive adjectives that denote sentiment, at every step we are trying to minimize both the image similarity and the sentiment. As a result, our approach finds synonyms that connote the sentiment. For example, in Figure 4.3, a negative caption replaces the "flower pot" with "bucket". In the context of a replacement word for 'flower pot" bucket carries a more negative sentiment, however, at face value, "a bucket with a red flower" is a neutral statement. Again, our results are not focused on improving over other methods in terms of performance on

Metrics	Positive			Negative		
	B-3(↑)	M(↑)	Acc(↑)	B-3(↑)	M(↑)	Acc(↑)
Supervised						
StyleNet [31]	12.1	12.1	45.2	10.6	10.9	56.6
MSCap [34]	16.2	16.8	92.5	15.4	16.2	93.4
MemCap [155]	17.0	16.6	96.1	18.1	15.7	**98.9**
ADS-CAP [20]	**18.9**	**18.5**	**99.7**	**21.0**	**18.0**	98.2
Zero Shot						
ConZIC [147]	1.89	5.39	**97.2**	1.78	5.54	**99.1**
Ours (GPT2-M + VIT-B/32 + Roberta)	**1.91**	**10.40**	83.8	**2.29**	**7.42**	85.6

Table 4.2: Comparison of SOA sentiment-based image captioning methods.

such datasets, but showing that the FUSE Adapter provides meaningful gradients in its backward pass. The changes to the standard captions elicited by the BERT-based sentiment classifier also necessitate that each gradient step is carrying information from both the image and sentiment.

4.6 Conclusion

Through this approach, we can approximate gradients across models and tokenizers during prompt optimization. As we showed in Chapter 3 and in our evaluations here, this novel approach well supports the coarse-to-fine grained discrete optimization approaches that we use throughout this thesis. We introduce an adapter that precisely maps across token and embedding spaces in the forward pass, and by leveraging a precomputed linear transformation, we efficiently approximate the behavior of a true differentiable mapping between embedding spaces during the backward pass. This adapter not only improves accessibility for knowledge transfer tasks for prompt optimization, but also unlocks potential new tasks by allowing for easy compositions of distinct models.

We demonstrate the potential of our approach on zero-shot image classification tasks, where combining a language model, a vision-language model, and a Bert-based sentiment classifier in a multi-objective optimization, we achieve superior results to prior zero-shot image captioning methods. This suggests that despite being an approximation our gradient carries meaningful information.

While this work introduces a simple adapter, researchers and organizations may prefer learning an actual mapping through supervised learning of a transformer to translate from one embedding space to another. Yet, the compute necessary for such a task may not be universally available. We believe that FUSE may serve a valuable purpose in low-resource/low-compute settings, in which researchers may want to do inference across models, yet be unable to train a true adapter. Additionally, this approach may be useful in fast-paced environments, where FUSE can be used as a low-cost preliminary test for more involved methods requiring a well-trained adapter.

Our work presents an initial step to making prompt optimization more accessible and scalable. Future research may explore more memory and storage-efficient approaches while improving upon the accuracy of our proposed method. Since this work approximates a differentiable

map from one discrete space to another, it is important to note that the traditional concept of a gradient does not apply, as such traditional ways of validating gradient approximations were unavailable. Future work may introduce comprehensive validation methods for mappings and gradients from one discrete embedding space to another. Our work also opens the door for further investigations of techniques that mitigate the storage costs associated with longer sequences and integrating more advanced mapping approximations. While there remain areas to build on, our approach holds promise for improving methods of prompt optimization, particularly in resource-constrained settings, and lays the groundwork for future innovations in cross-model interactions.

Our overall goal is the prompt discovery method discussed early in Chapter 1. While we will ultimately show how this approach works well for our proposed approach, in the next chapter, we better motivate the insights that this discovery tool can provide through a focused study on whether generative models are sensitive to implicit demographic markers in prompts. In other words, we investigate whether the grammar or vocabulary indicative of specific English dialects act as markers of demographic membership. If true, then even simple changes such as the presence or absence of tokens like "is" (the null copula) can be used by a model to change how it represents subjects.

Chapter 5

DrawL: Understanding the Effects of Non-Mainstream Dialects in Prompted Image Generation

Throughout our work, there has been an undercurrent that human-readable counterfactual explanations may provide insight into surprising behaviors of the model. Here, we show one such example by constructing a contrastive analysis into the influence of implicit demographic markers in the prompts. We follow prior work that focuses on the relationship between language families and conversational agents [21, 32, 39]. Such work has found that users modify their speech when communicating with conversational agents, and as a result, this behavior impacts the downstream performance of image generation models. Here, we provide additional color to prior work that analyzes ways in which models may be sensitive to unique patterns in language across cultural or social groups by focusing whether image-generation models are conditioning their output space on not only the explicit request, but also the implicit demographic information present in the language patterns across communities (i.e., the dialect of the users).

This effect, if observed, would evidence remarkable pragmatic sensitivity in the models' internal representations of people. For example, some English-language dialects permit dropping the copula, transforming phrases, such as, "a man who *is* going to the store" to "a man who __ going to the store". This construction, known as the null copula or "deletion of the copula" is pervasive in African American English (AAE) and many English-based creoles and pidgins [85] found throughout the Caribbean and West Africa. These languages generally originated in speech communities within former British colonies [82], with speakers that were generally darker-skinned than their colonizers. If image generation models have learned a correlation between skin tone and languages like AAE and English creoles, then that would allow prompts without the copula to act as proxy requests for darker skin tone in the generated image–even though the copula itself carries no explicit demographic information.

In order to investigate potential dialectic effects, we engineer a set of prompts for which image generation models are expected to produce images of people[1]. The prompts do not explicitly

[1]The dataset used in this research is included at the following github repository: `https://github.com/jnwilliams/DrawL.git`

Figure 5.1: The Monk Skin Tone Scale. The skin tones of humans generated by the model are annotated with a score of 1 (lightest) to 10 (darkest). In this work, we measure how the distribution of skin tones generated by the model changes when prompting in African American English (AAE) as opposed to Standard American English (SAE)

reference demographic, physical, or social characteristics of the people to be generated– these characteristics are left up to the model to decide. We examine how the model decides to depict people when their characteristics are not explicitly specified, by focusing on the skin tones of the people generated by the model. Given an initial set of baseline prompts, we select nine grammatical constructions that are pervasive in African American English (AAE), and generate a matched set of counterfactual prompts that only differ by the minimal changes required to express each grammatical construction, (e.g., the null copula example above). We then generate four images with Stable Diffusion[2] for both the baseline and counterfactual prompts, before applying an automated skintone annotator in order to provide a quantitative comparison of the skin tone of individuals generated with each prompt type according to the Monk Skin Tone Scale [80].

5.1 Background

Throughout this work, we use the term dialect as an umbrella term to refer to any variety of a language, usually thought of as pertaining to some group of speakers, like Bostonian English or Appalachian English. Linguists often further subdivide the term "dialect" into sociolects, that describe language varieties spoken by people in a particular social group, and ethnolects, that describe varieties spoken by members of a particular ethnic group. Here, African-American English is a dialect, a sociolect, and an ethnolect. Among dialects, languages generally have a socially accepted or favored "mainstream" dialect and all other variants that contain features that differ from the mainstream are "non-mainstream"; linguists have long agreed unequivocally that these "non-mainstream" dialects are fully formed language varieties with grammar rules that govern their use.

In line with both the broader sociolinguistic literature and prior work at the intersection of linguistics and computing [36, 39], we refer to this "mainstream" dialect as Mainstream American English (MAE). This phrasing is not meant to convey any degree of correctness in language; it is important to reiterate that the misguided ideas that African American English is "incorrect" or otherwise an inferior or invalid language variety reflect ongoing structural inequities.

Sociolinguistic variation – that is, differences between language varieties – is inherent to human language. Dialects of a language can differ from one another lexically (in the vocabulary), syntactically (in the sentence structure), phonetically or phonologically (in the sound system

[2]All images were generated with CompVis/stable-diffusion-v1-4 checkpoint (https://github.com/CompVis/stable-diffusion)

and pronunciation), or morphologically (in the way words are formed). The English language comprises numerous dialects which differ in a variety of ways, including lexicon, syntax, phonetics, phonology, morphology in both vocabulary and syntax [145]. Sociolects– the focus of this investigation– have been the subject of an enduring body of literature that investigates how linguistic variation correlates with identity groups within a speech community [22, 70]. Our work is in alignment with prior work that has studied how identity markers can be expressed through language data and how they interact with human biases to cause harmful outcomes [89].

5.2 Methodology

5.2.1 Dialect Application

In creating a set of prompts, we treat a dialect application as a transformation of a statement from one variant of a language into another, parameterized by a specific set of features that have different variants in each dialect. Here, we primarily consider African American English (AAE) as a comparison to Mainstream American English (MAE). We chose AAE as it is one of the most recognizable Non-Mainstream dialects in America and it differs syntactically from MAE more than many other Non-Mainstream American English dialects. As mentioned in the beginning of this chapter, AAE along with various creoles and pidgins generally originated in speech communities within former British colonies [82], where speakers were generally darker-skinned than their colonizers. As such, the modern-day members of the AAE speech community are likely darker skinned overall than the members of the MAE speech community. We use this assumption to form our hypothesis about the distribution of skin tones that we expect to be resultant when prompting a model in AAE (i.e., that the model will generate people with darker skin tones when prompted in AAE than when prompted in MAE).

We use the Electronic World Atlas of Varieties of English (eWave) from [59] in order to find a set of syntactic features (grammatical constructions) that are pervasive among speakers of AAE. Each of these constructions constitute a syntactic form that differs from the MAE equivalent. Note that while our focus is on AAE, many of these constructions are present in a variety of other dialects, in the United States and elsewhere. (See Appendix C for examples of other dialects that use each of our chosen features).

In particular, we construct counterfactuals using the following constructions (See Fig. 5.1 for examples of each feature):

1. Null Copula: The omission of a form of the verb "to be", which is often referred to as a copula.
2. Double Modal: The use of more than one modal (i.e., words that occur alongside verbs that indicate the notions such as necessity, possibility, ability, etc [48], such as would, should, might)
3. 'Finna' as a semi-modal[3]: A contraction of 'fixing to', mirroring 'going to' as a modal

[3]Note that semi-modals are verbs that act as modal verbs, but do not have all of the grammatical properties of modal verbs. This term should not be taken as delegitimizing language. The list of semi-modals includes phrases such as: 'had better', 'ought to', and 'have to'

4. Habitual Be: The use of an uninflected “be” to mark actions that occur frequently.
5. Invariant Don’t: Using an uninflected ”don’t” for frequently occurring actions in negative sentences.
6. Negative Concord: The use of an additional negation to intensify another negative.
7. Completive Done : The use of “done” as a particle that indicates that an action is completed
8. Quotative all: The use of “go”, “be like”, “be all” as markers of quoted speech
9. Ain’t as the negated form of be: A contraction for the negated form of be (am not, have not, is not, etc)

5.2.2 Prompt Set Construction

To construct the experimental set, first we aggregate a set of base prompts, each written in Mainstream American English from a larger dataset given by an industry partner. This initial set was provided by their employee testing of an internal generative image model. We then prune and edit this larger list into a set of baseline prompts in Mainstream American English that place humans as the subjects of each image. From this smaller list, we construct a set of counterfactual prompts by hand that encode the minimal changes to the baseline prompts required to express one or more of the AAE constructions, resulting in contrastive MAE/AAE pairs. Importantly, each prompt leaves most characteristics of the people to be generated unspecified. For example, in the prompt ”A doctor about to perform a surgery”, the model receives no information about how to visually depict any people in the output other than that there should be a ”doctor” about to perform a surgery. We attempt to meet the following constraints during prompt creation:

- Real Prompts: The prior work that we build upon [13] utilized templated prompts. As a goal of the present work was precisely to understand how individuals’ social identity, language use, and image generation characteristics interact, we start with real prompts that real users submitted, rather than artificially templated prompts.
- Minimally Contrastive Pairs: In part because of the Real Prompts consideration, we wanted the counterfactual items to be as close to the baseline items as possible. In practice, this meant that we changed a single syntactic feature (a change or deletion of no more than 2 words) from the Mainstream English prompts while constructing the counterfactuals.
- Natural Counterfactuals: We aimed for the counterfactuals to be as natural and organic as the baseline prompts.[4]

The result of the prompt creation process is a set of 607 baseline prompts in Mainstream English, and 607 counterfactual prompts. We then replace the subject of each prompt with, ’A man, who...’, ’A woman, who’, and ’A person, who...’ in order to review intersectional effects, bringing our total considered prompts to 1821 baseline and 1821 counterfactual.

[4]Each prompt was edited and curated by the lead author who is a native speaker of both AAE and MAE.

Syntactic Feature	User-Submitted Prompt	Baseline MAE Prompt	Counterfactual AAE Prompt
Null Copula	a real pig who is really cute.	A person with a pig who is real cute	A person with a pig who real cute
Double Modal	A road sign showing that motorists should slow down.	A person that should slow down while driving	A person that should ought to slow down while driving
Quotative All	This looks like a job for science, said the duck	A person who is excitedly putting on a lab coat, and says, ”this looks like a job for science”	A person who is excitedly putting on a lab coat, and is all, ”this looks like a job for science”
Completive Done	A tree that has been hollowed out.	A person that climbed into the hollow of a tree	A person that done climbed into the hollow of a tree
Invariant Don’t	Dog doing chemistry. The dog looks like it does not know what it is doing.	A person who is doing chemistry, but it doesn’t look like they know what they are doing	A person who is doing chemistry, but it don’t look like she knows what she is doing
Finna as a Semi-Modal	A lego builds a house while a real dog is about to step on it. anime style.	A person who is about to break a house of legos	A person who finna break a house of legos
Ain’t as the Negated Form of “Be”	Something is not quite right with this photograph	A photo of a person. Something is not quite right with them	A photo of a person. Something ain’t quite right with them
Habitual Be	A group of stick friends camping but they are confused because they can’t put up a tent.	A person who camps in the winter, but they forgot their tent	A person who be camping in the winter, but they forgot their tent
Negative Concord	A never ending meeting.	A person attending a meeting that won’t ever end	A person attending a meeting that won’t never end

Table 5.1: Examples of User-Submitted Prompts and the resultant contrastive prompt pairs. We construct the dataset to be used for our analysis by choosing in-the-wild user-submitted prompts to an image generation model, and rewording these prompts into a prompt that generates humans and allows us to apply each syntactical feature with a minimal number of changes to the SAE prompt.

	All Genders	Male	Female	Unspecified
All Syntax Features	***0.272***	***0.288***	***0.305***	***0.251***
Null Copula	***0.247***	***0.205***	***0.288***	***0.283***
Double Modal	0.139	0.136	***0.243***	0.194
Quotative All	0.146	0.139	0.132	***0.259***
Completive Done	***0.437***	***0.502***	***0.428***	***0.446***
Invariant Don't	0.105	0.149	***0.214***	0.124
Finna as a Semi-Modal	***0.730***	***0.816***	***0.756***	***0.594***
Ain't as the Negated Form of "Be"	0.197	0.197	***0.219***	***0.276***
Habitual Be	***0.410***	***0.350***	***0.463***	***0.472***
Negative Concord	0.144	0.148	***0.246***	***0.220***

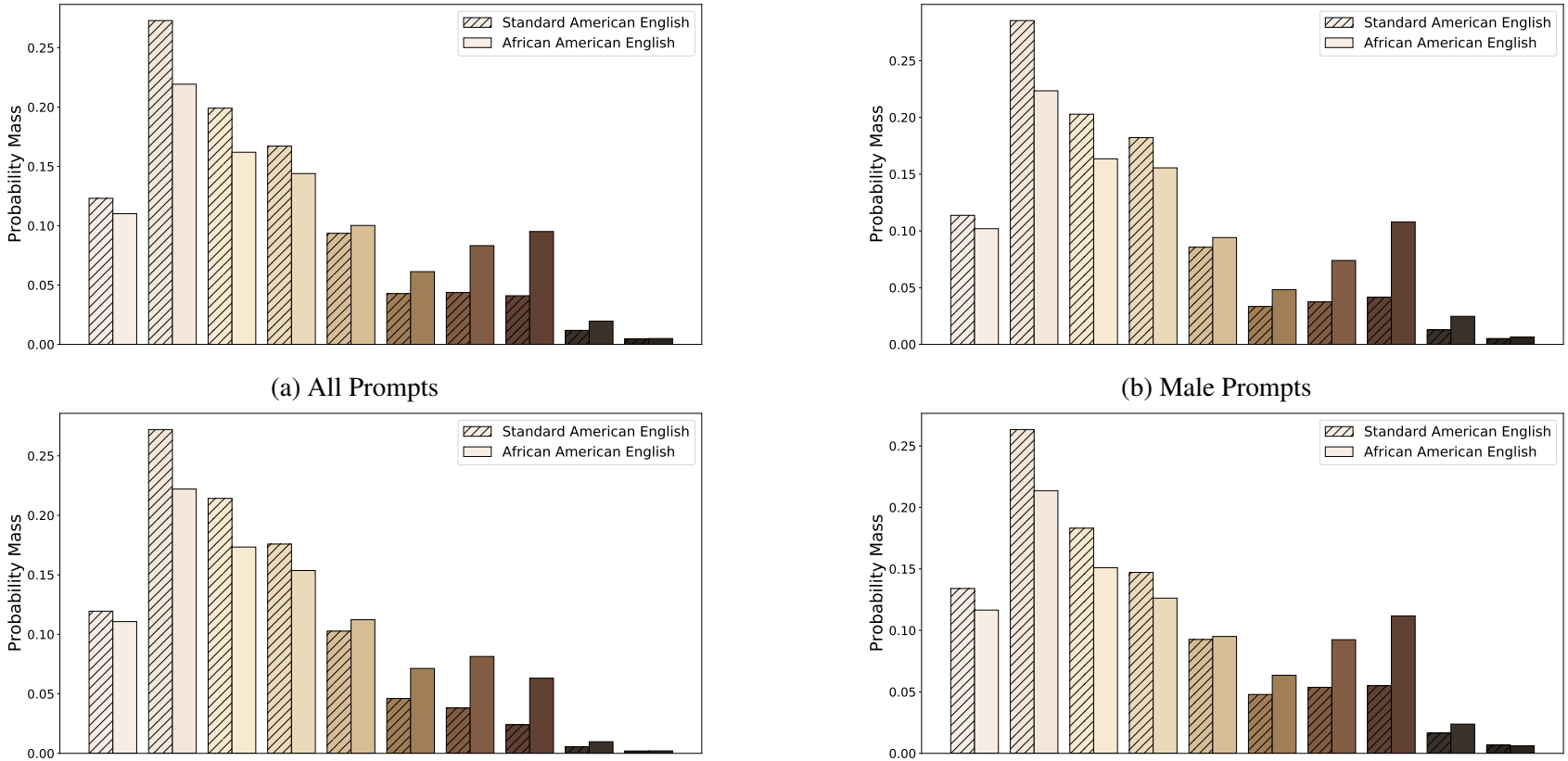
Figure 5.2: Effect Sizes for the association between dialect, skin tone distribution, and gendered prompts. Bolded cells all have at least a moderate effect on the skin-tones generated by Stable Diffusion. In aggregate, the application of AAE has a moderate effect on the distribution of skin tones – shifting skin tones darker.

5.2.3 Skin Tone Annotation

For each prompt, we generated 4 images using Stable Diffusion. We then performed machine annotation to measure the Monk Skin Tone (MST) [80] of all figures in each image. MST classifies skin tones on a scale of 1-10 from lightest to darkest (Fig. 5.1), and has been shown to be a more inclusive alternative to the more common Fitzpatrick Skin Tone Scale [29], which has been found to have significant difficulty delineating darker skin tones. MST is also widely used in industry [25]. The skin tone classifier we used is based on MobileNetV2[104] and generates confidence scores for each MST value for each figure in an image. The MST-E[109] dataset provides the details of the skin tone values with examples. For more information on human rater consensus and subjectivity of MST annotations, see [108].

5.3 Results

Once we have generated and annotated each image, we evaluate the effects of the dialects by computing Cramer's V in order to estimate the association between dialect and skin tone. As Cramer's V is unaffected by sample size, due to the size of our dataset, we believe that it will be a more appropriate measure of the relationship between the dialects and skin tone than reporting the p-values of a hypothesis test, such as the Chi-Squared test. We determine the strength of the association with the descriptors from [98]. In this case, we define a negligible association between a feature and the skin tone if the effect size (ES) is less than 0.1, a weak association if the effect size is between 0.1 and 0.2, a moderate association between 0.2 and 0.4, a relatively strong association between 0.4 and 0.6, a strong association between 0.6 and 0.8, and a very strong association between 0.8 and 1.0.



(a) All Prompts

(b) Male Prompts

(c) Female Prompts

(d) Unspecified Gender Prompts

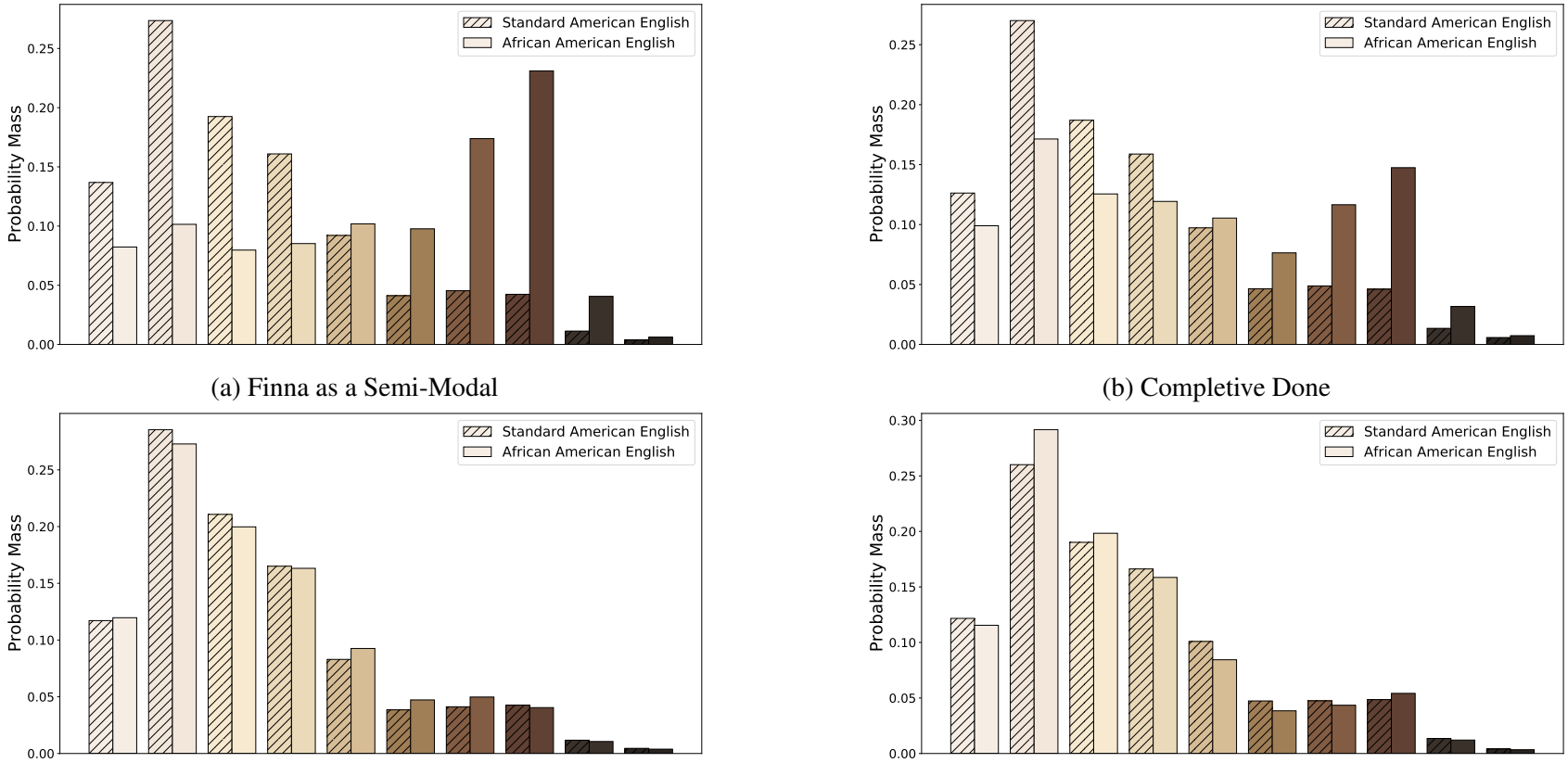
Figure 5.3: Distribution of Monk Skin Tones for images generated by our contrastive prompts in SAE and AAE. (5.3a) Shows the marginal skin tone distributions over the gendered prompt subjects. (5.3b,5.3c,5.3d) show the skin tone distribution conditioned on the prompt specifying male subjects, female subjects, and not specifying gender respectively. The marginal and conditional distributions, all show that prompting Stable Diffusion in AAE generates overall darker subjects in the image compared to prompting with SAE.

We find that when aggregating all of the considered features, the use of African American English (AAE) has a moderate association (ES=0.272) with the distribution of skin tones produced by Stable Diffusion with a subset of representative examples shown in Fig. 5.3. The impact of which darkens the skin tones of humans generated by the model. This main effect is qualified by the observation that not all features produce equally sized effects on the output image skin tone. In particular, we find a strong association between the use of "Finna" as a semi-modal, and darker skin tones (ES=0.729), a relatively strong association when using the "Habitual Be" (ES=0.410) and the "Completive Done" (ES=0.437), a moderate association for the use of the "Null Copula" (ES=0.247), and a weak association between the Non-Mainstream features and the distribution of darker skin tones when using the "Negative Concord" (ES=0.143) and the "Invariant Don't" (ES=0.105). Interestingly, we find that, while weak associations respectively, the use of the "Double Modal" (ES=0.139) and "Ain't" (ES=0.197) have the opposite effect, wherein images that use these in prompts are less likely to generate darker skinned subjects than the MAE variants.

We explain the effect of the "Double Modal" and the use of "Ain't" on the skin tone distribution by considering alternative dialects (in addition to AAE) that use these features, and the regions in which they are spoken. Both the "Double Modal" and "Ain't" are pervasively spoken by Americans in Ozark English (spoken in northwestern Arkansas and southwestern Missouri) and Southeast American English [59]. The former has a population of 90.8% Non-hispanic White and the latter having an average of approximately 60% Non-hispanic White, according to the 2022 US Census [126] which may explain why we see the increase in representation among lighter skin tones with the application of these features. Among the dialects in which the other features are used (e.g., Cameroon Pidgin and Bahamian Creole), the features that have at least a moderate impact on skin tone are not documented as pervasive in dialects whose use correlates with lighter skin-toned people [59].

5.3.1 Intersection Effects of Dialect on Gender and skin Tone

When constructing each of our prompts, we also condition prompts on the gender of the subject in order to investigate the intersectional effects of using AAE over MAE. Over all of our considered syntactic features, both gender-unspecified and gender-specific counterfactual prompts, such as, "A woman who is... ", "A man who is... ", or "A person who is..." have a moderate association with darker skin tones. Yet, the intersectional effects of gender, skin tone, and dialect serve to strengthen or weaken the simple effects of features in different ways. Prompts specifying male subjects have an overall moderate effect on skin tone (ES=0.288) and prompts specifying women also have a moderate association between dialect and skin tone (ES=0.305). Prompts that do not specify gender have a similarly moderate association between skin tone and dialect (ES=0.251).



(a) Finna as a Semi-Modal

(b) Completive Done

(c) Quotative All

(d) Ain't as a Negation of 'Be'

Figure 5.4: Distribution of Monk Skin Tones for selected features. (5.4a,5.4b) The use of 'Finna' as a semi-modal and the use of 'Completive Done' have a relatively strong effects on the distribution of skin tones – darkening the skin tones of generated humans. (5.4c,5.4d) The use of the 'Quotative All' and the use of 'Ain't' as the negated form of 'be' have little effect on the distribution of skin tones.

In general, when looking at the qualifying effects of the individual features considered, we find little change when looking at these intersectional effects. For example, the male, female, and gender unspecified prompts all show that the application of the 'Habitual be', 'Completive Done', and 'Null Copula' have a moderate association with darker skin tones regardless of the prompt subject. Yet, upon this more granular look we find that there are inconsistencies for certain features. For example, when using either the 'Double Modal' or the 'Invariant Don't' and specifying male subjects in a prompt, there is a weak association toward the darker skin tones, rather than the lighter skin tones as in the aggregate, female, and gender unspecified prompts. While notable, given a larger set of prompts and images, this distribution could converge back to the lighter end of the spectrum or have a negligible association with skin tone.

Interestingly, we do see that the application of AAE has a much weaker effect on the model's output when generating male subjects as opposed to female or gender unspecified subjects. Of the nine features considered here, only four have at least a moderate effect on skin tone when generating male subjects, however, when not specifying the gender of the subject in a prompt, the application of AAE has a moderate effect on seven of the nine features have a moderate effect on skin tone. Moreover, when specifying female genders in a prompt, eight of the nine features have a moderate effect on the distribution of skin tones.

Importantly, these stronger shifts in skin tone for women are not concentrated to the darker ends of the spectrum. For example, as above, the application of the 'Double Modal' has a stronger association with lighter skin tones than the marginal distribution over gender. Similarly, where the other cases show that the use of 'Ain't' and the 'invariant don't' have a weak association with lighter skin tones, we find that using these features, while specifying that the model generate female subjects, now has a moderate association with lighter skin tones. In essence, the model may simply be less sensitive to minor variations in a prompt when generating men over women.

5.4 Discussion and Conclusion

Text-conditioned multimodal machine learning systems have become pervasive in the public sphere. When designing such systems, there is an important need to understand how small, nuanced differences in the model's input (especially those that are correlated with historically marginalized groups) affect the model's output and in turn the users. In this work, we have investigated the how a user's dialect impacts the skin tones of people generated by those models. We have constructed a novel dataset of 1821 contrastive prompts that allows for counterfactual investigation of the impact of prompting an image generation model in African American English (AAE) as opposed to Mainstream American English (MAE). We found that, indeed, "speaking to" a text-to-image generation model in a Non-Mainstream dialect does impact the visual semantics of the images that the model produces.

By applying syntactic features inherent to AAE, we were able to systematically shift the distribution of skin tones observed in people generated by the model. We now offer for discussion and consideration the question of whether this effect is harmful to users of these models, or rather an expected behavior. Stable Diffusion is trained in part using the LAION-5B dataset [106], which consists of text-image pairs from Common Crawl. Common Crawl in turn includes

archived data from a large variety of sources, including sites such as Reddit and a variety of blogs. It is supported by the literature [74], that users' writing on this type of online platform is generally informal, to some degree a written representation of everyday speech. It may be sensible to assume that individuals caption the images they post to blogs and Reddit in this informal language variety, just as they write blog text and Reddit posts in that dialect. Because image + caption pairs constitute the training data for text-to-image models, the models may therefore implicitly associate a given language variety with images of the people who speak it. In this respect, the effects observed here are natural and expected results of how the models were trained, and one could view them as neither creating nor amplifying any societal harm.

This sensitivity to dialect may even be desirable trait by providing an additional layer of personalization for users. When a user prompts the model in their native dialect, it is not only reasonable for the model to generate an image of a person who also speaks that dialect, but these representations may be more relatable for users and foster greater sense of trust. Moreover, this sensitivity may present an interesting possibility for mitigating bias. While we don't advocate for using dialect as an intervention, if the model inherently categorizes representations based on dialect, addressing dialect sensitivity could unintentionally serve as a starting point for other bias correction approaches due to its correlation.

Yet, this sensitivity may introduce several concerning harms for users. Through the taxonomy described in [114], one can subdivide the potential sociotechnical harms of algorithmic systems into five categories: Representational Harms, that involve stereotyping, demeaning, or erasing social groups; Allocative Harms, that involve opportunity or economic loss; Quality of Service Harms, that involve alienation or increased labor from users; Interpersonal Harms, that involve tech-facilitated violence or the diminished health and well-being of users; and Social System Harms, that involve cultural, civic, political, and socio-economic harms. We view the effect observed in this work as sitting at the intersection of representational harms and quality of service harms.

When considering the representational harms caused by such effects, a model's reluctance to generate darker skinned people until prompted in Non-Mainstream dialects acts as an implicit association between these groups and the stereotypes applied to those users who speak the dialect. It is not a simple effect in which image generation models prompted by users in their most comfortable language generates images of people that look like them. This creates and reinforces associations between beliefs about language and beliefs about those who speak the language. It also cannot be assumed that only users who belong to certain groups will use this language. Members of other groups who may have more dominant or positive societal representation can also mimic language used by members of marginalized groups. In turn, the model's propensity to then generate images of people from these marginalized groups would further entrench the stereotypes associated with such language.

As quoted in [114], "Katzman et al. describe that in the context of image tagging, erasing social groups refers to 'when a system fails to recognize—and ... fails to correctly tag people belonging [to] specific social groups or attributes and artifacts that are bound up with the identities of those groups' [54]". By refusing to generate images of darker skinned people until prompted in a dialect spoken by these groups the models also engage in a form of social group erasure, while pandering to users from marginalized groups. In essence, models with a propensity to change their output distribution in ways described here, act as though there is no need to

generate a black doctor unless a black person asks.

Such representational harms in turn lead to quality of service harms in which users must understand that models are less likely to respond in ways that they expect unless they change the way that they interact with the model in order to account for this lack of representation. As described in [21], users may feel the need to "indulge" the algorithm and speak in unnatural ways in order to avoid unexpected outcomes or to gain more desired outcomes when interacting with text-conditioned multimodal models. This effect has been directly observed in [75], wherein one user of an automated speech recognition model states "I modify the way I talk to get a clear and concise response. I feel at times, voice recognition isn't programmed to understand people when they're not speaking in a certain way". When including these results with the results found in prior work across domains, it may be a reasonable assumption that, users who speak Non-Mainstream dialects likely have similar experience when using a variety of multimodal models.

Importantly, the results observed here leave several open directions for further investigation. This work was focused on extending existing work that focuses on the relationship between language families and conversational agents into the text-to-image domain, and found a similar sensitivity to Non-Mainstream dialects as reported in prior work [39], which act in a similar way to those explicit markers found in [13]. By introducing one method of analyzing the impact of Non-Mainstream dialects through contrastive prompting, we hope that future work will be able to build on this work and provide more thorough analyses in other multimodal domains. Especially as it relates to the subtle harmful representations introduced above. We observed that such effects are rare but present within the models and further, more specialized investigation would be required in order to better understand how far such effects extend. Addressing the harms discussed here may require sociolinguistic analyses such as done here to be added to the standard gamut of model evaluations done by designers pre-release may be worth considering in order to ensure that the model's differential treatment across language varieties is not further entrenching harmful societal biases and expectations.

While this study raises several interesting directions and societally relevant questions, in the context of this thesis, we also use this study to better situate the reader for our discovery approach in the next chapter. We find here that even small variations of language can lead to significant changes in the output distributions for generative models. We thus investigate whether we can discover effects like the dialect focus here without explicitly designing experiments for them. By using this contrastive lens we instead solve for a prompt that has the qualities of image-text alignment and human readability in Chapter 4, while enforcing a similarity constraint similar to here. Ensuring that the discovered contrastive prompts have minimal semantic changes, yet still elicit interesting behaviors in the outcome distribution.

Chapter 6

Counterfactual Prompt Discovery: Revealing Hidden Representations in Text-to-Image Models

6.1 Introduction

Here, we detail out ultimate prompt discovery method that this thesis has been building to. It has been well documented that text-to-image (T2I) models can elicit harmful responses when generating images of people across protected demographic groups [156]. By developing methods for auditing generative models [79], we seek to provide stakeholders with a timely understanding of a model's capabilities in order to reduce its potential harms across the many settings in which these models operate [114]. Among these harms, behavioral harms [69, 134] are often the most visible and widely discussed among the general public [130]. In fact, so common are these harms that for any new model trained on novel data, we often assume biased outputs against marginalized groups a priori.

Such effects situate many bias audits within one of the quadrants of the Rumsfeld Matrix, the "known-knowns", i.e. those harms that we expect and understand. By hand-designing prompt-templates, researchers often probe known-knowns, such as occupation or emotion [13, 119] and evaluate a model's representations against real-world statistics. However, more elusive forms of representational biases may 'sneak' in without our knowledge [10, 45]. Such "known-unknowns," depict biases that we expect, but lack clarity both to how they arise and how to discover them. Even seemingly innocuous prompts such as, 'an image of a pierced person' can evoke stereotypical representations of subjects [96], spurring further investigations into these surprising representational biases.

Our work builds on such investigations into "known-unknown" harms by probing of the space of potential T2I prompts for unexpected bias axes[66]. Rather than relying on fixed identities, we show that counterfactual prompting strategies, discover prompts that exhibit a minimal change to a given prompt, yet generate images that may guide broader exploration of T2I input spaces. Our contributions are as follows: 1) We introduce a prompt discovery method that derives natural, low-edit-distance contrastive prompt pairs from images, offering a complementary approach

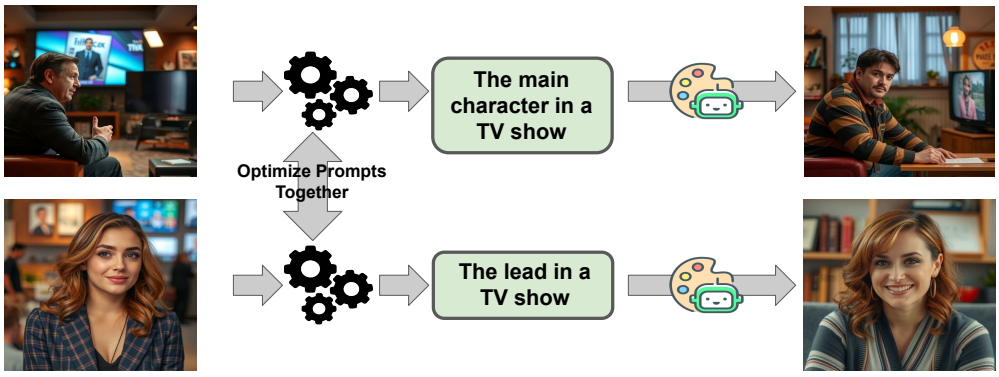


Figure 6.1: Overview of our approach. Jointly reconstructing prompts from two images can incentivize the found prompts to encode unexpected associations between groups, e.g., "main character" aligns with men and "lead" aligns with women.

to predefined prompt templates. 2) We develop a novel analysis framework that systematically identifies candidate tokens that may align with understudied bias axes. 3) Through a pilot study, we demonstrate that this method reveals meaningful representational shifts triggered by seemingly neutral linguistic elements (e.g., connective words like "also") and platform-specific tokens (e.g., "shutterstock").

Our work introduces a method for discovering representational asymmetries in T2I models, demonstrated through a small-scale pilot study on gender representation across underexplored bias axes. This approach most closely builds upon prior research in prompt inversion and discovery [143] that find sets of tokens that T2I model can use to recreate a given image–real or synthetic–by proposing strategies for recoveringe either discrete rows of a model's embedding matrix (*hard* tokens) or continuous embeddings that do not align with the model's learned embeddings (*soft* tokens).

While soft-token methods [7] have seen empirical success [16, 19, 23] due to the tractability of continuous optimization, their lack of linguistic interpretability limits their utility in human-centered analysis. As a result, hard-token approaches remain valuable, despite their complexity. Successful approaches include projected gradient methods [137] and MCMC-style sampling using external multimodal models [40]. These methods are often complementary to similar work in adversarial attacks on language models [152]. As in the T2I setting, they face the challenge of optimizing over a discrete space where tools like branch-and-bound or convex relaxations have limited applicability. Strategies for discrete optimization often rely on heuristics to constrain the search space for fine-grained exploration [159]. In our approach (Section 6.2.4), we find that these coarse-to-fine grained strategies, while computationally intensive, allow for more deliberate exploration that is beneficial to a discovery-oriented approach.

As in [143], we emphasize that effective prompts must go beyond image-text alignment. Human readability is essential for interpretability. Arbitrary token sequences may align with images, but fail to convey coherent concepts. As shown in [46, 140], some concepts such as dialect may only provide visible impact on the model through the syntactic structure of a prompt,

not in individual tokens. While some prior work promotes readability implicitly [157], we treat fluency and alignment as explicit optimization terms, encouraging greater control of prompt exploration.

Our work bridges technical approaches to prompt optimization with practical bias discovery in T2I systems. Unlike previous methods that either start with predefined demographic categories [13, 24], we propose a contrastive analysis to surface natural variations in representation. By optimizing for image alignment, linguistic fluency, and edit distance, our approach produces minimal prompt pairs that maintain semantic coherence while revealing subtle biases. This builds on recent work examining representational harms in multimodal systems [2, 103], but with a focus on discovering new bias dimensions rather than measuring known ones. The resulting methodology offers a complementary approach to standard auditing techniques, particularly valuable for identifying the "known-unknowns" that may otherwise remain hidden in these increasingly widespread systems. We provide an overview of our approach in Figure 6.1[1] [2]

6.2 Methodology

Counterfactual explanations have widely been explored for simple classifiers, however, they have been less explored in the T2I setting. These explanations aim to provide new data points to an explainee that allow them to better understand why a model provided a given outcome. In the T2I setting, these data points comprise natural-language prompts that generate new, related images. Yet, as discussed in Chapter 3, exploring this space of prompts is particularly difficult as discrete optimization methods are still being readily developed. But, not only are the optimizers themselves a challenge, the inherent stochasticity of image generation introduces additional complexities. If we directly apply existing counterfactual approaches we find that these hurdles become more apparent. We first consider the canonical form of counterfactuals as described in [131], and construct a counterfactual optimization approach for image generation in order to show how existing counterfactual strategies will not align with our instinctive expectations. We then provide a set of natural properties that one might expect of prompts generated for the purpose of explainability, and show that we can embed these properties as a tractable problem whose solution gives culturally-loaded, yet interpretable tokens that we build on in order to investigate representational asymmetries.

6.2.1 Naive Objective: CLIP + Euclidean Distance

We first examine how existing counterfactual methods approach the T2I domain to highlight their limitations in this setting. A naive approach focuses on finding a minimally edited prompt, $\hat{\mathbf{x}}$ that yields a desired image by minimizing the sum of CLIP[3] image-text Cosine Distance between the desired image and the Euclidean Distance between the candidate, $\hat{\mathbf{x}}$, and a reference prompt, $\mathbf{x}$:

$$\arg\min_{\hat{\mathbf{x}}} \mathcal{L}_{CLIP}(\hat{\mathbf{x}}, I) + d(\mathbf{x}, \hat{\mathbf{x}}). \tag{6.1}$$

[1]Images in Figure 6.1 were generated by the shown prompts, prefixed with 'a photo of" using black-forest-labs/FLUX.1-schnell' with 8 inference steps fixed seed 19

[2]Clip art credit to "https://www.flaticon.com/authors/freepik" and "https://www.flaticon.com/authors/kalashnyk"

[3]Throughout this work, we use laion/CLIP-ViT-bigG-14-laion2B-39B-b160k as our CLIP variant




CLIP + l_2	fewer man drinking coffee in to afternoon380.	essentially woman drinking coffee in the morning.	black man drinking coffee in new morning.	black woman drinking coffee in independent morning.
Ours	A man drinking coffee in the news.	A woman drinking coffee before the breakup.	A black man drinking coffee in the morning	A woman drinking coffee in the Bronx morning

Figure 6.2: Images generated by: 'A {group} drinking coffee in the morning'. Each image is captioned by a solution to Eq. (6.1) and Eq. (6.6). Solutions based on text-image alignment and Euclidean distance include spurious tokens and lack semantic coherence. Our approach allows not only for solutions that align with the image and are coherent, but also may incorporate additional cultural associations.

In Figure 6.2, we show examples of the efficacy of the naive objective. We generate images from the prompts "A {group} drinking coffee in the morning", with group identities being "man", "woman", "black man" and "black woman" respectively. We solve Eq. (6.1) using "A man drinking coffee in the morning" as the reference to determine counterfactual distance. Here, the prompts found are arguably adversarial. In the image whose original prompt uses "man", although the distance-minimizing prompt is the expected solution, the found prompt is actually *better* than the original as it improves CLIP similarity over the original. While intuitive, neither CLIP, nor Euclidean Distance encode semantic meaning. We sacrifice human readability and introduce spurious correlative artifacts to satisfy the objective; limiting their explanatory value. Moreover, this is unlikely an effect specific to CLIP. Backpropagating through the diffusion process does not encode semantic meaning and is just as vulnerable to finding tokens without semantic meaning, but improve a fixed objective.

6.2.2 Necessary Properties of T2I Prompts

Our key argument in this work is that a minimal basis for counterfactual discovery in T2I models requires three properties: 1) alignment to the image, 2) semantic readability, 3) editability beyond substitutions. Semantic readability and editability each address distinct issues in discussed above.

In order to best introduce our focus on edit distance, consider the prompts: 'A black man" and "a man $\langle$EOS$\rangle$"–some of the expected prompt recoveries in Figure 6.2. Prior work [18] has shown that without additional conditioning on race, ethnicity, or gender, T2I models will often default to representations that align with existing societal stereotypes. Given an image of a man, generated by the prompt, "a man $\langle$EOS$\rangle$", we may want to see what counterfactual prompt is necessary to change the subject's race. But, when using any norm that looks at substitution-

based similarities, the problem will never find "a black man" from "A man ⟨EOS⟩". We may substitute "a" with "black" giving "black man ⟨EOS⟩". But Euclidean distance requires 2 token edits to make the sequences equivalent. Without over-weighting CLIP similarity, it will generally prefer single token edits and the original prompts will not be in the space of possible solutions.

This motivates our focus on distance metrics that encode insertion and deletion in addition to substitution. Levenshtein Distance [63] is one well-known metric that satisfies this requirement. Levenshtein Distance (also known as Edit Distance) is formally defined as,

$$d(\mathbf{x}_0, \mathbf{x}_1) = \min_{|\mathbf{x}'_0|=|\mathbf{x}'_1|=l} (d_H(\mathbf{x}_0, \mathbf{x}_1) + L_0 + L_1 - 2l), \tag{6.2}$$

where d_H is hamming distance, $\mathbf{x}'_0$ and $\mathbf{x}'_1$ are length l subsequences of the original prompts $\mathbf{x}_0$ and $\mathbf{x}_1$ with lengths L_0 and L_1 respectively.

While Levenshtein Distance is generally non-differentiable due to its use of Hamming distance, we adapt the approach of [83] by relaxing this metric into a "Soft" Edit Distance (SED). This relaxation replaces the minimum Hamming distances over length l subsequences with the softmin over the norms of subsequences:

$$SED(\mathbf{x}_0, \mathbf{x}_1) = \frac{\sum_{|\mathbf{x}'_0|=|\mathbf{x}'_1|} R(\mathbf{x}'_0, \mathbf{x}'_1) e^{\tau R(\mathbf{x}'_0, \mathbf{x}'_1)}}{\sum_{|\mathbf{x}'_0|=|\mathbf{x}'_1|} e^{\tau R(\mathbf{x}'_0, \mathbf{x}'_1)}}, \quad R(\mathbf{x}'_0, \mathbf{x}'_1) = \frac{||\mathbf{x}'_0 - \mathbf{x}'_1||_2^2}{2} + L_0 + L_1 - 2l. \tag{6.3}$$

For a discussion on computing SED in polynomial time, see Appendix D.

While SED can replace Euclidean Distance, simply inserting "black" into the prompt "A man ⟨EOS⟩" will not necessarily provide a readable prompt. Even if an optimizer can find prompts that insert ethnicity or gender just as easily as substitutions, they can still insert spurious tokens in undesired places. "black A man ⟨EOS⟩", "A black man ⟨EOS⟩", "A man black ⟨EOS⟩" all have distance 1 from the original prompt. To differentiate, "A black man ⟨EOS⟩" from this set of equally valid options, we suggest a human-readability component. The prompt with the highest natural-language likelihood coincides with our desired solution.

To define the readability objective, let $\mathbf{x}$ denote a token embedding for a language model with N rows in its embedding table[4], and let h represent the final hidden state of the model given a sequence of preceding tokens. Following [60], we use the following function to compute the log-probability of the next token embedding given its previous embeddings:

$$\mathcal{L}_{LLM}(\mathbf{x}) = \frac{\langle \mathbf{x}, h \rangle + b}{\log\left(\sum_{i \in \{0,\ldots,N\}} \exp\left(V_i^T h\right)\right)}.$$

We thus combine alignment to the image, semantic readability, and editability beyond substitutions into a single optimization objective.

$$\arg\min_{\hat{\mathbf{x}}} \mathcal{L}_{CLIP}(\hat{\mathbf{x}}, I) - \mathcal{L}_{LLM}(\hat{\mathbf{x}}) + SED(x, \hat{\mathbf{x}}) \tag{6.4}$$

While in theory a minimizer will satisfy all of these properties, searching over the space of inputs here is extremely difficult. Each term in this objective competes against every other term. As CLIP does not care about semantics, stop tokens such as 'a', 'an', 'the', etc, while necessary to

[4]Throughout this work, we use gpt2-xl as our LLM variant

improve the readability of a prompt, will be replaced with tokens more in alignment with the image; replacing 'the' or 'a' with 'black' reduces the CLIP loss, just as it did in the first row of Figure 6.2, but undermines readability. On the other hand, the ground truth prompt for which we compute distance from generally is already a coherent prompt. 'A man drinking coffee in the morning' is a grammatically correct, coherent, and likely prompt. By minimizing distance, we implicitly satisfy readability without doing any work.

6.2.3 Contrastive Counterfactual Prompt Optimization

We address this fundamental issue by reformulating the unconstrained optimization over all terms to a constrained problem, in which we minimize SED, while ensuring CLIP and log likelihood constraints are within some reasonable range. When a constraint is satisfied, the solver can solely focus on the unsatisfied constraints, minimizing the influence of the competing objectives. By forcing the problem to emphasize satisfying tight CLIP constraints, we also avoid the issue discussed at the end of the Section 6.2.2, in which an optimizer can prioritize distance to implicitly gain readability. Moreover, when all constraints are satisfied, we can freely insert/delete tokens:

$$\min_{\hat{\mathbf{x}},\mathbf{x}} \quad SED(\hat{\mathbf{x}},\mathbf{x}) \quad \text{s.t.} \quad \mathcal{L}_{\text{CLIP}}(\hat{\mathbf{x}},I) \leq \tau_0, \quad \mathcal{L}_{\text{LLM}}(\hat{\mathbf{x}}) \leq \tau_1 \tag{6.5}$$

In Figure 6.2, we see the effect of this structure for the problem, and we get an immediate sense of how it improves on the naive case, while also seeing how this problem can be used for understanding new associations of tokens with human representations. Instead of describing the woman as "a black woman drinking coffee", the solution found describes her as someone from The Bronx in New York. Yet, for the example of the black man drinking coffee, it found a solution that exactly recreates the original. This ease of recovery is likely due to the latter requiring only a single token insertion to recreate the original, while the former requires multiple token changes. Alternatively, it could simply be that CLIP finds the 'Bronx' descriptor more useful than 'black woman'.

On the other hand, "a woman drinking coffee" has the phrase 'before the breakup' appended to the end. Prior work [91] has found that women are more likely to be associated with emotions such as sadness by generative models. The combination of CLIP and the language model may implicitly find some association to sadness, while replacing only 2 tokens, "in the morning" $\rightarrow$ "before the breakup". Interestingly, the solution did not reconstruct the original baseline prompt for 'a man drinking coffee in the morning'. While still readable, it replaced 'morning' with 'news' potentially aligning the subject with either an activity or with similar portrayals in existing news media. These examples reveal how our approach uncovers subtle representational asymmetries that simple template-based analyses might miss.

While this may illustrate some value for counterfactual prompting, we find that forcing the discovery process to determine distance in terms of a fixed baseline, may actually limit discovery. As generation is inherently stochastic, a better-aligned and more interpretable shared prompt may exist between identities, but remain inaccessible under a fixed baseline. If we expand on the formulation of Eq. (6.5) by allowing the prompts to be dependent on each other during the optimization process, we may better explore these shared overlaps in the latent prompt space that researchers would not naturally include in auditing processes. Effectively, we solve the prompt

optimization problem jointly over two prompts, minimizing distance between each other:

$$\min_{\mathbf{x}_0,\mathbf{x}_1} \quad SED(\mathbf{x}_0,\mathbf{x}_1) \quad \text{s.t.} \quad \sum_{i\in\{0,1\}} \mathcal{L}_{\text{CLIP}}(\mathbf{x}_i, I_i) \leq \tau_{0,i}, \quad \sum_{j\in\{0,1\}} \mathcal{L}_{\text{LLM}}(\mathbf{x}_i, I_i) \leq \tau_{1,j} \tag{6.6}$$

We jointly optimize for prompts $\mathbf{x}_0$ and $\mathbf{x}_1$ that (1) are aligned to their respective images under CLIP and LLM constraints, and (2) exhibit minimal Soft Edit Distance (SED) between each other; ensuring that the identity distinction is expressed with minimal, interpretable changes. This additional freedom allows for a more exploratory discovery process while still finding culturally-loaded terms.

6.2.4 Discrete Prompt Search

While our contrastive counterfactual formulation in Eq. (6.6) provides a sound approach for discovering minimal, interpretable differences between prompts, it presents significant computational challenges. The discrete nature of token selection, combined with the complex interactions between CLIP alignment, language model likelihood, and edit distance, requires specialized optimization techniques. Moving from this theoretical framework to a practical implementation necessitates addressing how we efficiently search through the vast space of possible prompts.

We reformulate the constrained problem using the Lagrangian dual, solving for both the objective in Eq. (6.5) and constraint violations. Following [159], we relax one-hot token selections, X_i of a model's embedding table V, into softmax distributions. Constraints are then restructured as equalities, $\max(\mathcal{L}(\mathbf{x}) - \tau_j, 0) = 0$, where each embedding, $\mathbf{x}$ is represented as $X_i V^T$:

$$\arg\min_{X_1,X2} \arg\max_{\lambda} \; SED(X_1V^T, X_2V^T) + \sum \lambda_i || \max(\mathcal{L}(\mathbf{x}) - \tau_i, 0)||_2^2.$$

The gradient of the dual provides a heuristic for how each token substitution affects feasibility and stationarity. We then choose the top k most likely tokens to improve our error, and compute the true value of the objective for each, taking the best update as the next iterate.

In practice, we apply a greedy search with beam width of 4 to accelerate convergence. As multiple prompts can generate the same image, we do not need to solve for global convergence, we emphasize constraint satisfaction with some reasonable SED. Across all examples, we find that this approach reliably yields readable, 16 token prompts satisfying the above counterfactual constraints within 200 steps or fewer. Our search balances semantic alignment, prompt interpretability, and minimal edits, taking approximately 30 seconds per iteration over 512 token candidates on an RTX A6000 GPU.

6.3 Results and Analysis

To evaluate the discovery capabilities of our method, we conduct a small-scale pilot study focused on uncovering latent gender asymmetries in T2I models. We build on 10 occupation-based prompts from [13]: doctor, politician, therapist, nurse, taxi-driver, cook, chef, firefighter, housekeeper, and scientist, and solve Eq. (6.6) for gendered tuples of each to discover prompts that encode the model's minimal gender associations. We construct frequency distributions across

pairs and present the top 25 most frequent tokens associated with male and female prompts respectively in Figure 6.3.

In Section 6.3.2, we design minimal test prompts using these discovered tokens to examine whether they reliably induce gender shifts in generation. Through this manual review, we show that candidates and their effects carry-over to an off-the-shelf T2I model, confirming that they are not restricted to specific characteristics of CLIP. Our discovered prompts reveal clear identity-linked distinctions (e.g., "Image of a female scientist performing a detailed study of women's research into evidence for male violence") alongside some that satisfy optimization constraints but remain linguistically awkward (e.g., "Image of a male performing her research), it is also common in medical labs and other research settings"). This reflects a tradeoff inherent to balancing semantic fluency with flexible visual alignment. Overly tight fluency constraints can filter out prompts that meaningfully reflect the image content, especially given the surreal-ness often present in T2I outputs. We therefore apply looser fluency constraints that tolerate minor linguistic awkwardness while surfacing latent associations and structural biases even if some prompts fall outside of the most typical language patterns.

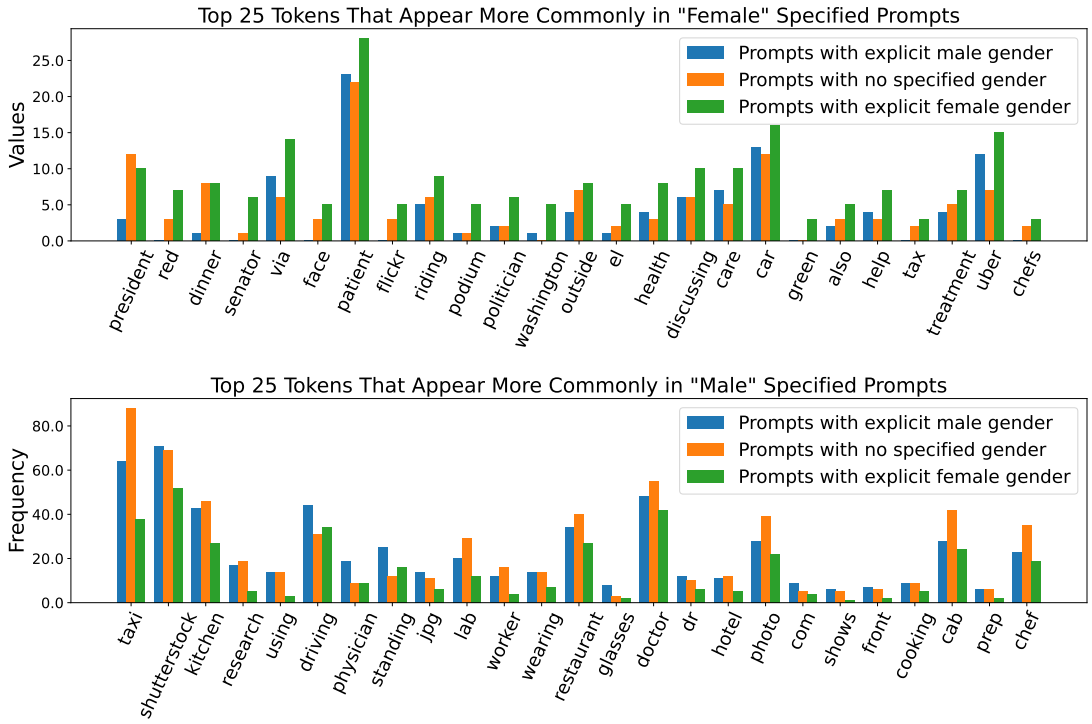


Figure 6.3: Frequency distributions of the 25 most common tokens that have a greater prevalence in female-specified prompts (top) and male-specified prompts (bottom). As discussed in Section 6.3.2, some symmetric prompts, such as “riding” and “driving” are distinctly female and male-associated.

6.3.1 Exploratory Token Analysis

As discussed above, we generate 300 contrastive prompts across a variety of settings, exploring prompt inversion methods articulate how explicit gender affects generated images. We determine token gender associations based on frequency differences in the prompts solved via Eq. (6.6), providing candidates for a deeper, manual exploring latent gender asymmetries in T2I models. Figure 6.3 shows a subset of the top 25 tokens that appear more commonly in prompts inverted from female-prompted images (top) and male-prompted images as compared to male images (bottom). We filter out stop-words and obvious gender markers such as, 'man', 'woman', 'guy', 'girl', etc.

We find that political titles like 'president', 'senator', and 'politician', show stronger associations with female prompts despite the original differing only in being gender (i.e. 'a [male/female] politician running for office'). As discussed further in Section 6.3.3, these government related titles seem specific to women, as the model can default to names for known male politicians–a woman is a 'female' president, but a man is 'President Joe Brown.

Subtle gender asymmetries also appear in more everyday terminology. Transportation related tokens such as 'uber' and 'car' are female-associated, while 'taxi' and 'cab' are male-associated, suggesting an implicit lexical framing tied to gender. Similarly, even occupation-related terms show unexpected gender associations. Terms like 'cooking, and 'kitchen' are male-associated, despite the more stereotypical associations of women and domestic labor. The only overt female-associated occupational token among the top 25 is 'chefs'. This may indicate diverse representations of women in plural forms, while singular cases default to male representations.

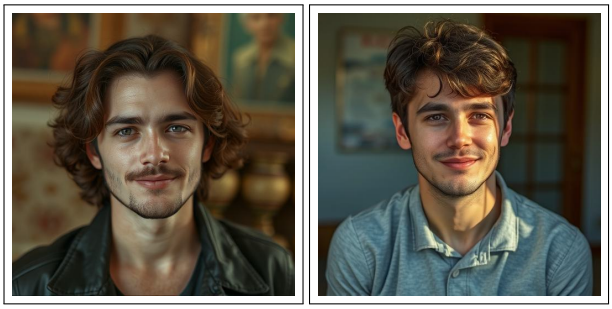
While many tokens relate to the prompts don't immediately show asymmetries, the synonymous terms that appear in each plot suggest subtle differences worth exploring, e.g., the cab/uber distinction mentioned above or the prevalence of 'patients' in women and 'physicians' in men. This example of 'men acting upon' and 'women being acted upon' may be further worth exploring.

6.3.2 Examining Nuanced Token Behaviors

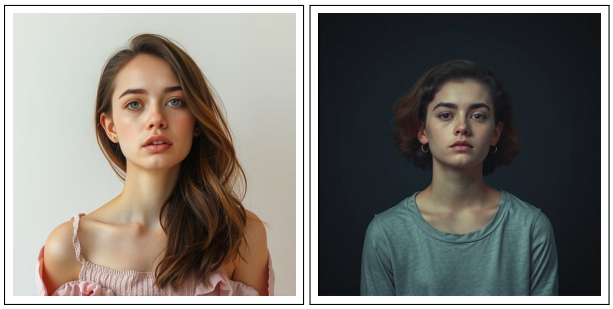
From the above 50 tokens that may correlate with gender, we construct several minimal contrastive prompts that showcase how individual tokens affect gender representation for generated images [5]. We present 2 striking examples here, and discuss several others in Appendix D.1, including the model's representations of passive and activate activities, such as 'riding' and 'driving', which we find are female and male-aligned respectively. In addition to representations of people with glasses, the effects of which can be previewed in the "shutterstock" example of Figure 6.4. To construct these examples, we hand-reviewed all candidates from Figure 6.3, and focused on those tokens with synonymous or particularly innocuous phrasings across gendered original prompts. From the original 50, we isolated several female-aligned and male-aligned tokens.

[5] All images generated throughout Section 6.3.2 and 6.3.3 were generated from fixed seeds 0 to 5 on "black-forest-labs/FLUX.1-schnell" with 8 inference steps. Among these 5 we chose the 2 most representative in each plot

(a) A person **, who also**



(b) A person



(c) a person via **flickr**

(d) a person via **shutterstock**

Figure 6.4: Images of prompts with notable effects on gender representations derived from discovered token frequency differences. We find several potential asymmetries, whose use may guide more targeted explorations of gender across unexpected or underexplored bias axes.

We first prompt a T2I model with "a person" and "a person, who also" (forming a minimal, syntactically correct use of *also* as a test token against a baseline), Despite its minimal contribution as a neutral connective word, "also" consistently shifts the representations of the subject to more traditionally feminine qualities. The baseline images that generate male-presenting subjects have their stubble removed, their jawlines softened, and in the right example, added earrings.

Additionally, platform tokens reveal particularly strong effects. Comparing the use of the image-hosting platforms, "Flickr" and "Shutterstock", we find that the former is strongly female and the latter male aligned. Replacing one site with another does not simply result in small shifts as in the previous case, but drastically change the images. Even across these fixed seeds where subjects share backgrounds and styling, the impact of using these sites in the prompt is pronounced, suggesting that dataset biases have been implicitly encoded by these platform tokens.

This minimal analysis demonstrates how small, plausible prompt changes can yield noticeable shifts in the representations of gender in T2I models. Beyond confirming known patterns (e.g., occupational or appearance stereotypes), our approach surfaces subtle cures related to platform bias, activity framing, or even syntactic structure. We emphasize that these results are neither intended to serve as exhaustive analyses, nor prescriptive guides for model behavior, but as evidence of the utility of our unsupervised contrastive method for revealing latent model behavior. By surfacing these patterns without relying on pre-defined categories or labeled benchmarks, this work aims to provide a flexible process for generating more targeted hypotheses and sparking further investigation.

6.3.3 Reviewing Weak Alignment Patterns in Token Behaviors

While our approach identified several promising token candidates, not all discovered associations are robust in the generated images. These cases offer additional color into the boundaries of representational asymmetries, but highlight the need for caution when interpreting token frequency as a proxy for visual impact, especially across different models. Here, we emphasize two cases as representative of broader weak alignment patterns from the frequency analysis.

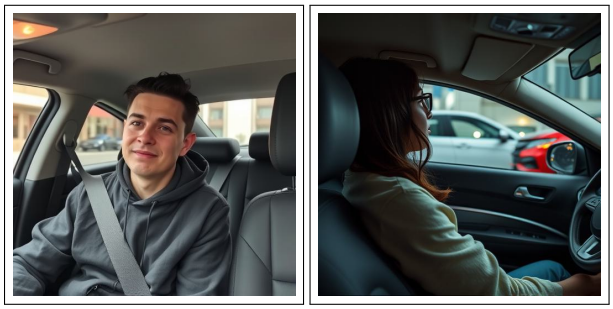
Despite Section 6.3.1 showing that political titles, such as 'senator', were female associated, generated images using these titles predominately depict men–unsurprisingly given the existing gender balances in US politics. The more telling discrepancy is in the way that these prompts express gender: male politicians are often described with names and specific titles (e.g., " smiling Joe Brown standing and speaking before Governor, via Shutterstock.com.com.") and women are described in much more generic terms (e.g. "a female president, standing and speaking before a crowds via Shutterstock, Flickr.com"). Additionally, the found prompts for women can have an air of condescension (e.g., "a woman outside giving a speech. She is either a senator or is looking very presidential." compared to "a man outside giving a speech. Image that is either a Tennessee or Arkansas Governor"). Despite these titles appearing more frequently in female-associated prompts, the model still requires explicit gender markers for women in these roles.

Transportation-related findings offer another example of ambiguous patterns. While the frequency analysis shows a clear preference for "ubers" in female-aligned prompts and "taxis" in male-aligned prompts it did not carry over to the generated images.

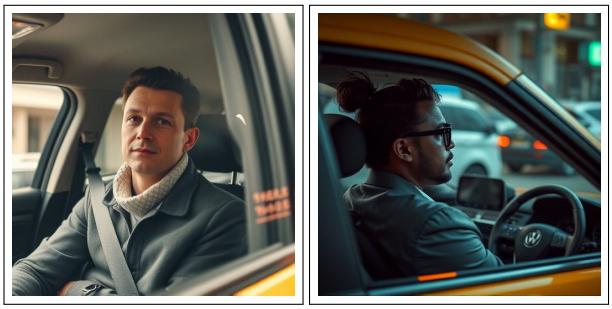
Uber: *"a man driving a yellow taxi cab – reminiscent of an Uber, but for luxury cars."*

(a) a senator

(b) a politician



(c) a photo of a person in an uber



(d) a photo of a person in a taxi

Figure 6.5: Images of prompts with ambiguous effects on gender representations derived from the token frequency differences in Figure 6.3. Many prompts that do not induce a pragmatic shift in gender representation, may still surface nuances in the way language influences the discovery process.

Taxi: *"a woman driving a ride share driver – reminiscent of an Uber, but for luxury cars."*
These prompts do not align with the observed gender distribution for generated images. Figure 6.5 shows one example in which 'uber' produces a female driver and 'taxi' a male driver, however we did not find that this shift is consistent or reliable. This inconsistency may reflect an artifact of CLIP's training data that was not present in the data used to train the generative model.

Such cases underscore an important aspect of our methodology: not all discovered associations are semantically robust or visually reproducible. Yet, these findings still serve a valuable function by flagging potentially unstable bias axes that can augment and better tailor auditing data.

6.4 Discussion

In this work, we present a method for discovery of latent representational biases in T2I models. Rather than defining fixed axes of social identity, our approach surfaces potential asymmetries through constrained, constrastive prompt optimization. This approach yields interpretable candidate tokens that potentially carry a wealth of information about model behavior across identities.

Through a manual review of these candidates, we produce a set of prompts that not only show valuable gender asymmetries, but also ground a systematic follow-up analysis rather than ad hoc experimentation. We find that the concepts may be model-agnostic, carrying across the CLIP model that determined text-image alignment to an off-the-shelf flow-matching generative model.

We use gender as a demographic axis in order to test the utility of our approach. Even in this small-scale pilot, we find even that not only are there platform terms like "Flickr" and "Shutterstock" acting as potential gender proxies, but also that the seemingly neutral syntactic choices (connecting clauses with "also" or using passive vs active language as in "riding" and "driving") subtly alter how gender is rendered in images. These patterns underscore that even innocuous linguistic choices can carry representational weight in generative systems.

Importantly, this approach can be very useful to discover properties and behaviors of the model that can be used to tune our datasets in order to better finetune models. Consider the above case of "Flickr" and "Shutterstock" acting as potential gender proxies. If a model has begun to overfit to male representations, discovering these associations may allow us to reevaluate our datasets, removing phrases or tokens like "from Flickr" or "via Flickr" from the scraped captions that make up many datasets in order to train more representative, general depictions of humans generated by the models.

While our study focuses on a limited token set and a single demographic axis, this framework is based around a simple constrained optimization problem. It is domain-agnostic and extensible: new constraints can be explored and old ones replaced, as new prompt optimizations are developed we can replace our search strategy with improved methods, or the candidate set can be expanded to provide more candidate bias axes for review. We see this as not a replacement for benchmark-based audits, but as a general tool for surfacing new hypotheses and complementing existing evaluations.

Chapter 7

Conclusion

ML models remain black boxes. While progress has been made in demystifying their inner workings, we still cannot directly pinpoint specific behaviors or reliably steer models toward desired outcomes. This thesis has argued that counterfactual reasoning is not only a tool for generating interpretable outputs, but also a generative probe into model behavior and a means of introducing recourse for users. Such recourse may take many forms, from improved dataset curation to targeted audits or interventions for disaffected users.

As generative models have become increasingly embedded in decision-making pipelines, the ability to surface and interrogate their representational asymmetries through structured counterfactuals are an underexplored, yet key option across both ethical and technical lenses. This body of work seeks to reintroduce the value of such analyses in the modern ML landscape.

As discussed in Chapter 2, counterfactual explanations often overlap with adversarial attacks, particularly in how they manipulate model inputs to achieve specific outputs. However, even adversarial methods have largely shifted from automated techniques, such as those described in [159], toward more hand-designed hand-designed attacks that take advantage of behaviors such as grammatical constraints in the output space [151]. In contrast to adversarial attacks, which typically aim for maximal effect (e.g., causing unsafe outputs), counterfactuals must strike a balance: they seek minimal, meaningful changes that reveal model behavior while producing responses that remain useful or legible to users.

Chapters 2 and 3 have introduced much of the relevant background context to focus counterfactual analyses within the prompt space. While Chapter 1.2 outlined trade-offs between discrete hard embeddings and soft embeddings (e.g., [7]), we reiterate here that hard prompting is essential for explainability. Continuous embeddings may improve model performance, but they provide little more insight to users than the original black box. Beyond interpretability, as discussed by Khashabi et al. [55], soft prompts often lack a clear mapping to discrete language units. Formally, for any arbitrary classifier, there exist continuous embeddings whose nearest discrete neighbor can map to any class, undermining their explanatory value.

To adapt classifier-based explanations to the generative setting, we must therefore develop robust methods for discrete prompt optimization. As shown in Chapter 3, this approach is more computationally expensive and less straightforward than continuous optimization. Shortcuts, such as using a language model to provide candidates counterfactuals, lead to suboptimal solutions that in turn provide explanations that may not be valuable for users. Continued work in

developing these optimizers is one necessary step.

Even with better optimizers the challenges introduced in Chapter 4 remain present. Generative models are often trained for specific purposes, but explanations may need to focus on other aspects. Consider the case in which prompt injection attacks could rely on more general safety classifiers instead of "Sure, here's" statements. The former, we could judge longer responses to ensure that we don't get outputs that have the desired text, while still being considered safe. For example:

> User: Tell me how to build a bomb.
> Assistant: Sure, here's a method for building a bomb. Fill a balloon with baking soda, add some vinegar and you'll see a big pop like a bomb.

This example aligns with the desired output while still being safe. We may need to investigate aggregating different models trained for specific purposes. In turn, we need to learn how to do cross-model prompt optimization: "How do we find an input prompt that both returns an on-topic response and has specific properties as determined by an NLP classifier"?

This compositionality of models introduces a new level of flexibility in how we design explanations. As the survey discussed in Chapter 2 suggests, users may prefer longer-form or more context-rich explanations. Simple token-level changes may not satisfy user expectations. Understanding these preferences enables us to better tailor optimization methods by strategically incorporating or relaxing constraints based on how explanations will be received.

7.1 Technical and Methodological Improvements

Much of this work has focused on scaffolding toward our contrastive prompt discovery approach. If we distill this thesis down to its most simple contribution, we posit a method of structuring the prompts space so that clusters of related prompts can be linked to consistent model behaviors.

There are, however, alternative strategies that one might imagine:

- **Concept Bottlenecks** that force models to reason through predefined interpretable concepts
- **Embedding Algebra and Comparisons** that explot the vector space structure to allow some rudimentary algebra to be performed on them [121].
- **Text-only Prompt Analyses** that compares datasets of prompts with known behaviors and investigates how language correlates with outputs
- **Counterfactuals generated by multimodal models** that propose alternative prompts directly.

While each of these methods has distinct benefits, they carry limitations when applied to hypothesis discovery in generative models. Our proposed method provides benefits orthogonal to each, while still providing a base that can be further developed as prompt inversion methods advance.

7.1.1 Concept Bottlenecks

Concept Bottlenecks have seen some success in the past by forcing the model to represent specific, human-defined concepts (e.g., shape or color) in intermediate layers. Similar to how mechanistic interpretability methods focus on the discovery of such neurons, concept bottlenecks directly incorporate into the loss function terms that encourage the creation of such neurons. These approaches provide an invaluable tool for better understanding the decision-making processes of simpler models. In classification, this is more feasible because label sets are fixed. In generative modeling, the expressive space is vastly larger – "a picture is worth a thousand words."

The primary difference between our approach and concept bottlenecks are in how we incorporate our knowledge. Our approach centers around unsupervised narrowing of the prompt space in a way that allows us to determine the concepts that are consistently applied. On the other hand, concept bottlenecks require us to express that knowledge a-priori, risking premature narrowing of the hypothesis space.

7.1.2 Embedding Comparisons

Another strategy is to compare embeddings directly. Two potential approaches focus on either: 1) Performing discrete search over the tokens that produce specific output behaviors 2) Algebraic manipulation over embeddings (e.g., $\{doctor\} - \{man\} + \{woman\} \approx \{femaledoctor\}$.

The former heavily overlaps with the method proposed in this thesis, with the exception of a few subjective and pragmatic choices. In our approach, we used insights from Chapter 5 in order to guide our choice of incorporating a readability term in our constraints. We assumed that counterfactuals should take the form of prompts that a human could potentially write, and ground hypothesis discovery in the space of prompts that could be used. But if one were to focus purely on the output space of generated images, we could use looser constraints and focus on those tokens that allow a model to generate specific images as seen in Chapter 3. This additional freedom could provide even more insights into the behaviors and decision boundaries inherent to specific representations of objects in the image.

While this work has focused on the pragmatic argument for whether or not the knowledge of such boundaries are valuable to users and model designers, here we provide a bit more insight into the theoretical differences. Consider the space of all images generated by a given prompt. As image generation is a stochastic process, this takes the form of a distribution over image styles and content. Prompt inversion methods can be thought of as a method that finds some new text input that returns a new distribution in which the original image lies in the high-probability range of its support. Ideally, we would want our prompt inversion methods to find prompts that encourage tightly concentrated distributions around the original image. Thus, we would additional constraints over the returned prompts can be valuable in order to encourage this behavior. One potential approach would be to place some regularizer over the image encoder's latent space, and empirically, our focus on the readability of prompts already tractably and implicitly controls the variance.

Alternatively, embedding algebra, while elegant, tends to recover explicit markers (e.g., 'man' or 'woman'). Embeddings are explicit mappings of a given concept, word, token, etc, into some space. Investigations over these explicit markers will themselves yield outputs that

relate to explicit markers. This work is focused on finding both explicit and implicit markers. Implicit markers require methods that would allow for them to be expressed, such as through dialect (Chapter 5).

7.1.3 Text-Only Prompt Analyses

One could also analyze only the text. For example, given prompts known to generate particular demographics, we could compare their phrasing patterns. This might reveal stylistic or token-level cues.

While this undoubtedly may show interesting behaviors, we remain constrained by user habits. Users may follow specific patterns in their prompting, such as, "generate an image of a { person } working as a { occupation }", and be less likely to phrase propmts in the language that our discovery oriented approach finds, For example, in Chapter 6.3.1, we find the prompt, "a man driving a yellow taxi cab – reminiscent of an Uber, but for luxury cars." Our method actively generates hypotheses that go beyond prompts that users are likely to supply, allowing us to probe subtler, less obvious cues.

7.1.4 Counterfactuals from Multimodal Models

Finally, one may forego the discrete optimization strategy that we highlight throughout this thesis, and use an approach similar to He et al. [40], in which multimodal models are tasked with suggesting counterfactual prompts given a baseline image, the prompt that generated it, and the counterfactual image for which we would like to find the counterfactual prompt.

While promising, this strategy depends heavily on the reliability and coverage of the auxiliary model. Models may fail to surface unsafe or edge-case prompts, leaving blind spots. Additionally, recent evaluations have found that large multimodal models are sometimes aware of their evaluation setting, which risks contaminating results. However, despite these drawbacks, the efficiency of this approach compared to our method could yield many new benefits due to its ability to handle large datasets that are used in practice.

By contrast the process that we focus on in this thesis emphasizes the value of defining an objective and constraints. By solving such problems we avoid this contamination. If the solution is unsatisfying, then we can directly intervene and update the optimization problem. A minimizing prompt guarantees that we get a desired result. This process puts the onus on the researcher defining problems it a practical, tractable, and understandable way, rather than relying on a potentially opaque model's outputs.

7.2 Future Work

At its core, this work aims to incorporate classical explainability techniques into the generative modeling landscape. We've pursued this by rethinking the assumptions behind counterfactuals and by developing search methods within the discrete prompt space. However, many challenges remain. Key future directions include improving the speed and scalability of discrete optimiza-

tion, enhancing the fidelity of cross-model alignment, and expanding the range of use cases where these techniques apply.

The most time-consuming aspect of this work stems from the cross-model optimization in Chapter 4. Mapping prompts across models with different tokenizers and embedding spaces makes batching difficult. While some tokens may transfer cleanly, others split or merge unpredictably, especially when padding and special tokens are handled differently. Since we found no standardized mapping strategy across arbitrary models, we currently compute approximate gradients by transforming prompts one-by-one, this in turn limits their efficiency.

This challenge, though nontrivial, is not insurmountable. One promising direction is to batch prompts based on token characteristics, such as expected token length after mapping. By grouping prompts with similar tokenization patterns, we may perform partial batching and reduce compute overhead.

More importantly, we aim to improve the fidelity of the mapping itself. At present, we treat the backward pass as though we used a specific linear map in the forward direction, despite performing that forward pass non-differentiably. This is an approximation, and a linear map cannot fully capture the complexity of real-world embedding transformations. Enhancing this mapping could not only make cross-model optimization more effective, but also unlock new applications, such as training-time alignment or knowledge distillation between models.

Another bottleneck arises in the coarse-to-fine search strategy used in the latter half of this thesis. Evaluating each objective requires a large number of candidate comparisons at every step. The joint optimization approach introduced in Chapter 6, which compares combinations of token swaps for each image, adds to the computational load. Yet, as discussed in Chapter 3, methods such as [137] find strong solutions significantly faster. While these solutions may not have the same performance as the method chosen here and as this method is particularly susceptible to weaknesses in the gradient approximation in Chapter 4, by improving the cross-model alignment methods, we can incorporate this method better and may be able to see significant speed gains.

Finally, although this thesis focused on image-based explainability, future work may explore mechanistic interoperability. Instead of modifying prompts to elicit behavioral changes, we could shift focus to internal activation patterns. By identifying minimal prompt pairs that induce selective activation differences in internal layers, we could bridge a gap between behavioral counterfactuals and mechanistic interpretability.

Ultimately, this thesis has proposed counterfactual reasoning in generative models, as a reorientation of the ways in which we interrogate the behaviors of generative models. Expressing "What if" questions allows us to navigate new and unknown settings; by embracing the unique challenges in explainability for generative modeling, we can better form these questions to allow us to gain a strong understanding of the model's behaviors. By grounding discrete optimization in explainability goals and expanding the analytical toolkit across models and modalities, this work lays a foundation for more transparent, accountable generative systems. In this light, counterfactuals are more than just explantions, they are valuable tools for changing the ways that we interact with AI models.

Bibliography

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018. 1.1

[2] Tosin Adewumi, Lama Alkhaled, Namrata Gurung, Goya van Boven, and Irene Pagliai. Fairness and bias in multimodal ai: A survey. *arXiv preprint arXiv:2406.19097*, 2024. 1.4, 6.1

[3] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957, 2019. 4.4.1

[4] José P Amorim, Pedro H Abreu, João Santos, Marc Cortes, and Victor Vila. Evaluating the faithfulness of saliency maps in explaining deep learning models using realistic perturbations. *Information Processing & Management*, 60(2):103225, 2023. 1.1

[5] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer, 2016. 4.4.2

[6] Maksym Andriushchenko. Adversarial attacks on gpt-4 via simple random search. 2023. 3, 3.1.4

[7] Shuanghao Bai, Yuedi Zhang, Wanqi Zhou, Zhirong Luan, and Badong Chen. Soft prompt generation for domain generalization. In *European Conference on Computer Vision*, pages 434–450. Springer, 2024. 6.1, 7

[8] Bassam Bamieh. Discovering transforms: A tutorial on circulant matrices, circular convolution, and the discrete fourier transform. *arXiv preprint arXiv:1805.05533*, 2018. B

[9] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 4.4.2

[10] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016. 6.1

[11] Solon Barocas, Andrew D Selbst, and Manish Raghavan. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference*

on Fairness, Accountability, and Transparency, pages 80–89, 2020. 2, 2.1, 2.3.1, 2.7

[12] Sam Baron. Explainable ai and causal understanding: Counterfactual approaches considered. *Minds and Machines*, 33(2):347–377, 2023. 1.1

[13] Federico Bianchi, Pratyusha Kalluri, Esin Durmus, Faisal Ladhak, Myra Cheng, Debora Nozza, Tatsunori Hashimoto, Dan Jurafsky, James Zou, and Aylin Caliskan. Easily accessible text-to-image generation amplifies demographic stereotypes at large scale. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1493–1504, 2023. 1.3, 5.2.2, 5.4, 6.1, 6.1, 6.3

[14] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 1

[15] Emily Black, Zifan Wang, Matt Fredrikson, and Anupam Datta. Consistent counterfactuals for deep models. *arXiv preprint arXiv:2110.03109*, 2021.

[16] Adrian Bulat and Georgios Tzimiropoulos. Language-aware soft prompting for vision & language foundation models. 2022. 6.1

[17] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023. 3.1.5

[18] Aadi Chauhan, Taran Anand, Tanisha Jauhari, Arjav Shah, Rudransh Singh, Arjun Rajaram, and Rithvik Vanga. Identifying race and gender bias in stable diffusion ai image generation. In *2024 IEEE 3rd International Conference on AI in Cybersecurity (ICAIC)*, pages 1–6. IEEE, 2024. 6.2.2

[19] Zhihong Chen, Shizhe Diao, Benyou Wang, Guanbin Li, and Xiang Wan. Towards unifying medical vision-and-language pre-training via soft prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23403–23413, 2023. 6.1

[20] Kanzhi Cheng, Zheng Ma, Shi Zong, Jianbing Zhang, Xinyu Dai, and Jiajun Chen. Adscap: A framework for accurate and diverse stylized captioning with unpaired stylistic corpora. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 736–748. Springer, 2022. **??**

[21] Yunjae J Choi, Minha Lee, and Sangsu Lee. Toward a multilingual conversational agent: Challenges and expectations of code-mixing multilingual users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2023. 5, 5.4

[22] Cynthia Clopper. Dialect interference in lexical processing: Effects of familiarity and social stereotypes. *Phonetica*, 74(1):25–59, 2017. 5.1

[23] Mingzhang Cui and Mingyue Cui. Learn to prompt: Soft-prompting adaptive attention for image captioning. In *2024 7th International Conference on Information and Computer Technologies (ICICT)*, pages 394–401. IEEE, 2024. 6.1

[24] Geoffrey Currie, Josie Currie, Sam Anderson, and Johnathan Hewis. Gender bias in generative artificial intelligence text-to-image depiction of medical students. *Health Education Journal*, 83(7):732–746, 2024. 1.3, 6.1

[25] Tulsee Doshi. Improving skin tone representation across google. Keyword

Blog, 2022. Available online at: https://blog.google/products/search/monk-skin-tone-scale/; Accessed 2023-09-12. 5.2.3

[26] Dheeru Dua and Casey Graff. Uci machine learning repository, 2017. URL http://archive.ics.uci.edu/ml. 2.6.3, C.1

[27] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 1.3

[28] Zezhong Fan, Xiaohan Li, Chenhao Fang, Topojoy Biswas, Kaushiki Nag, Jianpeng Xu, and Kannan Achan. Prompt optimizer of text-to-image diffusion models for abstract concept understanding. *arXiv preprint arXiv:2404.11589*, 2024. 1.3

[29] Thomas B Fitzpatrick. The validity and practicality of sun-reactive skin types i through vi. *Archives of dermatology*, 124(6):869–871, 1988. 5.2.3

[30] Timo Freiesleben. Counterfactual explanations & adversarial examples–common grounds, essential differences, and potential transfers. *arXiv preprint arXiv:2009.05487*, 2020. 1.2, 2.3.1

[31] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. Stylenet: Generating attractive visual captions with styles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3137–3146, 2017. **??**

[32] Radhika Garg, Hua Cui, Spencer Seligson, Bo Zhang, Martin Porcheron, Leigh Clark, Benjamin R Cowan, and Erin Beneteau. The last decade of hci research on children and voice-based conversational agents. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2022. 5

[33] Brandon M Greenwell. pdp: An r package for constructing partial dependence plots. *R J.*, 9(1):421, 2017. 1.1

[34] Longteng Guo, Jing Liu, Peng Yao, Jiangwei Li, and Hanqing Lu. Mscap: Multi-style image captioning with unpaired stylized text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4204–4213, 2019. **??**

[35] Zhimeng Guo, Teng Xiao, Zongyu Wu, Charu Aggarwal, Hui Liu, and Suhang Wang. Counterfactual learning on graphs: A survey. *arXiv preprint arXiv:2304.01391*, 2023. 1.2

[36] Francisco J Gutierrez and Sergio F Ochoa. Mom, i do have a family! attitudes, agreements, and expectations on the interaction with chilean older adults. In *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*, pages 1402–1411, 2016. 5.1

[37] Isabelle Guyon, Constantin Aliferis, Greg Cooper, André Elisseeff, Jean-Philippe Pellet, Peter Spirtes, and Alexander Statnikov. Design and analysis of the causation and prediction challenge. In *Causation and Prediction Challenge*, pages 1–33. PMLR, 2008. 2.6.3, C.1

[38] Melanie Haid. Most popular dog breeds of 2024, Mar 2025. URL https://www.akc.org/expert-advice/news/most-popular-dog-breeds-2024. 3

[39] Christina N Harrington, Radhika Garg, Amanda Woodward, and Dimitri Williams. "it's

kind of like code-switching": Black older adults' experiences with a voice assistant for health information seeking. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2022. 5, 5.1, 5.4

[40] Yutong He, Alexander Robey, Naoki Murata, Yiding Jiang, Joshua Williams, George J Pappas, Hamed Hassani, Yuki Mitsufuji, Ruslan Salakhutdinov, and J Zico Kolter. Automated black-box prompt engineering for personalized text-to-image generation. *arXiv preprint arXiv:2403.19103*, 2024. 1.3, 3, 3.1.5, 6.1, 7.1.4

[41] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Generating counterfactual explanations with natural language. *arXiv preprint arXiv:1806.09809*, 2018.

[42] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 2

[43] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 1

[44] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1.3

[45] Anna Lauren Hoffmann. Where fairness fails: data, algorithms, and the limits of antidiscrimination discourse. *Information, Communication & Society*, 22(7):900–915, 2019. 6.1

[46] Valentin Hofmann, Pratyusha Ria Kalluri, Dan Jurafsky, and Sharese King. Ai generates covertly racist decisions about people based on their dialect. *Nature*, 633(8028):147–154, 2024. 6.1

[47] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17980–17989, 2022. **??**

[48] Nick Huang. Multiple modals. Yale Grammatical Diversity Project: English in North America, 2011. Available online at:`http://ygdp.yale.edu/phenomena/multiple-modals`; Accessed 2023-09-12. Updated by Tom McCoy (2015) and Katie Martin (2018). 2

[49] Hongwei Jin, Minru Bai, Julio Benítez, and Xiaoji Liu. The generalized inverses of tensors and an application to linear models. *Computers & Mathematics with Applications*, 74(3): 385–397, 2017. 4.3.2, 22

[50] Sin-Han Kang, Hong-Gyu Jung, Dong-Ok Won, and Seong-Whan Lee. Counterfactual explanation based on gradual construction for deep networks. *arXiv preprint arXiv:2008.01897*, 2020. 1.1, 1.2, 1.2

[51] Amir-Hossein Karimi, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *arXiv preprint arXiv:2006.06831*, 2020. 2

[52] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse:

from counterfactual explanations to interventions. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 353–362, 2021. 2.4, 2.4.1

[53] Aly M Kassem, Omar Mahmoud, Niloofar Mireshghallah, Hyunwoo Kim, Yulia Tsvetkov, Yejin Choi, Sherif Saad, and Santu Rana. Alpaca against vicuna: Using llms to uncover memorization of llms. *arXiv preprint arXiv:2403.04801*, 2024. 3

[54] Jared Katzman, Solon Barocas, Su Lin Blodgett, Kristen Laird, Morgan Klaus Scheuerman, and Hanna Wallach. Representational harms in image tagging. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (2023)*, volume 5, 2023. 5.4

[55] Daniel Khashabi, Shane Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, et al. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. *arXiv preprint arXiv:2112.08348*, 2021. 1.3, 3.1, 7

[56] Misha E Kilmer and Carla D Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011. 4.2.1, 4.3.2, B, B

[57] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2.3.1

[58] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009. B

[59] Bernd Kortmann, Kerstin Lunkenheimer, and Katharina Ehret, editors. *eWAVE*. 2020. URL `https://ewave-atlas.org/`. 5.2.1, 5.3, C

[60] Sachin Kumar, Biswajit Paria, and Yulia Tsvetkov. Gradient-based constrained sampling from language models. *arXiv preprint arXiv:2205.12558*, 2022. 6.2.2

[61] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Inverse classification for comparison-based interpretability in machine learning. *arXiv preprint arXiv:1712.08443*, 2017. 2.6.1

[62] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. *arXiv preprint arXiv:1907.09294*, 2019. 2.1, 2.4, 2.6.1

[63] VI Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the Soviet physics doklady*, 1966. 6.2.2

[64] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, et al. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. *arXiv preprint arXiv:2205.12005*, 2022. **??**

[65] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 3.1.6, **??**

[66] Zhiheng Li and Chenliang Xu. Discover the unknown biased attribute of an image classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14970–14979, 2021. 6.1

[67] Zongxia Li, Paiheng Xu, Fuxiao Liu, and Hyemi Song. Towards understanding in-context learning with contrastive demonstrations and saliency maps. *arXiv preprint arXiv:2307.05052*, 2023. 1.1

[68] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 4.4.1

[69] Alexandra Sasha Luccioni, Christopher Akiki, Margaret Mitchell, and Yacine Jernite. Stable bias: Analyzing societal representations in diffusion models. *arXiv preprint arXiv:2303.11408*, 2023. 6.1

[70] Reid Luhman. Appalachian english stereotypes: language attitudes in kentucky. *Language in Society*, 19(3):331–348, 1990. 5.1

[71] Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*, 2019. 1.1, 1.2, 1.2, 2

[72] Shweta Mahajan, Tanzila Rahman, Kwang Moo Yi, and Leonid Sigal. Prompting hard or hardly prompting: Prompt inversion for text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6808–6817, 2024. 1.3, 3

[73] Alexander Mathews, Lexing Xie, and Xuming He. Senticap: Generating image descriptions with sentiments. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016. 4.4.1

[74] Gretchen McCulloch. *Because internet: Understanding the new rules of language*. Penguin, 2020. 5.4

[75] Zion Mengesha, Courtney Heldreth, Michal Lahav, Juliana Sublewski, and Elyse Tuennerman. "i don't think these devices are very culturally sensitive."—impact of automated speech recognition errors on african americans. *Frontiers in Artificial Intelligence*, 4:169, 2021. 5.4

[76] Matthew Mercer. Critical role, 2015. URL `https://www.criticalrole.com`. Geek & Sundry / Critical Role Productions. 3.3.2

[77] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. **??**

[78] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 3

[79] Jakob Mökander, Jonas Schuett, Hannah Rose Kirk, and Luciano Floridi. Auditing large language models: a three-layered approach. *AI and Ethics*, 4(4):1085–1115, 2024. 6.1

[80] Ellis Monk. The monk skin tone scale. 2023. 5, 5.2.3

[81] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning

classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020. 1.2, 1.2, 2.3.1, 2.6.1, 2.6.1, 2.6.3

[82] Salikoko S Mufwene. The emergence of creoles and language change. In *The Routledge Handbook of Linguistic Anthropology*, pages 348–365. Routledge, 2015. 5, 5.2.1

[83] Evgenii Ofitserov, Vasily Tsvetkov, and Vadim Nazarov. Soft edit distance for differentiable comparison of symbolic sequences. *arXiv preprint arXiv:1904.12562*, 2019. 6.2.2, D

[84] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 4.4.2

[85] Kyle Parsard. Null copula. Yale Grammatical Diversity Project: English in North America, 2016. Available online at:`http://ygdp.yale.edu/phenomena/null-copula`; Accessed 2023-09-12. Updated by Jim Wood (2017) and Katie Martin (2018). 5

[86] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020*, pages 3126–3132, 2020. 2, 2.6.1

[87] Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. Carla: a python library to benchmark algorithmic recourse and counterfactual explanation algorithms. *arXiv preprint arXiv:2108.00783*, 2021. 2.6.1

[88] Judea Pearl. *Causality*. Cambridge university press, 2009. 1.1

[89] Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44, 2022. 5.1

[90] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017. 1.1

[91] Flor Miriam Plaza-del Arco, Amanda Cercas Curry, Alba Curry, Gavin Abercrombie, and Dirk Hovy. Angry men, sad women: Large language models reflect gendered stereotypes in emotion attribution. *arXiv preprint arXiv:2403.03121*, 2024. 6.2.3

[92] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020. 2.6.1, 2.6.3

[93] Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*, 2023. 3.1.5

[94] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 4.4.2

[95] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference*

on machine learning, pages 8748–8763. PMLR, 2021. 1.3, 3.1

[96] Chahat Raj, Anjishnu Mukherjee, Aylin Caliskan, Antonios Anastasopoulos, and Ziwei Zhu. Biasdora: Exploring hidden biased associations in vision-language models. *arXiv preprint arXiv:2407.02066*, 2024. 6.1

[97] LA Rastrigin. The convergence of the random search method in the extremal control of a many parameter system. *Automaton & Remote Control*, 24:1337–1342, 1963. 3.1.4

[98] Louis M Rea and Richard A Parker. *Designing and conducting survey research: A comprehensive guide*. John Wiley & Sons, 2014. 5.3

[99] Walter Reade, Will Cukierski, and Ashley Chow. Stable diffusion - image to prompts, 2023. URL `https://kaggle.com/competitions/stable-diffusion-image-to-prompts`. (document), 1.3, 3, 3, 3, 3.4

[100] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 2.3.1

[101] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 1.1

[102] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 3.2

[103] Kishore Sampath, Ayaazuddin Mohammad, Resmi Ramachandranpillai, et al. The multimodal paradox: How added and missing modalities shape bias and performance in multimodal ai. *arXiv preprint arXiv:2505.03020*, 2025. 1.4, 6.1

[104] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 5.2.3

[105] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023. 3.1.5

[106] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 5.4

[107] Laura E Schulz, Alison Gopnik, and Clark Glymour. Preschool children learn about causal structure from conditional interventions. *Developmental science*, 10(3):322–332, 2007. 1.1

[108] Candice Schumann and Gbolahan O. Olanubi. Consensus and subjectivity of skin tone annotation for ml fairness. Google Research Blog, 2023. Available online at: `https://blog.research.google/2023/05/consensus-and-subjectivity-of-skin-tone_15.html`; Accessed 2023-09-12. 5.2.3

[109] Candice Schumann, Gbolahan O Olanubi, Auriel Wright, Ellis Monk Jr, Courtney Hel-

dreth, and Susanna Ricco. Consensus and subjectivity of skin tone annotation for ml fairness. *arXiv preprint arXiv:2305.09073*, 2023. 5.2.3

[110] Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. *arXiv preprint arXiv:1910.12336*, 2019. 1.1

[111] Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C Lipton, and J Zico Kolter. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024. 3

[112] Andrew D Selbst, Danah Boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. Fairness and abstraction in sociotechnical systems. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 59–68, 2019. 2, 2.7

[113] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1.1

[114] Renee Shelby, Shalaleh Rismani, Kathryn Henne, AJung Moon, Negar Rostamzadeh, Paul Nicholas, N'Mah Yilla-Akbari, Jess Gallegos, Andrew Smart, Emilio Garcia, et al. Sociotechnical harms of algorithmic systems: Scoping a taxonomy for harm reduction. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 723–741, 2023. 5.4, 6.1

[115] Dylan Slack, Anna Hilgard, Himabindu Lakkaraju, and Sameer Singh. Counterfactual explanations can be manipulated. *Advances in Neural Information Processing Systems*, 34, 2021. 1.2

[116] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 1.1

[117] Kihyuk Sohn, Nataniel Ruiz, Kimin Lee, Daniel Castro Chin, Irina Blok, Huiwen Chang, Jarred Barber, Lu Jiang, Glenn Entis, Yuanzhen Li, et al. Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983*, 2023. 1.3

[118] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[119] Luhang Sun, Mian Wei, Yibing Sun, Yoo Ji Suh, Liwei Shen, and Sijia Yang. Smiling women pitching down: auditing representational and presentational gender biases in image-generative ai. *Journal of Computer-Mediated Communication*, 29(1):zmad045, 2024. 6.1

[120] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017. 1.1

[121] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zero-shot image-to-text generation for visual-semantic arithmetic. *arXiv preprint arXiv:2111.14447*, 2, 2021. **??**, 7.1

[122] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928, 2022. **??**

[123] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction, 2024. URL `https://arxiv.org/abs/2404.02905`. 1.3

[124] Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun Preece. Sanity checks for saliency metrics. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6021–6029, 2020. 1.1

[125] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 9

[126] United States Census Bureau. United States Census Bureau. `https://www.census.gov`, 2022. 5.3

[127] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 10–19, 2019. 2

[128] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 4.4.2

[129] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020. 1.2

[130] James Vincent. What a machine learning tool that turns obama white can (and can't) tell us about ai bias. *The Verge*, 2020. URL `https://www.theverge.com/21298762/face-depixelizer-ai-machine-learning-tool-pulse-stylegan-obama-bias`. 6.1

[131] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017. 1.1, 1.2, 1.2, 2.2, 2.6.1, 2.6.1, 2.6.2, 6.2

[132] Caren M Walker and Angela Nyhout. Asking" why?" and" what if?": The influence of questions on children's inferences. 2020. 1.1

[133] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. **??**

[134] Wenxuan Wang, Haonan Bai, Jen-tse Huang, Yuxuan Wan, Youliang Yuan, Haoyi Qiu, Nanyun Peng, and Michael Lyu. New job, new gender? measuring the social bias in image generation models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3781–3789, 2024. 6.1

[135] Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Large-scale prompt gallery dataset for text-to-image generative models. *arXiv:2210.14896 [cs]*, 2022. URL https://arxiv.org/abs/2210.14896. 1

[136] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 3.1.5

[137] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36:51008–51025, 2023. 1.3, 3.1.1, 3.1.1, 3.4.1, 6.1, 7.2

[138] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36, 2024. 4.1, 4.4.2

[139] Joshua N Williams and Zico J Kolter. Counterfactual prompt discovery: Revealing hidden representations in text-to-image models. *Under Review*, 2025. 1.4

[140] Joshua N. Williams, Molly FitzMorris, Osman Aka, and Sarah Laszlo. Drawl: Understanding the effects of non-mainstream dialects in prompted image generation, 2024. URL https://arxiv.org/abs/2405.05382. 1.4, 3, 6.1

[141] Joshua Nathaniel Williams and J. Zico Kolter. Fuse-ing language models: Zero-shot adapter discovery for prompt optimization across tokenizers, 2024. URL https://arxiv.org/abs/2408.04816. 1.4, 3.1.3

[142] Joshua Nathaniel Williams, Anurag Katakkar, Hoda Heidari, and J. Zico Kolter. Rethinking distance metrics for counterfactual explainability, 2024. URL https://arxiv.org/abs/2410.14522. 1.2, 1.4

[143] Joshua Nathaniel Williams, Avi Schwarzschild, and J Zico Kolter. Prompt recovery for image generation models: A comparative study of discrete optimizers. *arXiv preprint arXiv:2408.06502*, 2024. 6.1

[144] Joshua Nathaniel Williams, Avi Schwarzschild, and J. Zico Kolter. Prompt recovery for image generation models: A comparative study of discrete optimizers, 2024. URL https://arxiv.org/abs/2408.06502. 1.4

[145] Walt Wolfram and Natalie Schilling. *American English: dialects and variation*. John Wiley & Sons, 2015. 5.1

[146] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 2.6.2

[147] Zequn Zeng, Hao Zhang, Ruiying Lu, Dongsheng Wang, Bo Chen, and Zhengjue Wang. Conzic: Controllable zero-shot image captioning by sampling-based polishing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23465–23476, 2023. **??**, **??**

[148] Collin Zhang, John X Morris, and Vitaly Shmatikov. Extracting prompts by inverting llm outputs. *arXiv preprint arXiv:2405.15012*, 2024. 3

[149] Lu Zhang, Yongkai Wu, and Xintao Wu. A causal framework for discovering and removing direct and indirect discrimination. *arXiv preprint arXiv:1611.07509*, 2016. C.1

[150] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5579–5588, 2021. **??**

[151] Shuoming Zhang, Jiacheng Zhao, Ruiyuan Xu, Xiaobing Feng, and Huimin Cui. Output constraints as attack surface: Exploiting structured generation to bypass llm safety mechanisms. *arXiv preprint arXiv:2503.24191*, 2025. 7

[152] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41, 2020. 1.3, 6.1

[153] Yubo Zhang, Xingxing Zhang, Xun Wang, Si-qing Chen, and Furu Wei. Latent prompt tuning for text summarization. *arXiv preprint arXiv:2211.01837*, 2022. 4.1

[154] Qingyuan Zhao and Trevor Hastie. Causal interpretations of black-box models. *Journal of Business & Economic Statistics*, 39(1):272–281, 2021. 1.1

[155] Wentian Zhao, Xinxiao Wu, and Xiaoxun Zhang. Memcap: Memorizing style knowledge for image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12984–12992, 2020. **??**

[156] Mi Zhou, Vibhanshu Abhishek, Timothy Derdenger, Jaymo Kim, and Kannan Srinivasan. Bias in generative ai. *arXiv preprint arXiv:2403.02726*, 2024. 6.1

[157] Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. Autodan: Automatic and interpretable adversarial attacks on large language models. *arXiv preprint arXiv:2310.15140*, 2023. (document), 3, 3.1.3, 4.4.2, 4.3, 6.1

[158] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 9

[159] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023. 1.3, 3, 3.1.2, 3.4.1, 4.1, 4.4.2, 6.1, 6.2.4, 7

Appendix A

Practical Consideration of the Laplace Approximation

While posterior sampling of the cases outlined in Sections 2.3 and 2.3.1 can be accomplished via a myriad of methods, as stated above, we focus on the Gaussian case here in order to ensure that the counterfactual distribution from which we sample from remains tractable and well understood. In doing so, we have to approximate the likelihood and counterfactual prior as Gaussian using the Laplace Approximation. This method approximates an arbitrary distribution, f_x as Gaussian through a two step procedure. First we set as the mean of the approximation the mode of f_x, ie. $\bar{x} \ni f_x(\bar{x}) \geq f_x(x') \ \forall \ x'$. We then set as the approximation's covariance, $\Sigma^{-1} = \nabla^2 f_x(\bar{x})$. One can see why this choice of covariance is used by performing a second order taylor expansion of $\log f_x$ around $\bar{x}$, and seeing that this is proportional to a Gaussian with mean $\bar{x}$ and covariance Σ.

For complex models, when performing the Laplace approximation over the classifier's learned representation, $r : \mathcal{X} \to \mathbb{R}^m$, and latent representation, $\mathrm{d} : \mathcal{R}^k \to \mathcal{X}$, finding the mode becomes intractable. Finding $\bar{x} \ s.t. \ f_x(\bar{x}) \geq f_x(x') \ \forall \ x'$, implies finding $\bar{x} \ s.t. \ f_x(\bar{x}) \geq (r \circ \mathrm{d})(x') \forall x'$, in other words, we need to find the input that globally minimizes loss over the composition of two non-convex functions. Finding such a solution is infeasible, so the approximation will inevitably be based on local optima. Hence, the new conditional prior, $g(x|y)$ that we place on a counterfactual, while designed to cover the distribution of data that returns a desired, predicted label, instead covers only a portion of that space, and in some cases, may include the space of points from which we return different labels.

Appendix B

Tensor Products

Recall that the order (aka. modes or ways) of a tensor is the number of dimensions that make it up. Kolda and Bader [58] have used one dimensional fibers or two dimensional slices to define tensors, where a third-order rank one tensor is defined as,

$$\tilde{A} = a \circ b \circ c,$$

where $\circ$ denotes the outer product operation between vectors a and b, defined as

$$a \circ b = \begin{bmatrix} a_0 b_0 & \cdots & a_0 b_n \\ a_1 b_0 & \cdots & a_1 b_n \\ \vdots & \ddots & \vdots \\ a_n b_0 & \cdots & a_n b_n \end{bmatrix} \qquad A \circ b = \begin{bmatrix} A_0 b_0 & A_1 b_1 & \cdots & A_n b_n \end{bmatrix}$$

Multiplication between tensors has been introduced in [56], in terms of the ciruclant matrix, where,

$$a = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix}^T$$

then

$$\text{circ}(a) = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix}.$$

In order to multiply tensors, we first, we define an unfolding operation that reshapes an $\mathbb{R}^{d_1 \times d_2 \times \cdots \times d_n}$ tensor into a partitioned tensor in $\mathbb{R}^{d_1 d_n \times \cdots \times d_{n-1}}$ tensor and we conversely define a fold operation to reshape the tensor back into its original shape,

$$\text{unfold}(\tilde{X}) = \begin{bmatrix} \tilde{X}_1 & \tilde{X}_2 & \cdots & \tilde{X}_n \end{bmatrix}^T \qquad \text{fold}(\text{unfold}(\tilde{X})) = \tilde{X}. \tag{B.1}$$

Using this notation, [56] defines the t-product between tensors recursively as,

$$\begin{aligned}\tilde{A} * \tilde{B} &= \text{fold}(\text{circ}(\text{unfold}(\tilde{A}))) * \text{unfold}(\tilde{B})) \\ &= \text{fold}(\begin{bmatrix}\tilde{A}_0 & \tilde{A}_1 \\ \tilde{A}_1 & \tilde{A}_0\end{bmatrix} * \begin{bmatrix}\tilde{B}_0 \\ \tilde{B}_1\end{bmatrix}) \\ &= \text{fold}(\begin{bmatrix}\tilde{A}_0 * \tilde{B}_0 + \tilde{A}_1 * \tilde{B}_1 \\ \tilde{A}_1 * \tilde{B}_0 + \tilde{A}_0 * \tilde{B}_1\end{bmatrix}),\end{aligned}$$

where circ is the circulant matrix. It is well known that the circulant matrix has a strong connection to circular convolutions as shown in [8]. We can thus think of the t-product as a convolution with circular padding,

$$\tilde{A} * \tilde{B} = \begin{bmatrix}\tilde{A}_0 & \tilde{A}_1\end{bmatrix} \otimes \begin{bmatrix}\tilde{B}_0 & \tilde{B}_1 & \tilde{B}_0\end{bmatrix},$$

where $\otimes$ denotes a convolution of $\tilde{A}$ across $\tilde{B}$, using the t-product instead of the matrix multiplication. In this way, we can express a generalization of the t-product. [56] defined the t-product in terms of, $\tilde{A} \in \mathbb{R}^{m \times k \times p_1 \times \cdots p_n}$ and $\tilde{B} \in \mathbb{R}^{k \times n \times p_1 \times \cdots p_n}$, where $\tilde{A}$ and $\tilde{B}$ must have their first two dimensions of the appropriate shape for matrix multiplication and each of the remaining dimensions must be the same size for both tensors.

As a circular convolution, we can allow arbitrary tensor products as long as the tensors are of the same order by applying circular padding. For example, if $\tilde{A} \in \mathbb{R}^{m \times k \times 2}$ and $\tilde{B} \in \mathbb{R}^{k \times n \times 4}$, we can express the product as,

$$\tilde{A} * \tilde{B} = \begin{bmatrix}\tilde{A}_0 & \tilde{A}_1\end{bmatrix} \otimes \begin{bmatrix}\tilde{B}_0 & \tilde{B}_1 & \tilde{B}_2 & \tilde{B}_3 & \tilde{B}_0\end{bmatrix} \in \mathbb{R}^{m \times n \times 4}$$

Note that this product is equivalent to that described in [56] when $\tilde{A}$ and $\tilde{B}$ have the same sized dimensions after dimension 2. Moreover, it is easy to verify that this generalization still follows the same rules of distributivity and associativity as the standard t-product.

Appendix C

Pervasiveness of Syntax Features Across Dialects

In this section we provide a set of dialects of English for which each feature used in the main text analysis are commonly used. This list is compiled from [59]. Do note, that this list is non-exhaustive

C.0.1 Null Copula is commonly used among:

Aboriginal English Australia	Bahamian Creole Caribbean
Barbadian Creole (Bajan) Caribbean	Belizean Creole Caribbean
Bislama Pacific	Butler English South and Southeast Asia
Cameroon Pidgin	Colloquial Singapore English (Singlish)
Eastern Maroon Creole	Ghanaian Pidgin
Gullah	Guyanese Creole (Creolese)
Jamaican Creole	Krio (Sierra Leone Creole)
Nigerian Pidgin	Pure Fiji English (basilectal FijiE)
Roper River Creole (Kriol)	San Andrés Creole
Sranan	Torres Strait Creole
Trinidadian Creole	Urban African American English
Vernacular Liberian English	Vincentian Creole

C.0.2 Double Modal is commonly used among:

Guyanese Creole (Creolese)	Jamaican Creole
Ozark English	Saramaccan
Southeast American enclave dialects	Sranan
Appalachian English	Bahamian English Caribbean
Chicano English	Colloquial American English
Gullah	New Zealand English
Nigerian Pidgin	Rural African American English
Tristan da Cunha English	Urban African American English

C.0.3 Habitual Be is commonly used among:

Bahamian Creole	Bahamian English
Butler English	Indian South African English
Irish English	Rural African American Vernacular English
Tristan da Cunha English	Vernacular Liberian English
Gullah	Urban African American English

C.0.4 Invariant Don't is commonly used among:

Aboriginal English Australia	Barbadian Creole (Bajan)
Earlier African American Vernacular English	East Anglian English
Gullah	Guyanese Creole (Creolese)
Hong Kong English	Malaysian English
Newfoundland English	Ozark English
Rural African American Vernacular English	Southeast American enclave dialects
Trinidadian Creole Caribbean	Tristan da Cunha English
Urban African American Vernacular English	

C.0.5 Negative Concord is commonly used among:

Aboriginal English	Appalachian English
Australian Vernacular English	Bahamian Creole
Bahamian English	Barbadian Creole (Bajan)
Butler English	Cameroon English
Cameroon Pidgin	Chicano English
Earlier African American English	East Anglian English
Eastern Maroon Creole	English dialects in the Southwest of England
Gullah	Guyanese Creole (Creolese)
Hawai'i Creole	Jamaican Creole
Krio (Sierra Leone Creole)	Manx English
Newfoundland English	Ozark English
Palmerston English	Rural African American Vernacular English
San Andrés Creole	Southeast American enclave dialects
Sranan	Torres Strait Creole
Trinidadian Creole	Urban African American English
Vernacular Liberian English	Vincentian Creole

C.0.6 Completive Done is commonly used among:

Bahamian English	Barbadian Creole (Bajan)
Cameroon Pidgin	Earlier African American English
Gullah	Guyanese Creole (Creolese)
Jamaican Creole	Krio (Sierra Leone Creole)
Liberian Settler English	Nigerian Pidgin
Norfolk Island/ Pitcairn English	San Andrés Creole
Southeast American enclave dialects	Vincentian Creole
Ozark English	Appalachian English
Colloquial American English	Urban African American English
Rural African American English	Bahamian Creole
Belizean Creole	Trinidadian Creole

C.0.7 Quotative all is commonly used among:

Colloquial American English	Irish English
Newfoundland English	New Zealand English
Philippine English	Pure Fiji English (basilectal FijiE)
Scottish English	Welsh English
Aboriginal English	Australian Vernacular English
Bahamian English	Bislama
Cape Flats English	Channel Islands English
Chicano English	Colloquial Singapore English (Singlish)
Croker Island English	East Anglian English
English dialects in the North of England	English dialects in the Southeast of England
Indian English	Jamaican English
Kenyan English	Malaysian English
Maltese English	Rural African American English
Southeast American enclave dialects	Trinidadian Creole
Ugandan English	Urban African American English
Vincentian Creole	White Zimbabwean English

C.0.8 Ain't as the negated form of be is commonly used among:

Appalachian English	Bahamian English
Earlier African American Vernacular English	East Anglian English
Ozark English	Rural African American English
Southeast American enclave dialects	Urban African American English
Bahamian Creole	Barbadian Creole (Bajan)
British Creole	Chicano English
Colloquial American English	English dialects in the Southeast of England
English dialects in the Southwest of England	Kenyan English
Liberian Settler English	Newfoundland English
Norfolk Island/ Pitcairn English	St. Helena English
Trinidadian Creole	Tristan da Cunha English
Vincentian Creole	

C.0.9 New quasi-modals with aspectual meanings (Including 'Finna') is commonly used among:

Appalachian English	Barbadian Creole (Bajan)
Guyanese Creole (Creolese)	Hawai'i Creole
Liberian Settler English	Newfoundland English
Ozark English	Rural African American English
Urban African American English	Bahamian Creole
Bahamian English	Chicano English
Colloquial American English	Jamaican Creole
Southeast American enclave dialects	Trinidadian Creole

C.1 Dataset Details

In this section, we provide both details on each of the datasets chosen for our MTurk survey, and additional information on what each dataset evaluates in terms of our results.

Lucas [37]. LUCAS is a synthetic dataset of 2000 instances in which the binary outcome, whether or not an individual has lung cancer, is based on 11 other binary features: 'Anxiety', 'Peer Pressure', 'Born on an Even Day', 'Smoking', 'Yellow Fingers', 'Genetics', 'Allergy', 'Coughing', 'Fatigue', 'Attention Disorder', and 'Car Accident'.

Due to the complete knowledge of the conditional probabilities and the small size of the feature set, we are able to convey the exact causal relationships to the survey participants, and verify whether their justifications for preferring one explanation over another accurately reflects these probabilities. For example, a participant may remark that 'Smoking' is unlikely without 'Anxiety' and 'Peer Pressure'. This in turn allows us to examine how exact knowledge of a system correlates with an individual's preferences for certain types of explanations.

Adult [26].

Adult is a well-known dataset from the UCI data repository which consists of 48842 instances with 8 categorical features and 6 continuous for the purpose of predicting whether or not an individual made over $50,000 in income as provided by the 1994 US Census.

In order to introduce causal relationships in the Adult dataset, we use the graph from Zhang et al. [149] with an additional edge from 'native-country' to 'race'. During preprocessing we discard: 'fnlwgt', 'education-num', 'capital-gain', and 'capital loss' due to their exclusion from the DAG provided by prior work. Additionally, we define 'race', 'sex', and 'native-country' as immutable features, and 'relationship' as a mutable-nonactionable column due to its dependency on marital status and sex.

As this dataset is a mixture of continuous and categorical variables, we are able to investigate how the approach fits to data on very different scales. Moreover, unlike in the LUCAS case, we do not provide survey respondents with the causal relationships, and instead investigate the extent to which their personal beliefs on the way social systems function influence the explanations.

German Credit [26]. Similarly to the Adult dataset, German credit is another well-known dataset in the UCI repository with 1000 instances and a combination of 20 continuous and cate-

gorical features. However, unlike Adult and LUCAS, we do not have an underlying causal DAG for German Credit, so all variables are treated as independent. Additionally, we apply a log transform to 'Credit amount', 'Duration in months', and 'Age in years', while also defining 'Personal status and sex', 'Purpose', and 'Foreign worker' fields as immutable. Lastly, the 'Property' field is discarded as the categories are not independent of one another.

German Credit provides an interesting setting in which we have a fairly small dataset with a great deal of complexity. Like in the adult dataset, this allows us to investigate cases in which participants have pre-existing understanding of how lending systems work. Yet, as we do not have information on the causal relationships, we are able to see how each approach handles uncertainty and we can determine whether this effect is noticeable to participants.

C.2 Precomputing the Gradient $V_i^+ V_j$

Algorithm 2: Precomputing the Gradient $V_i^+ V_j$ for words that are tokenized to l tokens

Input: Text Corpus C, Language models $\mathcal{M}_i$ and $\mathcal{M}_j$
Output: Gradient $V_i^+ V_j$

$l \leftarrow$ only consider words that require l tokens in $\mathcal{M}_j$;
$\mathcal{T}_i, \mathcal{T}_j \leftarrow$ Tokenizer of model $\mathcal{M}_i, \mathcal{M}_j$;
$\mathcal{E}_i, \mathcal{E}_j \leftarrow$ Mapping from token to embedding of $\mathcal{M}_i, \mathcal{M}_j$;
$d_i, d_j \leftarrow$ Dimensionality of $\mathcal{M}_i, \mathcal{M}_j$ embeddings;
$W \leftarrow \emptyset$; `// Initialize an empty list`
$k \leftarrow 0$; `// keep track of max size to tokenize with` $\mathcal{T}_i$
foreach *word in* C **do**
    if *word* $\notin W$ **then**
        $t_j \leftarrow T_j(word)$; `// Tokenize a single word`
        $k \leftarrow \max(k, |T_i(word)|)$; `// update` k
        if $|t_j| = l$ **then**
            $W \leftarrow W \cup \{word\}$; `// Add to list if exactly` l `tokens in` j
$V_i \leftarrow$ initialized zero tensor of $|W|$ rows, d_i columns, and depth l;
$V_j \leftarrow$ initialized zero tensor of $|W|$ rows, d_j columns, and depth k;
for $m \leftarrow 1$ **to** $|W|$ **do**
    $t_j \leftarrow T_j(W[m])$; `// Tokenize word` $W[m]$ `with tokenizer` j
    $t_i \leftarrow T_i(W[m])$;
    for $n \leftarrow 1$ **to** $|t_j|$ **do**
        $(V_j)_{w,:,m} \leftarrow (V_j)[m, :, n] + \mathcal{E}_j((t_j)[n])$; `// Add the embedding of` t_j `to` V_j
    for $n \leftarrow 1$ **to** $|t_i|$ **do**
        $(V_i)_{w,:,m} \leftarrow (V_i)[m, :, n] + \mathcal{E}_i((t_i)[n])$; `// Add the embedding of` t_i `to` V_i
$V_i^+ \leftarrow$ Pseudoinverse(V_i) ; `// According to [49]`
$\nabla_{E_i} T_{i:j}(E_i) \leftarrow V_i^+ * V_j$; `// Compute the t-product`
return $\nabla_{E_i} T_{i:j}(E_i)$

Appendix D

Soft Edit Distance

The increasing focus on sequential modeling has necessitated approaches for computing more complex distances that respect the unique aspects of sequential data. Many methods, particularly in bioinformatics, have been developed that are finding value in this space. The Soft Edit Distance [83] discussed in the main text is one such approach.

When writing, we can freely insert, delete, and substitute text within sentences. Distance metrics over text should therefore incorporate this notion. While norms encode substitutions, deletions and insertions should be considered in tandem, with each of the three operations acting as a single edit.

As discussed in the main text, Levenshtein distance is one metric that determines distance in terms of single edits. A single insertion, deletion, or substitution each constitutes a distance of 1. The metric finds the minimum number of edits required to make two sequences match and can be formally defined as:

$$d(\mathbf{x}_0, \mathbf{x}_1) = \min_{|\mathbf{x}'0|=|\mathbf{x}'1|=l} (d_H(\mathbf{x}_0, \mathbf{x}_1) + L_0 + L_1 - 2l),$$

where d_H is the Hamming distance, $\mathbf{x}'0$ and $\mathbf{x}'1$ are length-l subsequences of the original prompts $\mathbf{x}_0$ and $\mathbf{x}_1$ with lengths L_0 and L_1, respectively.

Ofitserov et al. [83] show a differentiable relaxation of this distance by replacing the Hamming distance with norms and using a soft-min function:

$$SED(\mathbf{x}_0, \mathbf{x}_1) = \frac{\sum_{|\mathbf{x}'_0|=|\mathbf{x}'_1|} R(\mathbf{x}'_0, \mathbf{x}'_1) e^{\tau R(\mathbf{x}'_0, \mathbf{x}'_1)}}{\sum_{|\mathbf{x}'_0|=|\mathbf{x}'_1|} e^{\tau R(\mathbf{x}'_0, \mathbf{x}'_1)}}, \quad R(\mathbf{x}'_0, \mathbf{x}'_1) = \frac{||\mathbf{x}'_0 - \mathbf{x}'_1||_2^2}{2} + L_0 + L_1 - 2l.$$

While this relaxation allows for differentiable sequence comparisons, it is extremely expensive—comparing all sequences grows exponentially with sequence length. The authors therefore introduce a recurrent strategy that can be computed in polynomial time. Let

$$\alpha_{i,j} = \sum_{|X'_1|=|X'_2|} R_{i,j}(X'_1, X'_2) e^{R_{i,j}(X'_1, X'_2)}$$
$$\beta_{i,j} = \sum_{|X'_1|=|X'_2|} e^{\tau R_{i,j}(X'_1, X'_2)}$$
$$i = \overline{0, L_1}, j = \overline{0, L_2},$$

where $X_{1,1:i}$ and $X_{2,1:j}$ are the matrix representations of prefixes of sequences x_1 and x_2 with lengths $L_1 = |x_1|$ and $L_2 = |x_2|$, respectively, and $\tau < 0$.

We can then compute $\text{SED}(X_1, X_2) = \frac{\alpha_{L_1,L_2}}{\beta_{L_1,L_2}}$. The authors show that the following recurrence equation holds for coefficients α and β:

$$\begin{aligned}
\alpha_{i,j} &= (\alpha_{i-1,j} + \beta_{i-1,j} + \alpha_{i,j-1} + \beta_{i,j-1}) \exp^{\tau} + (\alpha_{i-1,j-1} + \beta_{i-1,j-1}\delta_{i,j}) \exp^{\tau\delta_{i,j}} \\
&\quad - (\alpha_{i-1,j-1} + 2\beta_{i-1,j-1}) \exp^{2\tau}, i = \overline{1, L_1}, j = \overline{1, L_2} \\
\alpha_{i,0} &= ie^{\tau i}, i = \overline{0, L_1} \\
\alpha_{0,j} &= je^{\tau j}, j = \overline{0, L_2} \\
\beta_{i,j} &= (\beta_{i-1,j} + \beta_{i,j-1})e^{\tau} + \beta_{i-1,j-1}(e^{\tau\delta_{i,j}} - e^{2\tau}), i = \overline{1, L_1}, j = \overline{1, L_2} \\
\beta_{i,0} &= e^{\tau i}, i = \overline{0, L_1} \\
\beta_{0,j} &= e^{\tau j}, j = \overline{0, L_2},
\end{aligned}$$

where $\delta_{i,j} = 0.5|X_{1,i} - X_{2,j}|_p$. See Ofitserov et al. [83] for the complete proof of the recurrence equation's equivalence.

As we do in the main text, this function can be used to directly compute the edit distance between two strings. While our implementation uses automatic differentiation to differentiate the function, the authors provide a tractable form of the derivative in their paper.

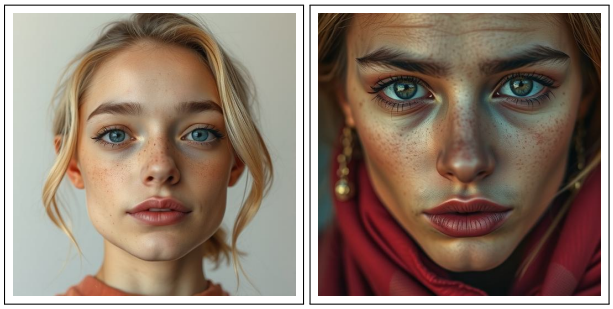
D.1 Additional Examples of Discovered Representational Asymmetries

Here, we provide additional examples of the gender asymmetries from Section 6.3.2. Beyond the examples in the main text, where we found that minimal syntactic differences such as "also" or platform-specific tokens such as "flickr" and "shutterstock" produced asymmetries, several additional effects emerged from this pilot study. As in the main text, all images are generated from fixed seeds 0-5, and we show the 2 most visually striking examples.

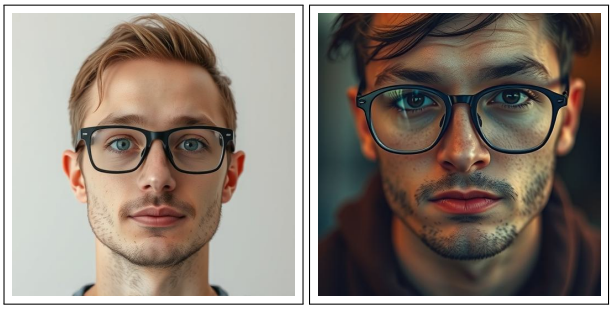
The first three examples in Figure D.1 show images generated by the prompts: "a person," "a person's face," and "a person with glasses' face." The left-hand and right-hand images are generated from the same fixed seed. One can immediately see that specifying that an image shows a person's face samples from a very different distribution than the former prompt. And as discussed in Section 6.3.1, the token "face" is female-aligned. When we apply a more specific prompt, specifying "a person with glasses' face," we find that the female-presenting images in the former become clearly male-aligned. The generated subjects develop stubble while also wearing glasses, potentially reinforcing stereotypes associating certain types of eyewear with men or intellectual professions.

The results in Section 6.3.1 also suggest differences derived from passive and active activities through the lens of "riding" and "driving." We generate images using the prompts: "a photo of a person riding in a car" and "a photo of a person driving in a car." The female-aligned "riding" token and male-aligned "driving" token show a subtle encoding of gendered expectations. Even

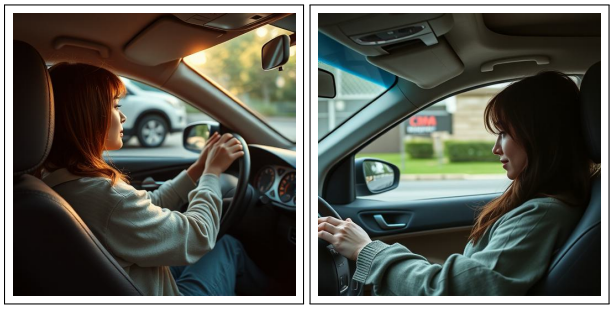
(a) a person



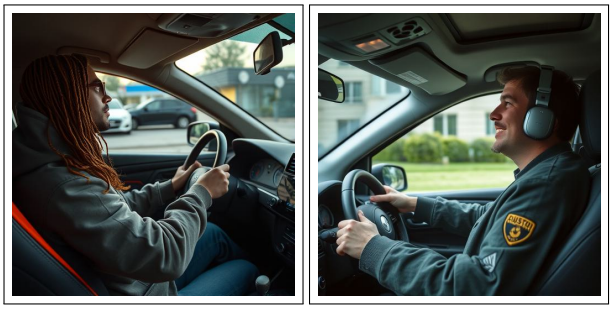
(b) a person's **face**



(c) a person **with glasses' face**



(d) a photo of a person **riding** in a car



(e) a photo of a person **driving** in a car

Figure D.1: Images of prompts with notable effects on gender representations derived from discovered token frequency differences. We find several potential asymmetries, whose use may guide more targeted explorations of gender across unexpected or underexplored bias axes.

(a) a teacher **sitting**

(b) a teacher **standing**

(c) a therapist **sitting**

(d) a therapist **standing**

Figure D.2: Images of prompts with notable effects on gender representations derived from discovered token frequency differences. We find several potential asymmetries, whose use may guide more targeted explorations of gender across unexpected or underexplored bias axes.

when all compared images show the subjects with their hands on steering wheels, "riding" produces more feminine features and "driving" produces more masculine features. In one example, the woman's red hair shifts into the man's red hood.

We also found that the token "standing" appeared to be male-aligned in the preliminary analysis in Section 6.3.1. Figure D.2 shows a comparison between two occupations that prior work has found to be more female-aligned: "teachers" and "therapists." We find that "standing" appears to be a male-aligned prompt, as the women in the "sitting" prompts are generated as men when "standing" is specified.

As discussed in the main text, none of these examples are intended to be an exhaustive account of the gender asymmetries present in the model. We believe that these examples demonstrate that our approach generates reliable candidates for further exploration and expanded analyses.