UCLA

UCLA Electronic Theses and Dissertations

Title

Exploring User-Centric Generative Models: Advancing User Control, Comprehension, and Creative Capacity

Permalink

https://escholarship.org/uc/item/2bg09326

Author

Evirgen, Noyan

Publication Date

2023

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA

Los Angeles

Exploring User-Centric Generative Models: Advancing User Control, Comprehension, and Creative Capacity

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical and Computer Engineering

by

Noyan Evirgen

2023

© Copyright by
Noyan Evirgen
2023

ABSTRACT OF THE DISSERTATION

Exploring User-Centric Generative Models: Advancing User Control, Comprehension, and Creative Capacity

by

Noyan Evirgen
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2023
Professor Xiang 'Anthony' Chen, Chair

The rapid advancement of generative models, particularly in image generation and manipulation, has opened new possibilities in creative and design fields. However, their complex 'black box' nature poses significant challenges for user interaction and control, especially for non-experts. This dissertation addresses these challenges by developing and evaluating user-centric tools that enhance interaction with Generative Adversarial Networks (GANs) and Text-to-Image (T2I) models.

Central to this work is the exploration of user-driven methods to improve the usability and accessibility of these models. The research introduces innovative tools that empower users to iteratively refine and control the generative process. These tools are designed to complement existing GAN architectures, allowing users to interact more intuitively with the models, particularly in tasks requiring precise image editing and creative content generation.

Empirical studies form a significant part of this research, evaluating the effectiveness of these tools in real-world scenarios. The studies involve user tasks in image editing and creative content generation. Findings demonstrate that the developed tools not only facilitate

a more intuitive interaction with generative models but also enable users to achieve superior results compared to existing state-of-the-art methods.

Additionally, this dissertation investigates ways to enhance user understanding of the generative process. By making the mechanisms of these models more transparent and comprehensible, the research contributes to a more informed and effective use of these technologies.

In conclusion, this dissertation focuses on making advanced generative models more accessible and user-friendly. It offers insights into the development of intuitive tools that bridge the gap between the complex capabilities of generative AI models and the creative and practical needs of users.

The dissertation of Noyan Evirgen is approved.

Bolei Zhou

Gregory J. Pottie

Quanquan Gu

Xiang 'Anthony' Chen, Committee Chair

University of California, Los Angeles

2023

To my family, Erhan, Nevin, and Nogay Evirgen

I couldn't have done this without you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

At the culmination of this incredible journey, my heart is filled with immense gratitude towards those who have been pivotal in making this thesis a reality.

Foremost, I extend my deepest thanks to my advisor, Professor Xiang 'Anthony' Chen. His unwavering guidance, profound expertise, and infectious enthusiasm for groundbreaking research have been the cornerstone of my academic journey. His unique perspective on choosing impactful research topics and relentless energy have been nothing short of inspirational.

My heartfelt appreciation also goes to the other distinguished members of my dissertation committee: Professor Bolei Zhou, Gregory J. Pottie, and Quanquan Gu. Their thorough evaluations, insightful feedback, and dedication have greatly enriched my work and academic growth.

A special note of thanks to my lab mates - Hongyan Gu, Ruolin Wang, Jiahao Li, Bruce Liu, and Youngseung Jeon. Their camaraderie, collaborative spirit, and willingness to engage in thoughtful discussions and peer reviews have been invaluable throughout this journey.

Lastly, but most importantly, I owe an immeasurable debt of gratitude to my family. To my parents and my brother, your unwavering love, encouragement, and belief in me have been the bedrock of my strength and perseverance. This thesis is not just a reflection of my efforts, but a testament to your endless support and sacrifices. I dedicate this achievement to you.

VITA

2011–2015 B.S. (Electrical and Electronics Engineering), Bilkent University

2015–2018 M.S. (Information Technology and Electrical Engineering) ETH Zurich

2018–present Ph.D. (Electrical and Computer Engineering) University of California, Los Angeles

PUBLICATIONS

GANravel: User-Driven Direction Disentanglement in Generative Adversarial Networks. **N. Evirgen**, X. Chen, CHI 2023.

GANzilla: User-Driven Direction Discovery in Generative Adversarial Networks. **N. Evirgen**, X. Chen, UIST 2022.

A Novel Method for Scheduling of Wireless Ad Hoc Networks in Polynomial Time. A. Köse, H. Gökcesu, **N. Evirgen**, K. Gökcesu, M. Médard, IEEE Transactions on Wireless Communications, 2020.

System and method for supervised learning of permeability of earth formations. R. Viswanathan, L. Venkataramanan, P. Srivastava, A. Prado, **N. Evirgen**, M. Loan, H. Datir, US Patent 17/310965, 2020.

An unsupervised learning algorithm to compute fluid volumes from NMR T1-T2 logs in unconventional reservoirs. L. Venkataramanan, **N. Evirgen**, D. Allen, A. Mutina, US Patent 62/574992, 2018.

The Effect of Communication on Noncooperative Multiplayer Multi Armed Bandit Problems. **N. Evirgen**, A. Köse, ICMLA 2017.

Performance Comparison of Algorithms for Movie Rating Estimation. A. Köse, C. Kanbak, **N. Evirgen**, ICMLA 2017.

Deep Learning Tools for Foreground-Aware Analysis of Film Colors. B. Flueckiger, **N. Evirgen**, E. G. Paredes, R. B. Ripoll, R. Pajarola, AVinDH SIG 2017.

Proof of concept for satellite attitude determination using GNSS. A. Şen, C. Tansu, E. C. Ünlüsoy, M. Yurt, **N. Evirgen**, Ö. C. Sakinci, B. Akbulut, RAST 2015.

CHAPTER 1

Introduction

1.1 A New Era in Artificial Intelligence: Generative Models

As we stand at the threshold of a new era in technology, the rise of artificial intelligence (AI) marks a turning point in human history, a period characterized by unparalleled advancements and transformative innovations. AI, once a domain relegated to the realms of science fiction and theoretical research, has rapidly evolved into an integral part of our daily lives, reshaping industries, redefining human interaction, and revolutionizing the way we perceive and interact with technology.

At the heart of this AI revolution lies the emergence of generative models, a groundbreaking development that has changed the landscape of AI. These generative models, driven by intricate algorithms and deep learning techniques, have ushered in a paradigm shift in various domains. In the realm of visual arts and design, models like generative adversarial networks (GANs) and text-to-image systems have empowered creators with tools to generate lifelike images, realistic animations, and innovative designs. Similarly, in the field of language and communication, LLMs have redefined the boundaries of text generation, enabling the creation of coherent, contextually relevant, and often insightful textual content. Their applications span from writing assistance to conversational agents, from content creation to language translation, representing a quantum leap in how we interact with and utilize language in the digital age.

These generative models have opened a myriad of possibilities across industries. In

healthcare, they aid in synthesizing medical data for research, diagnostic simulations, and training. In education, they serve as dynamic tools for learning and creativity, enhancing student engagement and understanding. In entertainment and media, they offer novel ways to produce content, tell stories, and engage audiences. The potential of generative models extends to sectors like finance, law, and engineering, where they offer solutions for data synthesis, predictive modeling, and problem-solving.

However, with all their capabilities and potential, these generative models also present a significant challenge: their complexity and 'black box' nature. This aspect often renders them inaccessible to most potential users, particularly those without specialized training in AI. The intricate workings of these models, while fascinating, remain a mystery for many, limiting their ability to leverage these tools to their full potential.

This observation, where the most advanced tools in AI are simultaneously the most enigmatic and out of reach for the general public, sets the stage for the research presented in this dissertation. In this environment of technological growth and the need to make AI tools available to everyone, the dissertation introduces innovative solutions and insights aimed at bridging the gap between the complex world of generative models and the everyday user.

1.2 Challenges and Research Questions in User-Centric Generative Models

Generative AI models present a unique set of challenges, distinct from those of traditional decision-making AI models, such as classification or regression. The effectiveness of decision-making models is typically gauged by performance, which can be quantified using specific metrics. These metrics can often be improved through dataset enhancements or adjustments in model architecture. Furthermore, the predictions made by such models can serve as 'advisory' inputs in high-risk fields like medicine, supplementing but not replacing human decision-making.

However, the dynamics of AI-human collaboration in the context of generative models are less straightforward. Assessing the quality of images generated by these models with standard metrics is challenging, and the integration of generated data into user workflows is often not clearly defined.

In tasks involving classification or regression, the algorithms' primary goal is to make predictions and, ideally, 'explain' the reasoning behind these predictions. This allows users to incorporate both the explanation and prediction into their workflow seamlessly. In contrast, the interaction between users and generative models is not immediately apparent. Consider, for instance, a game developer using a generative AI model to create human faces. The question arises: should they simply use a set of generated images directly in their games, or should they provide the model with specific conditions, such as generating faces with blue eyes and long hair? Further complexities emerge when considering alterations to an already created image. The inherently complex nature of generation tasks necessitates a reevaluation of how humans and AI models interact. As a result, addressing these issues gives rise to several pertinent research questions:

1. **Tailoring generative model interfaces for non-expert users:**
   How can the interfaces of generative models be designed or adapted to be more user-friendly and intuitive, especially for non-expert users, to facilitate effective control and guidance of the generative process?

2. **Aligning generative outputs with specific user creative goals:**
   How can generative models be tailored to better align their outputs with specific user goals, particularly in creative tasks that require detailed customization and nuanced control?

3. **Empirical evaluation of user-centric generative models:**
   What methodologies and criteria can be employed to empirically evaluate the effectiveness of user-centric tools in improving user experience and outcome quality in

interacting with generative models?

These research questions aim to address the fundamental challenges in bridging the gap between the advanced capabilities of generative AI models and the practical needs of their users. Exploring these questions is crucial for enhancing the utility and applicability of generative models in various real-world scenarios.

1.3 Contributions and the Outline of the Dissertation

To address the research questions outlined above, the dissertation presents research with three primary aims:

1. It aims to advance user control in generative models, with a particular focus on the image domain. This involves enhancing the ability of users, especially those without extensive technical expertise, to effectively manipulate and guide the generative processes.

2. The research seeks to increase creative capacity by developing user-centric tools. These tools are designed to enable deeper and more intuitive interactions between users and generative models, thereby broadening the creative possibilities available to them.

3. An essential goal is to improve users' understanding of the generative process. This involves elucidating the complex mechanisms underlying generative models, making them more transparent and comprehensible, which in turn can lead to more informed and effective utilization of these technologies.

Throughout this dissertation, every chapter adds to our understanding of how users can better interact with generative models. It tackles the main questions we've set out to answer and explores ways to make these complex technologies easier and more useful for everyone. In the end, the dissertation wraps up by looking at the bigger picture of this research, talking

about its limits, and suggesting ideas for what to study next in this area. Specifically, the outline of the dissertation is as follows:

- **Chapter 2** This chapter presents GANzilla, a user-driven tool designed to address the 'black box' nature of Generative Adversarial Networks (GANs). By employing a scatter/gather technique, GANzilla enables non-expert users to iteratively discover and refine editing directions in GANs. The effectiveness of GANzilla is validated through a study, demonstrating its ability to assist users in both specific and open-ended image editing tasks.

- **Chapter 3** This chapter introduces GANravel, a user-driven tool aimed at enhancing direction disentanglement in Generative Adversarial Networks (GANs). Addressing the challenge of GANs as 'black boxes', GANravel enables users to iteratively refine editing directions. The tool's effectiveness is demonstrated through user studies, showing its superiority in disentanglement performance over existing methods and its practical application in creating high-quality images and GIFs, such as dog memes.

- **Chapter 4** This chapter introduces PromptZEN, an innovative tool developed to enhance user interaction with text-to-image diffusion models, specifically Stable Diffusion. Addressing the complexity and 'black box' nature of these models, PromptZEN enables users to perform precise, localized edits on images through prompt scheduling. The chapter highlights PromptZEN's role in providing users with granular control over the image generation process, making advanced text-to-image models more interactive and accessible.

- **Chapter 5** This chapter addresses the final research question by exploring methods to enhance user understanding of the generative process in models like GANs and T2I systems. It examines approaches to make these models more transparent and understandable, thereby improving user comprehension and effectiveness in utilizing

these technologies. The chapter also discusses the implications of these findings for future research and the development of generative models.

- **Chapter 6** The final chapter concludes the dissertation, summarizing the key contributions and findings. It discusses the limitations of the current work and outlines potential directions for future research in the field of user-centric generative models, suggesting ways to further bridge the gap between technical complexity and user accessibility.

CHAPTER 2

User-Driven Direction Discovery in Generative Adversarial Networks

2.1 Introduction

Generative Adversarial Networks (GANs) promise to create new content by learning the characteristics of existing data, showing compelling results in various domains, from stylization [ZPI17], scene creation [VPT16], and improving the quality of scientific data [SYP19].

Unfortunately, despite its increasing widespread use, most GAN models to date remain a 'black box' to end-users with little transparency about what a model is capable of generating and little control over the generative process. Without transparency and control, when end-users have a creative intent (e.g., a caricaturist illustrating a character's facial expression in creative storytelling), they cannot see whether or how one can instruct a GAN model as-is to generate specific characteristics. In other words, there is little support for formulating user-defined *editing directions*[1]—a set of input parameters that steer the GAN model to generate contents with varying levels of characteristics (*e.g.*, making the hair on the input image more or less blonde).

To address the limited transparency and control, some prior work allows a user to browse GAN-generated results in an interactive gallery view [ZB21]; however, such an open-ended exploration is not intended to converge to a specific direction. Others propose methods to

[1]Hereafter simply referred to as 'direction'.

dissect a GAN model [BZS18] or to perform post hoc extraction of principal components [HHL20] or semantic controls [CBP20]. However, such directions are often pre-defined by algorithms, which do not permit a user to specify directions to generate their own desired characteristics.

We design and implement GANzilla—a tool that complements existing algorithm-driven approaches by enabling user-driven direction discovery in a GAN model. As a proof of concept, we focus on a common use case of stylizing faces based on the StyleGAN2 model [KLA20a]; however, the workflow in our tool is expected to generalize to other usages of GAN as well.

As shown in Figure 2.1a, a GANzilla user starts with brushing on a few exemplar images demonstrating specific areas they want to stylize, based on which the back-end samples a large number of directions. Then, GANzilla's front-end employs the classic scatter/gather technique [PSH96] to let a user iteratively filter directions[2] applied on exemplar images and iteratively narrow down to the ones that result in their desired characteristics. Specifically, a user can gather one or more clusters (Figure 2.1b), see whether the constituent directions interest them (Figure 2.1c), and, if so, scatter (Figure 2.1d) them into new clusters to refine their selection. Flexibly, at any given time, the user can go back to the previous iterations and scatter a different subset of the clusters (Figure 2.1e). Meanwhile, the user can test directions on some other images and see if the effects generalize (Figure 2.1f) and bookmark the ones they like (Figure 2.1g).

We validate GANzilla in a user study ($N = 12$) with two types of tasks: *(i)* In the closed-ended tasks, we controlled what a user intended to edit by providing participants a set of edited image pairs (references and targets). Results show that participants were able to find the GAN directions that closely replicated the edits—specifically, their discovered directions transformed the reference images into ones that are more similar to the target im-

[2]In GANzilla, a direction is represented as a thumbnail of an exemplar edited image, *e.g.*, an 'aging' direction is shown as a face older than the original image.

ages and such similarities rank high when compared to edits done by 1000 randomly-sampled directions (representing the latent space). *(ii)* In the open-ended tasks, we tested whether participants could use GANzilla to achieve personalized edits: given some high-level editing goals (*e.g.*, making the face happier), each participant would use GANzilla to find specific edits that they considered to achieve such goals. Results show that each participant felt satisfied with their edits, which also highly align with the goals when analyzed from a language perspective (comparing image and text embeddings); further, the resultant edits exhibit diversity across participants, indicating GANzilla's ability to enable personalized content creation.

Overall, GANzilla makes a **tool contribution**: in contrast to using a GAN model as a 'black box', we provide a comprehensive tool for a user to see what directions the GAN model is capable of and to iteratively discover and test directions that generate their desired characteristics. GANzilla's user-driven direction discovery aims to complement (rather than replace) existing algorithm-driven approaches [GAO19, SGT20, YSZ21] by providing users with an option to explore more editing options when the pre-computed controls do not fully meet their needs.

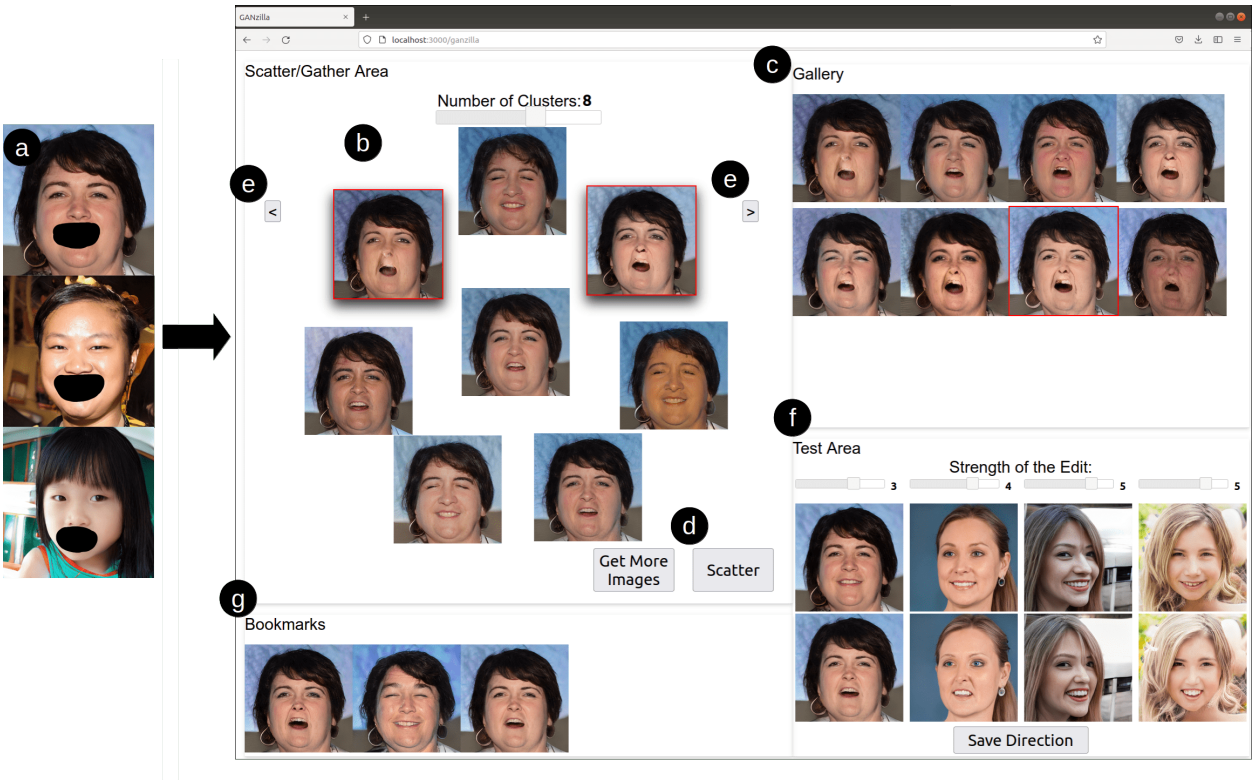

Figure 2.1: GANzilla is a tool that allows users to discover editing directions in Generative Adversarial Networks (GAN) via iterative scatter/gather interactions—a user-driven approach that complements many existing algorithm-driven methods. (a) A user starts by highlighting a region of interest (an optional step). (b) Based on the highlight (if there is), directions are sampled and clustered, each shown as an image edited by that direction. The user can gather clusters by selecting thumbnail images (indicated by a red border). (c) The user can see all the directions of the gathered clusters and (d) scatter them into new clusters. (e) The user can go back-and-forth across iterations to explore alternate choices of scatter/gather. (f) The user can test a selected direction (red border in c) on other images with individual sliders controlling the strength to apply the direction. (g) The user can bookmark directions that meet their editing goals.

2.2 Background & Related Work

In this section, we first provide background information on GAN and editing directions, then we review two areas of work that intersect with ours: existing algorithm-driven approaches for discovering GAN directions and enabling users to interact with GAN.

How does GAN work? As shown in Figure 2.2, GAN is a family of neural networks where a generator G is trained to create synthetic data (x_g) that simulates those from a certain domain (*e.g.*, images of human faces), a discriminator D is trained to distinguish between the generator's synthetic data (x_g) and real data (x_r), and the countering of D and G iteratively leads to the generator's ability to create synthetic data indistinguishable from real data. By default, GAN functions as a 'black box' where a large collection of data points (*e.g.*, images) are generated from randomly-sampled 'noises' (z), leaving users very little control of the generative process.

What is a GAN (editing) direction? The input 'noise' (z in Figure 2.2) to the generator is a point from GAN's latent space, which typically consists of a high-dimension of variables, each drawn from a Gaussian distribution [Bro20]. An editing direction is a vector d in the latent space along which we can move z so that the resultant $x_g = G(z)$ will change in a semantically meaningful way, *e.g.*, , given a closed-mouth image $G(z_0)$, making the mouth open in the image $G(z_1)$ where $z_1 = z_0 + \lambda d$. The coefficient λ is the strength to apply direction d.

Algorithm-driven approaches for discovering GAN directions. To discover GAN directions, early work employs supervised approaches that require sampling and labeling a large number of points in the latent space [JCI19, GAO19, SGT20, PBH20, YSZ21]. Recently, there has been a plethora of research focused on unsupervised methods [VB20, SZ21], each of which aims at controlling a specific aspect of the generated data. Härkönen *et al.* describe a PCA-based approach to decompose a GAN model into interpretable controls for users to specify desired attributes of the generated outcome [HHL20]. Wu *et al.* present a

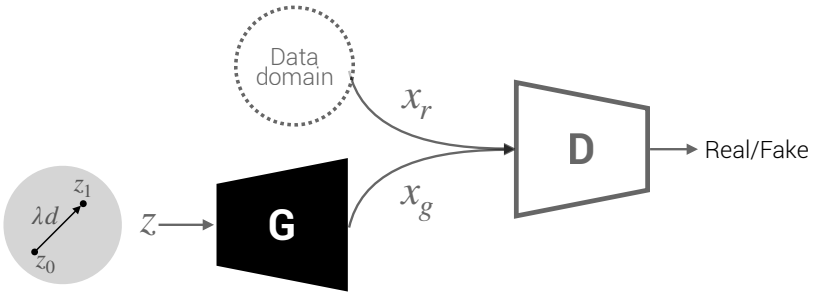

Figure 2.2: Overview of a typical GAN model.

method for computing style channels, each controlling a distinct, localized visual attribute without entangling with one another [WLS21a]. Collins *et al.* enable local semantic editing using GAN: by selecting a specific part of an image (*e.g.*, Person A's nose), GAN is able to transfer its style to another image (*e.g.*, making Person B's nose look like A's) [CBP20]. Although all the above work does enable high-level user-control of the GAN's output, such fully algorithm-driven approaches tend to result in one-size-fits-all directions and do not permit user input to specify customized directions. Complementarily, GANzilla enables each individual user to discover directions on their own, which provides a useful option when the pre-computed controls do not fully meet a user's needs.

Enabling users to interact with GANs. Even prior to the popularity of GAN, researchers have explored interactions with other generative processes, most of which are centered around generative design (*e.g.*, using topology optimization) via sketching [KGC17, CTW18] or visualization [MGB18] to explore a large design space. Given that most conventional ways of controlling GAN is via the use of sliders, Dang *et al.* conducted a comprehensive comparative study to understand the effects of regular sliders *vs.* sliders that provide a 'filmstrip' of feedforward information (preview images) [DMB22]. Other researchers go beyond sliders to consider alternative techniques to interact with GAN. For example, Zhang and Banovic present samples of generated images on a grid-like view wherein a user can zoom in/out or pivot to explore more images [ZB21]. Alternatively, it is also possible to consider other input modalities beyond conventional GUI elements. For example, Yu *et al.* demon-

strate a visual design assistant that takes in a user's natural language feedback to guide the GAN model's modification of the design [CGL20]. Ling *et al.* allow users to discover directions by changing segmentation masks of the generated images [LKL21]. Related to our goal of enabling users to discover directions, Or *et al.* develop StyleCLIP that takes the input of textual description (*e.g.*, "Mohawk hairstyle") and transforms the image accordingly. However, this approach cannot accept arbitrary editing requests; rather, acceptable textual descriptions are limited to a set of phrases the CLIP model produces to characterize a given image dataset. In contrast, GANzilla does not limit a user to a pre-defined vocabulary but allows them to discover their desired editing direction by navigating vast examples using the scatter/gather technique.

2.3 GANzilla: User-Driven GAN Direction Discovery for Image Editing

In this section, we present a detailed walkthrough of GANzilla's design and implementation using an exemplar use case of stylizing a human portrait image to make the face happier.

2.3.1 Highlighting an area to focus the edit on

To start, the user can choose to select a specific part of the face for the edit to focus on. For example, the user might wish to edit the mouth to make a big smile so the face would appear happier. To specify the mouth area, the user simply uses a built-in brush tool to paint over a few exemplar images GANzilla provides (Figure 2.1a). Note that this step is optional: without any selection, edits could occur indiscriminatively across the entire image.

Implementation. GANzilla runs the direction search on the StyleSpace of StyleGAN2. StyleSpace refers to the space that is defined by style parameters of StyleGAN2. These parameters control the individual strength of various filters in StyleGAN2. It is significantly more disentangled than the traditional latent space [WLS21b]. Depending on

whether the user performs this optional highlighting step, the backend either uses the entire or a subset of the style parameters based on the highlighted region. We use a similar method to Collins *et al.* [CBP20] to translate a highlighted region into a subset of style parameters. Each style parameter of StyleGAN2 is assigned an importance metric depending on the highlighted region and the activation maps of the highlighted image. This step allows us to *select* the filters of StyleGAN2. After doing this for few exemplar images, we take the union of the selected style parameters. These parameters are later used to sample directions.

2.3.2 Sampling and clustering directions

Next, based on the user's selected image region (if there is any), GANzilla samples a large number of directions, each of which is represented by an exemplar image edited along that direction. GANzilla clusters the sampled directions and displays each cluster using one of its directions' thumbnail images (Figure 2.3b) to avoid crowding the UI. Selecting one or multiple clusters shows the constituent directions in a separate view (Figure 2.3c). The user can change the number of clusters (the default number is six) and if they cannot find a direction that matches their editing intent, they can request GANzilla to resample more directions (Figure 2.3d).

Implementation. Sampling is done using the selected set of style parameters. We observed from StyleCLIP that the directions only needed a small number of style parameters to be diverse and expressive. Following that observation for each direction sample, we sub-sample the style parameters that come from the highlighting step. The sub-sampling rate is tuned as a hyper-parameter. The resulting set of style parameters are then randomly increased or decreased using a normal distribution, which defines the sampled directions. Each direction then changes a different set of style parameters to increase diversity. In order to cluster these directions, we first generate the resulting images. Then we extract latent codes for these images using an AI-model (CLIP model in our case). Next, these latent codes are clustered with the k-means clustering. Note that we can not use the directions

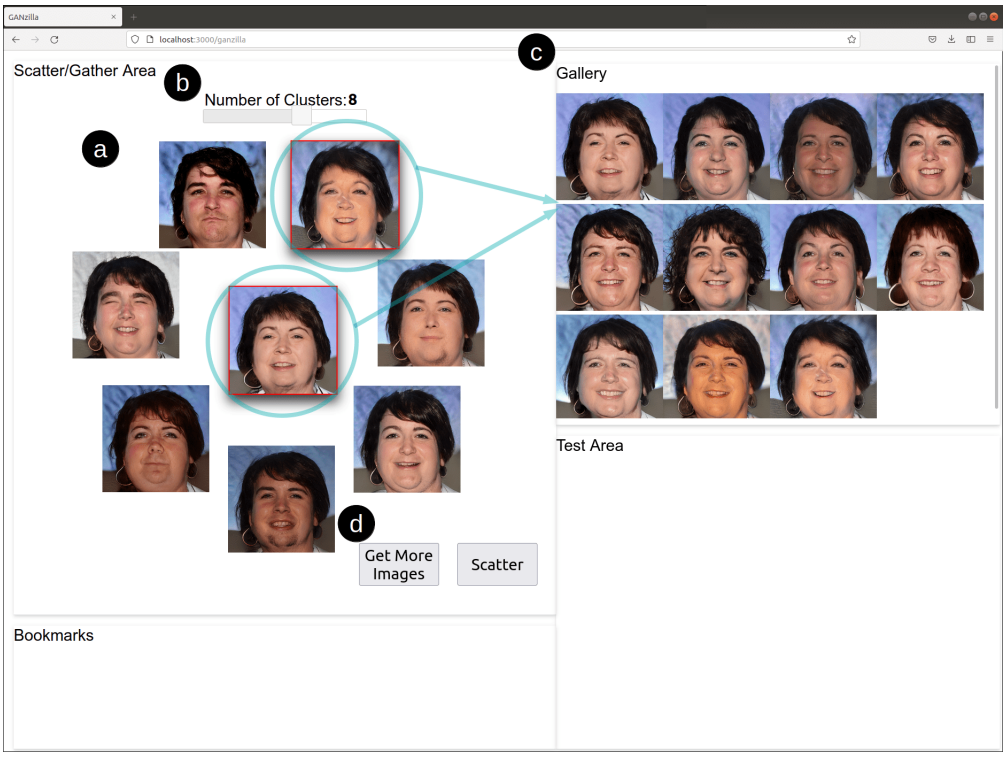

Figure 2.3: After highlighting, GANZILLA generates an initial set of directions. (a) By default it returns six clusters of directions which are all displayed with one representative image each. (b) The user can change the number of clusters. (c) Selecting multiple clusters (indicated by a red border) shows the constituent images of those clusters. (d) If the user cannot find a direction, they can request to sample more.

directly without generating the images, because the dimensions of the directions are not comparable with each other. As for the representative image of the cluster, we choose the one that is closest to the center of the cluster. If the user asks for more directions, we repeat the sampling step while giving a higher priority to the style parameters that are not chosen in previously sampled directions. This allows us to investigate a new subspace in StyleSpace that is not covered in previous directions.

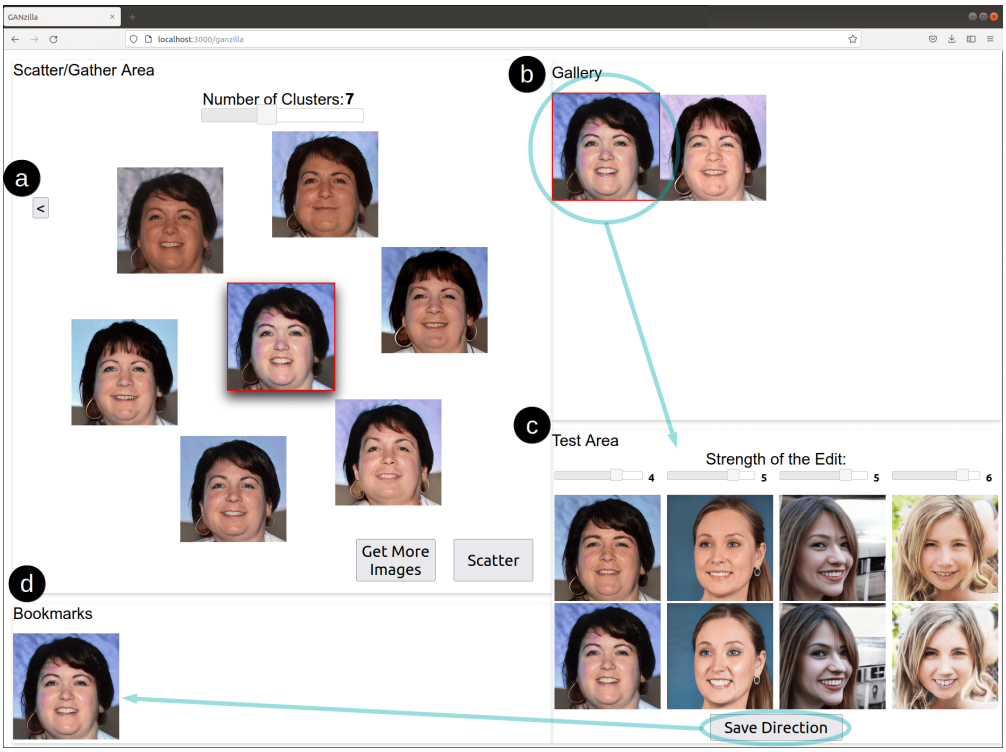

Figure 2.4: The resulting view after clusters gathered in Figure 2.3 are scattered. (a) If the user feels unsatisfied with the scatter results, they can click the back button to go to the previous clusters. (b) The user can select a direction (indicated by red border) and (c) test how it works on other images: the first row are the reference images and the second row edited by the direction being tested, whose strength can be adjusted for individual test images. (d) The user can bookmark a direction if they are satisfied with its edits.

2.3.3 Iterative scatter/gather of directions

Once a user identifies clusters of interest (*e.g.*, open-mouth images matching their intended editing goal), they can iteratively use the scatter/gather technique: first selecting those clusters (gather) and then clicking the 'scatter' button to re-cluster them; then the user can repeat this process with more scatter/gather. Figure 2.4 shows the updated UI after scattering the two clusters gathered in Figure 2.3. Here, GANzilla repurposes the classic scatter/gather technique [PSH96], which was used for browsing a large collection of text

documents, for enabling a user to iteratively converge their choices of direction. Further, GANzilla allows a user to step back (to previous iterations) and explore alternate 'branches' of scatter/gather. Specifically, if the user feels unsatisfied with the current batch of directions, they can click the '¡' button (Figure 2.4a), return to the previous clusters, gather and scatter a different subset of them.

Implementation. Scatter is implemented by first sampling new directions based on the gathered clusters and then clustering the new set of directions. In order to sample a new direction, a random pair is selected from all the gathered directions. Then these two directions are averaged, creating a new direction that combines the two. In order to increase the variation, a random vector (sampled from a normal distribution) is added to the resulting direction. Results of scatters are stored in a tree data structure. When the user wants to explore alternate branches, a new subtree is created from the current node.

2.3.4 Testing directions on more images

At any given time, the user can test a direction by selecting its thumbnail (Figure 2.4b) and see how this direction works on other images. Each of these test images comes with a slider for the user to adjust the strength of the direction (how strongly to apply its editing effects) (Figure 2.4c). Such a 'test field' allows a user to calibrate how strongly they should apply a direction and examine the direction's generalizability to make sure that it can generate the intended edits on other images as well. If a user is satisfied with the direction after the tests, they can bookmark it with the save button (Figure 2.4d). Later, the user can also bring back a 'bookmarked' direction to the test-area by clicking its thumbnail.

Implementation. When a thumbnail image is clicked to be tested, GANzilla scales that direction with a default strength of the direction. Then, the direction is applied to different test images that can be uploaded by the user. The resulting images and their reference images are shown to the user. Whenever there is a change to a slider, the resulting image is re-generated.

2.3.5 Other implementation details

We used Pytorch as our deep learning backend. For the rest of the back-end implementation we used Python. We used Flask as our web-framework, which handled the communication between our front-end and the back-end. For our front-end, we used a combination of Javascript, Node.js and React. The back-end ran on a Linux server equipped with an Nvidia GeForce RTX 3090 GPU.

2.4 User Study

We conducted a study to validate whether GANzilla can enable users to discover directions that steer a GAN model to edit images for specific purposes.

2.4.1 Participants

We used convenience sampling to recruit 12 participants from a local university (eight male, four female, aged 23 to 31). Eight participants majored in electrical and computer engineering, one in bioengineering, one in medicine, one in mechanical engineering and one in engineering management. Eleven participants had programming experiences (3 to 15 years) and three had programmed or used GAN-enabled applications before (P5, P6 and P7)[3].

2.4.2 Tasks & Procedure

Each participant performed two blocks of editing task using GANzilla, each consisting of three trials.

- **Closed-ended tasks**. In each trial, we provided a participant with a set of image pairs

[3]We decided not to exclude these participants because GANzilla was meant for complementing existing algorithm-driven direction discovery and users who work on GAN development should be able to use and benefit from GANzilla as well.

where each pair showed an image before and after editing (Figure 2.5), which hereafter are referred to as *reference* and *target* images, respectively. The participant's goal was using GANzilla to find a direction that could replicate the edits, *i.e.*, transforming the reference images into ones that were as similar to the target images as possible.

- **Open-ended tasks**. In each trial, we provided a participant with a set of images and a high-level editing goal—specifically, making all the faces old, happy, and surprised. The goals were intentionally open-ended so that the participant had to come up with specific edits based on their own interpretation of the goal and use GANzilla to discover directions accordingly.

Each study started with an introductory tutorial of GANzilla, followed by a brief practice session for each participant to try out GANzilla using a toy dataset. We then continued with the block of open-ended tasks, after which the participant would take a short break before performing three trials of closed-ended tasks[4] (Figure 2.5). The order of the three trials within each block were counter-balanced across participants. Finally, we concluded the study with a semi-structured interview to elicit participants' qualitative feedback of interacting with GANzilla. The entire study took place over Zoom and lasted for about one hour and each participant was compensated with a $25 gift card.

2.4.3 Data & Apparatus

For the back-end we used the state-of-the-art StyleGAN2 [KLA20a]. Together with its predecessor, StyleGAN [KLA19], StyleGAN2 has been used for various applications including style transfer, data augmentation and image-editing. For data we used the Flickr-Faces-HQ (FFHQ) dataset, which was originally created for the StyleGAN as a benchmark. As our deep learning model, we used a pretrained model that is trained on FFHQ and released by

[4]In an earlier pilot study, participants found closed-ended tasks much more challenging; thus we always started with open-ended tasks to ease participants' learning curve.

Figure 2.5: Closed-ended tasks: the three sets of image pairs given to the participants where each column consists of a reference and a target images. A participant's task was to discover a direction that edits each reference image to best approximate the corresponding target image.

Nvidia for StyleGAN2 on their github page[5]. Other implementation details are introduced in § 2.3.5. We conducted the study virtually where each participant used Zoom's remote desktop control to interact with the GANzilla front-end running in a Chrome Web browser on the experimenter's desktop computer. The front- and back- ends were connected via a local area network to minimize latency.

[5]https://github.com/NVlabs/stylegan2-ada-pytorch

Figure 2.6: Open-ended tasks. Participants were given a text description of the tasks (making the face old, happy, and surprised). The reference images are given at the first column. Rest of the columns are generated by directions discovered by participants. For the same editing goal, participants discovered a wide variety of directions using GANzilla.

Figure 2.7: Closed-ended tasks. Participants were given the reference images (first column) and the target images (last column) to match as well as they can. The columns in the middle are generated by participants' discovered directions.

2.4.4 Measurement

We recorded the entire Zoom meeting including the screen recording. In addition to that, we saved every image participant generated throughout the study. We also logged all of the user actions with timestamps including what they highlighted, which buttons they clicked, which directions they have tested and saved.

For qualitative measures of open-ended tasks, immediately upon finishing each trial, we asked each participant to rate (along a 7-point Likert scale) how successful they thought

they had achieved the editing goal with the direction they found.

In the exit interview, participants started with an overall assessment of GANzilla based on their overall experience for both closed- and open-ended tasks. We asked *(i)* whether the tool is easy to use and *(ii)* whether the user can find directions that match their editing goal. Next, participants rated the cognitive load using the mental demand, effort and frustration dimensions in the NASA TLX questionnaire [Har86]. Next, we asked participants to ablatively evaluate the usefulness of GANzilla's individual UI elements: highlighting, the scatter/gather technique, changing the number of clusters, going back-and-forth across iterations, asking for more images, and live-testing directions on multiple images. All questions were rated along a seven-point Likert scale.

2.5 Quantitative Results

Figure 2.6 and Figure 2.7 show sample images edited by participants' discovered directions for open- and closed-ended tasks, respectively. Below we provide quantitative analyses to better understand participants' performance and behavior using GANzilla.

2.5.1 Closed-ended tasks

2.5.1.1 User performance

We calculated the cosine similarity between the target image and a participant-generated image (*i.e.*, edited by the participant's discovered direction) using VGG-Face's latent vector [PVZ15] extracted from the last layer. Cosine similarity results in between 0 and 1 where 1 represents a closer match between the vectors. Specifically, we ran two different analyses using this VGG-Face similarity metric.

In the first analysis, we tested whether a participant-generated image was more similar to the target image than the reference image is. As shown in Table 2.1, the similarities between

	Task 1	Task 2	Task 3
Reference	0.388	0.252	0.377
User-Generated	0.446 ± 0.123	0.422 ± 0.112	0.544 ± 0.140

Table 2.1: Closed-ended tasks: user-generated images (edited by participants' discovered directions) are more similar to the target images across all three tasks compared to reference images. Higher value represents a closer match between the vectors.

the participant-generated images and the target images (averaged across participants) were 0.446 ± 0.123, 0.422 ± 0.112 and 0.544 ± 0.140 for the three trials, respectively, all of which were higher than the similarities between the reference and the target images, which were 0.388, 0.252, and 0.377.

In the second analysis, we first sampled 1000 random directions to edit a reference image. We then calculated where a participant-generated image ranked amongst the 1000 randomly-edited images, in terms of their similarity to the target image. Results show that, in 33 out of 36 tasks (three tasks per participant $\times$ 12), the participant-generated images ranked top-5, which suggests that the participant-discovered directions got very close to the target image amidst the large GAN latent space (represented by the random samples).

2.5.1.2 User behavior

Overall, the average time to complete a closed-ended task is seven minutes and 16 seconds. Participants spent 37.4% of their time on performing scatter/gather interactions, 45.0% on testing a direction, and 17.6% on highlighting.

The average number of scatters per task is 1.64 and for an average of 0.53 times a participant went back to undo a scatter. We noted that some users scattered more than the

others. Four participants contributed to almost half (49.2%) of the total number of scatters. The average number of directions tested per task is 6.61. On average when these directions were being tested, the strength of the direction was changed 3.23 times.

On average, participants requested 1.81 times per task to change the number of clusters, which ranged from six to nine across all tasks. In comparison, participants only asked to sample more images 0.56 times per task.

2.5.2 Open-ended tasks

2.5.2.1 User performance

We conducted three folds of analyses:

First, we employed an open-source face analyzing tool called Deepface [SO21]. We extracted the age and emotion predictions as well as their respective confidence levels. Using this tool, we can analyze whether a user-generated image (compared to the reference image) resulted in an increase in age (old) or the confidence level in emotions (happy and surprised). Results show that on average age increased by 10 ± 3.12 years and confidence values for happy and surprised increased by $45.1\% \pm 25.7\%$ and $56.7\% \pm 22.9\%$ respectively.

Second, we used the CLIP model [RKH21] that embeds both text and image into the same latent space to make them comparable. We first computed the *image-originated* embeddings, which correspond to each participant's discovered directions by performing a subtraction between the CLIP embedding of the participant-generated image and that of the reference image. Next, we computed the *text-originated* embeddings: for each editing goal, we used the corresponding keyword ('happy', 'old', and 'surprised') to create the text embeddings in an approach similar to StyleCLIP [PWS21]. We then compare the image- and text-originated embeddings by calculating their cosine-similarities, which were 0.314 ± 0.131, 0.341 ± 0.192 and 0.500 ± 0.072, respectively, for the three editing goals.

Third, to put these similarity numbers in perspective, we sampled 1000 random directions

	Task 1	Task 2	Task 3
Top-10*	0.194 ± 0.099	0.285 ± 0.176	0.387 ± 0.172
User-Generated	0.314 ± 0.131	0.341 ± 0.192	0.500 ± 0.072

*Amongst 1000 random samples that approximate GAN's latent space

Table 2.2: Open-ended tasks: Similarity of user-generated images and top-10 randomly generated images compared to the CLIP *text-originated* embeddings. User-generated images are more similar to the given text ('old', 'happy', and 'surprised') across all tasks. Higher value represents a closer match between the vectors.

to represent the latent space and used the aforementioned Deepface tool to search for the top-10 directions that generated images with the highest increase in age/confidence in emotions. We then compared the embeddings of these top-10 directions with the aforementioned text-originated embeddings. We found that their averaged cosine-similarity values are 0.194 ± 0.099, 0.285 ± 0.176 and 0.387 ± 0.172, respectively, which are all lower than those achieved by user-generated images, as shown in Table 2.2.

The three analyses above suggest that participants' discovered directions reached a high semantic proximity to the editing goal. In addition, participants also felt positively about how they succeeded in achieving the editing goals, with reported scores (on a seven-point Likert scale) of 6.08 ± 0.76, 5.83 ± 0.99 and 5.75 ± 0.83, respectively for the three tasks.

2.5.2.2 User Behavior

We report the same set of behavioral measures as in the closed-ended tasks and compare the two, reporting statistical significance whenever there is any (otherwise any difference should be assumed as not reaching statistical significance).

Overall, the average time for a participant to complete an open-ended task is eight

minutes and 55 seconds, with 27.5% of their time spent on scatter/gather, 58.9% on testing directions, and 13.6% on highlighting. Participants used the test-field about two minutes more per task than the closed-ended tasks ($W = 32, p = 0.03$ based on a Wilcoxon signed-rank test), probably to ensure that the editing goal actually applied to more than one face for the open-ended tasks.

The average number of scatters per task is 1.17, about 0.5 smaller than that in the closed-ended tasks ($W = 8, p = 0.02$). Scattering more allows one to further refine a direction to better match the target image, which probably explained the higher number in the closed-ended tasks. Unsurprisingly, the same participants who scattered more in closed-ended tasks also scattered more here, contributing to over half (54.8%) of the total number of scatters. On average, users went back 0.5 times to undo scatter—a similar number as in the closed-ended tasks.

The average number of directions tested per task is 7.11. When these directions were being tested, the strength of the direction was changed for an average of 3.67 times. On average, participants requested 2.86 times to change the number of clusters, which ranged from eight to ten. Participants only asked for more images 0.388 times per task.

2.5.3 Workload measured by NASA TLX

For the mental demand dimension, the four participants (P2, P4, P7, and P9) who rated higher than four (neutral) considered the main workload as inspecting the small changes and differences amongst images during the scatter/gather process. As mentioned by P2, "You need to take a look at the images, small changes in them. Then you have to envision what to combine which increases the mental load". P2 is also the only participant who gave a higher-than-neutral rating of effort and the only one that rated frustration higher than three, which in part due to the need to spend a lot of time on some tasks because there were many options. Rating of P2 on frustration is an outlier based on the IQR analysis. Similar concerns are shared by P1 and P4: "Workflow is easy but still I had to click different

	User Ratings 1: Strongly Disagree; 7:Strongly Agree												
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	Mean & Std
Ease of Use	7	6	5	5	5	5	6	6	3	7	6	7	5.67 ± 1.15
Perceived Success	6	5	5	6	6	7	5	5	6	5	4	6	5.5 ± 0.80
Mental Demand	2	6	2	5	3	3	6	3	5	2	2	2	3.42 ± 1.62
Effort	3	6	3	5	4	2	1	4	3	5	4	2	3.5 ± 1.45
Frustration	1	5	3	2	3	1	2	1	2	1	1	1	1.92 ± 1.24
Highlighting	6	6	6	5	5	4	5	6	7	6	6	7	5.75 ± 0.87
Scatter/Gather	7	7	4	6	6	6	6	5	5	6	7	4	5.75 ± 1.06
Number of Clusters	3	4	6	2	5	6	5	5	6	5	6	6	4.92 ± 1.31
Back-and-Forth	5	4	4	6	5	7	6	6	7	6	6	7	5.75 ± 1.06
More Images	7	6	7	4	4	7	4	6	6	4	6	4	5.42 ± 1.31
Live-Testing Directions	7	5	5	7	7	7	7	7	7	7	7	7	6.67 ± 0.78

Table 2.3: The participants' ratings (Row 1-2: overall experience; 3-5: workload; the rest: usefulness of individual components). The questions are explained in § 2.4.4. All questions used a seven-point Likert scale. The outliers are found using interquartile range (IQR) analysis. P9 ease of use, P2 frustration, P4 number of clusters, P2 and P3 live-testing directions are outliers.

images and pay attention. If there were no images I liked, it got harder because I had to start thinking about which images have the right parts to scatter" (P1). "Paying attention to the details can be demanding, focusing on the patterns as well" (P4). P7 commented on the challenges of visualization of data: "The task is really useful but visualizing really high dimensional data is challenging which increases the mental load" (P7).

2.5.4 Summary of quantitative results

Analyses of the closed-ended tasks show that participants' discovered directions transformed the reference images into ones that are more similar to the target images and such similarities rank high when compared to edits done by 1000 randomly-sampled directions.

Analyses of the open-ended tasks show that participants' discovered directions achieve

the given editing goals as validated by Deepface's age and emotion detection; further, such directions highly align with the interpretation of these goals from a language perspective.

Analyses of user behavior across both tasks show that *(i)* a few (four) participants contributed to about half the total number of scatter/gather interactions; *(ii)* participants scattered/gathered significantly more in closed-ended tasks; and *(iii)* participants spent significantly more time testing directions in open-ended tasks.

Table 2.3 shows participants' ratings of GANzilla with respect to ease of use, perceived success, workload, and an ablative assessments of each component's usefulness. Next, we report participants' qualitative feedback behind these ratings.

2.6 Qualitative Findings

We employed a method akin to the Affinity Diagram approach [HB97], based on which we aggregated participants' responses to summarize their perceived ease and success of using our tool (§2.6.1) and surfaced recurring themes regarding how participants assess the usefulness of GANzilla 's individual components (§2.6.2). Specifically, the first author transcribed participants' responses to develop the initial codes, which were then reviewed by the second author. Disagreements were resolved via discussion between the two authors.

2.6.1 Overall assessment

2.6.1.1 Ease of using the tool

When asked how GANzilla was easy to use, all but one (P9) participant gave a rating above five. For example, P1 said: "I don't have to remember most of the workflow. It is just highlight and then click on images based on what you are searching." Both P3 and P5 commented on the intuitiveness of the UI. Some participants pointed out that there was a learning curve mainly due to the inevitable randomness of the sampling process (P3 and

P6), *i.e.*, participants needed to learn how to develop a strategy of using GANzilla based on the sampled directions given to them. P9 gave the only below-five rating and thought that it was hard to know how to improve the directions without getting overwhelmed by the sheer amount of information ("too many faces"). Rating of P9 is an outlier based on the IQR analysis.

2.6.1.2 Perceived success of the tasks

When asked how they felt successful that they found the directions to achieve their goals, all but one (P11) participant gave a rating above five. Even P9 who did not feel GANzilla was easy to use considered the task successful: "At the end of all tasks, I found a direction. They were not exact but close". Participants' responses also pointed out nuances between types of tasks—"Open-ended tasks were easy to achieve. Closed-ended were harder" (P4) and nuances between different stages of a task—"It is easy to find the main direction (bigger smile) but getting all the secondary changes are challenging" (P7). P11, who gave the only below-five rating, reflected on their usage strategy—"Sometimes I felt like it did not match the target very well. Maybe, I needed to iterate/scatter more. This is especially the case for closed-ended tasks" (P11).

2.6.2 Ablative assessment of individual components

2.6.2.1 Highlighting an area to focus the editing on presented limited usefulness

For open-ended tasks, sometimes participants did not know which part to edit before starting explorations, as mentioned by P1: "It seems if I know exactly what I am looking for it is helpful. But there are some cases where it is not obvious so I can't imagine where to highlight." Perhaps a more noticeable issue of this component was GAN's entanglement problem (discussed in more details in § 2.7), as participants noticed that sometimes the highlighted part was not guaranteed to be majorly edited (P2 and P4) and sometimes non-

highlighted parts were also changed (P6 and P7). Interestingly, one participant (P2) reported using this component, not to instruct the GAN model, but to remind themselves to focus on specific parts they wanted to edit.

2.6.2.2 Scatter/gather helps to combine directions with different features

Multiple participants (P1, P4, P5, P10, and P11) mentioned this usage, *e.g.*, "There were a lot of cases that it was useful. For example I wanted to change both eyes and mouth but some clusters had only eyes and some clusters only had the mouth. I could leverage scatter to get both." (P1) Participants also pointed the need for finer-grained gather, *e.g.*, "It would be nice if we could choose individual images instead of clusters. Because sometimes I did not want to choose the entire cluster." (P2) Interestingly, one participant pictured scatter as "zooming to that region" (P12), which could inform them if the gathered directions "are bad".

2.6.2.3 Changing the strength both positively and negatively in the testing field helps one better understand the direction

The ability to live-test a direction on multiple images was rated the highest amongst all components. Foremost, participants valued such a test of a direction's generality (P4, P5, and P6), as pointed out by P6: "I could also see the directions on other images. So I have an idea about how well it works generally." Participants also realized the importance of exploring the right strength of applying a direction, *e.g.*, "The previous steps help me to find the direction but you still need to figure out the strength" (P1), "You can experiment with the magnitude of the vector. It allowed me to try different combinations and helped me build my intuition about the direction" (P12). To our surprise, many participants (P1, P4, P5, P6, P8, P10, and P12) heavily used the functionality of setting a negative value on the strength of the direction, which essentially allowed them to observe what happens if

they go in the reverse direction. Specifically, seeing how a direction works in reverse was "informative" (P10), helped participants "validate" (P5) or "understand" (P4, P8, and P11) a direction better and "convince" (P12) themselves that it was the right direction. P6 even employed a strategy that leveraged such negative strength: "... in the last task I could not find the asked direction. Instead I found an opposite direction and used a negative weight on the test area." Amongst the two outlying scores (5), while P3 did not state anything specifically negative, P2 pointed out that he did not find testing highly useful because he could already anticipate how the direction would likely fail in some cases.

2.6.2.4 Changing the number of clusters and requesting more samples help mitigate randomness

Multiple participants (P1, P2, P5) pointed out the inherent randomness of sampling directions and considered that asking for more images was a back-up solution (P3, P6, P8, and P9) that helped when they could not find a direction that they were looking for. Sampling randomness also affected the quality of the clusters, as pointed out by a few participants (P1, P4, and P7). Changing the number of clusters helped them make sense of the clusters. For example, P12: "It helped me to choose better groups because when you increase number of clusters, clusters become better refined". However, as pointed out by P1, one trade-off was having to track changes in the clusters as the number was changed. P4 did not find changing the number of clusters useful because he 'would rather see as many images' as he could and thus always set the max number (10). Rating of P4 is an outlier based on the IQR analysis.

2.7 Discussions & Future Work

We discuss several issues in the current system and possible solutions for future work.

Limitations of the current study. First, future work could increase the number

of participants and the number of tasks (*e.g.*, via a out-of-lab deployment) beyond the current controlled study. Second, three of our participants had prior knowledge of GAN. Although anecdotally we did not observe any difference in how these three participants used GANzilla, future work should still strive to focus on a narrower user group (*e.g.*, product designers who use GAN to formulate ideas). Finally, to ease participants' learning curve, we fixed the order of the tasks to be open-ended (easier) first then closed-ended (harder). To verify whether there is an ordering effect, future work could extend our study with a counter-balanced design. For stylizing faces, mouth area is usually the most expressive. However, there were other changes in our closed-ended tasks. For example, Task 3 had more 'squinty' eyes and Task 2 had more makeup after the direction was applied. We do recognize such changes are more subtle compared to mouth and will address this limitation in our future work by introducing tasks that involve more significant changes in non-mouth areas.

Addressing entanglement issues in GANs. Entanglement refers to a long-standing phenomenon in GAN direction discovery: if a feature is changed and another unintended feature is also changed, these two features are said to be *entangled*. For example, while trying to make someone look happier, the image might also appear younger. In this example, the feature happy and young are entangled. In general, we want to discover directions that are disentangled.

To mitigate this issue, we used the state-of-the-art StyleGAN2 and found the directions in StyleSpace which is significantly more disentangled than the latent space. However, entanglement still existed in our study and was pointed out by two participants: "... when I try to make someone happy, their skin tone also changes" (P4); "I focus on the mouth but I get variety of eyes." (P5) Interestingly, some participants used the scatter/gather technique to mitigate entanglement issues. For example, if the goal is to find a direction that results in a bigger mouth, by scattering a cluster that has the bigger mouth feature, all the entangled features in the cluster are scattered too. This allowed participants to choose a more disentangled direction.

In the future, we can also preprocess the StyleSpace itself to address entanglement. Instead of randomly sampling various dimensions, we can be more selective about the sampling procedure. This can be achieved by uncovering dependencies between dimensions, so that related dimensions are selected together rather than entangled ones. Another idea is to let users inform the system about the entanglement issue in a secondary highlighting step. Then we can try to remove the StyleSpace parameters that are related with what user highlighted. An iterative process between the tool and the user can result in more disentangled directions, which is left for future work.

Providing users with more guidance and explanation. While GANzilla makes the generative process controllable as a whole, participants nonetheless requested more guidance and explanation on the specific steps in this process. For example, participants wished the tool could provide some guidance when they were stuck (*i.e.*, faced with clusters that contained no directions related to their editing goal). One possible idea for future work, in this case, is to guide the user with a shortcut that jumps to a previous step that contains directions most different from the current ones. Another popular suggestion is to help users to keep track of how clusters change and differ. For example, displaying a heatmap next to each image so that the changes/differences become more salient to human eyes. One participant (P7) also suggested feedforward visualizations to help users preview what they will get if they perform certain actions. Further, future work can introduce a recommender component that retrieves directions in previously-unsampled space based on what clusters a user currently gathers.

Integrating GANzilla with algorithm driven direction discovery. Currently, we do not leverage algorithm-driven directions and instead let the users discover them from scratch. Although this approach worked well in our studies, using prior work to discover directions can also be beneficial to the user experience. Some of the sampled directions can be filtered or better understanding of the editing space can be achieved. In the future, users can look through algorithm-driven directions and then can decide to use GANzilla if

their needs are not satisfied. These directions can be analyzed by our back-end to improve GANzilla sampling and they can even become part of scatter/gather functionality. We consider GANzilla complementary to existing approaches [HHL20, JCI19] and thus did not compare their results. To address this, our future studies will incorporate state-of-the-art methods to show how user-driven directions might result in different edits than algorithm-driven approaches.

Improving scatter/gather with more engagement We observe that not all participants fully leverage scatter to 'dive deep' into the StyleSpace but rather only scattered a few times. This is partly because it is challenging to manage highly-branched out scatter/gather paths as the number of iterations increases. Future designs could incorporate a tree-like UI structure of directions: every time the user scatters, a new branch is created. Another approach in the future can be to build the tree to guarantee hierarchical semantics. This can allow users to traverse a semantically more meaningful tree and and go 'deeper' in to the StyleSpace for more specific directions. As it becomes easier to manage the scatter/gather iterations, we can further support exploring multiple directions at the same time (*e.g.*, making a mouth open and eye brows raised).

CHAPTER 3

User-Driven Direction Disentanglement in Generative Adversarial Networks

3.1 Introduction

Generating complex data is a long-standing problem in computer sciences. In recent years, generative adversarial networks (GANs) are shown to be suitable for such tasks including medical imaging [YWB19], data enhancement [PBS17, WYW18] and image editing [ZKS16]. Moreover, human-AI collaboration is shown to be useful in various areas including medicine [GLX22, GHH21] and art creation [MH21]. Unfortunately, GANs function as 'black boxes' by their nature. As a result, end-users have little control over the generative process which limits human-AI collaboration. When an end-user with creative intent uses a GAN model for editing, the lack of control over the capabilities of the model can lead to inadvertent and inconsistent results. Moreover, there is little support for the end-users to *improve* said controls (*editing directions*)– a set of input parameters that steers the GAN model toward the intended characteristics with varying levels (e.g. making the eyes on the input image bigger or smaller while preserving other characteristics). Specifically, a common problem is that directions can be *entangled*, *i.e.*, while the direction changes the desired attribute, it might change other unintended attributes as well (*e.g.*, the direction that adds glasses to the people can also change their gender). Entanglement can also result in a direction that only works on a certain type of image (*e.g.*, the direction that adds glasses to the people, may not be able to apply on young people).

Improving an entangled direction, *i.e.*, *disentanglement*, is an active research area in GAN research. InterFaceGAN [SYT20] finds entangled directions using pre-trained classifiers and disentangles them via subspace projection. However, InterFaceGAN requires developers to come up with disentanglement rules for each domain and use human annotators to label large datasets. There are also GAN architectures that have improved disentanglement properties over the traditional GAN [GPM20] such as StyleGAN [KLA19], and GANformer [HZ21]. StyleGAN achieves better 'attribute separation' by introducing 'styles' which is inspired by the style transfer literature. GANformer has better 'spatial decomposition' since it leverages transformers [VSP17]. Despite their promises, these algorithm-driven solutions cannot capture a user's intent of disentangling specific attributes of a given image; thus it is important and complementary to support a *user-driven* approach, especially when algorithms fail to achieve their desired disentanglement effects.

To this end, we design and implement GANravel, a tool that allows users to interactively and iteratively disentangle directions. In order to highlight how GANravel can enable users to disentangle directions, we focus on two studies: *(i)* the common task of editing human faces and *(ii)* the creative task of creating memes of dogs. GANravel can use any image-generating GAN model that has disentangled directions. We complement different GAN models –StyleGAN2 [KLA20b] and FastGAN [LZS20]– in the studies to underline the model-agnostic nature of GANravel.

GANravel achieves disentanglement through two main approaches, *global* and *local* disentanglement. Global disentanglement focuses on the holistic attributes (*e.g.*, gender, age, lighting, etc.) and aims to disentangle a direction by balancing these attributes. This is achieved by users' selecting *exemplary* images that carry the entangled attribute and adjusting the weights of these images. Complementarily, local disentanglement focuses on attributes of specific components (*e.g.*, hair style, eye size, smiling, etc.) which can be more subtle than the global attributes. Therefore, they are inherently harder to balance by tuning exemplary images. Instead, users disentangle local attributes through masking where the

user highlights a region of entanglement. Then, the masks are used to find the entangled attribute in a one-shot manner (from a single mask without training).

To validate GANravel, we conducted two user studies with 16 participants each. The first user study had coarse- and fine-grained tasks. The second user study had three tasks where we compared GANravel with state-of-the-art user-driven direction discovery method GANzilla. In the first user study, coarse-grained tasks were adding glasses, making the face smile more, and making the face older. Fine-grained tasks were adding lipstick, making eyes bigger, and making hairs curlier. Participants were asked to find the target direction. Results show that participants were successful in finding disentangled directions. Specifically, in the coarse-grained tasks, participants found directions that were more disentangled than the directions that are found with state-of-the-art methods. Disentanglement was measured with two main analyses: *(i)* facial identity preservation and *(ii)* facial attribute classifier based metric. Results show that our metrics were 'similar' across tasks and participants felt satisfied with their edits and disentanglement performance. Participants' perceived disentanglement success was also aligned with our iterative disentanglement metrics which showed that a directions got more disentangled as participants iterated it more with GANravel. In the second user study, participants were tasked to create dog memes by finding disentangled directions. Participants used GANravel to disentangle directions they found using GANzilla. Disentanglement was measured through a classifier based metric. Results show that participants were able to disentangle directions which was also aligned with their perceived success.

GANravel makes a **tool contribution**, enabling a user to interactively and iteratively disentangle a direction and improve generated results. GANravel provides two user-driven approaches for disentanglement, encompassing global and local disentanglement. Overall, GANravel complements existing GAN architectures and the user study showed that resulting directions were more disentangled compared to the state-of-the-art direction discovery methods.

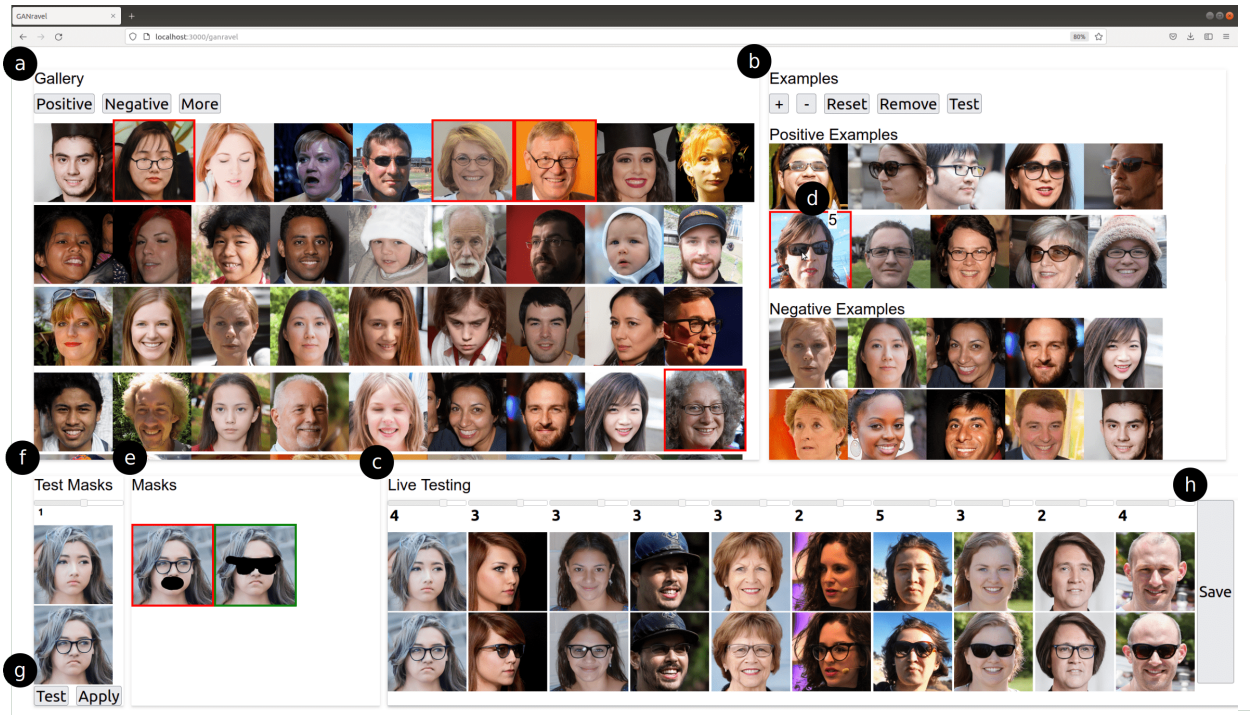

Figure 3.1: GANravel enables users to disentangle editing directions in generative adversarial networks (GAN) using global and local disentanglement approaches. (a) A direction is often entangled when created by selecting exemplary images from the gallery. (b) The weights of the exemplary images can be adjusted to disentangle global attributes such as age and gender. (c) The direction can be tested on the live-testing section using multiple test images. (d) The user can hover over an exemplary image to see its weight and go back and forth between weight adjustments and live-testing until global attributes are disentangled. (e) The user can use masks to disentangle local attributes such as glasses and closed mouth. (f) The masks can be combined to either preserve or discard a region of interest and they can be tested. (g) Resulting disentangled direction can be applied to other test images in the live-testing section. (h) The final disentangled direction can be saved and applied in other future images.

3.2 Background & Related Work

In this section, we first review background information about GAN, editing directions, and entanglement. Then, we review two areas of prior work: *(i)* existing algorithm-driven approaches for direction discovery and *(ii)* existing approaches that enable users to interact with GAN.

Generative adversarial networks As shown in Figure 3.2, a typical GAN [GPM20] consists of two neural network models: the generator (G) and the discriminator (D). G is trained to generate synthetic data x_g from a data domain such as human faces. At each training step, the goal of G is to 'trick' D by creating x_g that is indistinguishable from the data from the aforementioned data domain (x_r), and the goal of D is to distinguish synthetic data x_g from the real data x_r. By playing this zero-sum game, both networks are trained based on the prediction of D until G is trained to generate realistic data. By default, G generates x_g from a randomly-sampled noise vector (z). GAN functions as a 'black box' because the space where z resides is considered to be highly nonlinear. As a result, end users have very little control over the generative process.

GAN editing direction Formally, the generator learns a mapping function $f : Z \mapsto X$ where $Z \in \mathbb{R}^n$ and X is the space of the data domain. n is the dimension of the input vector which depends on the generator model. Z is referred to as the 'latent space'. Typically, when G is trained, latent space is sampled from a Gaussian distribution [GPM20]. The resulting vector (z in Figure 3.2) can be moved in the latent space to change the output in a semantically meaningful way along the editing direction (d in Figure 3.2). For example, given a face without glasses $f(z_0)$, the initial vector can be edited so that the output $f(z_1)$ is the same face as $f(z_0)$ but wearing glasses, where $z_1 = z_0 + \lambda d$. The coefficient λ is the 'strength' of the edit when d is normalized.

Entanglement in editing directions Entanglement is when a direction d changes multiple semantic attributes of the output simultaneously. For example, if $f(z_0)$ outputs a

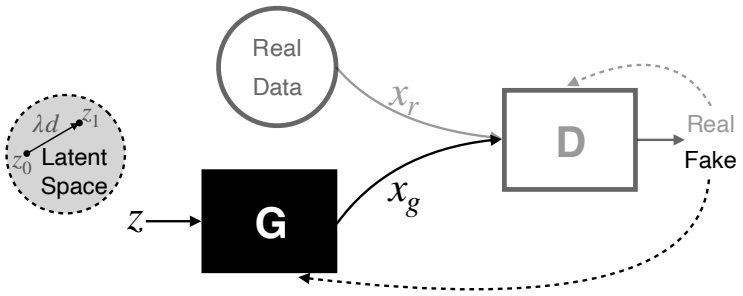

Figure 3.2: A typical GAN model and its' training scheme. Dashed lines indicate the gradients that train the generator (G) and discriminator (D). In this training step, G failed to trick D and the weights of G are updated accordingly. z controls the generated data x_g. The direction (d) is defined as the vector that changes the input z with addition. When d is normalized, λ is referred to as the strength of the direction.

face without glasses, and $f(z_0 + \lambda d)$ outputs the same face but with glasses and curlier hair, then the facial attributes 'glasses' and 'curly hair' are considered to be 'entangled' for the direction d. In this example, while trying to edit the face to add glasses, another attribute is unintentionally changed. Entanglement can happen through the direction discovery process or the generative model can introduce entanglement due to bias. *Disentanglement* is the process of improving the entangled direction, resulting in a direction that only changes the intended attributes.

Algorithm-driven discovery in GAN There are many supervised direction discovery approaches that require a large labeled dataset [JCI19, GAO19, SGT20, PBH20, YSZ21, RMC15, TTP21]. For example, InterFaceGAN finds disentangled directions by using classifiers and subspace projection [SGT20]. The downside of these approaches is that they rely on attribute predictors or human annotators. Relying on predictors or annotators severely limits the number of directions, while GANravel provides arbitrary range of directions through disentanglement. Recently, there has been some interest in finding directions in an unsupervised manner. Voynov *et al.* finds directions in an unsupervised manner by training a 'reconstructor' that predicts the strength and the index of a randomly sampled

direction [VB20]. SeFa reveals underlying variation factors in an unsupervised manner using closed-form factorization [SZ21]. GANSpace finds unsupervised directions using PCA on the feature space [HHL20]. Collins *et al.* applies k-means to the hidden layer activations of the generator to find a decomposition of the generated output into semantic objects. Then, the generative model is able to transfer a style of a facial attribute in one image to another image using the decomposition and the respective style parameters [CBP20]. Although there are many algorithm-driven direction discovery methods, there is little support for improving the directions through disentanglement. GANravel enables users to disentangle a given direction with simple interactions.

Enabling users to interact with GAN Interactions with generative adversarial networks have been an active research area. Typically, the users interact with GANs through sliders similar to GANravel. We decided to use sliders in the live-testing area following prior work [SYT20, AZM21]. Dang *et al.* compared the regular sliders and sliders that provide feedforward information ('filmstrips') in a comparative study [DMB22]. Zhang *et al.* enabled users to explore the latent space with a grid-like view of sampled images [ZB21]. The users can zoom in or out, pivot, and pan to explore the latent space. Even though the gallery section of GANravel is not spatially meaningful, it also leverages a grid of images for selecting exemplary images. Chiu *et al.* created a tool that allows the user to search through the GAN latent space interactively with a one-dimensional slider [CKL20]. There are also various prior works that can enable new user interactions. Heim showed that GANs can iteratively accept inputs to 'generate an image more like A than B' [Hei19]. Chen *et al.* showed that it is possible to generate human faces from human face sketches [CSG20]. Cheng *et al.* developed a visual design assistant that interacted with the users through natural language and edited the GAN outputs [CGL20]. Ling *et al.* allows users to edit images leveraging segmentation masks of the images [LKL21]. StyleCLIP accepts a textual description of the direction and finds it using the CLIP model [PWS21, RKH21]. GANzilla allows users to discover directions via iterative scatter/gather interactions and complements

other direction discovery methods [EC22]. GANzilla is a 'complementary' solution to the algorithm-driven discovery methods. It does not find a target direction or improve disentanglement, whereas GANravel is a user-driven disentanglement tool that provides more flexibility to disentangle user-defined directions. GANzilla uses thumbnail images to represent directions and uses a brush tool to highlight a region of interest which is similar to how users navigate in GANravel and highlight a mask for local disentanglement.

3.3 Design & Implementation

In this section, a detailed walkthrough of GANravel's design and implementation is given using an exemplary use case of adding glasses to the face. Similar to GAN Dissection [BZS18], GANravel uses the *filters*[1] in GAN models to edit images. Specifically, we extract feature maps from each filter and define the directions at the filter level. Specific to StyleGAN, we define the directions in StyleSpace. StyleSpace refers to the space of style parameters that scale the outputs of convolutional filters of StyleGAN2. StyleSpace is considered to be more disentangled than the latent space which makes it suitable for GANravel [WLS21a]. For the other GAN models, we define a space similar to StyleSpace. Each filter has a value associated to it which is found by averaging their feature map. When all of the associated values are concatenated, it creates a vector which we define as the direction. In other words, each dimension of the direction represents a filter in the GAN model similar to StyleSpace. These directions can then be applied to other reference images similar to [BZS18], where the feature maps are increased or decreased by the respective dimension of the direction. GANravel consists of two main disentanglement approaches, global and local disentanglement which are detailed in § 3.3.1 and § 3.3.2 respectively.

[1] referred to as neurons or units in [BZS18] interchangeably.

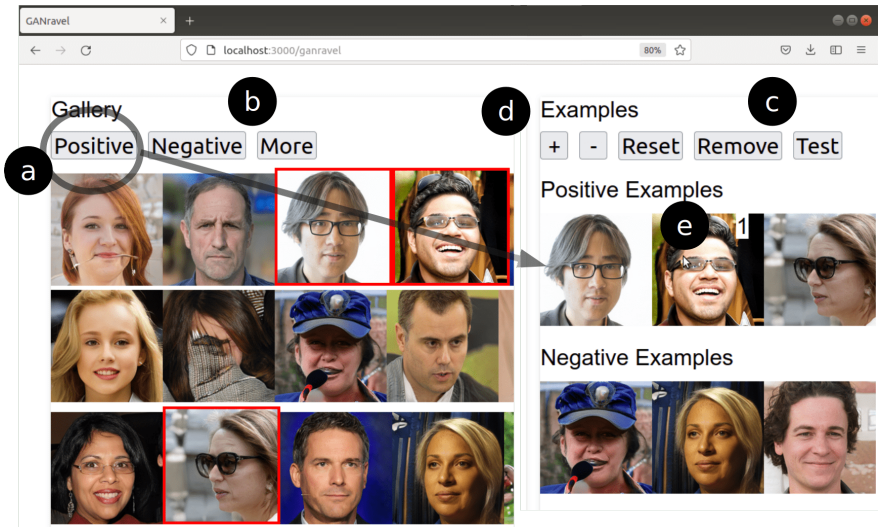

Figure 3.3: Exemplary image selection in GANravel. (a) The positive and negative examples can be selected from the gallery. (b) Users can request more images. (c) The resulting direction can be tested on test images using the 'test' button. (d) The weights of the examples can be changed using the '+' and '-' buttons.

Starting with an entangled direction The workflow of GANravel starts with an entangled direction which can be achieved by selecting a handful of positive and negative exemplary images from the image 'gallery' (Figure 3.3a). Users can also request more images (Figure 3.3b). Positive examples carry the target attribute (glasses) whereas negative examples do not. After the selection is complete, the current entangled direction can be tested in the 'live-testing' area (Figure 3.1c) using the 'test' button (Figure 3.3c).

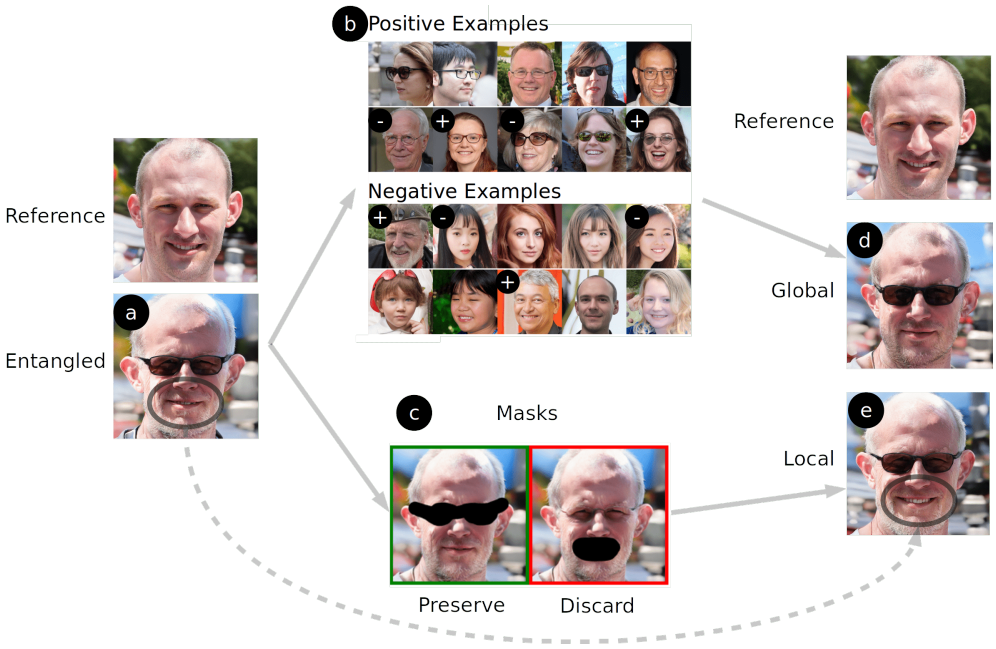

Figure 3.4: Global and local disentanglement. (a) The resulting image when an entangled direction (glasses entangled with age and mouth) is applied to the reference image. (b) Global disentanglement. The weights of the young images with glasses are increased to disentangle age. (c) Local disentanglement. The area of the intended direction (glasses) is preserved while the area of the entangled attribute (mouth) is discarded. (d) The resulting image when the globally-disentangled direction is applied to the reference image. Age is disentangled but mouth is still entangled. (e) The resulting image when the locally-disentangled direction is applied to the reference image. Mouth is disentangled but age is still entangled.

3.3.1 Global Disentanglement

The edited test image with the entangled direction can be seen in Figure 3.4a. As it can be seen, the glasses direction is entangled with age (person gets older). In order to disentangle glasses direction from age, the weights of exemplary images need to be adjusted which can be done by the '+' and '-' buttons (Figure 3.3d). The weights can be seen when the mouse hovers over the images (Figure 3.3e). Since the entangled attribute is age, the weights of

young positive images are increased (Figure 3.4b). Another approach is to increase the weight of old negative images. In an earlier pilot study, we discovered that global disentanglement requires back and forth between weight adjustments and live-testing. It is also beneficial to look for new exemplary images (*e.g.*, positive images that are young and have glasses). After changing the weights to balance out the entangled feature age, the edited test image with the globally-disentangled direction can be seen in Figure 3.4d. As it can be seen, the resulting glasses direction is disentangled from age, but still has some entanglement issues (*e.g.*, mouth is closed, beard, etc.). These subtle issues are harder to disentangle with positive/negative examples and weight adjustments. They are instead fixed with local disentanglement.

Implementation First, the vector (associated with the filters) of each selected image are extracted. The weighted average vector of positive examples is subtracted from the weighted average vector of negative examples, resulting in the discovered direction. Initially, all the positive examples have a weight of 1 and all the negative examples have a weight of -1. By increasing the absolute value of positive or negative weights, the user can increase the effect of individual vectors or vice versa. Since the user only selects a handful of images, it is not required for the user to select negative examples to save time and effort. Instead, each selected vector is subtracted by an 'average' vector which is created by averaging 10000 vectors that are extracted from random image samples. As a result, even if the user only selects positive examples, the difference between selected vectors and average vector carries enough information to find the target direction.

3.3.2 Local Disentanglement

In addition to global disentanglement, GANravel can also disentangle local attributes. This is achieved by double clicking the test image with entangled attributes which pops up a brushing tool for highlighting. After the region of interest is highlighted, the mask will appear in the 'masks' section (Figure 3.1e). The masks can be used in two ways: *(i)* to 'discard' the attribute and *(ii)* to 'preserve' the attribute. The discard feature changes the direction

to be *less* affected by the masked area and the preserve feature allows the direction to be *more* affected by the masked area. For example, considering the same entangled direction in Figure 3.4a, it can be seen that glasses direction is entangled with not only age (global) but also closed mouth (local). The masks that are used to 'preserve' the glasses and 'discard' the closed mouth can be seen in Figure 3.4c. The edited test image with the locally-disentangled direction can be seen in Figure 3.4e. As can be seen, the edited image still has the glasses which is achieved by the 'preserve' mask. Moreover, the mouth is no longer closed similar to the reference image which is achieved by the 'discard' mask. However, the direction is still entangled with age which is a global entanglement issue. The masks can be tried in combination by selecting them and clicking the 'test' button (Figure 3.1g). Each click cycles through green, red, and no selection for the masks. If the user wants to test the masked direction on all the test images, the 'apply' button can be used (Figure 3.1g). Finally, the directions can be saved with the 'save' button (Figure 3.1h).

Implementation. When a region is highlighted, the filters that are responsible for that region can be found. We extract all the outputs of the filters as feature maps. We then calculate the overlap between the images and the mask, scaling the mask for each layer. After normalizing the overlap, we use the overlap value as the 'importance' metric for that particular filter. Higher values indicate that the filter is highly 'responsible' for the region of interest. For the 'preserve' feature, we scale the vector with the importance metric which results in more 'important' filters having stronger influences on the output. For the 'discard' feature we use the same importance metric but we inversely scale the vector which results in more 'important' filters having weaker influences on the output. This simple yet effective trick allows us to disentangle a direction in a one-shot manner without any training or data in real-time.

3.3.3 Other Implementation Details

We used Nvidia's implementation of StyleGAN2 in PyTorch and their pre-trained model for the first user study. We trained a FastGAN model for the second user study. We extended the PyTorch code of StyleGAN2 and FastGAN to be able to extract style parameters (only in StyleGAN2) and filters which are used in disentanglement. We used Python for all of the back-end calculations and Flask for our web-framework. The front-end of GANravel was developed on Javascript, Node.js, and React. The back-end ran on a Linux 18.04 server which is equipped with an Nvidia GeForce RTX 3090 GPU.

3.4 User Studies

We conducted two user studies to validate whether GANravel can enable users to disentangle directions that edit images for *creative purposes*. In the first user study, we asked participants to find and disentangle directions for human faces using GANravel in two sets of tasks: *(i)* coarse-grained and *(ii)* fine-grained tasks. The goal of the first user study is to analyze the *disentanglement performance* of GANravel with respect to the state-of-the-art methods on the standard face editing task. In the second user study, we asked participants to create dog memes using GANravel, GANzilla and combination of both. The goal of the second user study is to put emphasis on the tool contribution of GANravel by enabling users in a creative task of generating memes. The second user study also shows how GANravel can disentangle directions that are found by another direction discovery method such as GANzilla which is the most similar work to ours.

3.4.1 Editing Human Faces

The details of the first user study is provided below.

Participants. We used convenience sampling to recruit 16 participants from a local uni-

versity. Out of 16 participants, twelve were male, four were female, and they were aged from 23 to 33. Seven participants majored in electrical engineering, two in biomedical engineering, five in economics, and two in computer science. All of the participants had programming experiences from five to 10 years and none of them had programmed or used GAN-enabled applications before.

Tasks & Procedure. Each participant performed two sets of tasks (coarse-grained and fine-grained tasks) using GANravel, and each task consists of three trials. In both tasks, the participant's goal was to use GANravel to find a direction that steers the output of the GAN towards the given editing goal while preserving other attributes.

- **Coarse-grained tasks.** In each trial, participants were given a generic editing goal. Specifically, the goals were: adding glasses to the faces, making the faces smile more and making the faces appear older[2]. The goals were intentionally generic so that they can be compared with the state-of-the-art face editing methods.

- **Fine-grained tasks.** In each trial, participants were given a specific editing goal. Specifically, the goals were: adding lipstick to the faces, making eyes bigger, and increasing the curliness of the hair[3]. The goals were intentionally specific which are not naturally supported by the prior works to show the flexibility of GANravel.

Each user study started with an introductory tutorial of GANravel. After the tutorial, the participants are given a brief practice session to try out GANravel using a toy example. We then continued with a block of tasks (either coarse- or fine-grained) which is followed by a short break. After the break, the participants are given the remaining block of tasks. The order of the tasks and the three trials within each block were counter-balanced across participants. We concluded the study with a semi-structured interview to elicit participant's

[2]Hereafter simply referred to as glasses, smile and age

[3]Hereafter simply referred to as lipstick, eye, and curliness

qualitative feed-back of GANravel. The entire study took place over Zoom and lasted for about an hour. Each participant was compensated with a $25 gift card.

Data & Apparatus. We used StyleGAN2 as our GAN model, specifically we used a pre-trained model which is trained on Flickr-Faces-HQ (FFHQ) dataset. The model parameters and its PyTorch code are available on Nvidia's github page[4]. Other implementation details are given in § 3.3.3. The user studies are conducted virtually over Zoom. Each participant used Zoom's remote control feature to interact with the computer of the experimenter. GANravel ran on the same computer to minimize latency.

3.4.2 Generating Dog Memes

The details of the second user study is provided below.

Participants. We used convenience sampling to recruit 16 participants from a local university. The participants of the first and second user studies did not overlap. Out of 16 participants, ten were male, six were female, and they were aged from 21 to 29. Five participants majored in electrical engineering, one in industrial engineering, three in biomedical engineering, two in economics, and five in computer science. All of the participants had programming experiences from five to 10 years and none of them had programmed or used GAN-enabled applications before.

Tasks & Procedure. Each participant performed three tasks using *(i)* GANravel, *(ii)* GANzilla and *(iii)* combination of both. Specifically, the participants were asked to generate two dog memes for each task. A meme consisted of two images side-by-side and text underneath them. The images were either the unedited reference images, or an edited image, which is generated by applying the discovered direction to the reference image. The text underneath them was written by the participants. We also let participants turn these

[4]https://github.com/NVlabs/stylegan2-ada

two images into a single GIF to animate the meme [5]. For the first two tasks, participants generated dog memes using GANravel and GANzilla separately. For the last task, participants disentangled the directions they found before (with GANzilla) using GANravel. In other words, they improved the quality of the memes that they already found before. The goal in the first two tasks is to compare the disentanglement quality of GANravel with a similar work to ours GANzilla. The goal in the last task is to show how GANravel can be used with another direction discovery method to disentangle (improve) directions.

Each user study started with an introductory tutorial of GANravel and GANzilla. After the tutorial, the participants are given a brief practice session to try out GANravel and GANzilla using a toy example. We then continued with a block of tasks (creating two memes using either GANravel or GANzilla) which is followed by a short break. After the break, the participants created two more memes using the remaining tool. After the second task, participants are given another short break. After the second break, for the third task, the participants disentangled the memes (previously found with GANzilla), using GANravel. The order of the first two tasks were counter-balanced across participants. Similar to the first user study, we had a semi-structured interview at the end. The entire study took place over Zoom and lasted for about an hour.

Data & Apparatus. We used FastGAN [6] as our GAN model, specifically we used Projected GAN [7] for training the model on Animal Faces-HQ (AFHQ) Dog [CUY20] dataset. GANzilla is publicly available on Github [8]. We adapted GANzilla for the trained FastGAN model. Other implementation details are given in § 3.3.3. Similar to the editing human faces user study, the user studies are conducted virtually over Zoom.

[5]We only reported the images in the paper. The GIFs as well as the source code will be made available on GitHub if the paper gets accepted.

[6]https://github.com/odegeasslbc/FastGAN-pytorch

[7]https://github.com/autonomousvision/projected_gan

[8]https://github.com/noyanevirgen/GANzilla-UIST22

3.4.3 Measurement

In both user studies as the participants interacted with GANravel and GANzilla, we saved every image generated by them. We also saved all of the user interactions, such as which buttons are clicked, and which images are selected with timestamps. We also recorded the entire session over Zoom.

In the exit interview, first we asked participants to assess GANravel based on their overall experience. Specifically, they are asked to rate (on a 7-point Likert scale) *(i)* whether GANravel is easy to use, *(ii)* whether GANravel can find directions that match their editing goal, and *(iii)* whether GANravel can disentangle an initially entangled direction. Next, participants rated the cognitive load using the mental demand, effort and frustration dimensions of the NASA TLX questionnaire [Har86]. Finally we asked participants to evaluate the usefulness of GANravel's individual UI components: selecting positive and negative examples, changing weights of positive or negative examples, live-testing directions on multiple images, highlighting and masking to create new directions.

3.5 Quantitative Results

In this section, we provide quantitative analyses to understand participant performance and behavior using GANravel and compare it with state-of-the-art baselines. There are multiple comprehensive analyses for the tasks including: *(i)* disentanglement performance comparison between the user-edited images and the images edited by the state-of-the-art baselines (§ 3.5.1), *(ii)* further analyses into disentanglement to surface trends in entanglement (§ 3.5.2), and *(iii)* user behavior (§ 3.5.3). The study of human faces in AI literature allows us to access labeled datasets, classifiers, and facial feature extraction tools. This makes it easier to analyze disentanglement performance using the first user study. We also performed analyses on the second user study, in which participants were asked to find editing directions for dogs, although these analyses were not as detailed.

3.5.1 Disentanglement performance

Measuring disentanglement accurately is an open question in GAN research. For the first user study, we measured disentanglement through two main analyses: *(i)* similar to [KGM22], we measured how well a direction preserves the facial identity, and *(ii)* we measured how much the intended facial attribute changed compared to unintended features using facial attribute classifiers. A disentangled direction should preserve the facial identity more since there are fewer facial attributes changing compared to an entangled direction. With a disentangled direction, the intended facial attribute should change more than the unintended features which can be measured with classifiers. For the second user study, we measured how well a direction preserves the breed of the dog. A disentangled direction is expected to preserve the breed more, similar to facial identity analysis in the first user study.

Baseline Methods. We quantitatively compared GANravel with four state of the art methods for editing human face user study: InterFaceGAN [SYT20], GANSpace [HHL20], StyleFlow [AZM21] and GANzilla [EC22]. GANSpace and StyleFlow are unsupervised direction discovery methods and InterFaceGAN is a supervised direction discovery method that leverages classifiers. GANzilla is a complementary user-driven direction discovery method. GANravel does not require a dataset or a classifier similar to GANSpace, StyleFlow, and GANzilla. On the other hand, GANravel can disentangle a given direction, similar to the conditional manipulation of InterFaceGAN. Although InterFaceGAN, GANSpace and StyleFlow do not have user interaction as their contribution, they produce state-of-the-art directions that are made available by the developers. Other than GANzilla, baseline methods are not available for the second user study. Instead we directly compared the disentanglement performance of GANravel, GANzilla, and the combination of both.

Calibration. One of the challenges was to calibrate the strength of the directions across methods. The individual strengths of the directions needed to be adjusted per method and direction so the faces' changes were comparable. For the first user study, we followed an

approach similar to [KGM22]. We leveraged VGG-Face [PVZ15] to find the smallest and largest limits of the applicable strengths where the faces could still be detected. We then divided this range into five intervals and used the resulting six images as the edited images. In total, we used 1000 reference images for the three aforementioned directions. Therefore, for each method, we had 1000 reference images $\times 3$ (coarse-grained) tasks per reference image $\times 6$ resulting images per task $= 18000$ total number of images for the analysis. We applied the same principle to the directions found by our participants. For the second user study, we applied the same principle where a simple *dog detector* is used which is trained with Kaggle's Dog dataset [Cuk13].

Analysis. We re-trained InterFaceGAN for StyleGAN2 since it was originally released for StyleGAN. The directions of GANSpace[9] and StyleFlow[10] are already available for StyleGAN2 on Github. For the editing human faces user study, we implemented GANzilla's scatter/gather functionality to find directions, since participants did not interact with GANzilla in the first user study. Due to the nature of unsupervised direction discovery, not every direction can be found by the baselines. Coarse-grained tasks (glasses, smile and age) are all supported by the baselines and they are used in our analyses to compare GANravel with the baselines for disentanglement performance. We also ran our analyses on fine-grained tasks and reported the metrics, even though they were not comparable with the baselines. For the second user study, we trained a FastGAN using the AFHQ Dog dataset and extended the code of GANzilla. The details of the user study can be found in § 3.4.2.

Facial Identity. Although facial identity is used as a measure of disentanglement in the literature, it suffers from certain entanglement types. A subtle entanglement (*e.g.*, bigger eyes) does not change the face as much as a global entanglement (*e.g.*, getting older). As a result, the facial identity metric should not be compared across tasks. We used a different face recognition model for facial identity analysis (FaceNet [SKP15]) than the model that is

[9]https://github.com/harskish/ganspace

[10]https://github.com/RameenAbdal/StyleFlow

used for calibration (VGG-Face). Because the calibration step can bias the facial identity similarity metric if they use the same model. First, we extracted latent vectors from the last layer of FaceNet for the reference images. Next, we extracted latent vectors from the six edited images that originated by calibration. Then, we calculated the cosine similarity between the reference latent vectors and the six edited latent vectors. We averaged the results across tasks. Cosine similarity is between zero and one. Higher values represent a closer match between the vectors and therefore represent a more disentangled direction. The results can be seen in Table 3.1. As it can be seen GANravel and InterFaceGAN have better facial identity retaining compared to GANSpace and StyleFlow. GANravel outperforms all the baselines. The values also differ across tasks. For example, age has lower values compared to glasses and smile. This can be explained by images going through more major changes with the age direction making it harder to retain identity. For the fine-grained tasks: lipstick, eye, and curliness the GANravel facial identity metrics are $.84 \pm .19$, $.85 \pm .21$, and $.73 \pm .22$ respectively. According to Mann-Whitney U test, there are no statistically significant differences between coarse and fine-grained tasks. According to Friedman-Nemenyi test, only the age direction has statistically significant difference after Bonferroni correction (p=.03), which indicates age direction changes the face more than the other directions.

Classifier-Based. Although facial identity is a useful metric for disentanglement, it does not entirely measure disentanglement. A face can go through minimal changes or the direction can be subtle in which case the face retains most of its identity. Another way to quantify disentanglement is to use facial attribute classifiers and measure the cosine similarity of their latent vectors after they are edited. First, we can extract the latent vectors of the edited and the reference images with a classifier that is trained for the goal of the direction. For example, if the direction is age, we can extract the latent vectors of the reference images and the edited images using an age classifier. Then, we can calculate the cosine distance between the vectors coming from the age classifier. We can do the same calculation using

Coarse-Grained	Glasses	Smile	Age
interfaceGAN	$.74 \pm .11$	$.78 \pm .13$	$.60 \pm .21$
GANspace	$.65 \pm .21$	$.76 \pm .15$	$.42 \pm .34$
StyleFlow	$.55 \pm .24$	$.69 \pm .15$	$.48 \pm .40$
GANzilla	$.58 \pm .22$	$.71 \pm .22$	$.45 \pm .39$
GANravel	$\mathbf{.84 \pm .19}$	$\mathbf{.86 \pm .18}$	$\mathbf{.67 \pm .23}$

Table 3.1: Facial identity metrics of GANravel with baselines IFG, GS, and SF for wearing glasses, smiling, and increasing age tasks. Higher values indicate higher disentanglement.

different face attribute classifiers such as baldness, hair color, face roundness, facial hair, etc. Since a disentangled direction should not change other facial attributes, it is expected to have a higher cosine similarity when different face attribute classifiers are used compared to the age classifier. However, in practice, it is costly to train many different facial attribute classifiers. Instead, we used an open-source face attribute classifier model called FAN [HFZ18] that can detect 40 binary attributes with one model including all six tasks in our study. FAN takes an image as an input and outputs a vector of size 40 that consists of 1s and 0s indicating whether that attribute is present or not in the image. We used FAN and recorded: *(i)* whether the targeted attribute (old) was detected, *(ii)* percentage of attributes that were lost (out of 40), and *(iii)* percentage of new attributes (out of 40) that were detected after the direction is applied. We averaged the results across all the coarse-grained tasks and the results can be seen in Table 3.2. Ideally, after the image is edited, the output of FAN should not lose any attribute, and it should not find any new attributes other than the targeted attribute.

According to Friedman's test, after Bonferroni correction, there is no statistically sig-

Coarse-Grained	Success (%) (↑)	Lost (%) (↓)	Found (%) (↓)
interfaceGAN	72.34 ± 18.15	6.32 ± 3.64	8.82 ± 4.12
GANspace	69.68 ± 21.64	8.74 ± 6.61	10.12 ± 7.12
StyleFlow	69.13 ± 27.16	11.98 ± 6.85	13.87 ± 6.91
GANzilla	66.99 ± 35.98	12.12 ± 7.35	10.33 ± 5.22
GANravel	$\mathbf{74.81 \pm 12.66}$	$\mathbf{3.41 \pm 4.33}$	$\mathbf{4.59 \pm 4.12}$

Table 3.2: Facial attribute classifier metrics of GANravel with baselines IFG, GS, and SF for wearing glasses, smiling, and increasing age tasks. Success percentage indicates how successful the direction is in adding the target attribute when applied. The lost percentage indicates how many facial attributes are lost when the direction is applied. The found percentage indicates how many facial attributes are introduced when the direction is applied. Lower values for lost and found indicate higher disentanglement.

nificant difference in success rates. This indicates that all methods introduce the targeted attribute with similar percentages. However, InterFaceGAN and GANravel have statistically significant lower lost and found attributes which is a result of their disentanglement capabilities (p=0.04 and 0.02 respectively). GANravel outperforms all the baselines. For the fine-grained tasks: the GANravel success, lost and found metrics are 63.91 ± 23.18, 6.00 ± 7.91, and 5.61 ± 11.23 respectively. According to Mann-Whitney U test, there are no statistically significant differences between coarse and fine-grained tasks. According to Friedman-Nemenyi test, only the success metric has statistically significant difference (p=.02) after Bonferroni correction. This can be explained by fine-grained tasks being more subtle by definition. That being said, lost and found metrics are similar which again shows that

the directions are disentangled regardless of the task.

Dog Breed. In order to measure disentanglement in the second user study, we used a dog breed classifier which is trained with the Oxford Dog dataset [PVZ12]. Similar to the facial identity metric in the first user study, a disentangled direction is expected to preserve the breed more. It should be noted that human faces are better studied and have better models (like FAN) than dogs. As a result, the analyses in this section should be viewed as complementary to the previous disentanglement performance analysis.

In the second user study, we asked participants to use GANravel, GANzilla, and the combination of both (GANzilla+GANravel) to create dog memes. In the last task, participants disentangled directions that they already found with GANzilla. Similar to facial identity metric, we extracted latent vectors of the reference images and the edited images and calculated cosine similarity for each task. Cosine similarities are between zero and one, where higher values represent a more disentangled direction. The results for GANravel, GANzilla, and the combination of both (GANzilla+GANravel) are $.91 \pm .12$, $.45 \pm .32$, and $.90 \pm .15$, respectively. According to Friedman-Nemenyi test, across three tasks, only GANzilla has statistically significant difference (p=.02) after Bonferroni correction. This shows that GANzilla is significantly worse at creating disentangled directions than GANravel. More interestingly, participants were able to disentangle the directions they discovered with GANzilla using GANravel. This analysis highlights how GANravel can be either used by itself or complementary to other direction discovery methods for disentangled direction discovery.

3.5.2 Iterative disentanglement

User-driven disentanglement is one of the biggest contributions of GANravel. Previously in § 3.5.1, we showed the disentanglement performance of the directions that users found at the end of each trial. In this section, we analyze how the disentanglement changes over time as the participants interact with GANravel. We provide insight into interactive

Over Time	Success (%) (↑)	Lost (%) (↓)	Found (%) (↓)
Glasses	0.29 ± 1.38	-7.61 ± 4.51	-5.80 ± 4.73
Smile	0.60 ± 0.69	-4.69 ± 6.54	-3.02 ± 3.51
Age	0.38 ± 1.01	-6.95 ± 7.82	-4.00 ± 5.51
Lipstick	1.08 ± 1.11	-7.95 ± 7.99	-6.12 ± 7.65
Eye	1.37 ± 0.92	-6.57 ± 4.66	-7.69 ± 8.10
Curliness	0.58 ± 1.56	-5.89 ± 5.99	-5.78 ± 6.11

Table 3.3: Facial attribute classifier metrics of GANravel for each trial when the final 'disentangled' direction results are subtracted from the initial 'entangled' direction. Positive values in success indicate improvement in the target facial attribute over time. Negative values for lost and found indicate higher disentanglement over time.

disentanglement in using classifier-based disentanglement over time.

For each trial when the users selected exemplary images and tested a direction for the first time, we saved the direction as the 'entangled' direction. After they interacted with GANravel to improve the direction, we saved the final version as the 'disentangled' direction. Previously in § 3.5.1, we reported the results for the 'disentangled' direction. We did the same analysis for the 'entangled' direction. Then, we subtracted the success, lost, and found percentages of the 'disentangled' direction from the 'entangled' direction and reported it for each trial. Positive values in success imply the *final* direction introduces the target facial attribute more often than the *initial* direction. Whereas, negative values in lost and found imply that the *final* direction is more disentangled than the *initial* direction. The results can be seen in Table 3.3.

According to Mann-Whitney U test, there is a statistically significant difference between the initial and final directions for both loss and found percentages (p = 0.02, 0.04, 0.02, 0.02, 0.02, and 0.03 respectively for lost) (p = 0.03, 0.03, 0.04, 0.02, 0.03, and 0.03 respectively for found). As the users interacted with GANravel, they disentangled the initial direction consistently for each trial. However according to the Mann-Whitney U test, there is no statistically significant difference between the initial and final directions for the success percentages. In other words, the target facial attribute is present when the initial direction is applied as well as the final direction. The improvement is in the disentanglement performance over time.

We also analyzed how the disentanglement performance improved over time. Every time the participants applied a disentanglement, global or local, we extracted the current direction and ran the same analysis we ran earlier in this section. The results can be seen in Figure 3.5. As it can be seen, the disentanglement performance increases over time (percentages get lower). Interestingly, on average it took 4.12 and 5.32 actions for participants to reach at least 90% of the maximum disentanglement they could achieve. This can be explained by participants focusing on global disentanglement at the beginning which is easier to observe with the metrics, since global disentanglement change the image more than local disentanglement. 73.9% of the first 5 actions consist of global disentanglement, which supports the previous observation.

3.5.3 User behavior

In this section, we report how the users interact with GANravel. Overall, the average time to complete a coarse and fine-grained task were 8 minutes 54 seconds and 8 minutes 29 seconds respectively. For the second user study, the average time to create a dog meme was 9 minutes and 12 seconds for GANzilla, 9 minutes and 21 seconds for GANravel, and 6 minutes and 42 seconds for disentangling the directions of GANzilla using GANravel. The faster times in the last task can be explained by participants not needing to choose

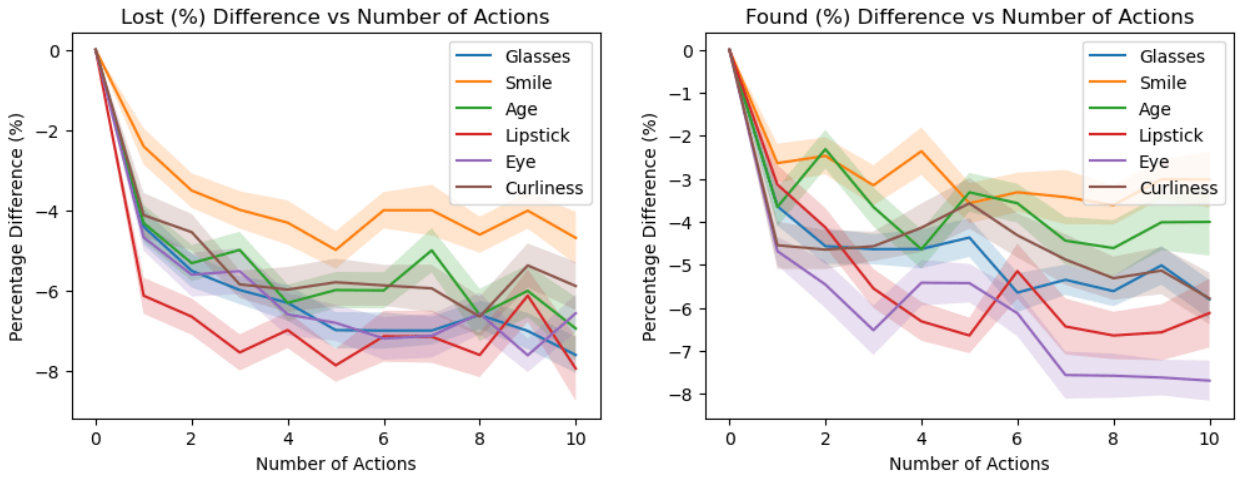

Figure 3.5: Lost and found percentages over user actions when the current direction metrics are subtracted from the initial entangled direction metrics. Lower values indicate higher disentanglement. The shaded regions represent 0.2 times the standard deviation to make plots readable. The values of tasks should not be compared with each other as discussed.

exemplary images. The results show that using GANravel from scratch to find a disentangled direction takes less time than disentangling an entangled direction that is found with GANzilla, including the time that it takes to find a direction with GANzilla. However, it can also be argued that algorithm-driven approaches can find a direction significantly faster than a user-driven tool like GANzilla. Therefore using algorithms in combination with GANravel for disentanglement can yield better results. We did not analyze the optimal disentanglement procedure in this work and left it as future work. Participants spent 32.32% of their time selecting exemplary images, 46.34% on live-testing the directions, 7.66% on highlighting, 8.17% on weight adjustments, and the remaining 5.51% on applying masks. The average number of positive or negative examples selected per trial is 15.32. The average number of highlighting per trial is 2.75. The average number of times the weights of the exemplary images are adjusted per trial is 8.54. The average number of directions tested on live-testing per trial is 3.72. However, participants tested each direction thoroughly and changed the individual weights of the test images on live-testing 8.56 times per direction.

The average number of masks the participants tested per trial is 2.53. The average number of times the participants applied the mask to all the test images is 1.82 per trial.

Participants were able to find state-of-the-art disentangled directions from scratch in under 10 minutes 81.19% of the time. This shows how GANravel can efficiently enable the end user to find disentangled directions when a dataset or a classifier is not available. Moreover, participants got better at disentanglement as they spent more time with GANravel. For the first user study, the initial directions they found were 41.33% more disentangled (according to classifier-based metrics) after the first three trials which is statistically significant (p=0.04) according to to Mann-Whitney U test. They achieved this by spending 13.1% more time on the initial image selection which is also statistically significant (p=0.04) according to to Mann-Whitney U test.

3.6 Qualitative Results

We employed a method akin to the Affinity Diagram approach [HB97], and we aggregated participants' responses. We summarized their perceived ease, the perceived success of disentanglement, and the perceived success of finding the directions using GANravel in § 3.6.1. We also report the cognitive load responses using the mental demand, effort and frustration dimensions in the NASA TLX questionnaire in § 3.6.2. Additionally, we extracted recurring themes regarding how participants assess the usefulness of the individual components of GANravel in § 3.6.3. Specifically, the first author transcribed participants' responses to develop the initial codes, which were then reviewed by the second author. Disagreements were resolved via discussion between the two authors. Figure 3.6 shows the average ratings of the participants for GANravel on ease of use, perceived disentanglement success, perceived trial success, cognitive load, and ablative assessment of each component's usefulness.

We also show some qualitative images showing the disentanglement performance of GANravel for the first user study. In Figure 3.7, we compare the resulting images of

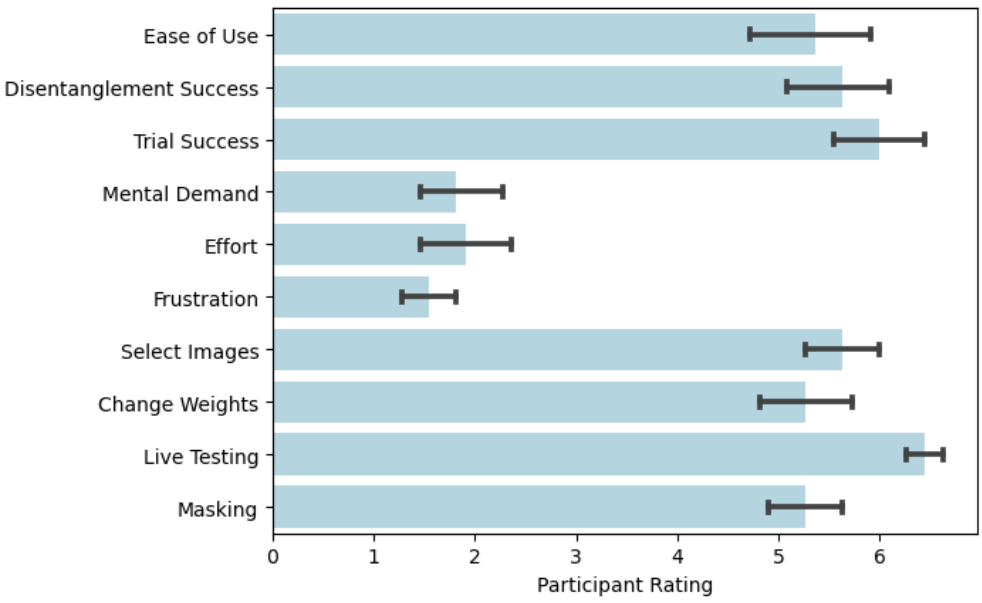

Figure 3.6: The participants' average ratings. The questions are explained in § 3.4.3. All questions used a seven-point Likert scale.

GANravel with the baselines. As can be seen, GANravel has better disentanglement. Additionally, we show directions that participants found during various trials in Figure 3.8 (which uses the same reference image as Figure 3.7). As can be seen, the participants have successfully disentangled directions in all of the tasks. In Figure 3.9, we show the improvement of the direction 'glasses' as the participant interacts with GANravel. As can be seen, the participant disentangles age and gender with global disentanglement. Then, the participant disentangles the remaining local attributes with local disentanglement. For the second user study, we show some dog memes the participants created in Figure 3.10. In Figure 3.11, we also show some directions the participants found using GANravel, GANzilla and both. As it can be seen, the entanglement issues improved after the participants used GANravel.

Figure 3.7: Comparison of state-of-the-art direction discovery methods and GANravel. The reference row is the original image. GANravel has better disentanglement than other methods.

Figure 3.8: Participants found various disentangled directions using GANravel. Each image is generated from a disentangled direction, found by the participants. Each column is for a different trial. Participants successfully disentangled the directions.

Figure 3.9: Improvement of the direction glasses as the participant interacts with GANRAVEL. The 'entangled' direction has entanglement with age, gender, and other various local attributes. After the global disentanglement, the direction is disentangled from age and gender but still entangled with local attributes (*e.g.*, hairstyle, mouth, etc.). Finally, after the local disentanglement, the glasses direction is disentangled.

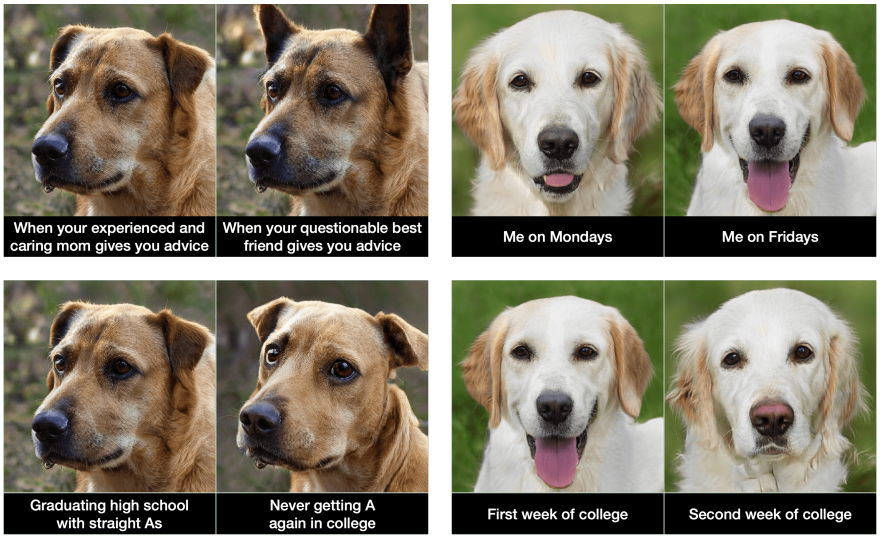

Figure 3.10: Dog memes created by the participants. A disentangled edited image makes a higher quality meme.

Figure 3.11: Comparison of different tasks in the dog meme user study. GANzilla creates entangled directions which result in worse quality edited images. GANravel can disentangle a direction that is found with GANzilla.

3.6.1 Overall assessment

P1-16 represents the 16 participants from the first user study. P17-32 represents the 16 participants from the second user study.

3.6.1.1 Ease of using the tool

All of the participants except (P9, P17, and P26) gave a rating equal or greater than five, when they were asked to rate how easy GANravel was to use. For example P1 commented on the intuitive workflow of GANravel: "The steps are really intuitive. You just select images that have the target attribute and then highlight the region if the results are entangled". P3, P7, P17, P11, P14, P30, and P31 commented on how fast and easily they could find the directions. P4, P5 and P22 pointed out that they had to fine-tune the individual strengths on some of test images for the target attribute to appear, which made it harder to access the directions easily. When the directions are entangled, they can fail to work on some test images. For example P22 said: "Initially, the direction was not working on all of

the dogs. When I disentangled the direction, it started to work". P1, P2, P23, P12, and P32 mentioned it was intuitive to figure out which images to select for global entanglement. P9 gave below-five rating (four) and mentioned the lack of guidance in the tool: "When I got a direction that I did not expect, the next step to take was not always clear". P17 and P26 shared similar concerns. Ratings of P9, P17 and P26 (below-five) are outliers based on the IQR analysis. Overall, participants thought the tool was easy to use.

3.6.1.2 Perceived success of disentanglement

All of the participants except P8 and P29 gave a rating greater than four, when they were asked to rate their perceived success of disentanglement. P9, who previously rated four for the ease of use, thought the disentanglement was successful: "I could get rid of most of the entanglement problems as I changed the weights". P26, who also rated four for the ease of use, said: "I could clearly see the disentanglement when I improved the image that I found before with GANzilla." P8, who rated three, thought that it was hard to preemptively avoid entanglement and the resulting directions were not always as they were envisioned. P8's and P29's ratings are outliers based on the IQR analysis. P1, P10, P15, P23, and P31 pointed out that they could disentangle the directions better as they got more experience with GANravel. Participants also pointed out variations in entanglements: "Some entanglements were more obvious, for example, gender and age. I tried to disentangle them first" (P3) and different strategies to overcome them: "Some entanglements were more subtle such as gaze direction or chin roundness. I tried to ignore those and focus on the more obvious ones first" (P5). "Sometimes I could see the breed of the dog changing. After balancing out the exemplary dogs, It got much better." (P21). Overall, the participants thought they were successful in disentangling the directions which is also backed by the quantitative analysis.

3.6.1.3 Perceived success of the trials

All of the participants gave a rating of six or seven, when they were asked to rate their perceived success in finding the direction for trials. Participants had different reasons why they were confident with their directions: "I could see that the direction was working on all of the test images" (P1), "I could build a diverse set of exemplary images in all trials, so the resulting directions were successful" (P3), and "I could see the improvement in directions after I spend some time on them, progression was convincing" (P10). Some participants felt successful after changing the strength of the direction and saw the transition from the initial image (P2, P7, P8, P12, P16, P20, P30). P17 and P32 pointed out the open-ended nature of creating memes and commented that they were surprised that they could find the directions they envisioned while creating the memes. P29 mentioned how he adjusted for the tool: "It was not possible to find everything I was looking for, instead I got inspired by what I could find.". Overall, participants felt like they were able to finish the tasks.

3.6.2 Cognitive load by NASA TLX

Only one participant (P9) gave a rating higher than four (neutral) for the mental demand dimension and pointed out that the main mental demand was to come up with ways to disentangle a direction when the entanglement was not very clear. As explained by P9: "Sometimes I could see there were some entanglement problems with the direction, the face looked like someone else, but I did not know how to fix it". P9 also gave a rating higher than neutral for the frustration dimension citing the same reasoning. Rating of P9 on frustration is an outlier based on the IQR analysis. Most of the participants found the tasks were not mentally demanding or frustrating as mentioned by P18: "It was easy and a lot of fun". P8: "It behaved as I expected and that was really satisfying". P4, P7 P13, and P31 mentioned that they could see their progress over time and were motivated by it. P8 was the only participant who rated higher than neutral for the effort dimension and said: "For some

trials, I had to do multiple iterations until I was confident". Rating of P8 on the effort is an outlier based on the IQR analysis. The rating of P4 for the frustration dimension was five because of how the directions could behave unexpectedly. As mentioned by P4: "Sometimes the directions did not work on all the test images or the directions required specific strengths to appear. I had to test a lot of different values in some trials". Rating of P9 (five) and P4 are outliers based on the IQR analysis. Most of the participants were not frustrated, in fact, they "enjoyed" (P3, P7, P17, P18, P19, P21) the tasks.

3.6.3 Ablative Assessment

In this section, we summarized the recurring themes based on participants' responses on the usefulness of individual components of GANravel.

3.6.3.1 Participants could preemptively avoid entanglement issues by curating a set of positive and negative examples.

Some participants (P1, P2, P5, P7, P10, P13, P17, P19, P21, P22, P24, P25, P28-32) pointed out that they got better at disentanglement as they learned more about the model and more about the entanglements in the directions. As P1 mentioned: "After the first couple of trials, I was looking to avoid some entanglements proactively. I was trying to balance out concepts such as gender, age, and glasses by choosing a diverse set of examples". Participants also had different strategies to avoid entanglement, for example as P2 mentioned: "For the curliness direction, I had a hard time finding male examples. So instead, I selected most of my positive and negative examples as females so the final direction was not entangled with gender". Another example was from P23: "I was selecting just a couple of exemplary images to see the initial entanglement and then try to select the next example based on the entanglement. From there, I added examples one-by-one". P5 pointed out that creating a 'balanced' set requires tuning through trial and error: "It was not as easy as selecting

the same number of female and male positive examples to disentangle gender. It required tuning". P7, P10, P23, P27, and P31 mentioned that with more available time, they could tune out the entangled features more.

3.6.3.2 Participants struggled with entangled directions that do not introduce the target facial attribute on all the test images.

P4, P5, P11, P13, P15, P23, and P29 pointed out the lack of consistency in the live-testing area when the direction is entangled. Specifically, they talked about the cases when a direction did not work on certain test images, but worked on others, as mentioned by P5: "When a direction did not work on a couple of test images, it was hard to believe in the direction". P4 talked about the same issue with more nuance: "Because of the entanglement, lipstick direction did not work on all the test images. But as I disentangle the direction and make the direction purer, it started to work on more images". P15 also mentioned the same issue: "Some directions changed the breed of the dog but not for all of the test images. It was a little confusing". To our surprise, some participants were more positively reinforced when the direction worked well on a couple of test images but not on all of them. They thought that it was "not possible" (P1 and P19), "a limitation of the model" (P2 and P32), and they tried to "get right" as many images as they can (P9, P13, and P22). Similarly, some participants mentioned the lack of quantitative metrics in the live-testing area (P3, P10, and P25), as mentioned by P10: "I wish there were metrics at testing, so I knew I was improving the direction more objectively". P6 (and P31) had a different strategy: "I focused on the worst looking test examples and tried to improve them. The rest of the test examples usually followed".

3.6.3.3 Some participants favored one of the disentanglement approaches over the other.

To our surprise, some participants found the global disentanglement more intuitive, as mentioned by P21: "Changing the weights to improve images were really helpful, I did not use the masking a lot." and P5: "Changing the weights of selected images was helpful whenever disentanglement was necessary and the results were as expected". Similarly, P1 preferred to use changing weights over masking, saying: "I exactly knew what 'changing weights' did, but I was not sure how masking worked under the hood". However, some participants preferred masking, as mentioned by P2: "Masking is easy to use, I just highlighted the area of interest and the directions were disentangled". P7 and P29 mentioned masking was "faster" and "easier" than changing weights to disentangle the direction. Some participants utilized both and even created their own system, as mentioned by P3: "If the entangled features were obvious such as gender or age, I fixed them with changing weights. For more subtle entanglements, I used masking". All of the participants gave ratings of above four to the *changing weights* component.

3.6.3.4 Participant used local disentanglement with various masks.

Masking was used in various ways by some participants, as P2 mentioned: "Whenever the lighting on the face was changing, I highlighted the entire face and used the masking to reduce lighting entanglement" (using the discard feature). Some participants highlighted more subtle changes such as cheeks (P3) or forehead (P4). However, according to P4, subtle masks did not always work and bigger masks were more "consistent". P5 gave a more nuanced explanation of why masking was harder to use in certain situations: "Trying to disentangle age from gender was harder with masks because both directions were not contained to an area". Some participants did not use the *discard feature* of the masks to eliminate entanglement (P8, P9, and P10) but instead used the *preserve feature* which is

explained in § 3.3.2. For example, P8 said: "For the eyeglass trial, I chose the best test example that had glasses. Then I used the masking on the glasses and kept them (preserve). The direction got significantly more disentangled". P6 only used the *discard feature* to improve the "worst-looking" test examples. A similar trend can be seen in the second user study, where participants had different strategies. For example P26 said: "Masking the ears were resulting in change of the breed, so instead I focused on the discard feature". P17 and P21 mentioned 'using different combination of masks' to end up with a disentangled direction.

3.7 Discussions

This section discusses several issues in the current tool and possible solutions for future work.

Limitations of the current study. First of all, future work can increase the number of participants beyond the current user studies. Moreover, we can also use a narrower user group (*e.g.*, artists trying to use GANs for creative purposes). In the 'age' trial of the first user study, participants had various interpretations of the task, *e.g.*, some participants found directions for wrinkly faces, some found white hair, etc. This trial can be improved so that it is easier to compare it with prior work.

Improving disentanglement with prior analysis. Currently, GANravel is designed to iteratively disentangle a direction. Typically, the user identifies an entanglement problem and then tackles it with either global or local disentanglement. While fixing one type of entanglement, the disentanglement process can introduce a new type of entanglement. For future work, one possible idea is to analyze entanglements in the model via an unsupervised algorithm prior, such as clustering. This would allow the model to predict which type of entanglements appear together and can be used to feed-forward information about the entanglements while the users interact with the system.

Providing guidance through exemplary image selection. Image selection plays

an important part in global disentanglement. Currently, the user goes through the image gallery which consists of randomly sampled images. We decided to go with random sampling because, in an earlier version of GANravel, user recommendations for selections created bias and entanglement in the resulting direction. In the future, non-biased guidance through image selection can be implemented to help the users. This can be achieved by a text prompt indicating what the user is searching for (*e.g.*, young blond males) or if the search is about more subtle attributes, it can be through highlighting a region.

Providing guidance and metrics on disentanglement. In the future, more guidance on disentanglement can be provided to the user. This can be achieved with heat maps that show the changed regions as well as metrics (*e.g.*, facial identity) that indicate the disentanglement performance. As another solution, the user can indicate when a disentanglement works and does not work, which can then be used to learn the correlations in the GAN model. The information can then be used for further guidance as the users disentangle a direction. As a result, the tool can adapt to user feedback.

3.8 Conclusion & Future Work

In this section, we summarize key insights of GANravel to help future work.

- Curating a balanced set of images to find a disentangled direction is a trivial task in disentanglement. Surprisingly, as the participants interacted with GANravel, they got better at avoiding the initial entanglement issues.
- Directions can fail to generalize across test images due to entanglement. But this failure can also be caused by the limitation of the model. In other words, some test images could not be successfully edited with the direction. This phenomenon was confusing to the users since they did not know if their direction was entangled or it was a limitation of the generative model. In the future, figuring out the model limitations automatically

and communicating it to the user can become an important task.

- A future research direction can be figuring out how to utilize algorithmic direction discovery methods in user-driven direction discovery. In the second user study, we discovered that starting with a direction can speed up disentanglement. Using algorithm-driven direction discovery methods to initialize the search process or leveraging them throughout the human-AI interaction is left as future work.

- As the participants disentangled directions, they learned the common entanglement issues specific to the generative model. With each new task, they had to disentangle the same issues such as age and gender. In the future, an algorithmic way to get rid of this repetitive task can be investigated.

- Subtle entanglement issues are harder to measure as discussed in § 3.5.2. In the future, new metrics can be investigated specifically tailored for local entanglement issues.

- Throughout the workflow of GANravel, selecting exemplary images was time consuming as discussed in § 3.5.3. Users were 'communicating' with the model through selecting images. For example by selecting images with blonde hair, the users were interacting with the model to find the respective direction that changes the hair color. In prior work, instead of exemplary images, StyleCLIP [PWS21] leverages text prompts for the same interaction, which is faster than selecting images. However the downside of natural language is, it does not have the fine-grained optimization a set of images can provide. In the future new interaction methods should be investigated that provides both fast and detailed communication between the users and AI models.

CHAPTER 4

User-Driven Prompt Scheduling in Text-to-Image Diffusion Models

4.1 Introduction

Foundational deep learning models are large machine learning models trained on a vast quantity of data at scale. Due to this large dataset they can be easily adapted to wide range of downstream tasks. Specifically, text-to-image models like Dall-e[RPG21] and Stable Diffusion[RBL22] have many applications including art [Opp22], medicine [WWA22], and education [VT23].

Despite these advancements, a challenge persists for non-expert users. These cutting-edge models, while potent, present themselves as intricate 'black boxes' to the end user. This effectively restricts users to basic interactions, preventing them from tapping into the full potential and versatility of the models. The primary user control remains the textual prompt, which serves as the singular input point, thus limiting the level of influence a user can exert on the output. Addressing this limitation, the image generation community has collaboratively created 'prompt books' as comprehensive guides. The community has also seen the rise of "prompt engineers" - professionals who specialize in crafting optimal textual inputs to achieve specific model outputs. The commercial value of this expertise has even led to a market for these services. Concurrently, there are also prompt search engines, further emphasizing the collective pursuit for refined input controls. While methods like the Prompt-to-prompt[HMT22], Prompt mixing[PGA23], and ControlNet[ZRA23] have enabled

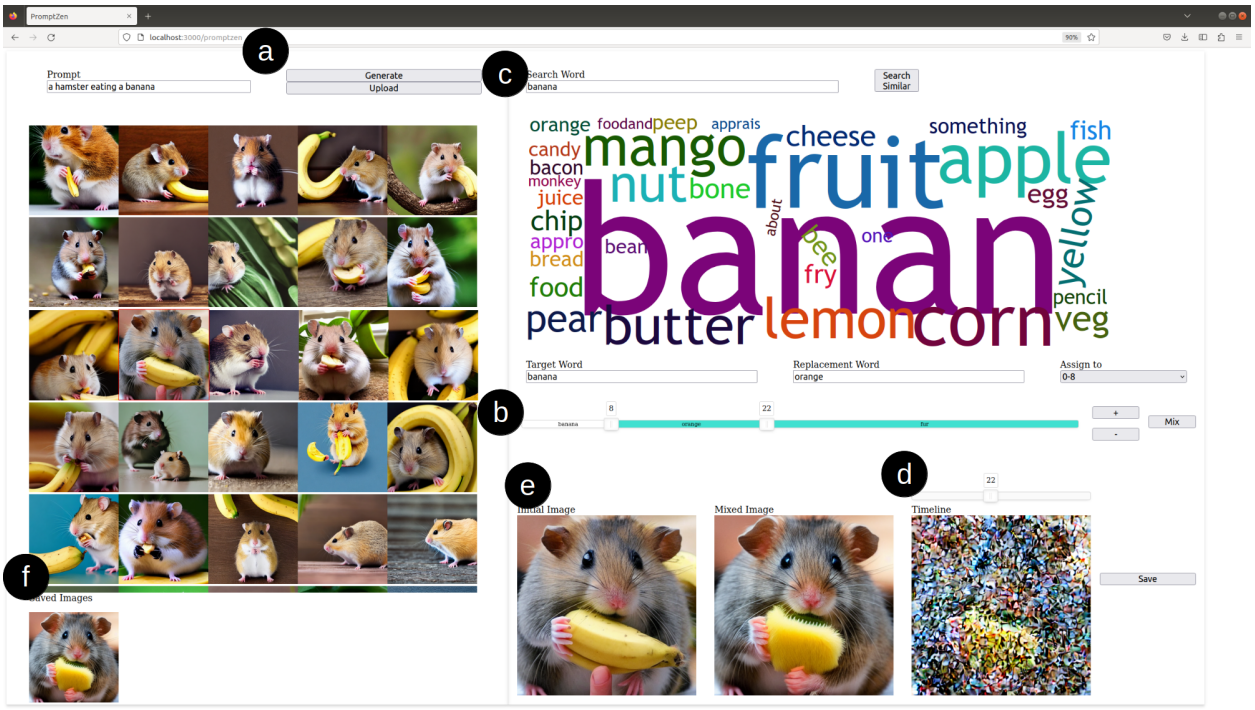

Figure 4.1: PROMPTZEN enables users to edit images through prompt scheduling. (a) User can choose a generated image or upload a real image for inversion. (b) User can schedule prompts to control image generation process. (c) User can find related keywords to influence prompt scheduling. (d) User can investigate image generation through denoising timesteps for precise prompt scheduling. (e) User can see the result of their schedule as well as the original image. (f) User can save and return to a saved image for subsequent trials.

more intricate interactions with the models, there remains a gap in making such advanced techniques readily accessible to the everyday user.

We design and implement PROMPTZEN—a tool that enables users to edit images through prompt scheduling. While our initial implementation is tailored for the Stable Diffusion model and its intrinsic cross-attention mechanisms, the workflow in our tool is expected to generalize to other diffusion-based text-to-image models as well.

Beginning their interaction, users can opt to select a pre-existing model-generated image or upload their personal real-world image, supplementing it with a related inversion

prompt (Figure 4.1a). Diving deeper into PromptZEN's functionality, users can pinpoint a keyword of interest and suggest its replacements for various timesteps[1]. Through a user-friendly slider bar (Figure 4.1b), they can methodically dictate the temporal sequence for their keyword replacements, providing a nuanced influence on the image's generation. To enhance this experience, PromptZEN incorporates two key features: *(i)* a context-driven word cloud (Figure 4.1c) that offers alternative keywords based on the original prompt, and *(ii)* a denoising visualization (Figure 4.1d) that reveals the image's evolution from pure noise to its final intricate form. This progression aids users in determining the optimal timings to introduce their keyword changes, refining localized edits in the process. The synergy of the word cloud and denoising visualization empowers users to navigate and execute previously inaccessible local edits with precise prompt schedules. After scheduling, they can test their schedule (Figure 4.1e) and save the edits for future generation (Figure 4.1f).

In our evaluation of PromptZEN, we conducted a user study involving 12 participants, focusing on its efficacy across two distinct tasks. Firstly, in closed-ended tasks, participants were provided with edited image pairs, consisting of references and targets, to discern their ability to replicate specific edits. The results showed that when utilizing PromptZEN, participants created prompt schedules that resulted in images that closely mirrored the desired target images. Moreover, in comparison to three other benchmark methods, the use of PromptZEN led to superior edits that maintained the background's integrity while aligning the local changes closely with the target in terms of shape, texture, and color distributions. Secondly, in open-ended tasks, participants used real images of their living spaces, allowing for freeform edits without a set directive. This was to understand PromptZEN's real-world application and gather actionable insights for future design decisions. The outcome was consistent with our previous findings: participants displayed a clear inclination towards PromptZEN over other baselines. In short, we showed both quantitatively and qualitatively, PromptZEN enabled participants to achieve successful image edits and also

[1]which we refer to as prompt scheduling

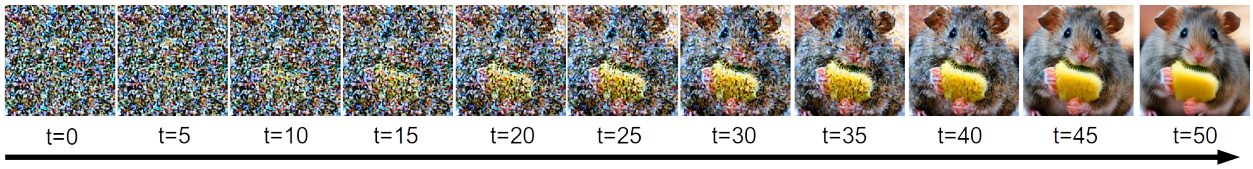

Figure 4.2: Diffusion models iteratively denoises a given noise over multiple timesteps. End users do not have access to this process hindering their control over the generation.

outperformed the existing benchmark methods in the process.

Overall, PROMPTZEN makes a **tool contribution**. In contrast to using diffusion models as 'black boxes', we introduce a comprehensive tool that empowers users with granular control over the image generation process. Through PROMPTZEN, users are guided at each step with intuitive visualizations, enabling a clearer understanding of how their input influences the final output. Importantly, the intent behind PROMPTZEN is not to overshadow or replace other methods that enhance user control. Instead, it aims to augment these techniques, showcasing the potential of user interaction within the context of user-driven diffusion models.

4.2 Background & Related Work

Diffusion models. Diffusion models operate based on the principle of iterative denoising. At the beginning of this process, the model starts with an image dominated by noise, as depicted in Figure 4.2. As the iterations progress, this noise gradually diminishes, revealing the underlying image structure. Each successive timestep refines the image further, reducing the noise and enhancing the clarity. By the conclusion of the sequence, clear and detailed features emerge from what was initially a chaotic noise pattern. Specifically, the underlying deep learning model responsible for the denoising process (U-net in the case of Stable Diffusion) is trained using a unique methodology. Artificial noise is introduced to an image from the training dataset, and the model is then tasked with predicting this introduced noise.

During the inference phase, the noise predicted by the model is subtracted from the original noisy image. This approach iteratively reduces the noise level, progressively revealing a clearer version of the image. However, a limitation in the current application of diffusion models is the availability of these denoising timesteps to the user. End users typically interact only with the final output and are denied access to these informative intermediate stages, restricting their understanding and control over the image evolution process.

Enabling users to interact with generative models. User interactions with generative models have been explored in various research studies. A common method used in this domain is the slider interface, as documented in several works [SYT20, AZM21]. An extension to this was presented by Dang *et al.*, who compared traditional sliders with enhanced versions showing previews, known as 'filmstrips' [DMB22]. Moving beyond sliders, Zhang et al. used a grid-based view for the latent space, allowing users to interact and explore generated samples more intuitively [ZB21]. A different approach was introduced by Chiu *et al.*., where a single-dimensional slider was employed to navigate the latent spaces of generative models [CKL20]. In terms of diverse input methods, works by Cheng *et al.* and Yu *et al.* incorporated natural language feedback to influence the output of generative models [CGL20, CSG20]. Further, Ling *et al.* introduced a system that lets users edit generated images using segmentation masks [LKL21]. Another method involves using textual descriptions for image transformations, as seen in StyleCLIP by Or *et al.* [PWS21] that uses CLIP embeddings [RKH21]. Tools such as GANravel [EC23] and GANzilla [EC22] have also contributed to the field, with GANzilla offering a scatter/gather technique and GANravel focusing on user-driven disentanglement in directions. Drag Your GAN, is an interactive point-based manipulation tool that lets user edit an image [PTL23]. In summary, various methods and tools have been developed to enhance user interactions with generative models, reflecting the ongoing growth in this area.

Controlling image diffusion models. In our exploration of image diffusion models, we employed a Stable Diffusion model in PromptZEN as a proof of concept. Controlling

these image diffusion models has gained significant traction recently. Various methodologies have emerged, ranging from the straightforward addition of noise to an image followed by its denoising using a guiding prompt [ALF22, MHS21, CVS22], to the more complex manipulation of model internals, such as attention maps, to maintain the image's structure [HMT22, GLK23, PKZ23, PGA23]. Innovations like Magicmix [LYZ22] are venturing into semantic mixing, blending distinct semantics for a fresh concept, while ControlNet [ZRA23] integrates spatial conditioning controls to upscale pretrained text-to-image diffusion models. MakeAScene [GPA22] and SpaText [AHG23] harness segmentation masks, translating them into tokens or localized token embeddings for controlled image generation. InstructPix2Pix performs the appropriate edit using an image and an instruction for how to edit that image [BHE23]. Further advancements include GLIGEN [LLW23] adjusting attention layers' parameters for grounded generation, Textual Inversion [GAA22] and DreamBooth [RLJ23] refining the diffusion model with user images for personalized content. Yet, despite these technical leaps, a discernible gap persists between user engagement and the optimal presentation of these control mechanisms for the end user. Complementarily, PromptZEN enables end users to control the generation process through intuitive user interactions built on top of attention-based techniques.

4.3 PromptZEN: User-Driven Prompt Scheduling in Stable Diffusion

In this section, we present a detailed walkthrough of PromptZEN's design and implementation using an exemplar use case of editing a scene with a hamster holding a banana. PromptZEN's implementation can be divided into four main groups: selecting an image, prompt scheduling, word cloud, and denoising visualization.

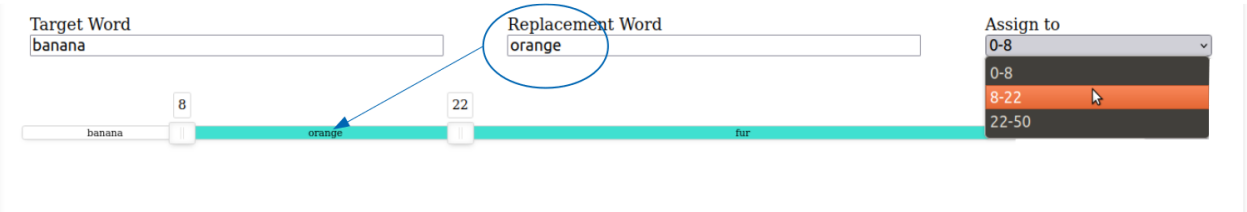

Figure 4.3: Users can schedule the replacement keywords using the dropdown menu. The schedule will be visualized as a slider bar where blue color indicates a replacement word. The timing of the replacement can be changed using slider handles.

4.3.1 Selecting an image to edit

Initially, a user provides a textual prompt to produce a variety of images generated from different seeds. As demonstrated in Figure 4.1a, these generated images are displayed in a gallery format. The user then selects their preferred image for further editing. In addition to using generated images, users also have the option to upload a real image for editing purposes.

Implementation. For generating images through Stable Diffusion, we utilize a conventional generation pipeline. This pipeline produces images based on the provided prompt using the model in the absence of prompt scheduling or scene modifications. When users upload their own images, we apply null-text inversion [MHA23]. Despite uploading an image, users are still required to input a prompt for inversion as well as to facilitate the functioning of PromptZEN's pipeline. Every time an image is generated, it is saved on a local device with the latent codes so that subsequent runs with identical prompts run faster.

4.3.2 Prompt Scheduling

Stable Diffusion creates images by sequentially denoising an initial noisy image across multiple timesteps. With prompt scheduling, Stable Diffusion is enabled to incorporate different prompts at specific timesteps. PromptZEN emphasizes modifying a singular keyword by

blending the original prompt with alternate keywords, ensuring the entire scene remains coherent. This design decision was taken to streamline user interaction, especially since our preliminary studies showed that users found scheduling multiple keywords challenging to navigate. As depicted in Figure 4.3, users begin by entering the term they aim to adjust into the 'Target Word' section. This keyword should be consistent with one from the original prompt. Users then enter the replacement keyword in the 'Replacement Word' section. In our example, we are editing 'banana' with keywords 'orange' and 'fur', the rationale for which will be elaborated in subsequent sections. This new keyword can be scheduled using the adjacent dropdown menu. The schedule is represented graphically with a sliding bar. This bar contains handles that segment the bar into sections, with each section representing a specific keyword. Between these handles, users can see and adjust the keywords for various timesteps. Users can adjust the number of segments (and thus the number of keywords) with '+' or '-' buttons. By dragging the handles, users can determine the duration and the timing of each keyword in the schedule. Each handle is also marked with numbers, indicating the associated timestep. After finalizing their desired schedule, users can activate the 'mix' button to produce the edited image.

Implementation. Once the prompt schedule is established, it is passed to the backend. Here, based on the replacement keywords, the prompts undergo modification. Within the Stable Diffusion framework, these prompts are channeled into the U-net via cross-attention layers. These layers comprise matrices named Key, Value, and Query, inspired by information retrieval systems [VSP17]. In the conventional implementation, prompts influence the Key and Value matrices, with the Query originating from a preceding layer in the U-net. In contrast, our methodology employs the new prompts (sourced from the prompt schedule) to compute only the Value matrices. However, the Key matrices remain computed using the original prompt, a procedure aligning with [PGA23]. Only updating Value matrices demonstrated to better retain the original image's layout, resulting in better disentangled edits. As a result, even if the entire prompt schedule diverges from the original in terms

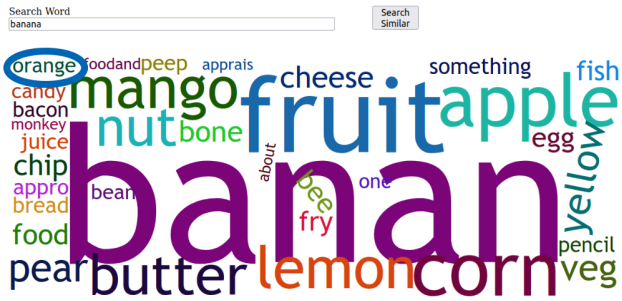

Figure 4.4: Users can search for keywords by providing a search word. Using CLIP embeddings, PROMPTZEN returns a word cloud where the most related words have bigger fonts.

of keywords, our framework still upholds the original image's layout. Once processed, the modified image, alongside its associated schedule, is stored on the local device, ensuring swift retrieval in subsequent operations.

4.3.3 Word Cloud

PROMPTZEN incorporates a word cloud within its pipeline, assisting users in discovering related keywords for any given keyword. As depicted in Figure 4.4, participants have the ability to input a keyword into the 'Search Word' field. This keyword subsequently guides the generation of a word cloud. This visual representation showcases words related to the user-provided keyword, with the font size indicating the degree of similarity. Users can either click on a word, automatically populating the 'Replacement Word' field, or initiate a new keyword search. In our example, we are searching for similar words to 'banana', in order to not dramatically influence the resulting image. Then we can use the keyword 'orange' for scheduling which can be achieved by clicking on it.

Implementation. When provided a keyword, we derive the prompt's embedding with the provided keyword substituted for the 'target word'. This embedding is subsequently compared with embeddings of all possible terms from the original CLIP vocabulary. This comparison is conducted by replacing the target word' with a vocabulary word and subse-

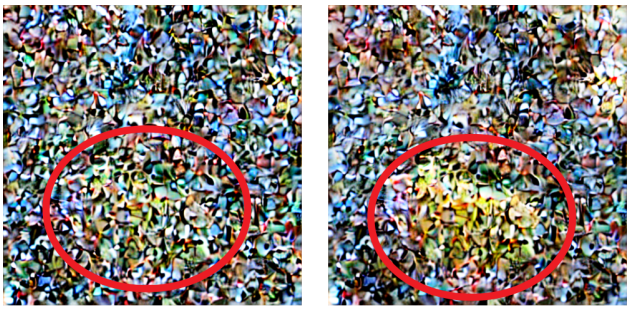

Figure 4.5: Users can use the denoising visualization to see how the model generates the final image over multiple timesteps. They can use this information for precise scheduling. On the left (at timestep t=8), the object the hamster is holding is barely visible indicating that the location of the object is set. Scheduled keywords after this timestep will not change the outline of the scene. On the right (at timestep t=22), the shape of the object is barely visible indicating that the shape of the object will not be affected by the scheduled word after this timestep. These visualizations enable the user to precisely schedule and control the generation of the image.

quently determining the resultant embedding. The measure of similarity, determined using cosine similarity, informs the font size allocation within the word cloud. Both the determined font sizes and the associated words are relayed to the frontend for visualization.

4.3.4 Denoising Visualization

PromptZEN integrates an interactive denoising visualization feature. This visualization includes a slider bar, which users can adjust. Altering the position of the slider reveals the image at a specific timestep, given the current prompt schedule. The primary objective of this feature is to allow users to witness the generation progression, transitioning from pure noise to the final image. As depicted in Figure 4.5, by timestep $t = 8$, the banana held by the hamster starts to become vaguely distinguishable. Such observation indicates that at this particular timestep, scheduling a specific keyword can influence various details without

drastically altering the overall location of the object, especially since the object's outline is already somewhat apparent. In a similar vein, by timestep $t = 22$, the object's color, as well as its overall structure, are prominently visible. Thus, introducing a different keyword at this stage can primarily impact very minute details, such as texture, while leaving other aspects largely unchanged. Notably, each denoising example possesses different timing requirements. Hence, the denoising visualization tool is instrumental in aiding users to precisely time their prompt schedules tailored to individual examples. In our provided example, we strategically scheduled keywords 'orange' and 'fur', influenced by the denoising visualization. As can be seen, the outcome is an image retaining the location of the original banana, yet incorporating the overall shape of an orange and the texture of fur.

Implementation. The Stable Diffusion U-net does not directly denoise the entire image. Instead, it denoises a latent code that represents the image. After the denoising process concludes, this latent code is channeled through a decoder to produce the full-resolution image. In our implementation, we maintain a record of all the latent codes from every timestep during the denoising phase. When a user adjusts the slider bar, the corresponding latent code associated with that specific timestep is passed to the decoder. The decoder then creates the high-resolution image corresponding to that denoising stage. For efficiency and quick access in subsequent interactions, we store this image and forward it to the frontend.

4.3.5 Other Implementation Details

For our deep learning tasks, we employed PyTorch as the backbone. Our codebase builds upon the official implementation of Prompt-to-prompt. This foundation proved beneficial, given its pre-existing functionality pertaining to attention map access and modification. We developed the remainder of the backend using Python. To manage interactions between our frontend and backend systems, we integrated Flask as our chosen web framework. Our frontend is a combination of technologies: it leverages Javascript, Node.js, and React to deliver a responsive interactive user experience. To ensure robust performance and seamless

computation, the backend is hosted on a Linux server outfitted with an Nvidia GeForce RTX 3090 GPU and the frontend is served on a web browser.

4.4 User Study

We conducted a user study to examine the capabilities of PromptZEN in enabling users to modify scenes for diverse and creative outcomes. Participants were tasked with editing objects in images using PromptZEN across two distinct task types. Following the methodology of other user-driven generative model evaluations [EC22, EC23], we incorporated both closed-ended and open-ended tasks. For the closed-ended tasks, participants had a specific editing objective to meet, requiring them to navigate and utilize PromptZEN to achieve a predefined image outcome. This design not only demonstrated the practical utility of PromptZEN for end users but also permitted a performance comparison with baseline approaches. Conversely, the open-ended tasks presented participants with a more ambiguous goal. The primary intent of these tasks was to highlight the broad range of scene variations possible with PromptZEN, thus emphasizing the tool's potential in enabling creativity and underscoring its diversity. In essence, the study aimed to both evaluate participant success in generating varied scene modifications and to highlight the innovative capacity of PromptZEN in facilitating diverse creative tasks.

4.4.1 Participants

We employed convenience sampling to enlist 12 participants from a nearby university. Of these participants, eight were male and four were female, with ages ranging from 21 to 32. The academic backgrounds of the participants were diverse: four specialized in electrical engineering, four in biomedical engineering, three in economics, and one in archaeology. While participants had programming experience spanning three to 10 years, none had prior exposure to Stable Diffusion or its related applications.

Figure 4.6: Reference and target images that are given to the participants for the closed-ended tasks. The goal for the participants was to recreate the target images.

4.4.2 Tasks & Procedure

Participants were required to complete two types of tasks with PROMPTZEN: closed-ended and open-ended. Each task type comprised three trials. In both, the aim was for participants to use PROMPTZEN to modify an object within an image, generating scene variations while retaining other features. The details of the tasks are detailed below:

Closed-ended tasks. The primary metrics of interest in our study were creativity and performance. Closed-ended tasks were designed with a particular emphasis on gauging performance. In each trial of this task type, participants were presented with an original image alongside its edited counterpart. Their objective was to reproduce the edited image as accurately as possible. Figure 4.6 showcases the original and corresponding edited images provided to the participants. The closed-ended tasks were intentionally restrictive, designed to constrain creativity and solely assess the participant's ability to meet a defined editing target with PROMPTZEN. To further evaluate the effectiveness of PROMPTZEN, participants were also tasked with three baselines. The first baseline involved replicating the image edit by solely modifying the prompt. In addition to this baseline, we incorporated two comparative methods: Prompt-to-prompt [HMT22] and Prompt mixing [PGA23]. Though the primary contributions of these methods do not center on user interaction, they focus instead on the intrinsic mechanism. Further details on the implementation of these comparative methods

can be found in section § 4.4.3. In other words, given the original prompt that generated the image, participants aimed to emulate the edited image using iterative prompt modification, Prompt-to-prompt, Prompt mixing, and PromptZEN. This task aims to investigate whether user-driven tools, such as PromptZEN, can offer improvements or variations to the outcomes achieved by baseline performance methods.

Open-ended tasks. The primary objective of the open-ended tasks was to explore the creative possibilities facilitated by PromptZEN and to gauge the diversity of variations achievable. Unlike the closed-ended tasks, participants weren't directed towards a specific editing outcome. Instead, they were encouraged to experiment and produce a varied array of images. Specifically, participants were prompted to submit pictures from their living spaces for editing. Recognizing that real images can introduce complexities due to image inversion, especially if the model misinterprets the scene, participants also had the option to generate interior scene images through prompts. Each participant was tasked with editing three scenes, either real or generated, aiming to introduce variations while retaining the essence of the original image. The study's design was deliberately open-ended to assess PromptZEN's efficacy in supporting creative tasks. The insights from this task could guide future designers considering the incorporation of Stable Diffusion tools in creative settings. Mirroring the closed-ended tasks, participants were also tasked to use Prompt-to-prompt and Prompt mixing methods and were asked to highlight their preferred outcomes. Notably, iterative prompt modification was not employed in this task since the images were produced through image inversion, rather than directly originating from provided prompts.

Procedure. Before each study, participants completed a preliminary questionnaire to gather background information. Each study was initiated with an introduction to PromptZEN, followed by a tutorial. Subsequent to the tutorial, participants were given a brief practice session using a toy example to become acquainted with PromptZEN's functionality. As Prompt-to-prompt and Prompt mixing lack dedicated tool support, participants were also introduced to an example notebook and received a tutorial explaining the operation of these

two baseline methods. Following this, participants started with a block of trials (either closed-ended or open-ended) which was followed by a short break. After the break, the participants are given the remaining block of trials. The order of tasks, as well as the order of trials within each block, was counter-balanced among participants. After the conclusion of each trial, participants responded to questions pertaining to the task, the details of which are elaborated upon in §4.4.5. The study was concluded with a semi-structured interview, aimed at garnering participants' qualitative feedback regarding their experience with PromptZEN. Conducted in person, the study spanned approximately one hour, with participants being compensated with a $25 gift card.

4.4.3 Baseline Implementations

Our study employed three baselines to compare and contextualize the capabilities of PromptZEN.

Prompt editing. This baseline method involves participants directly modifying the prompts to change the scene. When the edited prompts are closely related to the original, and identical seeds are applied, the generated images often remain similar to the original images. However, this approach lacks a mechanism to guarantee consistent image preservation. Using prompt editing as a baseline highlights the critical role of attention injection, without these injections user control over the generation is limited.

Prompt-to-prompt. The official implementation of this method offers many parameters and diverse means of injecting attention maps, such as *replacement*, *refinement*, and *re-weight*. It allows users to replace, introduce, or adjust the weight of words. For this study, we adopted the prompt refinement feature of prompt-to-prompt, as it most closely resembles the PromptZEN implementation. For other parameters, such as cross and self-attention replacement steps, we conducted a parameter sweep, presenting the resulting images to users for them to select their favored option.

Prompt-mixing. The official prompt-mixing feature employs CLIP embeddings to auto-

generate variations by sampling similar prompts. In our approach, we leveraged this mechanism to produce edited images, allowing participants to select their preferred variation.

4.4.4 Data & Apparatus

For the backend Stable Diffusion model, we used pretrained CompVis Stable Diffusion v1.4, the model parameters are publicly available on Hugginface website[2]. It should be noted that there are newer Stable Diffusion models publicly available and the methods introduced in this paper can still be applied to them since these models still leverage the same cross attention mechanisms. We decided to use v1.4 to be able to compare our results with other methods that are officially supported on v1.4. The user studies are conducted in person. The participants interacted with the local server directly that the backend was running to minimize the latency. Other implementation details are given in §4.3.5.

4.4.5 Measurement

We saved every image the participant generated throughout the study. We also logged all of the user actions with timestamps including keywords that they have searched on the word cloud, the prompt scheduling they have created, all the buttons they have pressed, and different timesteps they have tested on the denoising visualization as well as the resulting image. For qualitative measures, immediately upon finishing each trial, we asked each participant to rate (along a 7-point Likert scale) how successful they thought they had achieved the editing goal with the prompt scheduling they used. In the exit interview, participants started with an overall assessment of PromptZEN based on their overall experience for both closed- and open-ended tasks. We asked *(i)* whether the tool is easy to use and *(ii)* whether the user can find prompt schedules that match their editing goal. Next, participants rated the cognitive load using the mental demand, effort and frustration dimensions in the NASA TLX

[2]https://huggingface.co/CompVis/stable-diffusion-v1-4

Figure 4.7: Closed-ended tasks results. The first column in each task (1st, 4th, and 7th) has the target images given to the participants. The other columns are generated by participants. The images generated by PROMPTZEN closely match with the target image compared to other baselines. Better viewed in full screen.

questionnaire [Har86]. Next, we asked participants to ablatively evaluate the usefulness of PROMPTZEN's individual UI elements: word cloud, denoising, and prompt schedule. All questions were rated along a seven-point Likert scale.

4.5 Results

The results section is split into two subsections: quantitative and qualitative findings. Figure 4.7 and Figure 4.8 show sample images edited by participants for closed- and open-ended tasks, respectively.

4.5.1 Quantitative Findings

Below we provide quantitative analyses to better understand participants' performance and behavior using PROMPTZEN.

Figure 4.8: Open-ended tasks results. The first row is the real images participants used in open-ended tasks from their living spaces. Using PROMPTZEN participants were able to create variations in their images. Compared to baseline methods, the edited images with PROMPTZEN preserve the background better and they are more natural-looking. Better viewed in full screen.

4.5.1.1 Editing Performance

In this section, we evaluate the editing performance of participants using both PROMPTZEN and the baseline methods on closed-ended tasks. The editing objective can be categorized into two main goals: *(i)* modifying the object while maintaining the integrity of the rest of the image, and *(ii)* altering the object so it aligns with the intended editing target. We ran three analyses to investigate the editing performance of PROMPTZEN. The first objective aligns more with the intricacies of the attention-mixing technique and its associated hyperparameters. While identifying a disentangled editing approach – where "disentangled" means changes are isolated to the intended portion only – is valuable, it is not the primary focus of PROMPTZEN. Nevertheless, our initial analysis delves into how effectively the different methods retain the untouched aspects of the image. In our subsequent analysis, the emphasis shifts to gauging how accurately participants can modify the object to match

Method	Background	Object	HOG	GLCM	HSV
Prompt Editing	$.56 \pm .30$	$.36 \pm .23$	$.22 \pm .33$	$.40 \pm .35$	$.47 \pm .31$
Prompt-to-prompt	$.81 \pm .19$	$.51 \pm .29$	$.47 \pm .25$	$.59 \pm .37$	$.69 \pm .37$
Prompt Mixing	$.88 \pm .17$	$.58 \pm .31$	$.50 \pm .20$	$.66 \pm .35$	$.81 \pm .35$
PromptZEN	$\mathbf{.94 \pm .04}$	$\mathbf{.91 \pm .12}$	$\mathbf{.85 \pm .09}$	$\mathbf{.94 \pm .05}$	$\mathbf{.96 \pm .04}$

Table 4.1: Comparison of cosine similarities for PromptZEN and baseline methods. Higher number represents higher similarity to the target image. participants were able to accurately match the object's shape, texture, and color using PromptZEN.

their editing intentions. This embodies the core contribution of PromptZEN: empowering users to realize their editing visions via user interactions. Here, we compare PromptZEN against baseline methods, zeroing in specifically on the object being edited. In our third analysis, we examine the resilience of the prompt scheduling, aiming to elicit the extent to which participants can deviate from the ideal prompt scheduling while still producing a closely aligned image.

Background preservation. First, we manually created a mask for the object for each target image and images generated by participants. We then zero out the object of interest to create an image for the background. We extracted features from the pre-trained VGG16 deep learning model. Subsequently, we computed the cosine similarities between the features of the generated images and those of the target images for both PromptZEN and the baselines. The results can be seen in Table 4.1. As can be seen, the background is preserved in both PromptZEN and prompt mixing. This is mostly due to the underlying attention injection mechanism.

Object modifications. We conducted several analyses to evaluate the edits made to the

objects. Leveraging the masks from our prior analysis, we first isolated the images' objects by zeroing the backgrounds. To evaluate the overarching performance of PromptZEN against the baseline methods, we again leveraged the VGG16 deep learning model. Similarly, we computed the cosine similarity between the features of the generated images and those of the target images for both PromptZEN and the baselines. To get a deeper understanding of object performance at both high levels (*e.g.*, , shape) and low levels (*e.g.*, , texture and color), we extracted several features using different methodologies:

- For shape analysis, we used the Histogram of Gradients (HOG).
- For texture evaluation, the Gray Level Co-Occurrence Matrix (GLCM) was employed.
- For color assessment, a 2D color histogram using HSV was extracted.

These methods were chosen because they are robust to masking. Subsequently, we computed the cosine similarity for these features[3], facilitating a comparison between PromptZEN and the baselines. This in-depth analysis provided a comprehensive insight into performance across various facets of the object's representation. A comparison of these results is detailed in Table 4.1. As can be seen, participants were able to accurately match the object's shape, texture, and color using PromptZEN. In contrast, despite the underlying attention injection mechanisms being similar in PromptZEN, Prompt-to-prompt, and Prompt mixing, participants encountered challenges achieving comparable object edits when using the baseline methods. These results underscore the importance of user-driven prompt scheduling to enable users to control the generation of a scene.

Prompt scheduling performance. We conducted an analysis to assess the proximity of participants' selections to the ground truth prompt schedules. To gauge the accuracy of prompt scheduling, we introduced a novel metric. For every keyword at timestep t, we

[3]The results were consistent with Chi-squared distance on histogram features, we decided to report cosine similarity for all the features since it has a range between -1 and 1.

determined its CLIP embedding and subsequently computed the cosine similarity with the corresponding ground truth keyword for that specific timestep. The average cosine similarity was then determined across all denoising steps. The underlying rationale for this metric is that even when participants opt for a keyword that's similar, but not identical to the ground truth, high cosine similarities are observed due to the properties of the CLIP embedding. Participants exhibited a similarity score of $\mathbf{.73 \pm .24}$ when compared to the ground truth prompt schedule. In contrast, the resulting images (focused solely on the object) presented a similarity score of $\mathbf{.91 \pm .12}$. While these similarity measures are not directly analogous, the data suggests the robustness of the prompt scheduling mechanism. This indicates that even when faced with sub-optimal schedules (varying keywords or non-optimal timesteps), the desired images could still be generated with accuracy.

4.5.1.2 User Behavior

In this section, we delve into the ways participants interacted with PromptZEN and examine the statistical significance of their interactions between open and closed-ended tasks. If no significance is reported, differences should be understood as not reaching statistical significance, based on the Wilcoxon signed-rank test. These findings highlight the distinct approaches participants took, with an emphasis on creativity in open-ended tasks and precision in closed-ended ones.

Time Spent. On average, participants took five minutes and four seconds to complete closed-ended tasks, whereas open-ended tasks took slightly longer at five minutes and 17 seconds.

Word Cloud. Participants dedicated 31.3% of their time for open-ended tasks and 23.7% for closed-ended tasks. The increased time spent on the word cloud for open-ended tasks was significant ($W=32$, $p=0.02$), indicating a preference to brainstorm various keywords.

Prompt Schedules. 40.5% of the time was spent on this for open-ended tasks, slightly

less than the 45.21% for closed-ended tasks. Participants, on average, tested 10.12 prompt schedules for closed-ended tasks and 6.78 for open-ended tasks, a difference that was statistically significant (W=36, p=0.02). This discrepancy likely stems from the more meticulous effort to match the target output in closed-ended tasks. Supporting this hypothesis is our prior metric on prompt scheduling: participants adjusted prompt schedules by an average of 0.05 and 0.23 across subsequent schedules for closed and open-ended tasks respectively—a statistically significant variation (W=46, p=0.02). This further emphasizes the experimental approach in open-ended tasks and the required precision in closed-ended tasks.

Denoising Visualizations. The remaining time was utilized for denoising visualizations. Participants significantly prioritized their time on denoising visualizations for closed-ended tasks to align their generated images closely with the target image (W=32, p=0.03), with averages of 28.12 visualizations for closed-ended and 45.87 for open-ended tasks, a difference that reached statistical significance (W=51, p=0.01).

Other Metrics. On average, participants tested 4.12 words on the word cloud versus 3.01 for open-ended tasks (W=32, p=0.02). They also used 3.12 and 3.08 prompts (number of mixed keywords) per schedule for closed and open-ended tasks, respectively.

4.5.2 Qualitative Findings

We adopted a method similar to the Affinity Diagram approach [HB97]. The responses from participants were aggregated, and a summary of their perceptions regarding ease of use and the perceived success of editing scenes using PromptZEN can be found in Section4.5.2.1. We have also presented the findings related to cognitive load using the mental demand, effort, and frustration dimensions from the NASA TLX questionnaire in Section 4.5.2.2. In addition, we identified common themes on how participants evaluated the utility of the individual components of PromptZEN, which are detailed in Section 4.5.2.3. The transcription of participants' responses was carried out by the first author to develop initial codes. These codes were then reviewed by the second author, and any disagreements were resolved

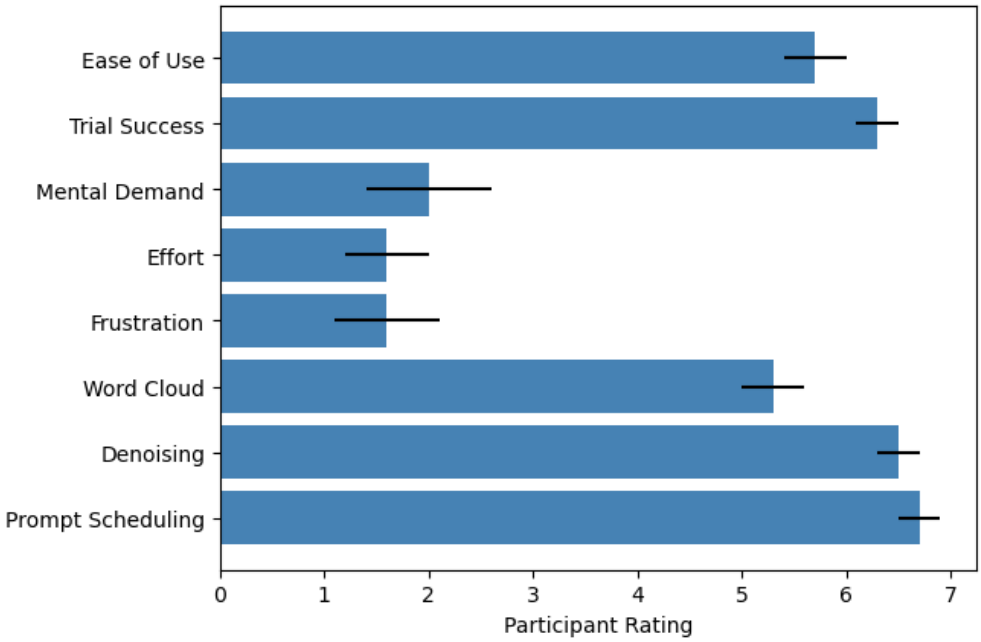

Figure 4.9: The participants' average ratings. The questions are explained in §4.4.5. All questions used a seven-point Likert scale.

through mutual discussion. Figure 4.9 provides the average ratings given by participants for PromptZEN in terms of ease of use, perceived trial outcomes, cognitive load, and an ablative analysis of the utility of each component.

We present qualitative images to showcase the editing capabilities of PromptZEN. In Figure 4.7, we compare the images from the user study with those from the baseline methods. This comparison indicates that participants more accurately matched the target images in the closed-ended tasks than the baseline methods did. In Figure 4.8, we highlight the diverse living space variations participants created using PromptZEN in open-ended tasks compared to the baseline methods. These generations exemplify the participants' ability to create varied scenes from authentic images of their living spaces with PromptZEN.

4.5.2.1 Overall assessment

In this section, we give an overall assessment of the tool in terms of ease of use and perceived success of the tasks.

Ease of using the tool. Upon querying about the ease of using PromptZEN, most participants expressed that they found the tool user-friendly and intuitive. The majority, eleven out of twelve, rated their experience above five on a scale of ten. P2 remarked, 'The process was fairly straightforward. Once I got the hang of it, editing became second nature.' Both P6 and P8 expressed appreciation for the user-friendly interface of PromptZEN. P6 noted, 'The interface was intuitive and did not hinder my creativity at all.' P4 and P7 particularly appreciated the interactive UI elements, highlighting the visual feedback as a strong feature of the tool. P4 said, 'The visual feedback on the sliding bars and the image generation visualizations helped me focus on the prompt schedule and iterate over various schedules.' P12 also praised the scheduling sliding bar and explained it as 'a way to summarize the generation'. P10, who gave a rating below five, faced a slight learning curve, expressing, 'While the tool is undoubtedly powerful, initially navigating through its functionalities required some attention.' Based on our analysis, the feedback from P10 is an outlier as determined through IQR analysis.

Perceived success of the tasks. In terms of perceived success in accomplishing the given tasks, eleven participants gave ratings above five. Moreover, all of the participants preferred PromptZEN compared to other baselines in all of the tasks. P3 commented, 'The results I achieved with PromptZEN, especially in the open-ended tasks, were beyond my expectations.' Even P10, who initially found the tool somewhat challenging, expressed satisfaction with the outcomes: 'Though I began hesitantly, by the end of my tasks, I felt I had a good grasp and was pleased with the scene modifications I achieved.' Another interesting feedback came from P7, who stated, 'The closed-ended tasks were a bit challenging, but the open-ended tasks were a joy. I felt I could truly experiment.' However, P11, who gave

the only below-five rating, felt a bit constrained. They remarked, 'In some cases, especially in closed-ended tasks, I felt I was not getting the exact result I had in mind. Maybe I needed to be more patient and try different combinations.' Based on our analysis, the feedback from P11 is an outlier as determined through IQR analysis. Notably, even those who found the tool easy to use highlighted the nuanced challenges in the closed-ended tasks, hinting at the trade-offs between precision and creativity. For example, P1 said, 'The closed-ended tasks required precision, whereas the open-ended ones allowed more freedom, but in both cases, I felt successful.' Meanwhile, P8 believed there was room for improvement in terms of achieving the precise outcomes desired in closed-ended tasks. This feedback might suggest the need for clearer instruction or further refining tool functionalities to accommodate users' varying expertise levels.

4.5.2.2 Cognitive load by NASA TLX

A unique participant, P5, rated higher than the neutral score of four concerning the mental demand dimension. P5 identified the primary cognitive challenge as discerning the optimal moments to schedule prompts, especially during the initial noisy phases of image generation. P5 described, 'There were instances when determining the best timesteps to influence the image's texture or shape, I struggled. Especially when the diffusion process was still in its early stages.' In the dimension of frustration, P5 also notched a rating surpassing the neutral threshold, attributing it to similar nuances in scheduling prompts. According to the IQR analysis, P5's score on frustration is an outlier. However, the overarching sentiment amongst participants was predominantly positive. For instance, P1 found the tool intuitive, and the denoising visualization particularly fun. Another participant, P7, reflected, 'The results resonated with my anticipations, making the entire exercise fulfilling.' P10 emerged as the only participant who perceived a heightened effort, citing, 'At times, I needed numerous iterations, especially when utilizing the word cloud for inspiration, to ensure the desired image outcome.' P11's rating on the effort dimension is an outlier as per the IQR analysis.

P8's frustration rating was a notable five, predominantly due to occasional inconsistencies between the provided prompts and the final generated images. As P8 expressed, 'On several occasions, the prompts, even after meticulous scheduling, didn't seem to significantly influence the final image or required exhaustive fine-tuning.' Ratings from both P8 (scoring five) is an outlier according to the IQR analysis. Nonetheless, the general consensus pointed towards a gratifying experience, with many, including P2, P3, and P4, P7, P11 enjoying their engagement with PromptZEN. In summary, most participants resonated with the promising capabilities of PromptZEN, and the mental demand, effort, and frustration levels were mostly minimal across the tasks.

4.5.2.3 Ablative assessment

In this section, we consolidate evaluations of the tool's individual components to provide insights for future work.

Word cloud empowers creativity through synonym exploration. A majority of the participants appreciated the word cloud's function. P4 said, 'Especially when I'm out of ideas, the word cloud sparks inspiration.' This sentiment was shared by P6, who felt it widened the scope of creative possibilities. However, P8 and P10 had reservations. P8 felt, 'The word cloud is informative, no doubt. But there's also a risk of over-relying on it, leading to possibly repetitive outcomes.' P10 added, 'While I used it often, I wish there were a way to filter or group the suggestions.' P2 used the word cloud as a way to discover new edits, mentioning, 'Whenever I felt stuck, the word cloud became my go-to. It not only gave me alternate terms but sometimes resulted in unpredicted and beautiful outcomes.' In essence, the word cloud facilitated participants in diversifying their keyword choices. However, its reliance on an initial keyword to generate related terms might narrow down the exploration capabilities. Future iterations could explore mechanisms to expand the breadth of word suggestions.

Prompt scheduling enhances control but requires precision. The feature of

prompt scheduling was widely acknowledged for the control it offered. P1 commented, 'The ability to schedule prompts allowed for a nuanced touch to the images. It's particularly empowering when I have a clear vision.' However, as P7 remarked, the requirement for precision was unmistakable. 'It's like having a double-edged sword. While I can influence the image, it's so easy to skew it if I'm not exact.' The challenge of maintaining the image's coherence when introducing terms at the early stages, related to Stable Diffusion's process, was echoed by P3 and P5. For example, P5 said, 'Earlier keywords had a bigger impact on the image than the later keywords. It took a while for me to realize that the later stages had almost no effect on the image'. Similarly, P3 commented, 'Prompt scheduling makes it look like timesteps have linear influence but in reality, the impact on the final image is far from linear.' In summary, while participants easily understood and utilized prompt scheduling, they also offered insights for potential refinements in future iterations.

Denoising visualization aids in pinpointing scheduling times. The denoising visualization received commendations for its transparency and insights into the image's progression. P7 found it invaluable, stating, 'From pure noise to a complete image, this visualization acted as a guide, letting me intervene at the right moments.' P9 felt more in control, sharing, 'It's a representation on how the image is created, helping me plan my edits.' Yet, for P11 and P12, there was a learning curve. P11 commented, 'It took me a few tries to understand and utilize the visualization effectively.' Also, some participants, like P2, P3, and P4, felt that the early stages of the visualization, where there is mostly noise, did not give them much to work with. P3 said, 'At first, I thought I should wait until I see something clear in the image before adding keywords.' In summary, participants leveraged denoising visualization to schedule their prompts successfully. However, there was an initial learning curve.

Prompt scheduling moves from a broad to a detailed effect on the image. Many participants noticed the step-by-step progression of Stable Diffusion. It starts by setting up the general scene, then focuses on shapes, and finally refines textures. P6 described it like

this: 'With Stable Diffusion, it is like painting. You start with the big picture, like the horizon, then add details like trees, and finally, you focus on the small things like leaves.' P2 observed, 'The early steps set the scene, and they have a big impact on the final image.' Going into more detail, P9 shared, 'Once the scene's set, you can not change it much even if you add new keywords later. But once I understood this order of creation, everything made sense.' P10 emphasized, 'It's important to get the shapes right early on because that sets the foundation.' P7 pointed out the role of the later stages, saying, 'Textures come in towards the end. So, changing things in the later steps can really change how an object looks.' To sum up, understanding how Stable Diffusion works—from setting the scene to adding textures—helps users know when and how to make edits. This understanding lets users get the most out of PromptZEN, making it a more powerful creative tool.

4.6 Limitations and Future Work

This section discusses several issues in the current tool and possible solutions for future work.

Limitations of the user study. Future work can increase the number of participants and the number of tasks beyond the current study. Moreover, it can include tasks that include subsequent image edits. Testing PromptZEN with alternative mechanisms would provide insights into its dependency on cross-attention map injection and its generalization capabilities. Evaluations with updated versions of Stable Diffusion and their novel mechanisms warrant consideration.

Image inversion. The image inversion method employed in this research encountered difficulties when applied to intricate scenes. This highlights a recognized limitation within current image editing methodologies. Advancements in the field may yield enhanced image inversion strategies. Consequently, exploring these methods with detailed, real-world scenes is an avenue for future research. Complex scenes may necessitate varied user interactions, such as employing heatmaps for inversion verification or specifying particular objects for

more accurate pixel-to-keyword alignment. The possibility of adjusting prompt directives based on different image regions exists. It should be noted that as the field progresses, adaptation in methodology may be necessary.

Choice of baseline methods. While there are alternative baselines to compare with PromptZEN, such as InstructP2P [BHE23], it's essential to understand that PromptZEN primarily serves as a supplementary tool, enhancing user controls in existing techniques. The attention injection mechanism, which is the foundation of our work, is not fixed; it can be substituted with other techniques in future developments. Our primary contribution lies in the user interactions we've introduced and the evidence that they can enhance editing performance. We selected our baseline methods with this central premise, as they too rely on the attention mechanism for their modifications.

Number of scheduled keywords. While participants had the autonomy to explore intricate scheduling patterns, they typically employed three keywords in their schedules. The task of formulating intricate schedules and interpreting scene generation from the final image is challenging. Future work can include enhanced feedback mechanisms for prompt schedules. Potential interactions could encompass systematic comparisons of prompt schedules, and the formulation of data trees to visualize and monitor schedule intricacies. Moreover, automated prompt schedule suggestions aligned with user preferences might aid in optimizing scheduling choices.

Word cloud improvements. Future iterations might re-evaluate the method for keyword discovery. Word clouds often return similar terms, limiting users in discovering novel keywords. An alternative approach could offer dissimilar words, potentially clustering them for enhanced user experience. A shift from traditional word clouds to a 'keyword discovery' mechanism would aim to facilitate users' exploration within the keyword space. In return, this can increase the diversity of the generated images.

4.7 Conclusion

Large diffusion models, particularly in text-to-image applications, have broadened the horizons of machine learning applications. Yet, their intricate structures have limited non-expert users. PromptZEN addresses this challenge, offering a tool that provides enhanced, intuitive control mechanisms for users, transcending the limitations of simple textual prompts. By facilitating prompt scheduling and integrating user-friendly features like context-driven word clouds and denoising visualizations, PromptZEN enables users to conduct local edits with precision. Our empirical study, involving 12 participants, confirmed the tool's effectiveness. In closed-ended tasks, PromptZEN surpassed three benchmark methods in producing images that closely resembled the target images. Similarly, in open-ended tasks, participants predominantly preferred PromptZEN to other methods. Thus, PromptZEN not only provides enhanced control to users but also objectively outperforms existing solutions. The results and capabilities of PromptZEN pave the way for future work, suggesting a strong potential for further advancements in refining and expanding user interactions in text-to-image diffusion models.

CHAPTER 5

Enhancing User Understanding through Text-to-Image Model Explanations

5.1 Introduction

Recent advancements in text-to-image models (T2I), including Dall-e [RPG21, RKH21], Imagen [SCS22], and Stable Diffusion [RBL22] have enabled the generation of intricate and contextually accurate images from textual descriptions. Applications of T2I models span diverse areas such as art generation [Opp22], design prototyping [KPJ23], visual aids in education [VT23], medicine [WWA22], and entertainment [LPL20]. Nevertheless, for the inexperienced end user, these models largely operate as a 'black-box', obscuring the underlying mechanisms that dictate their behavior. In the absence of explanations regarding the generation of specific images, users are deprived of a nuanced understanding of the model's operation, thereby inhibiting their capacity to exploit its full potential. Generating detailed images via computational models necessitates specific and well-structured prompts. To facilitate this process, the image-generation community has developed 'prompt books' as reference guides. Moreover, the emergence of prompt engineers, specialists dedicated to the formulation and refinement of prompts, underscores the complexity of achieving desired outputs. Nonetheless, for those unfamiliar with the intricacies of the method, attaining the intended image remains a formidable task. Significantly, a unique aspect of T2I models is that understanding the generation process can directly enhance the quality of the images produced. Grasping the generation mechanics and nuances results in superior prompts, thereby yield-

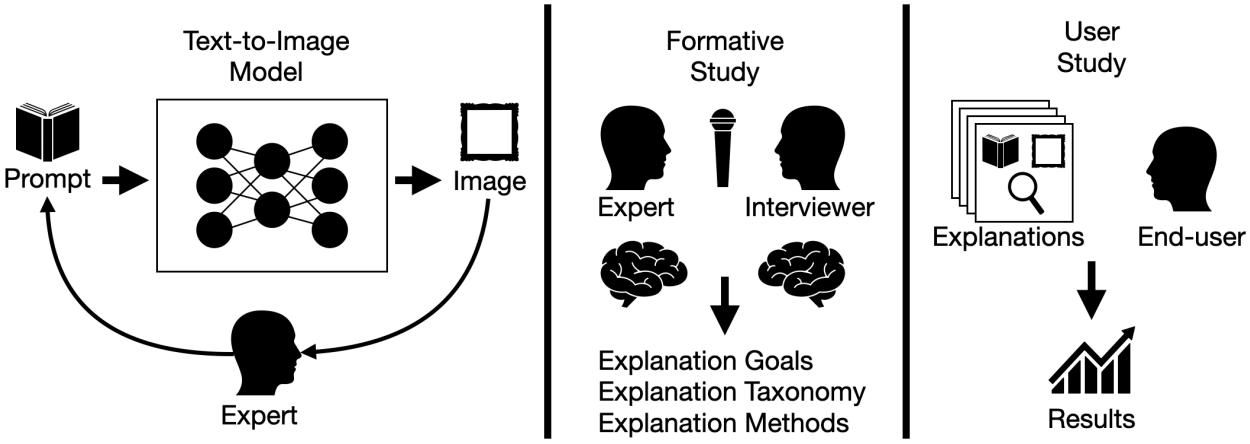

Figure 5.1: Visual representation of our process: Experts go through many iterations of prompts to create an image. Expert-led iterations illuminate text-to-image(T2I) model explanation goals and methods. Drawing from these expert insights, we designed and conducted a comprehensive user study with 473 participants, unveiling preferences and challenges in T2I explainability. Our efforts bridge the gap between complex T2I model operations and the understanding of novice end-users.

ing better images. Consequently, developing explanation techniques for T2I models directly contributes to improved image quality in downstream tasks for the user. Consequently, it is pivotal to identify explanation goals and devise techniques that best serve the user in the context of image generation.

Wang *et al.* [WYA19] investigated explanation goals in traditional Explainable AI (XAI) methods, which are predominantly applied to classification or regression tasks. However, Text-to-Image (T2I) generation represents a distinct domain and, consequently, has different objectives. Addressing these varied goals calls for the utilization of a range of explanation techniques. To pinpoint unique explanation goals inherent to T2I models, we executed a formative study involving eight experts. These experts identified a subset of goals originating from traditional XAI objectives. Specifically, participants sought explanations to 'filter causes,' 'generalize and learn' a mental model of the generation process, and 'pre-

dict and control' future generations. Following this, we facilitated a participatory design session with the participants to formulate explanation techniques in alignment with these identified goals. The techniques suggested by the participants can be categorized into four main groups: sensitivity-based, model-intrinsic, surrogate model, and example-based, similar to the 'functioning approach' taxonomy introduced by Speith [Spe22]. Using these four main explanation groups, we designed four distinct explanation methods for T2I models: redacted prompt explanation, keyword heat map, keyword linear regression, and keyword image gallery. These explanation methods are representative of the explanation techniques that are elicited during the formative study.

Subsequently, we implemented and evaluated these four explanation methods on a larger cohort of 473 participants (with 5676 responses) sourced from Amazon Mechanical Turk (AMT) across six distinct tasks. The choice of AMT was motivated by our interest in assessing the effectiveness of these explanations among inexperienced end-users. From our user study, we derived insightful findings that we anticipate will steer subsequent research related to explainable methods for T2I models. These insights not only hold relevance for tool designers but also for ML researchers aiming to enhance the explainability of T2I systems. Additionally, we have contributed a dataset comprising prompt and image pairs for future explainability research. Our data indicate that users are able to grasp new keywords via explanations and exhibit a preference for example-based explanations. Conversely, they encounter difficulties with keywords that alter the overarching theme and often overestimate their comprehension of the explanations provided. Moreover, users typically favor one to two explanation methodologies concurrently, with any further addition diminishing their performance.

Our main contributions can be summarized as:

- A curated dataset tailored to evaluate explainability techniques within T2I models.
- Empirical results from a large AMT user study, offering a resource for researchers

seeking a deeper understanding of user interactions with T2I explanations.

- Key observations and insights poised to inform and shape future research in T2I model explainability.

5.2 Related Work

Text-to-Image explanations. There are some introductory progress on trying to gain more insight to T2I models using AI models. For example, What the DAAM, explains T2I models by upscaling and aggregating cross-attention word–pixel scores [TPJ22]. X-IQE leverages visual large language models (LLMs) to evaluate text- to-image generation methods by generating textual explanations [Che23]. Liu *et al.* conducted a study exploring what prompt keywords and model hyperparameters can help produce coherent outputs. They present design guidelines that can help people produce better outcomes from text-to-image generative models [LC22].

Text-to-Image controls. On the other hand there are also various methods that try to gain more control over T2I models and offer certain perspective regarding their capabilities. For example, T2I-Adapter propose to learn simple and lightweight T2I-Adapters to align internal knowledge in T2I models with external control signals [MWX23]. Similarly, ControlNet allows users to add conditions like Canny edges, human pose, etc., to control the image generation of large pretrained diffusion models [ZRA23]. ImageReward is a general-purpose text-to-image human preference reward model to effectively encode human preferences [XLW23]. AnimateDiff is an effective framework for extending personalized T2I models into an animation generator without model-specific tuning [GYR23]. On the other hand Wu *et al.* proposed a simple yet effective method to adapt Stable Diffusion to better align with human preferences based on Human Preference Score (HPS) [WSZ23]. Observing that T2I models still lack precise control of the spatial relation corresponding to semantic text. Yan *et al.* takes text and mouse traces for image generation, as traces provide a more

natural and interactive way than layouts to ground the text into the corresponding position of the image [YJW22]. Promptist proposes prompt adaptation, a general framework that automatically adapts original user input to model-preferred prompts with llm [HCD22]. TIME receives a pair of inputs: a 'source' under-specified prompt for which the model makes an implicit assumption (*e.g.*, , 'a pack of roses'), and a 'destination' prompt that describes the same setting, but with a specified desired attribute (*e.g.*, , 'a pack of blue roses') [OKB23].

Traditional XAI explanations. The use cases of traditional XAI methods on classification and regression tasks differ from the use cases of generative tasks. Lim *et al.* [WYA19] summarize the explanation goals in traditional XAI as: *(i)* filter to a small set of causes to simplify their observation [Lom06], *(ii)* generalize these observations into a conceptual model to 'predict and control' future phenomena [Hei13]. Additionally, Nunes and Jannach suggested transparency, improving the decision-making when AI is used as a decision aid, and debugging as goals of traditional XAI methods [NJ17]. Jeyakumar *et al.* conducted a cross-analysis Amazon Mechanical Turk study comparing the popular state-of-the-art explanation methods to empirically determine which are better in explaining deep neural network model decisions [JNC20]. Fok *et al.* argue that explanations rarely enable complementary performance in AI-advised decision-making. Interestingly, they argue explanations are only useful to the extent that they allow a human decision-maker to verify the correctness of an AI's prediction, in contrast to other desiderata. This sentiment is relevant to explanations in T2I models, because explanations in T2I models are inherently more verifiable compared to traditional XAI methods since the image output carries more information [FW23]. Buçinca *et al.* argues that the limitations of contemporary explainable AI solutions are not appreciated because the most commonly-used methods for evaluating AI-powered decision support systems are likely to produce misleading (overly optimistic) results [BLG20]. In another work, Buçinca *et al.* argues that people over-rely on AI and they accept an AI's suggestion even when that suggestion is wrong. More interestingly, adding explanations to the AI decisions does not appear to reduce the overreliance [BMG21]. On the other hand Jacobs

et al. shows that there is mounting evidence that human+AI teams often perform worse than AIs alone.[JPJ21]. These observations should be considered again in the context of T2I explanations considering the underlying differences between the T2I explanations and traditional XAI explanations.

5.3 Formative Study

In order to comprehend the distinct explanation goals and pinpoint potential techniques for text-to-image models, and to elucidate their differences from traditional XAI methods, we conducted a formative study with eight participants who use text-to-image models regularly. These participants were invited to the study from the stable diffusion[1] and Midjourney[2] discord channels. Six of the participants were male and two of them were female, aged 21 to 37. Five participants had programming experience and four of them were familiar with XAI methods from traditional tasks. All of the participants used text-to-image generation tools regularly for at least a year.

Each session was a brainstorming discussion between the first author and a participant. We followed think-aloud protocol [Lew82] where the first author was the interviewer and note-taker, whereas the participant was the primary contributor. The interviewer focused on prompting the participant to clarify and broaden their ideas. The main purpose was to elicit potential explanations for T2I models that can be beneficial for the users. Each discussion lasted for around an hour. For the first ten minutes, participants were asked to brainstorm the potential **explanation goals** of T2I models. The remaining discussion revolved around how these goals can be achieved through various explanation ideas, similar to participatory design [Spi05]. We employed a method akin to the Affinity Diagram approach [HB97], based on which we aggregated participants' responses to summarize their ideas. Specifically, the

[1] discord.gg/stablediffusion

[2] discord.gg/midjourney

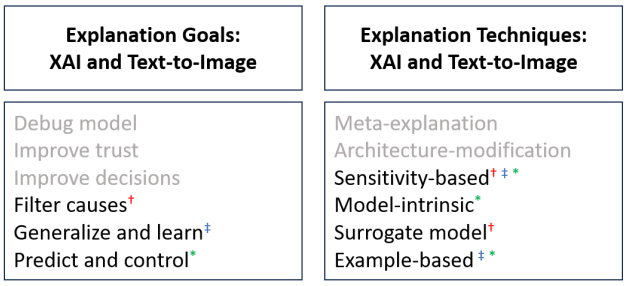

Figure 5.2: All goals and techniques presented here originate from traditional XAI. However, our objective was to contrast these with those of T2I. Explanations and techniques that are grayed out are exclusive to traditional XAI, while the others are applicable to both XAI and T2I. Elicited explanations and techniques are matched using symbols. Participants determined these relationships between goals and techniques upon being prompted with specific goals by the interviewer.

first author transcribed participants' responses to develop the initial codes, which are then reviewed by the second author. Disagreements were resolved via discussion between the authors.

5.3.1 Explanation goals for text-to-image models.

To derive potential explanations for T2I models, participants initially brainstormed the explanation goals specific to these models. As indicated by the participants and illustrated in Figure 5.2, the explanation goals of T2I models exhibit differences from those of XAI methods, while also presenting certain similarities. Interestingly, even though we did not show traditional XAI methods to the participants, there were many similarities in the brainstormed explanation goals. Unlike classification tasks, T2I models do not make or aid decisions but create images. Since these models are not used as decision-aid systems, the users are not interested in improving their 'decision-making process'. Rather, their aim is to deepen their understanding of the model to craft more effective prompts, thereby leading to the generation of superior images. Similarly, the users do not seek explanations for the purpose of 'debug-

ging' the model or improve their 'trust' in the model. For instance, P1 pointed out, 'There is no right or wrong when you are doing something creative. The output quality is highly subjective naturally.' P4 extended on this with more nuance and stated that, 'If a model is incapable of generating certain phrases, it is typically due to lack of training data and it does not change my confidence in other generations.' In a similar fashion, P7 pointed out that generated images themselves can be considered as proof that the model understands the context and the complex relationships in a scene. The variation in explanation goals between T2I models and XAI methods can influence the selection of suitable explanation techniques.

On the other hand, similar to traditional XAI methods, the users want to 'generalize' their observations into a conceptual model to 'predict and control' future phenomena, *i.e.*, future image generations. For example, P2 said, 'After a while, you get a feeling of which keywords[3] go together and how they change images.' P3 extended this idea by stating the main goal of image generation is to investigate the range of feasible images, and the conceptual model is crucial for this exploration. Similarly, the users want to 'filter causes' to a small set for simplification. P1 said, 'One of the first things I do is to figure out which words in my prompt contributed to the image.' P8 pointed out that the shorter prompts are easier to understand and improve on. P4 iterated on this notion and said, 'I want to understand how the changes to the prompt affect the image so that I can improve the prompt.', highlighting the importance of explanations for future image generations. P5 also talked about the iterative nature of T2I by stating, 'It is impossible to improve a prompt if you do not understand why it results in that particular image.' The participants tend to not 'debug' the model but 'debug' their prompts by analyzing them.

In conclusion, the users want the explanation in T2I models to simplify their conceptual model similar to traditional XAI methods. The difference is that they do not seek these

[3]Typically prompts consist of 'keywords' that are separated with commas. Therefore, a keyword in the context of this paper refers to multiple words such as, '..., digital painting, highly detailed, ...'.

simplifications to identify when the models fail or to debug the model, but to improve the model output iteratively with better prompts. The traditional XAI methods in classification tasks are not iterative by nature whereas T2I models require careful iterative prompting. Understanding a classifier's decision-making process does not directly increase the classification quality, whereas understanding a generative model's generative process can improve the prompts which leads to higher-quality outputs.

5.3.2 Participatory Design: Potential explanation techniques in text-to-image models

With the explanation goals for T2I models in mind, participants brainstormed potential explanations that can help the users. The explanation ideas that surfaced in interviews can be summarized in four groups: *(i)* sensitivity-based explanations, *(ii)* model-intrinsics, *(iii)* surrogate models, and *(iv)* example-based explanations. The *explanation taxonomy* we elicit from the study is similar to the *functioning approach* taxonomy introduced by Speith [Spe22]. As illustrated in Figure 5.2, the explanation techniques that surfaced during discussions, when prompted with the explanation goals, are highlighted.

Sensitivity-based explanations. Sensitivity-based explanations observe the change in the output when the prompt is slightly changed. During our study, participants mentioned that they investigate the effect of words by removing or adding them. For example, P6 mentioned how he builds his conceptual model by stating, 'Sometimes I remove a keyword to see how it changes the image, it helps the mental model I have about the generation process.' P4 also mentioned a similar strategy where she changes the strength of individual words to understand their effect on that particular image. Interestingly, P8 pointed out a common issue he faces, 'Sometimes adding a phrase changes the image as I expected, but then when the same phrase is added to another prompt, it does not have the same effect.' P8 concludes his thought process by arguing that prior knowledge about a phrase is not enough to confidently add it to a prompt and it needs to be tested iteratively for each new prompt.

While all participants utilized sensitivity-based explanations in their workflow, many (P1, P3, P4, P6, P8) highlighted the cognitive challenge of monitoring different tests due to the lack of a structured method. This underscores the importance of introducing tools that can assist in tracking and managing sensitivity-based explanations.

Model-intrinsic explanations. Explanations that use model-intrinsics leverage the model architecture. Rather than treating the model's generation process as a black box, these explanations delve into the model to offer insights into its workings. To our surprise, although none of the participants knew about the specifics of the model architecture, they frequently suggested explanation methods that would use model-intrinsics. Several participants (P1, P4, P5, P6, P7) offered varied suggestions based on the way the model processes words. For example, P1 mentioned, 'I would like to see the effected regions in the image for each word.' P4 extended on this idea by suggesting that these regions can be shown to the user as heatmaps, which can inform the user when a word is ineffective. P5 suggested how the words are combined in the model could explain the resulting image, 'I want to check if the model understands the context by mixing the contextually related words.' P2 and P8 suggested similar systems that would warn the user about certain phrases when they are not effective and suggest replacements. Despite these ideas, explanations that utilize model-intrinsics currently remain unavailable, leading to none of the participants incorporating such explanations into their workflows.

Surrogate model explanations. Surrogate model explanations involve using simpler or more interpretable models to approximate and explain the behavior of a more complex model. Some participants (P5, P6, P7, P8) suggested using surrogate models for explanations. For example, P5 suggested that an image can be explained as a linear combination of the words in the prompt. In contrast, P6 and P7 favored a non-linear approach, recommending the use of non-linear surrogate models such as decision trees. P8 recommended developing a distinct dialogue-based large language model (LLM) that can elucidate the rationale behind the produced image, shedding light on the generation process.

All these suggestions revolve around developing a new, more human-interpretable model to explain T2I models. Interestingly, both P1 and P2 argued that using surrogate models for explanations adds another layer of uncertainty. P1 expressed, "If we employ a separate model for explanations, how can we be sure it accurately represents the original model's decision-making process?" This highlights concerns about the reliability of surrogate models as true reflectors of primary model behavior. In classification and regression tasks, the accuracy of the surrogate model can be quantified using various metrics. However, for generative models, this straightforward assessment is not readily available. When prompted, P2 argued that in such cases, we can rely on qualitative analysis, such as human observation, which inherently has a degree of subjectivity. Similar to model-intrinsic explanations, surrogate model explanations are currently unavailable, leading to none of the participants incorporating such explanations into their workflows.

Example-based explanations. Traditionally, example-based explanations involve presenting instances that the model has previously encountered or processed, helping users to understand the model's behavior and decision-making patterns. In the context of T2I generation, this can have multiple interpretations as indicated by the participants. For example, P1 referenced the use of prompt search engines[4] to better comprehend the impact of specific keywords. P2, P5, P7, and P8 also emphasized the utility of these search engines to both discover and comprehend new keywords through multiple examples. Notably, P4 mentioned that when keywords are similar across different prompts, their resulting images can also appear alike, making differentiation challenging. To address this, P4 combines sensitivity-based explanations with example-based ones to discern subtle differences between similar keywords. In a different approach, P3 recommended showcasing related prompts and their corresponding images when given a specific prompt. Instead of browsing through an image gallery for specific keywords, the emphasis here is on contrasting entire prompts using text embeddings for a more discernible comparison. P5 emphasized the challenges with creating

[4]For example prompthero.com and lexica.art

prompt-image pair galleries: not only is it costly, but with each new model, the utility of keyword searches diminishes due to a lack of relevant data. While one might consider using older datasets as a workaround, it's no guarantee that keywords influencing older models will have the same effects on newer ones. It underscores the need for a balance between retaining older examples and generating new datasets from user interactions. Additionally, P5 stated: 'Given that these foundation models are frequently fine-tuned, relying on prompt search engines may not be a sustainable strategy.' An alternative approach could be utilizing LLMs to generate similar prompts and craft example images, even if it demands higher computational resources. All participants incorporated example-based explanations into their workflow. Interestingly, several participants, namely P3, P4, and P6, highlighted using these explanations as initial inspiration in their processes.

In summary, participants identified four primary explanations considering the *explanation goals* of a T2I model. Some of these explanations, such as sensitivity-based and example-based explanations, are practical. This is because participants already incorporate similar methods into their workflows. However, some explanations are not currently in use, but participants believe they would be beneficial, aligning with the explanation goals established earlier in the study.

5.4 Explanations for Text-to-Image Models

In laying the groundwork for explanation methods in text-to-image models, we introduce four explanation techniques, each representing a distinct type of explanation that was elicited from the formative study. While the study presented a range of ideas, we centered our implementation on the most prevalent concepts. The explanation methods for each type are explained and implemented as follows:

(i) **Redacted Prompt Explanation.**

Redacted prompt explanation (RPE) is a sensitivity-based explanation that frequently

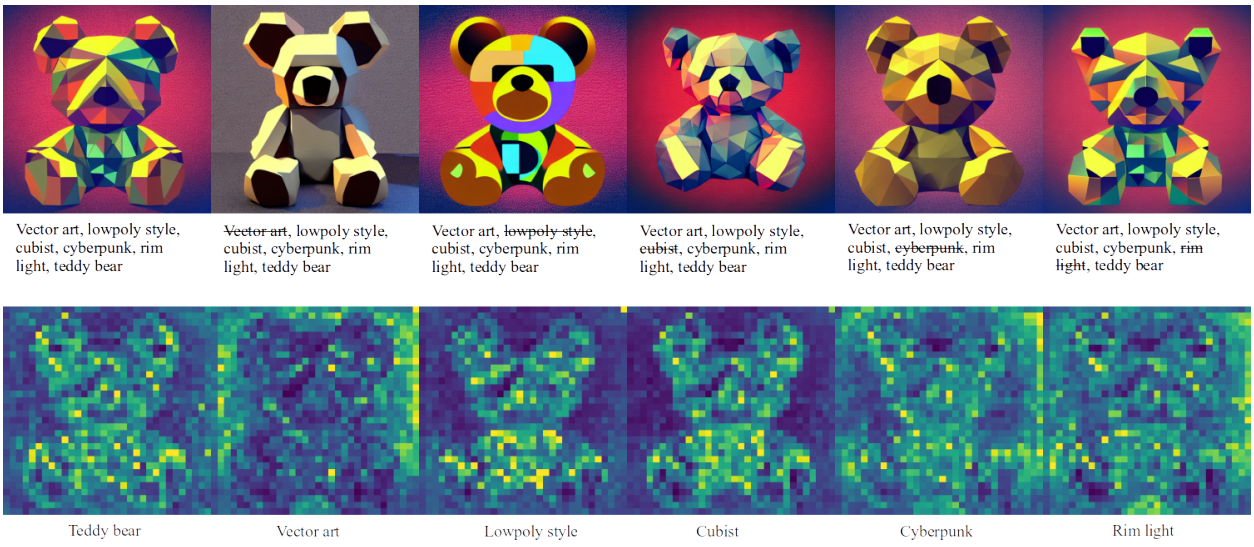

Figure 5.3: Redacted prompt explanation and keyword heat map. The first row displays a sample redacted prompt explanation. On the left, we present the original prompt alongside its resulting image. Subsequent images demonstrate the outcome when a keyword is omitted, providing insights into the impact of that particular keyword on the image's formation. The second row showcases keyword heat maps for the same original image and prompt. Each column corresponds to a distinct keyword, labeled below. For each keyword, cross-attention heatmaps highlight where the model's attention is concentrated. For instance, the keywords 'vector art' and 'cyberpunk' appear to influence the background, a finding that aligns with the redacted prompt explanation for those specific keywords.

came up during the participatory design process. An example RPE can be seen in the first row of Figure 5.3. This technique systematically redacts or removes keywords from prompts to gauge their impact on the generated image. RPE evaluates the similarity between the original and altered images. RPE operates by randomly removing a subset of keywords from the original prompt for each image-prompt pair, until the resultant image undergoes a 'significant' change. Formally, let P represent the original prompt, which is a set of keywords:

$$P = \{k_1, k_2, \ldots, k_n\}$$

Let $I(P)$ denote the image generated by using the entire prompt P. For any subset $P' \subseteq P$, $I(P')$ is the image generated by using the prompt subset P'. Let $S(I_1, I_2)$ be a similarity metric between two images I_1 and I_2. For a given threshold θ, which determines the 'significant' change in images, we randomly select a subset P' from P until the similarity satisfies:

$$S(I(P), I(P')) < \theta$$

Different methods can be used to define the sub-sampling, similarity, and θ. In our implementation, we adopted random subsampling, used CLIP embeddings [RKH21] to measure similarity, and set a predetermined constant for θ. The value of θ is qualitatively adjusted by reviewing examples to determine if image alterations are substantial enough to infer the influence of the excluded keywords. After applying RPE, both the original and altered image-prompt pairs are displayed to shed light on the effects of the redacted words. The primary objective of RPE is to elucidate the influence of a redacted keyword through image comparison.

(ii) **Keyword Heat Map.**

The Keyword Heat Map (KHM) is a model-intrinsic method that leverages Stable Diffusion parameters to craft a heatmap, serving as a visual explanation. An example KHM can be seen in the second row of Figure 5.3 Drawing inspiration from the Prompt-to-prompt technique [HMT22], KHM averages the cross-attention maps corresponding to each keyword present in the prompt, presenting the outcome as detailed heatmaps. These maps effectively elucidate which pixel regions in the image are more influenced or "attracted" by specific keywords, offering insights based on the model's internal parameters. A multitude of visualization strategies can be employed to illuminate the model's intricate interactions with the keywords during the image generation process. For instance, an exploration of these cross-attention maps across distinct denoising timesteps can provide clarity on the model's evolving focus throughout the generation journey. Such a timestep-based breakdown could reveal reasons why certain visual concepts, although present initially, may not

dominate the final image. In fact, techniques like Attend-and-Excite harness this temporal information, driving the model to ensure no keywords are overlooked in the generation process [CAV23]. While timestep-based heatmaps offer valuable insights, our implementation prioritizes a holistic view. We average across all timesteps and adopt a 32x32 resolution, a decision supported by prior work highlighting the efficacy of this resolution in semantic clustering [PGA23]. Formally, let P represent the original prompt, which is a set of keywords. For each keyword k in P, we extract cross-attention maps, $\mathcal{A}_{t,h}(k)$, for each timestep t and attention head h. Then we average the cross-attention maps across both timesteps and attention heads to produce a non-normalized heatmap, $\mathcal{H}'(k)$:

$$\mathcal{H}'(k) = \frac{1}{T \times H} \sum_{t=1}^{T} \sum_{h=1}^{H} \mathcal{A}_{t,h}(k)$$

where T is the total number of timesteps and H is the total number of attention heads. Finally we normalize $\mathcal{H}'(k)$ to get the heatmap $\mathcal{H}(k)$ such that the values lie in the range [0, 1]:

$$\mathcal{H}(k) = \frac{\mathcal{H}'(k) - \min(\mathcal{H}'(k))}{\max(\mathcal{H}'(k)) - \min(\mathcal{H}'(k))}$$

After applying KHM, the resultant heatmaps are closely associated with their source keyword, underscoring their integral influence in guiding the image synthesis. The primary aim of KHM is to provide a basic understanding of how keywords influence the final image outcome.

(iii) **Keyword Linear Regression.**

The Keyword Linear Regression (KLR) is a surrogate-model explanation that approximates the image generation process as a linear combination of its constituent keywords. KLR lacks visual representation capabilities compared to other methods. Instead of visual cues, this method directly reports numerical values corresponding to the significance of each keyword. Although there are other methods available, such as decision trees, we opted for linear regression due to its straightforwardness. Though various methods can be used to fit the linear model, we used a large prompt-image dataset that is publicly available. Specifically, we

used DiffusionDB dataset which contains 1.8 million unique prompts and 14 million images generated with Stable Diffusion v1.4 [WMM22]. Our approach is characterized by first identifying the 20 most related images in the dataset for each keyword. These related images are identified by comparing the CLIP embeddings. A notable benefit of utilizing CLIP embeddings is that even if a prompt does not explicitly contain a particular keyword, the resultant image can still resonate with the essence of that keyword. Once these images are identified, their VIT image embeddings[5] are averaged, ensuring that the emphasis remains squarely on the characteristics of the images themselves, rather than being constrained by the scope of CLIP [DBK20]. Using the embeddings for each keyword, linear regression is applied to predict the embedding of the original image. The weights associated with each keyword are viewed as an approximation of their contribution to the final image representation. After the linear regression the weights are normalized so that they collectively sum to 1. Finally, the weights for each keyword are then provided to the user to illustrate the contribution of each keyword to the generated image. Formally, given a dataset $\mathcal{D}$ consisting of images $\{I_1, I_2, ..., I_m\}$, we denote the embeddings for each image I and keyword k as $E_{\text{img}}(I)$ and $E_{\text{keyword}}(k)$, respectively using CLIP embeddings. The cosine similarity between an image and a keyword is defined by

$$\text{sim}(I, k) = \frac{E_{\text{img}}(I) \cdot E_{\text{keyword}}(k)}{\|E_{\text{img}}(I)\| \|E_{\text{keyword}}(k)\|}$$

For each keyword k, we determine the top 20 images from $\mathcal{D}$ with the highest similarity, termed TopImages(k). The refined embedding for the keyword k is then

$$E'_{\text{keyword}}(k) = \frac{1}{20} \sum_{I \in \text{TopImages}(k)} E'_{\text{img}}(I).$$

where $E'_{\text{img}}(I)$ is the VIT image embedding. Applying linear regression, the embedding of the generated image is modeled as

$$E_{\text{img}}(I_{\text{generated}}) = \beta_0 + \sum_{j=1}^{n} \beta_j E'_{\text{keyword}}(k_j) + \epsilon,$$

[5]We used google/vit-base-patch16-224 from Huggingface

where β_j indicates the weight of keyword k_j, β_0 is the constant term, and ϵ captures the model's prediction error. Lastly, to normalize the coefficients, we compute

$$\beta'_j = \frac{\beta_j - \min(\beta)}{\sum_{j=1}^{n}(\beta_j - \min(\beta))},$$

ensuring that $\sum_{j=1}^{n} \beta'_j = 1$.

In the absence of available prompt-image dataset, the embeddings for each keyword can be derived by calculating the discrepancy between the embeddings of the full-prompt resultant image and that with the omitted keyword, similar to sensitivity-based explanation. The pool of embeddings can be augmented by modifying this prompt to reduce variance. After implementing KLR, the keywords paired with their weights give a linear interpretation of how each keyword contributes to the image creation. The primary aim of KLR is to simplify the intricate generative process into a more understandable linear model.

(iv) **Keyword Image Gallery.**

The Keyword Image Gallery (KIG) offers an example-based explanation, elucidating the influence of each keyword through a curated gallery of images. An example KIG is given in Figure 5.4 for the teddy bear image presented in Figure 5.3. While there are several methods to curate this gallery, our approach centers on utilizing the prompts directly. Essentially, for each keyword, the gallery showcases images from the dataset that feature that keyword in their prompts. The selection criteria for these images revolve around the similarity of their entire prompts to the original prompt, provided the keyword in question is present. Formally, given a dataset $\mathcal{D}$ of images and associated prompts, and an original prompt P_{original}. Embeddings of prompts are represented as: $E_{\text{prompt}}(P)$. The cosine similarity between two prompts is defined as:

$$\text{sim}(P, P_{\text{original}}) = \frac{E_{\text{prompt}}(P) \cdot E_{\text{original}}(P_{\text{original}})}{\|E_{\text{prompt}}(P)\| \|E_{\text{original}}(P_{\text{original}})\|}$$

For a keyword k present in P_{original}, the subset of prompts containing k is denoted as:

$$S(k) = \{P \mid k \in P\}$$

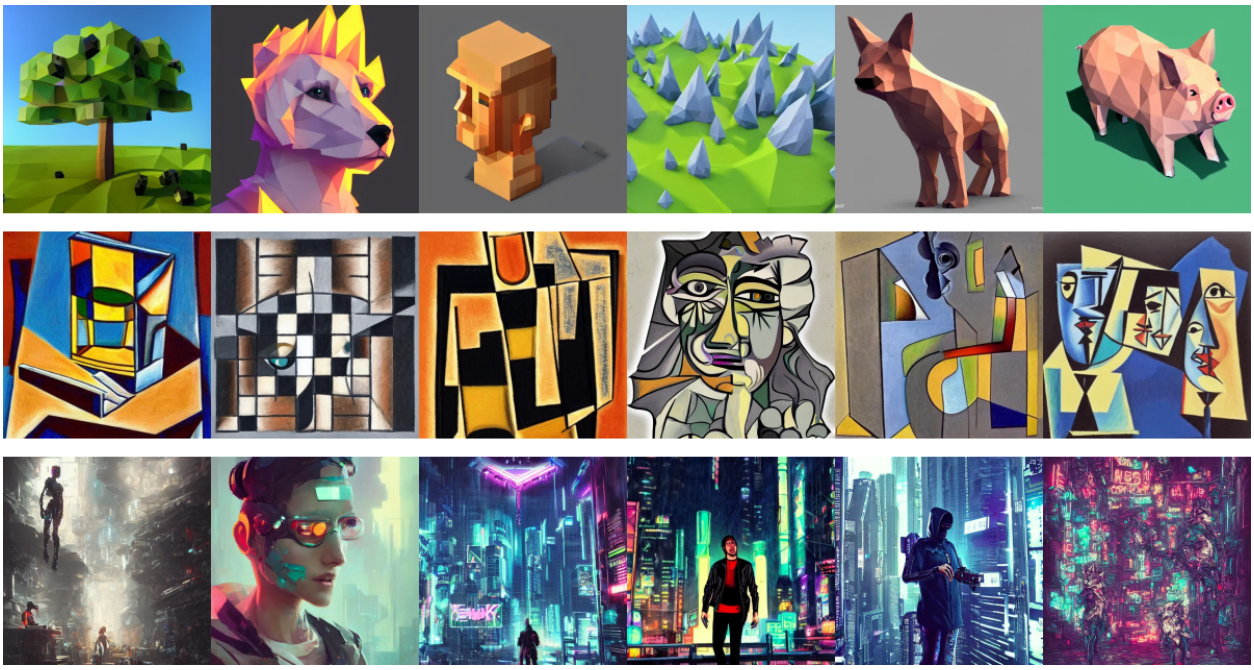

Figure 5.4: Keyword Image Gallery (KIG). KIG explains the 'teddy bear' image shown in Figure 5.3. Each row in the gallery corresponds to a specific keyword, organized from top to bottom as 'lowpoly style', 'cubist', and 'cyberpunk'. The primary objective of the KIG is to showcase exemplary images associated with each keyword. This provides context, helping users understand the influence and interpretation of each keyword.

Let n be a predefined number indicating the top n prompts to be selected based on similarity. The gallery for keyword k, curated based on similarity to P_{original}, is then:

$$G(k) = \text{argmax}_n \left(\text{sim}(P, P_{\text{original}}) \mid P \in S(k)\right)$$

In scenarios where prompt-image datasets for specific models are unavailable, images can be generated using prompts extracted from the DiffusionDB dataset. Alternatively, in the absence of prompts, a method akin to the one outlined in KLR can be employed to pinpoint images that align closely with the given keyword. The end result is a tailored collection of images for every keyword, highlighting the keyword's influence on the image generation process. The primary aim of KIG is to convey the impact of keywords using a range of

example images previously generated by the model.

5.5 User Study

In our formative study, while we did identify various explanation types and even proposed specific techniques for each in §5.4, the real-world efficacy of these methods remains to be explored. The effectiveness of an explanation method can be evaluated from various angles. However, as a pioneering effort in the realm of explainable text-to-image methods, we directed our focus towards end users, especially those unfamiliar with text-to-image models. With the rising ubiquity and accessibility of these models, understanding their reception and utility among mainstream users becomes crucial. For instance, with Apple now officially incorporating Stable Diffusion support in iPhones [OSW22], it underscores the importance of ensuring these explanations resonate with the general public. Our approach was to conduct a comprehensive user study using Amazon Mechanical Turk (AMT). The choice of AMT stemmed from its ability to quickly scale and accommodate a large number of participants, ensuring a diverse representation of end users. Additionally, AMT has been leveraged in previous research on user preferences in XAI methods, making it a reliable platform for our empirical analysis [JNC20].

5.5.1 Participants

473 participants contributed across an array of tasks. Each participant, on average, provided 12 responses (four per task), culminating in a comprehensive set of 5,676 responses. This extensive collection forms a solid base for extracting significant insights and conclusions about the practicality of our explanation methods.

Validating responses. Two filtering criteria were implemented to exclude participants who submitted illegitimate responses. First, similar to [JNC20], participants who submitted responses more quickly than the minimum threshold necessary are excluded, ensuring the

removal of submissions potentially auto-completed by bots. Second, each test input included an additional question: a random prompt and text pair from our dataset were presented to the user. For half of these instances, the original prompt was replaced with a random one, and users were then asked if the presented prompt corresponded to the image. Participants who failed more than 20% of these validation questions were subsequently excluded from the published results. To guarantee that the substituted prompt was not related to the original, CLIP embeddings were used. Prompts with low cosine similarities were selected as replacements.

5.5.2 Data & Apparatus

In some of the explanation methods, we leveraged a prompt-image dataset. Specifically, we used DiffusionDB dataset which contains 1.8 million unique prompts and 14 million images generated with Stable Diffusion v1.4 [WMM22]. We also proposed alternative explanations in §5.4 when these datasets are not available. We used the same Stable Diffusion model as our text-to-image model since it is open-source [RBL22]. We precomputed the images and explanations using a local machine equipped with an Nvidia GeForce RTX 3090 GPU. These results were then stored in an AWS S3 bucket, ensuring asynchronous delivery to the AMT participants during the study.

We curated a dataset of prompt-image pairs derived from the DiffusionDB dataset using a specific sampling method. This dataset can serve as a valuable resource for future researchers aiming to advance explainable AI techniques in text-to-image models. A straightforward random sampling from the DiffusionDB dataset proved suboptimal due to its imbalanced nature. For instance, certain terms like "cats" or "people" appear with greater frequency than others. While there exist various strategies to address dataset imbalance, we adopted a unique approach. Initially, we clustered images based on embeddings computed using a pretrained VIT model. During the sampling of prompt-image pairs from the various clusters, we ensured that each new sample's embedding was compared to those of existing ones. Only

when the distance between the new sample's embedding and the existing ones was higher than a predefined threshold was the sample added to the dataset. This iterative sampling continued across numerous rounds, ceasing only when a complete round failed to identify any new embedding that met the threshold criterion. Formally, $\mathcal{D}$ is the DiffusionDB dataset and $\mathcal{S}$ is our curated dataset which is initialized as an empty set, $\mathcal{S} = \emptyset$. $E(I)$ is the image embedding of image I, $\forall I \in \mathcal{D}$. $\mathcal{C}$ is the cluster sets, $\mathcal{C} = \text{cluster}(\{E(I)|I \in \mathcal{D}\})$. I^i_{sampled} is a sample from cluster i. Then,

$$\text{If } \min_{J \in \mathcal{S}} \text{dist}(E(I^i_{\text{sampled}}), E(J)) > \theta : \mathcal{S} \leftarrow \mathcal{S} \cup I^i_{\text{sampled}}$$

where θ is the threshold parameter. This cycle continues until $\mathcal{S}$ is stable over $\mathcal{C}$. Ultimately, our methodological approach yielded a diverse dataset comprising 516 distinct prompt-image pairs. The dataset's size can be modulated based on the study's requirements by adjusting the aforementioned threshold. For distinct research needs, we are making three distinct datasets of varying sizes publicly available, each generated utilizing different threshold values.

5.5.3 Tasks & Procedure

Our study was designed with three tasks for each participant. To ensure fairness and eliminate any potential biases, the sequence of these tasks was shuffled for different participants. The study commenced with a brief overview of what prompt and image pairs were. To familiarize participants with the study's mechanics, they were presented with two prompt-image pairs but weren't told which prompt corresponded to which image. They were then tasked with pairing the prompts with their rightful images by selecting a checkbox adjacent to the prompts. Before progressing to the main study, participants were required to correctly match the pairs on two separate occasions. Those who couldn't accomplish this preliminary task were not allowed to proceed further in the study. Prior to each task, participants are

presented with a sample correct response to familiarize them with the task.

The user study had three main goals. First, we wanted to find out which explanation method was preferred by users without technical backgrounds. Next, we aimed to measure how well these users actually understood the explanations given to them. Lastly, we were interested in seeing which combinations of explanations worked best together. In examining these facets, our objective is to optimize the design of text-to-image models to better cater to a broader, non-expert audience. These objectives are crucial for several reasons. Firstly, identifying user preferences can provide valuable insights to UX designers aiming to develop more user-friendly text-to-image models. Additionally, it is essential to assess the comprehensibility of the provided explanations. If explanations are not easily understood by users, their utility is reduced. Consequently, the results of this study can guide designers in selecting the most effective explanations, or combinations thereof, for inclusion in their applications.

In our formative study, we explored the potential of explanations in aiding users to refine their image prompts over subsequent interactions. However, an earlier pilot study indicated that familiarity with text-to-image models is important for prompt enhancement, even when equipped with explanations. As a consequence, our user study emphasized the comprehension and utility of explanations among non-expert users. In the future, the interplay between explanations and the iterative refinement of prompts warrants further investigation. Presently, our research establishes a foundational set of methods, validated by non-expert participants, serving as a reference point for future explorations.

Next, we introduce the three tasks we had in the user study: *(i)* user preference, *(ii)* performance, and *(iii)* combined explanations. Prior to each task, participants are provided with a demonstrative example to familiarize them with the task.

Task 1: User Preference on Explanations. To assess participants' preferred explanations, they were presented with the original prompt and image pair. Subsequently, two randomly selected explanations (out of a possible four) were shown, akin to the methodology

in [JNC20]. Participants were then prompted to choose the explanation they favored. The decision to only compare two explanations at a time is to minimize the cognitive load on participants.

Task 2: User Performance on Explanations. A particular method may appeal to participants due to various factors, such as visual intuitiveness or simplicity. However, this appeal might not guarantee effective conveyance of the underlying model's behavior. For an explanation to be considered effective, it should not only be favored by users but should also enhance their understanding of the model's workings. To assess the efficacy of the various explanation methods delineated in §5.4, we employed a binary choice paradigm. Below is a concise overview of each implementation:

- **Redacted Prompt Explanation.** Participants were presented the original image alongside two generated images, each missing distinct keywords. Their task was to associate each image with its corresponding redacted prompt.
- **Keyword Heat Map.** Participants were presented with two heatmaps, each associated with a different keyword. They were required to correctly match these heatmaps to their respective keywords, gauging their understanding of heatmap representations. The selected heatmaps for this task were pre-verified to ensure discernible differences, making the task solvable.
- **Keyword Linear Regression.** Participants were presented with a linear explanation for a prompt and a randomly generated linear explanation. They were tasked with identifying the correct linear explanation.
- **Keyword Image Gallery.** Participants were presented with two sets of image galleries, each related to a distinct keyword. They were required to pair each gallery with the correct keyword.

Additionally, after each task, participants were prompted to rate their confidence in their

predictions on a scale from 1 to 10. For tasks involving the use of two keywords, we employed CLIP embeddings to ensure the semantic impacts of the keywords were distinguishable.

Task 3: Combined Explanations. For each instance, participants were provided with two distinct sets of explanations, with each set comprising between one to four explanations. They were tasked with identifying and selecting their preferred set from the presented options. To maintain distinctiveness and clarity between the sets, no explanation appeared in both sets simultaneously, unless one set encompassed all four explanations. In such instances, the contrasting set faced no restrictions on its content.

5.6 Results

We performed several quantitative analyses on the data gathered from the AMT study. As previously highlighted, the primary focus of this study was to assess user preferences, evaluate performance, and determine the synergistic effectiveness of combined explanation methods. Additionally, we conducted analyses to examine the impact of different keyword types on user performance. In the subsequent sections, we detail these analyses and present key observations from the study. The summarized results can be found in Section § 5.6.3.

5.6.1 Preference vs Performance

We initially assess the correlation between participants' preferences and their actual performance in comprehending the explanations.

Participants prefer example-based explanation. The participants exhibited a preference for the Keyword Image Gallery (KIG) explanation method, selecting it 83.7% of the time when presented with choices. The results of pairwise preference percentages are presented in Table 5.1. Based on the Chi-Squared test with a Bonferroni correction, the KIG method is statistically preferred over the other methods. Based on the same analysis, Keyword Heat Map (KHM) performed better against KIG compared to other two methods.

	Chosen over RPE (%)	Chosen over KHM (%)	Chosen over KLR (%)	Chosen over KIG (%)	Average (%)
RPE	-	48.1	24.9	4.9	26.0
KHM	51.9	-	52.8	33.1	45.9
KLR	75.1	47.2	-	10.9	44.4
KIG	**95.1**	**66.9**	**89.1**	-	**83.7**

Table 5.1: Percentage preference of the method in the row over the method in the column when presented with a binary choice. Participants prefer KIG over other methods. The least favored method is RPE even though it is commonly used by experts.

This heightened performance is likely due to KHM's approach of elucidating the generation process using images, similar to KIG. Nonetheless, when assessed against the other two explanation method (RPE and KLR), KHM did not demonstrate a statistically significant preference.

Participants' least favored explanation is sensitivity-based. Interestingly, despite its frequent use by experts in our formative study, the Redacted Prompt Explanation (RPE) emerged as the least favored explanation method statistically. This disparity can be attributed to the difference in expertise levels; sensitivity-based explanations, like RPE, necessitate a certain degree of experience to be preferred.

Participants overestimate how well they understand the explanations. We conducted a paired-samples t-test with Bonferonni correction to compare participants' confidence scores with their actual performance on the second task. Across all explanation methods, participants consistently rated their confidence higher than their demonstrated performance levels. The average confidence and accuracy percentages can be found in Ta-

	Confidence (0-10)	Scaled Confidence (%)	Actual Performance (%)
RPE	4.7	73.3	65.2
KHM	7.8	89.0	65.9
KLR	5.1	75.5	59.9
KIG	8.6	93.1	78.8

Table 5.2: Average confidence and performance of different explanation methods. Scaled confidence is linearly scaled so that 0 confidence corresponds to 50%, mimicking random chance. Participants overestimate how well they understand the explanations.

ble 5.2. Importantly, a random guess would result in a 50% accuracy rate, and considering that confidence scores range from 0-10, we linearly scaled these scores for our analysis. In this scaling, a 50% accuracy corresponds to a confidence score of 0, as reflected in the table. We also conducted a repeated measures ANOVA test followed by a paired-samples t-test with Bonferroni correction, revealing that the Keyword Heat Map (KHM) exhibited the largest statistically significant disparity between participant confidence and actual performance. This discrepancy can likely be attributed to the inherent simple visualizations of KHM compared to KIG and RPE where the participants need to deduct the keyword effect by comparing images. Notably, this heightened confidence due to visual explanations was also statistically greater than that observed with the KLR (the only method that does not have a visual representation), as corroborated by a Chi-Squared Test with Bonferroni correction.

It is essential to highlight that comparing performances across different tasks is not appropriate due to inherent disparities in task difficulties. Consequently, results from the second task should be interpreted in isolation, without cross-comparison to other tasks.

For example, the KLR task necessitated participants to distinguish between the correct explanation and a decoy. Accomplishing this solely based on linear parameters is inherently challenging, which is reflected in participants' near-random success rate of 59%. However, KLR still provides important information to the participants by explaining the generation process through a linear model. For a more objective evaluation of the explanation methods, a comprehensive user study is necessary, focusing on how these explanations influence user objectives, such as determining which explanation methods enhance prompting effectiveness.

Participants' preference peaks with two explanations. In the third task, as the number of explanations increased, the most favored combination comprised two explanations. The preference percentages for one to four explanations were as follows: 35.4%, 66.1%, 54.1%, and 44.4%, respectively. Although a greater number of explanations inherently provide more information, it appears the cognitive burden imposed on participants inversely affected their preferences. Based on Chi-Squared test with Bonferroni correction, KIG with KLR is the most preferred combination with the preferred rate of 74.6%. This finding is intriguing given that the KHM explanation was the biggest rival to KIG during individual assessments in the first task. A plausible interpretation for this phenomenon is the overlap in insights offered by both KHM and KIG, as both are rooted in visual explanations. In contrast, KLR effectively complements KIG, enabling participants to visualize the influence of specific keywords (through examples by KIG) and comprehend their linear integration (demonstrated by KLR). This observation is further underscored by the preference rate of 28.1% for the RPE and KLR combination, which emerged as the least preferred combination statistically.

5.6.2 Keyword Types

To delve deeper into participants' performance concerning specific keywords, we examined all the prompts used throughout the study and segmented the keywords into two distinct categories. For the first category, the focus was on the impact scope of the keywords on images. Here, keywords causing localized alterations like 'cat', 'table', 'door', and 'lamp'

	Local (%)	Global (%)	Known (%)	Magic (%)	Known Confidence	Magic Confidence
RPE	88.2	61.3	67.0	62.4	4.7	3.6
KHM	83.6	60.8	69.0	63.1	8.3	6.7
KLR	65.6	52.9	58.3	60.2	6.8	4.1
KIG	84.6	75.5	82.0	77.3	9.4	7.6

Table 5.3: Performance distinction between local vs global keywords as well as known vs magic keywords for each explanation type.

were isolated. In contrast, those keywords governing overarching changes or dictating the broader theme of an image, including 'oil painting' and 'very detailed', were also identified. In the second category, keywords were differentiated based on their intuitiveness and their association with Stable Diffusion. While some keywords were deemed 'known', others were uniquely tied to Stable Diffusion, which we termed as 'magic' keywords. These 'magic' keywords, exemplified by phrases like 'trending on artstation' or specific references such as 'by Stanley Artgerm Lau and Alphonse Mucha', likely fall outside the user's existing knowledge base. Using these labels, we average the performance of the explanation methods across keywords.

Global keywords significantly influence participants' performance. Based on the results from a paired-samples t-test with Bonferroni correction, participants exhibited greater difficulty in comprehending global keywords compared to local ones. These results can be seen in Table 5.3. Keywords that influence the overarching theme or appearance of an image pose a greater challenge for participants than those affecting a specific segment of the image. This observation aligns with intuitive reasoning: subtle modifications across the

entirety of an image are more challenging to discern than distinct changes to a particular object or section.

Familiarity with keywords influences user confidence, not performance. Despite encountering unfamiliar keywords, participants were able to interpret them effectively, as shown in Table 5.3. According to a paired-samples t-test with Bonferroni correction, participants' performance with both familiar and "magic" keywords (those unique to Stable Diffusion) remained consistent. However, their confidence varied significantly, being notably lower when interpreting the "magic" keywords. This underscores that while participants can adapt to unfamiliar terms with the help of explanations, their assurance in their understanding diminishes.

5.6.3 Summary of Results

The study conducted a comprehensive analysis to evaluate user preferences, assess performance, and determine the combined effectiveness of different explanation methods, with a specific focus on the impact of different keyword types on user comprehension. Results showed that the Keyword Image Gallery (KIG) method was the most favored explanation technique, selected 83.7% of the time in binary choices, even though the Redacted Prompt Explanation (RPE), frequently used by experts, was the least preferred. A consistent pattern emerged where participants consistently rated their understanding of the explanations higher than their actual performance. Specifically, the Keyword Heat Map (KHM) method was observed to boost participant confidence more than other methods, though it didn't necessarily improve actual comprehension. Furthermore, when participants were exposed to multiple explanations, they showed a marked preference for combinations of two, with the KIG and KLR combination being the most favored. It is interesting to note that KLR, when standalone, does not convey substantial information as it simplifies the intricate generation process into a simple linear model. On the matter of keyword comprehension, global keywords, which influence the overarching theme of an image, posed a greater challenge

than local ones that only affect specific sections. Interestingly, while participants' performance remained consistent regardless of keyword familiarity, their confidence was notably diminished when dealing with unfamiliar, or "magic," keywords. This data underscores the importance of visual explanations and the need to manage user confidence in line with actual understanding.

5.7 Limitations and Future Work

Gap in explanation categories relative to traditional XAI. Our taxonomy lacks two categories present in Speith's function approach taxonomy [Spe22]: meta-explanation and architecture modification. Participants didn't present ideas aligning with these categories. Meta-explanation, which derives explanations by leveraging other explainability methods, is yet to be explored, given the current limited state of T2I explainability methods. Likewise, architecture modification, which aims to simplify models by altering their structure, remains unaddressed in the T2I context. However, the other categories were recurrent in participant discussions, indicating their intuitive nature.

Potential simplification in model-specific explanations. Our technique involved averaging attention maps across all attention heads, which might offer a simplified perspective of the actual image generation process. The diverse attention heads in Stable Diffusion focus on varied aspects of generation. Combining them might obscure information. As future text-to-image models could employ distinct architectures, it's essential to continually assess the relevance and limitations of model-specific explanations.

Evolving nature of Text-to-Image models. Many of the explanations proposed in this study hinge on keywords. As T2I models evolve, they might transition to a more conversational style or accept diverse input formats. Therefore, this study's insights should be interpreted with an emphasis on general end-user behavior rather than the specifics of existing explanation techniques. As T2I models undergo significant changes, reevaluation of

explanation techniques becomes essential.

Challenges in comparison of explanation methods. The task of comparing XAI methods remains formidable, and this area is still an active research area. Depending solely on participant feedback to determine the best explanation method is insufficient as shown in our work. Participants may not always possess the expertise or insight to judge explanations accurately. Therefore, it is crucial for future studies to define novel metrics or methodologies to comparatively analyze explanation techniques within the T2I framework.

Prompt iteration over time with explanation methods. Our study did not evaluate the potential improvement in participants' prompts over time. A crucial aspect of explanations in T2I models revolves around their capacity to guide users towards enhancing prompts, resulting in higher quality image outputs. Future work could concentrate on establishing metrics for 'improved prompts' and examining which explanations most effectively facilitate better image generation.

5.8 Conclusion

The development of text-to-image (T2I) models has expanded the ability to produce images from textual descriptions. However, for many users, especially those less familiar with the domain, the processes behind these models remain unclear. Our work aims to address this challenge by introducing and testing specific explanation methods designed to elucidate the workings of these models. Generating images from textual prompts requires precision and clarity in the prompts given. Through our initial study, we identified primary explanation goals specific to T2I models and subsequently developed explanation techniques aligned with these goals.

Our evaluation, conducted via Amazon Mechanical Turk, provided insights into how users interact with and perceive these explanations. Notably, the data revealed a preference among users for example-based explanations and indicated that users benefit most from a

limited set of explanation methods. Additionally, the study showed that certain keywords, which significantly change the theme of an image, are harder for users to understand.

In summary, this research contributes to the understanding of how users interact with explanations for T2I models. The dataset and findings presented provide a basis for further research in this area. As T2I models continue to evolve and find wider applications, making their operations transparent and understandable will be essential. This work is a step towards that objective.

CHAPTER 6

Summary

This dissertation has journeyed through the evolving landscape of generative models, highlighting the challenges and potentials of these technologies. The focus has been on developing and evaluating user-centric tools to make advanced AI more accessible and intuitive for a diverse range of users. This summary chapter synthesizes the key insights and contributions from each chapter, providing a cohesive overview of the entire work.

In Chapter 2, I introduced GANzilla, a tool designed to demystify Generative Adversarial Networks (GANs) for non-expert users. By implementing a scatter/gather approach, GANzilla allows users to intuitively explore and refine editing directions, enhancing their creative control over the image generation process. User studies demonstrated the tool's efficacy in enabling both specific and open-ended image editing tasks.

In Chapter 3, I introduced GANravel, a tool developed to aid users in disentangling editing directions within GANs. Addressing the 'black box' nature of these models, GANravel provided a user-driven approach for iterative refinement of image edits. The chapter detailed its successful application in user studies, including creative tasks like meme generation.

In Chapter 4, I introduced PromptZEN, a novel tool tailored for text-to-image diffusion models, particularly Stable Diffusion. PromptZEN empowers users to make precise, localized edits via prompt scheduling, offering an enhanced level of control and interaction. The chapter highlighted the tool's ability to facilitate user-guided image generation, substantiated by positive outcomes from user studies.

In Chapter 5, I tackled the challenge of making generative processes more transparent and understandable. It explored methods to enhance user comprehension of GANs and T2I systems, aiming to demystify the underlying mechanisms of these models. The chapter discussed the broader implications of these findings for future research and the ongoing development of generative models.

6.1 Limitations and Future Work

The sample size in some studies, though adequate for initial exploration, may not capture the full spectrum of user interactions and experiences. Additionally, the demographic representation in the studies might limit the generalizability of the findings to broader populations.

The field of AI, especially generative models, is rapidly advancing. Tools and methods developed in this dissertation might need continuous updating to stay relevant and effective in the face of new advancements. As new models and techniques emerge, some of the specific findings and tools discussed may become less applicable or require significant adaptation to align with future technologies.

The user-centric tools developed, while innovative, have inherent limitations in their features and capabilities. These limitations might impact their effectiveness in certain complex scenarios or for specific advanced user needs. Despite being designed for simplicity, there remains a learning curve associated with these tools. Users might require time and guidance to fully utilize their capabilities, which could be a barrier for some.

The research touches upon the democratization of AI but does not deeply explore the ethical implications, such as potential biases in model outputs or fairness in accessibility. Future research can focus on aligning generative models to eliminate biases. The tools and methods proposed may also have broader implications for employment and skill development in creative industries, which are not covered in this dissertation.

This dissertation contributes to the growing field of user-centric AI by offering practical

solutions and insights into enhancing user interaction with generative models. The tools developed and evaluated in this work represent a significant step towards democratizing advanced AI technologies, ensuring they are accessible and beneficial to a wide range of users. In summary, this dissertation underscores the importance of user-centered design in the realm of AI and generative models. It opens up new avenues for future research and development, aiming for a future where advanced AI is not just a tool for the few but an empowering technology accessible to all.

REFERENCES

[AHG23] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. "Spatext: Spatio-textual representation for controllable image generation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18370–18380, 2023.

[ALF22] Omri Avrahami, Dani Lischinski, and Ohad Fried. "Blended diffusion for text-driven editing of natural images." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18208–18218, 2022.

[AZM21] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. "Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows." *ACM Transactions on Graphics (ToG)*, **40**(3):1–21, 2021.

[BHE23] Tim Brooks, Aleksander Holynski, and Alexei A Efros. "Instructpix2pix: Learning to follow image editing instructions." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023.

[BLG20] Zana Buçinca, Phoebe Lin, Krzysztof Z Gajos, and Elena L Glassman. "Proxy tasks and subjective measures can be misleading in evaluating explainable ai systems." In *Proceedings of the 25th international conference on intelligent user interfaces*, pp. 454–464, 2020.

[BMG21] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z Gajos. "To trust or to think: cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making." *Proceedings of the ACM on Human-Computer Interaction*, **5**(CSCW1):1–21, 2021.

[Bro20] Jason Brownlee. "How to explore the gan latent space when generating faces.", Sep 2020.

[BZS18] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. "Gan dissection: Visualizing and understanding generative adversarial networks." *arXiv preprint arXiv:1811.10597*, 2018.

[CAV23] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. "Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models." *ACM Transactions on Graphics (TOG)*, **42**(4):1–10, 2023.

[CBP20] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. "Editing in Style: Uncovering the Local Semantics of GANs." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[CGL20] Yu Cheng, Zhe Gan, Yitong Li, Jingjing Liu, and Jianfeng Gao. *Sequential Attention GAN for Interactive Image Editing*, p. 4383–4391. Association for Computing Machinery, New York, NY, USA, 2020.

[Che23] Yixiong Chen. "X-IQE: eXplainable Image Quality Evaluation for Text-to-Image Generation with Visual Large Language Models." *arXiv preprint arXiv:2305.10843*, 2023.

[CKL20] Chia-Hsing Chiu, Yuki Koyama, Yu-Chi Lai, Takeo Igarashi, and Yonghao Yue. "Human-in-the-loop differential subspace search in high-dimensional latent space." *ACM Transactions on Graphics (TOG)*, **39**(4):85–1, 2020.

[CSG20] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. "DeepFaceDrawing: Deep generation of face images from sketches." *ACM Transactions on Graphics (TOG)*, **39**(4):72–1, 2020.

[CTW18] Xiang 'Anthony' Chen, Ye Tao, Guanyun Wang, Runchang Kang, Tovi Grossman, Stelian Coros, and Scott E Hudson. "Forte: User-Driven Generative Design." In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 496. ACM, 2018.

[Cuk13] Will Cukierski. "Dogs vs. Cats.", 2013.

[CUY20] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. "StarGAN v2: Diverse Image Synthesis for Multiple Domains." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[CVS22] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. "Diffedit: Diffusion-based semantic image editing with mask guidance." *arXiv preprint arXiv:2210.11427*, 2022.

[DBK20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929*, 2020.

[DMB22] Hai Dang, Lukas Mecke, and Daniel Buschek. "GANSlider: How Users Control Generative Models for Images using Multiple Sliders with and without Feedforward Information." *arXiv preprint arXiv:2202.00965*, 2022.

[EC22] Noyan Evirgen and Xiang'Anthony' Chen. "GANzilla: User-Driven Direction Discovery in Generative Adversarial Networks." *arXiv preprint arXiv:2207.08320*, 2022.

[EC23] Noyan Evirgen and Xiang'Anthony Chen. "GANravel: User-Driven Direction Disentanglement in Generative Adversarial Networks." In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2023.

[FW23] Raymond Fok and Daniel S Weld. "In Search of Verifiability: Explanations Rarely Enable Complementary Performance in AI-Advised Decision Making." *arXiv preprint arXiv:2305.07722*, 2023.

[GAA22] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. "An image is worth one word: Personalizing text-to-image generation using textual inversion." *arXiv preprint arXiv:2208.01618*, 2022.

[GAO19] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. "GANalyze: Toward Visual Definitions of Cognitive Image Properties." In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[GHH21] Hongyan Gu, Jingbin Huang, Lauren Hung, and Xiang 'Anthony' Chen. "Lessons Learned from Designing an AI-Enabled Diagnosis Tool for Pathologists." *Proc. ACM Hum.-Comput. Interact.*, **5**(CSCW1), apr 2021.

[GLK23] Hyojun Go, Yunsung Lee, Jin-Young Kim, Seunghyun Lee, Myeongho Jeong, Hyun Seung Lee, and Seungtaek Choi. "Towards practical plug-and-play diffusion models." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1962–1971, 2023.

[GLX22] Hongyan Gu, Yuan Liang, Yifan Xu, Christopher Kazu Williams, Shino Magaki, Negar Khanlou, Harry Vinters, Zesheng Chen, Shuo Ni, Chunxu Yang, Wenzhong Yan, Xinhai Robert Zhang, Yang Li, Mohammad Haeri, and Xiang 'Anthony' Chen. "Improving Workflow Integration with XPath: Design and Evaluation of a Human-AI Diagnosis System in Pathology." *ACM Trans. Comput.-Hum. Interact.*, dec 2022. Just Accepted.

[GPA22] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. "Make-a-scene: Scene-based text-to-image generation with human priors." In *European Conference on Computer Vision*, pp. 89–106. Springer, 2022.

[GPM20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks." *Communications of the ACM*, **63**(11):139–144, 2020.

[GYR23] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. "Animatediff: Animate your personalized text-to-image diffusion models without specific tuning." *arXiv preprint arXiv:2307.04725*, 2023.

[Har86] Sandra G Hart. "NASA task load index (TLX)." 1986.

[HB97] Karen Holtzblatt and Hugh Beyer. *Contextual design: defining customer-centered systems*. Elsevier, 1997.

[HCD22] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. "Optimizing prompts for text-to-image generation." *arXiv preprint arXiv:2212.09611*, 2022.

[Hei13] Fritz Heider. *The psychology of interpersonal relations*. Psychology Press, 2013.

[Hei19] Eric Heim. "Constrained generative adversarial networks for interactive image generation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10753–10761, 2019.

[HFZ18] Keke He, Yanwei Fu, Wuhao Zhang, Chengjie Wang, Yu-Gang Jiang, Feiyue Huang, and Xiangyang Xue. "Harnessing Synthesized Abstraction Images to Improve Facial Attribute Recognition." In *IJCAI*, pp. 733–740, 2018.

[HHL20] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. "Ganspace: Discovering interpretable gan controls." *Advances in Neural Information Processing Systems*, **33**:9841–9850, 2020.

[HMT22] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. "Prompt-to-prompt image editing with cross attention control." *arXiv preprint arXiv:2208.01626*, 2022.

[HZ21] Drew A Hudson and Larry Zitnick. "Generative adversarial transformers." In *International conference on machine learning*, pp. 4487–4499. PMLR, 2021.

[JCI19] Ali Jahanian, Lucy Chai, and Phillip Isola. "On the" steerability" of generative adversarial networks." *arXiv preprint arXiv:1907.07171*, 2019.

[JNC20] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. "How can i explain this to you? an empirical study of deep neural network explanation methods." *Advances in Neural Information Processing Systems*, **33**:4211–4222, 2020.

[JPJ21] Maia Jacobs, Melanie F. Pradier, Thomas H. McCoy Jr, Roy H. Perlis, Finale Doshi-Velez, and Krzysztof Z. Gajos. "How machine-learning recommendations influence clinician treatment selections: the example of antidepressant selection." *Translational Psychiatry*, **11**, 2021.

[KGC17] Rubaiat Habib Kazi, Tovi Grossman, Hyunmin Cheong, Ali Hashemi, and George Fitzmaurice. "DreamSketch: Early Stage 3D Design Explorations with Sketching and Generative Design." In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, p. 401–414, New York, NY, USA, 2017. Association for Computing Machinery.

[KGM22] Siavash Khodadadeh, Shabnam Ghadar, Saeid Motiian, Wei-An Lin, Ladislau Bölöni, and Ratheesh Kalarot. "Latent to Latent: A Learned Mapper for Identity Preserving Editing of Multiple Face Attributes in StyleGAN-generated Images." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3184–3192, 2022.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[KLA20a] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and Improving the Image Quality of StyleGAN." In *Proc. CVPR*, 2020.

[KLA20b] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and improving the image quality of stylegan." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020.

[KPJ23] Hyung-Kwon Ko, Gwanmo Park, Hyeon Jeon, Jaemin Jo, Juho Kim, and Jinwook Seo. "Large-scale text-to-image generation models for visual artists' creative works." In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pp. 919–933, 2023.

[LC22] Vivian Liu and Lydia B Chilton. "Design Guidelines for Prompt Engineering Text-to-Image Generative Models." In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[Lew82] Clayton Lewis. *Using the" thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY, 1982.

[LKL21] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. "EditGAN: High-Precision Semantic Image Editing." *Advances in Neural Information Processing Systems*, **34**, 2021.

[LLW23] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. "Gligen: Open-set grounded text-to-image generation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023.

[Lom06] Tania Lombrozo. "The structure and function of explanations." *Trends in cognitive sciences*, **10**(10):464–470, 2006.

[LPL20] Jiadong Liang, Wenjie Pei, and Feng Lu. "Cpgan: Content-parsing generative adversarial networks for text-to-image synthesis." In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pp. 491–508. Springer, 2020.

[LYZ22] Jun Hao Liew, Hanshu Yan, Daquan Zhou, and Jiashi Feng. "Magicmix: Semantic mixing with diffusion models." *arXiv preprint arXiv:2210.16056*, 2022.

[LZS20] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. "Towards faster and stabilized gan training for high-fidelity few-shot image synthesis." In *International Conference on Learning Representations*, 2020.

[MGB18] Justin Matejka, Michael Glueck, Erin Bradner, Ali Hashemi, Tovi Grossman, and George Fitzmaurice. *Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets*, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2018.

[MH21] Deborah Mateja and Armin Heinzl. "Towards Machine Learning as an Enabler of Computational Creativity." *IEEE Transactions on Artificial Intelligence*, **2**(6):460–475, 2021.

[MHA23] 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*, pp. 6038–6047, 2023.

[MHS21] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. "Sdedit: Guided image synthesis and editing with stochastic differential equations." *arXiv preprint arXiv:2108.01073*, 2021.

[MWX23] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. "T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models." *arXiv preprint arXiv:2302.08453*, 2023.

[NJ17] Ingrid Nunes and Dietmar Jannach. "A systematic review and taxonomy of explanations in decision support and recommender systems." *User Modeling and User-Adapted Interaction*, **27**:393–444, 2017.

[OKB23] Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. "Editing implicit assumptions in text-to-image diffusion models." *arXiv preprint arXiv:2303.08084*, 2023.

[Opp22] Jonas Oppenlaender. "The creativity of text-to-image generation." In *Proceedings of the 25th International Academic Mindtrek Conference*, pp. 192–202, 2022.

[OSW22] Atila Orhon, Michael Siracusa, and Aseem Wadhwa. "Stable Diffusion with Core ML on Apple Silicon.", 2022.

[PBH20] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. "Controlling generative models with continuous factors of variations." *arXiv preprint arXiv:2001.10238*, 2020.

[PBS17] Santiago Pascual, Antonio Bonafonte, and Joan Serra. "SEGAN: Speech enhancement generative adversarial network." *arXiv preprint arXiv:1703.09452*, 2017.

[PGA23] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. "Localizing Object-level Shape Variations with Text-to-Image Diffusion Models." *arXiv preprint arXiv:2303.11306*, 2023.

[PKZ23] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. "Zero-shot image-to-image translation." In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023.

[PSH96] Peter Pirolli, Patricia Schank, Marti Hearst, and Christine Diehl. "Scatter/gather browsing communicates the topic structure of a very large text collection." In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 213–220, 1996.

[PTL23] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. "Drag your gan: Interactive point-based manipulation on the generative image manifold." In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023.

[PVZ12] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. "Cats and dogs." In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

[PVZ15] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. "Deep face recognition." 2015.

[PWS21] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. "StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery." In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2085–2094, October 2021.

[RBL22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[RKH21] 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*, pp. 8748–8763. PMLR, 2021.

[RLJ23] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434*, 2015.

[RPG21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. "Zero-shot text-to-image generation." In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

[SCS22] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. "Photorealistic text-to-image diffusion models with deep language understanding." *Advances in Neural Information Processing Systems*, **35**:36479–36494, 2022.

[SGT20] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. "Interpreting the Latent Space of GANs for Semantic Face Editing." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[SO21] Sefik Ilkin Serengil and Alper Ozpinar. "HyperExtended LightFace: A Facial Attribute Analysis Framework." In *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pp. 1–4. IEEE, 2021.

[Spe22] Timo Speith. "A review of taxonomies of explainable artificial intelligence (XAI) methods." In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2239–2250, 2022.

[Spi05] Clay Spinuzzi. "The methodology of participatory design." *Technical communication*, **52**(2):163–174, 2005.

[SYP19] Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks." *Scientific reports*, **9**(1):1–9, 2019.

[SYT20] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. "Interfacegan: Interpreting the disentangled face representation learned by gans." *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[SZ21] Yujun Shen and Bolei Zhou. "Closed-Form Factorization of Latent Semantics in GANs." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1532–1540, June 2021.

[TPJ22] Raphael Tang, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Jimmy Lin, and Ferhan Ture. "What the daam: Interpreting stable diffusion using cross attention." *arXiv preprint arXiv:2210.04885*, 2022.

[TTP21] Christos Tzelepis, Georgios Tzimiropoulos, and Ioannis Patras. "WarpedGANSpace: Finding non-linear RBF paths in GAN latent space." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6393–6402, 2021.

[VB20] Andrey Voynov and Artem Babenko. "Unsupervised Discovery of Interpretable Directions in the GAN Latent Space." In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9786–9796. PMLR, 13–18 Jul 2020.

[VPT16] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. "Generating videos with scene dynamics." *Advances in neural information processing systems*, **29**, 2016.

[VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *Advances in neural information processing systems*, **30**, 2017.

[VT23] Henriikka Vartiainen and Matti Tedre. "Using artificial intelligence in craft education: crafting with text-to-image generative models." *Digital Creativity*, **34**(1):1–21, 2023.

[WLS21a] Zongze Wu, Dani Lischinski, and Eli Shechtman. "StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12863–12872, June 2021.

[WLS21b] Zongze Wu, Dani Lischinski, and Eli Shechtman. "Stylespace analysis: Disentangled controls for stylegan image generation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12863–12872, 2021.

[WMM22] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. "Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models." *arXiv preprint arXiv:2210.14896*, 2022.

[WSZ23] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. "Better aligning text-to-image models with human preference." *arXiv preprint arXiv:2303.14420*, 2023.

[WWA22] Zifeng Wang, Zhenbang Wu, Dinesh Agarwal, and Jimeng Sun. "Medclip: Contrastive learning from unpaired medical images and text." *arXiv preprint arXiv:2210.10163*, 2022.

[WYA19] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. "Designing theory-driven user-centric explainable AI." In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–15, 2019.

[WYW18] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. "Esrgan: Enhanced super-resolution generative adversarial networks." In *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.

[XLW23] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. "Imagereward: Learning and evaluating human preferences for text-to-image generation." *arXiv preprint arXiv:2304.05977*, 2023.

[YJW22] Kun Yan, Lei Ji, Chenfei Wu, Jianmin Bao, Ming Zhou, Nan Duan, and Shuai Ma. "Trace controlled text to image generation." In *European Conference on Computer Vision*, pp. 59–75. Springer, 2022.

[YSZ21] Ceyuan Yang, Yujun Shen, and Bolei Zhou. "Semantic hierarchy emerges in deep generative representations for scene synthesis." *International Journal of Computer Vision*, **129**(5):1451–1466, 2021.

[YWB19] Xin Yi, Ekta Walia, and Paul Babyn. "Generative adversarial network in medical imaging: A review." *Medical image analysis*, **58**:101552, 2019.

[ZB21] Enhao Zhang and Nikola Banovic. "Method for Exploring Generative Adversarial Networks (GANs) via Automatically Generated Image Galleries." In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[ZKS16] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. "Generative visual manipulation on the natural image manifold." In *European conference on computer vision*, pp. 597–613. Springer, 2016.

[ZPI17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks." In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[ZRA23] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. "Adding conditional control to text-to-image diffusion models." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.