PromptPaint: Steering Text-to-Image Generation Through Paint Medium-like Interactions

John Joon Young Chung*
SpaceCraft Inc.
Los Angeles, USA
jjyc@spacecraft.inc

Eytan Adar
University of Michigan
Ann Arbor, USA
eadar@umich.edu

ABSTRACT

While diffusion-based text-to-image (T2I) models provide a simple and powerful way to generate images, guiding this generation remains a challenge. For concepts that are difficult to describe through language, users may struggle to create prompts. Moreover, many of these models are built as end-to-end systems, lacking support for iterative shaping of the image. In response, we introduce PromptPaint, which combines T2I generation with interactions that model how we use colored paints. PromptPaint allows users to go beyond language to mix prompts that express challenging concepts. Just as we iteratively tune colors through layered placements of paint on a physical canvas, PromptPaint similarly allows users to apply different prompts to different canvas areas and times of the generative process. Through a set of studies, we characterize different approaches for mixing prompts, design trade-offs, and socio-technical challenges for generative models. With PromptPaint we provide insight into future steerable generative tools.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; • **Applied computing** → **Arts and humanities**; • **Computing methodologies** → **Computer vision**.

KEYWORDS

generative model, text-to-image generation, painting interactions

ACM Reference Format:
John Joon Young Chung and Eytan Adar. 2023. PromptPaint: Steering Text-to-Image Generation Through Paint Medium-like Interactions. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23), October 29-November 1, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3586183.3606777

1 INTRODUCTION

New diffusion-based techniques [58, 66, 68, 70] are enabling a wide array of text-to-image (T2I) models. Prompt-driven image creation allows even those without drawing or painting skills to produce high-quality images. Unfortunately, simple text prompts are not always useful for getting what the user imagines in their mind. While the proliferation of cutting-edge tools and demos make new features available (e.g., Midjourney [57], Dream Studio [80], Gradio demos [33], demos in Google Colab Notebooks [19]), guiding them is still challenging.

Artists often create images in a step-by-step procedure: fixing, refining, and improving their ideas as they go. People usually follow specific workflows to produce visual arts, with intermediate decisions between steps [26, 28, 91]. A key problem with generative models is that they work largely in an end-to-end fashion: a prompt goes in and an image comes out, with little chance to intervene in between. For example, in digital comics, artists create the piece in multiple steps: sketching, flatting, shadowing, drawing backgrounds, and adding special effects [91]. Each step allows for refinement and control. AI systems often hide these intermediate steps. Similarly, images generated only with the user's initial prompts would limit what the user can do during artifact production. A second problem is that natural language prompts are not expressive enough for all intents. Just as we would be challenged to describe the art we see, users may find it impossible to describe the art they imagine. This is particularly hard when concepts are ambiguous or don't yet exist (e.g., a style with elements of both Impressionism and Arte Nouveau). The user might not have sufficient natural language descriptions for what they want. Such natural language prompts also lack the ability to specifically control parameters (e.g., how do I get an image with a 'flatness' of 60%?).

To address these challenges, new technical approaches have emerged to enable the gradual editing of visual content. For example, we now see methods to 'in-paint' and 'out-paint,' adding or revising visual elements on the existing image [7, 58, 69]. Users can now also give an initial image to build up on the generation [7] or the visual structure [93]. Researchers have also investigated technical approaches to mix prompts [50, 52] or edit images based on natural language prompts [20, 34, 43, 84]. While the underlying algorithms can help end-users control the images they produce, there is very little consideration for how interactions should be modeled to support the creation experience.

In this work, we explore how users can interact with T2I models to enable the gradual building of artifacts while allowing flexible exploration in the 'art space.' To facilitate the steering of T2I models, we take inspiration from how artists interact with paint mediums (e.g., oil paint or watercolor). The main characteristic of the paint medium we leverage is the flexibility in the use and combination of colors. While we start painting with discrete colors in color tubes, we do not limit ourselves to those tubes but explore colors beyond them by mixing them on the palette. Moreover, we apply them on the canvas flexibly, either by overlaying different colors with each other or by using different colors on different canvas areas. With

*Work done at the University of Michigan.

UIST '23, October 29-November 1, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0132-0/23/10…$15.00
https://doi.org/10.1145/3586183.3606777

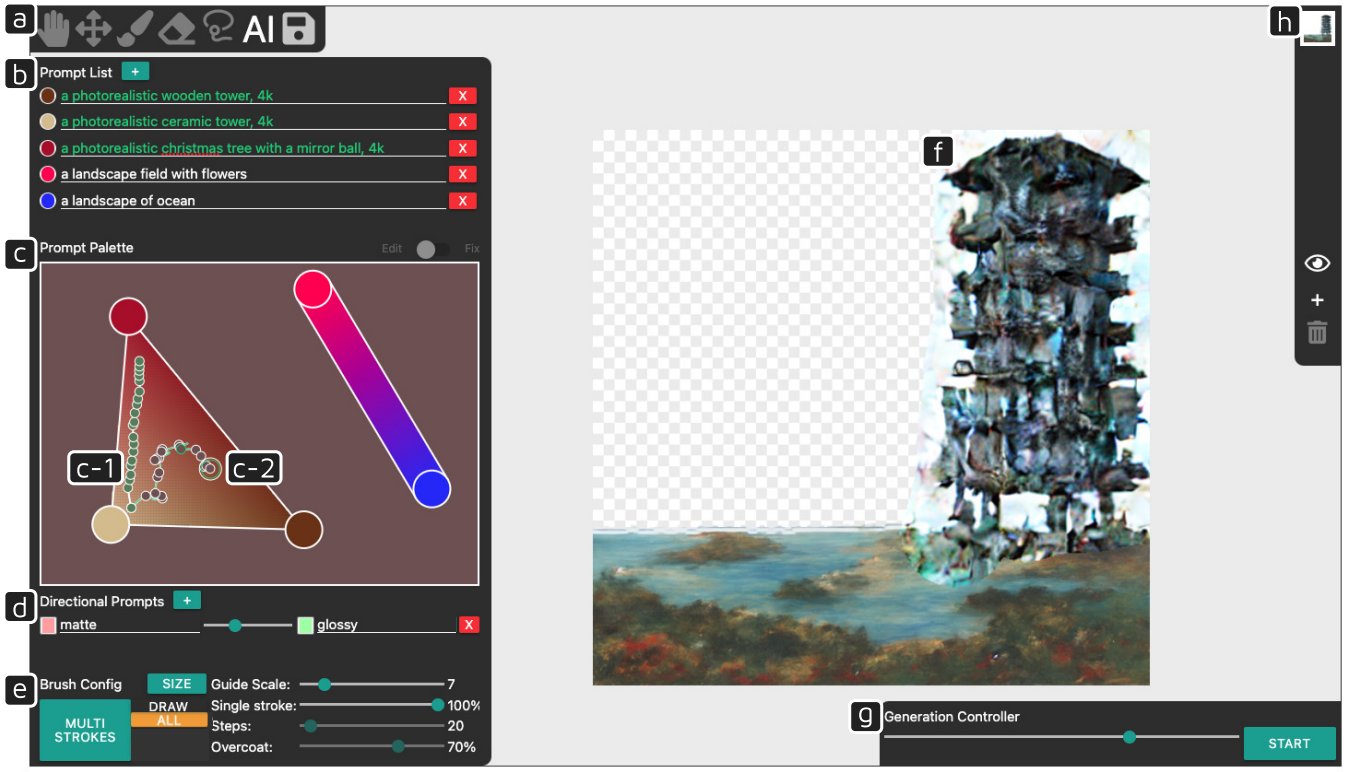

Figure 1: PromptPaint allows flexible steering of diffusion-based text-to-image generation by combining prompt-based generation with paint medium-like interactions (e.g., oil painting, watercolor). After the user defines their prompts of interest (b), the tool allows users to mix prompts as they would paint by: 1) interpolating them in the prompt palette (c, *prompt mixing*); or 2) adding attributes (d, *directional prompt*). PromptPaint also allows for gradual shaping of the artifact by allowing 1) changing of prompts during generation (c-1, g, *prompt intervention*) and 2) spatial selection of generation area (f, *prompt stencil*). Users can make detailed generation specifications with configuration widgets (e) along with other image editing functions such as moving image content, brushing, erasing, lassoing, and layer edits (a, h). The circle in the palette (c-2) indicates the user's selection of the mixed prompt on the palette.

PromptPaint we were inspired by this idea and implemented the system to allow users to interact with prompts as they would with colors. PromptPaint turns prompts into flexible materials that can even target verbally indescribable concepts with *prompt mixing* and *directional prompt*. PromptPaint modularizes image generation by allowing users to apply varying prompts to different parts of the canvas (*prompt stencil*) and different parts of the generation process (*prompt intervention*).

With PromptPaint, we characterized different approaches in their effectiveness for adding new attributes to an existing image. We found that different strategies have different strengths—prompt mixing and directional prompt were effective in adding a new attribute, and prompt intervention and prompt stencil tend to transform the image while maintaining the visual similarity to the original image. We also conducted a user study to identify how users interact with PromptPaint. From the user study, we found that different ways to steer T2I generation could allow users to generate images that align well with their intentions through iterations. However, we also identified design trade-offs between 1) focused iteration and curation and 2) manual editing and automation. Furthermore, the high complexity and randomness of AI models could result in a misalignment between AI behaviors and user expectations. Lastly, while users had some sense of ownership of the resulting artifacts, their expertise and alignment of the produced artifact with expectations can impact that sense of ownership. From the findings, we discuss insights into adopting paint-medium interactions in designing future versions of generative tools.

2 BACKGROUND AND RELATED WORK

2.1 Painting From Physical to Digital

The act of rendering an image by applying paint to canvas is an important form of creative expression [28]. "Personal causation" [24], or the change in the world by an individual, is an intrinsic satisfaction of painting. Painting enables unique ways of expressing ideas and emotions than other mediums, such as poetry or music. Critically, painting often involves continuous judgments during the process [28], which can be routinized in a workflow for a specific artifact type [26, 91]. These characteristics hint at the limitations of existing T2I models. First, not all visual ideas can be described

with text. Ideally, the way we craft an image should be closer to the medium itself. Second, the gradual judgments and iterations inherent in the painting are difficult with T2I models. Third, removing the physical act of painting, as T2I models do, reduces the feeling of "personal causation." With PromptPaint, our goal is to address these concerns using painting interactions. The combination of painting interactions with generative approaches supports the balance of direct manipulation with intelligent interfaces [74].

Researchers have designed many tools for painting and drawing. Some tools guided novice users without directly intervening in user drawings [39, 87]. Others augment by adding corrections to the drawn results [30, 51, 82, 90]. There are tools to target specific sub-problems in painting (e.g., flexible exploration of colors with color mixing interactions [75, 76]). Instead of supporting "existing drawing/painting practices," some systems enabled users to generate novel types of artifacts with computationally enhanced brushes [9, 40, 72]. With AI, systems can now support co-creation, where humans and machines take turns in drawing [23, 59]. PromptPaint builds upon these approaches by bringing diffusion-based T2I models closer to interactions with paint mediums.

2.2 AI Image Generation

There have been many approaches to generating images (beyond T2I diffusion models) with neural networks ranging from style transfer algorithms [27, 31, 73] to generative adversarial networks (GAN) [16, 32]. The most recent approaches include diffusion models that learn to recover images from noisy images [35]. These generate higher-quality images compared to other approaches, and researchers have devised ways to guide their generation with specific classes [25, 36].

In parallel to these techniques, new models include trained representations that combine text and images. CLIP [64], for example, enables natural language guidance for image generation [21, 48]. Diffusion-based T2I models are some of the most popular due to their flexibility, ability to follow input prompts, and high-quality output [58, 66, 70]. However, these models are largely end-to-end in their approach (prompt in, image out). Therefore, imbuing more human intention into the generated results can be challenging. Various approaches have tried to tackle this problem, from seeding an initial image to be transformed [7, 56] to combining two different prompts to realize them in the image [52], generating an image of one prompt while having the overall form of another prompt [50], editing or expanding images with visual masking [7, 58, 69], editing images with prompts [10, 20, 34, 43, 62, 84], giving visual structures [37, 49, 93], and automatically refining prompts [86]. Although these introduced technical approaches to gradually and iteratively shape images, they are largely unconcerned with the interaction model. We address this limitation by combining diffusion approaches with novel interaction techniques inspired by physical acts of painting.

2.3 Interaction with AI Generation

There are numerous approaches to *steer* AI generations. Controls vary from category selection [16, 47] (e.g., happy vs. sad face) to sliders on a fixed continuous semantic scale [15, 22, 55, 61] (e.g., melody on a positive-negative scale). More flexible control of continuous scales includes explorable galleries [92], user-definable sliders [17], or visual sketches [18]. Although more flexible, these approaches limit options to a somewhat constrained set. An alternative to widget-based controls is using examples as inputs [60, 73] (e.g., generating an image similar to the example). Although technically flexible in receiving "any examples," steering these models can be challenging as searching for another desirable example can be difficult. With advances in language models [11] and contrastive learning between text and other mediums [2, 64, 88], natural language prompts are used for model steering. Prompts can steer generation of texts [11, 81, 89], images [52, 54, 66, 70], UI designs [45], codes [13], 3D models [41, 63], music [2], and even videos [77, 85]. Prompting has comparative advantages over other approaches, as it does not limit the input the user can make. That is, with the obvious exception that they need to be able to say it. Textual prompts can also be challenging due to: 1) the wide variety of ways to describe something; and 2) the difficulty in describing some concepts due to ambiguity or vagueness (e.g., "a bit less vivid color"). Mixing prompts can help overcome these challenges by providing the grounding of a set of 'base' textual prompts but with the ability to select the vague semantic spaces between the prompts. Some interfaces explored this approach by showing multiple results from prompts with different mixing weights [5]. PromptPaint adopt paint-medium-like interactions to allow users to visually explore and iterate mixed prompts.

Finally, we should consider how the generation processes and results are embedded into the human art creation process. Generative models should be able to provide intermediate representations [91], which align with the user workflow and allow easy edits and iterations. However, generative models are often designed to produce high-fidelity, final artifacts. With GAN models, Endo explored one approach to enable edits, by allowing iteration on high-fidelity image generation with the user's direct manipulation input [29]. In diffusion-based T2I model contexts, researchers and practitioners investigated ways to repurpose generation results as a *modularized* unit in the human creation process. As introduced in Section 2.2, editing with seed images, masking, or prompts would be specific examples. However, not many approaches have looked at how to allow users to intervene during generation. We investigate both approaches to 1) repurposing generated *results* as a modularized unit for human artistic creation and 2) allowing user interaction during the generation *process*.

3 INTERACTING WITH GENERATIVE MODELS LIKE PAINT MEDIUM

While the goals of generating artifacts might not directly correspond to those of manual painting, we propose that analogies from painting interactions can facilitate the design of steering interactions for generative models (Figures 2 and 4). We focus on two different aspects. The first is going beyond discrete semantics (e.g., categories, prompts) for specifying generation and flexibly exploring semantics in the vector space. With the paint-medium analogy, we connect this to color-mixing interactions. The second is allowing the gradual generation of artifacts, similar to how we gradually apply colors when we paint. In the following, we detail the connections between the steering of generative models and paint-medium interactions.

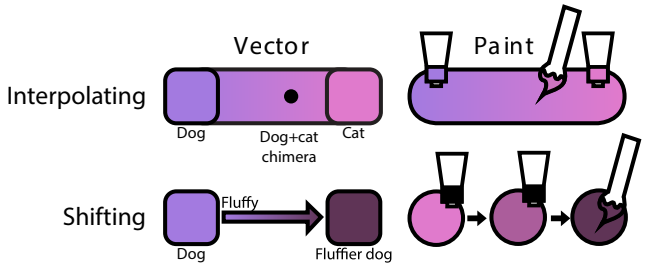

Figure 2: Mapping vector space exploration to paint color mixing. Discrete semantics (e.g., categories, prompts, or examples) are represented as a rounded square in Vector. They can map to discrete color tubes in Paint. Using the analogy, the user can explore semantics between discrete ones in a way similar to how they would explore colors by mixing.

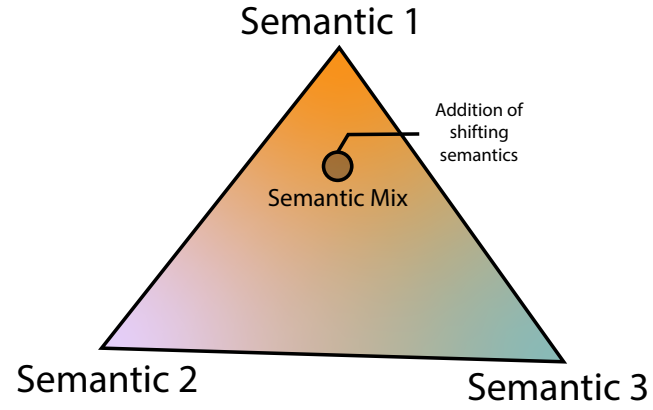

Figure 3: Using palette interaction for semantic mix has the benefit in that the interpolation and the shifting can be represented in the same interface.

3.1 Mixing Colors: Exploring Vector Spaces

Discrete input modalities of categories, examples, and prompts specify the semantics of generation while being easily comprehended by users. Generative models, such as language models [18, 47], style transfer algorithms [73], GANs [3, 42], or diffusion models [66, 70], first transform these inputs into vector representations. As a vector is a continuous representation, describing representations *between* discrete semantics would be difficult with only discrete interfaces (e.g., varying degress of 'chimeras' of *cat* and a *dog*). However, there are cases where users want to work with such semantics. For example, in some situations that are difficult to verbally describe, users might want to use vector representation spaces. Such a need would also arise when the user wants to do fine-grained control of an attribute, like adjusting the roughness of image textures or the fluffiness of a dog. Moreover, exploration of intermediate semantics would facilitate realizing eclecticism, where the artist tries to mix different styles together [38]. Previous work has shown that such manipulation is doable by interpolating discrete semantics [18, 44, 65] or shifting semantics with directional vectors about concepts [61, 71]. However, these approaches have not generally offered ways of turning vector manipulations into accessible interactions, specifically when the user can flexibly specify different discrete semantics (e.g., prompts).

To explore vector representations, we introduce the idea of interacting with discrete semantics in a way similar to how we *mix* physical paint colors. When paints are used, they come into our hands in color tubes, each having one discrete color. However, when applying them to the canvas, we do not limit ourselves to those discrete choices. Rather, we create new colors by mixing on a palette. Using this idea, we introduce the interaction of mixing different discrete semantics in a *semantic palette*. Analogically, each discrete semantic of categories, examples, or prompts would map to a discrete color for the paint mediums (Figure 2).

The first specific approach to mixing discrete semantics is to *interpolate* them, by mixing two or more discrete semantics on the semantic palette and exploring the space between them (top row of Figure 2). For example, to render an image of a chimera of a cat and a dog with T2I models, the user would interpolate the semantics of a cat and a dog. This would be similar to spreading colors to mix them and using intermediate gradients of those colors.

The second approach is *shifting*—adding a directional semantic for fine-grained control (bottom row of Figure 2). With a paint medium, this would be like adding a small amount of a different color to change the characteristics of the used color (e.g., making the green color darker by adding a bit of black pigment). In our case, the user can render an image of a dog with a certain level of fluffiness by adding the semantics (analogically, "pigments") of fluffiness to the semantics of a dog. Note that these interactions could be adapted to those generative techniques that can turn discrete 'user-facing' concepts into the vector space and then perform generation with the vector representations.

Naturally, there can be other interactions to mix discrete semantics. For example, we can mix prompts with sliders, each representing the weight of each prompt. Compared to such interactions, palette interactions can represent the mix of semantics with two visual signals: positions and colors on the palette. With palettes, specifically, both interpolation and shifting can be shown in a single interface. As in Figure 3, the palette interface can represent the interpolation with a point in the mixed-color gradient and show the shifting by adding the color to the selected interpolated point. On the other hand, conventional sliders may become more complex as the weights for interpolation and shifting would need to be represented in separate sliders.

3.2 Colors onto Canvas: Gradual Generation

Generation models do not often allow user interventions during the generative process. Thus, the experience of using generative models can be far from "creation," where the painter gradually shapes the artifact, making decisions as they go. Instead, we suggest that interventions could be applied by the end-user *during* the generative process. Again, we take the analogy of painting, focusing on how we apply paint to the canvas. When we apply colors to the canvas, we do not use the same paint for the whole area. Instead, we gradually build the artifact with multiple paint strokes and overlapping layers.

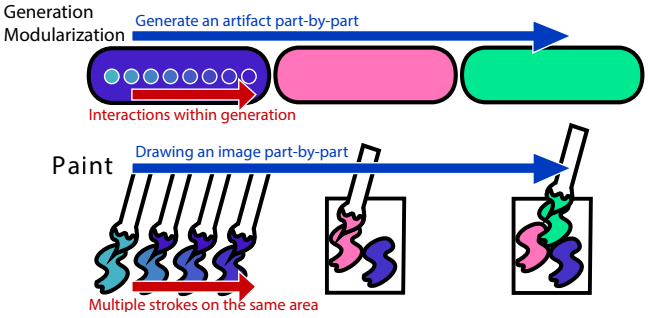

Figure 4: Mapping generation modularization to gradual painting of an artifact. Within-generation interventions would correspond to multiple strokes applied on the same canvas area, and generating the artifact part-by-part would map to drawing the image part-by-part.

We propose that modularizing the generation model temporally and spatially would allow for interactive changes to steer the model.

We consider two forms of modularization. The first allows interactions (temporally) within the generation process (red arrows in Figure 4). One example interaction can be changing the guiding prompt *during* generation. In our paint metaphor, this would be similar to overlaying different paint strokes on the same area to decide the final rendition. Not all models support this kind of intervention, though the diffusion-based T2I model does. As we will demonstrate, in diffusion-based T2I models, prompts that are used in the earlier stage can decide the overall form of the image while those in the later part decide the details. For example, brushing with a "banana on the ground" prompt-as-color first and then switching to "a futuristic car" color would result in a futuristic car in the shape of a banana.

The second form of modularization is the spatially partial generation of content (blue arrows in Figure 4). Analogically, this would be equivalent to how people draw an image part by part. Again, not all models are capable of this kind of focused generation. However, in-painting and out-painting in diffusion-based T2I model can support this functionality [7, 58, 69]. For example, with T2I models, the user can first generate the overall background and specific objects later. In the language of prompts-as-color, a brush could be loaded with an 'ocean' color and applied to the background to be followed by the targeted application of the 'boat'-color to certain areas.

4 PROMPTPAINT: INTERFACE

Using the interactions described in Section 3, we built PromptPaint, an image creation tool powered by a diffusion-based T2I model (Figure 1). PromptPaint supports the flexible steering of the generative model with 1) exploration and fine-grained control of prompt space with *prompt mixing* and *directional prompts*, and 2) the gradual building of images with *prompt intervention* and *prompt stencils*.

4.1 Canvas and Basic Editing Functions

PromptPaint presents a canvas where the user can create raster images (Figure 1h) with basic image editing functions. This includes moving/rotating images inside the canvas, brushing, erasing, and lassoing (from left to right of Figure 1a, except the right two). Furthermore, the user can add layers, change their ordering, hide, or even delete them (Figure 1h).

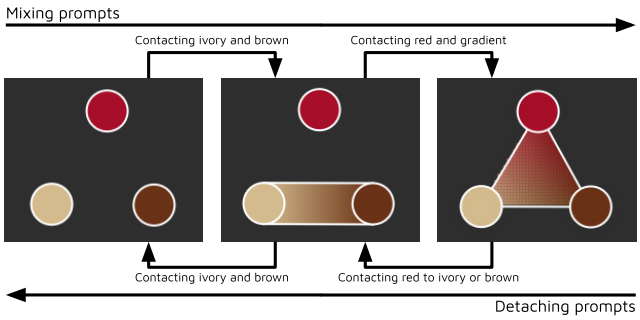

Figure 5: Interactions to mix/detach prompts in the Prompt Palette.

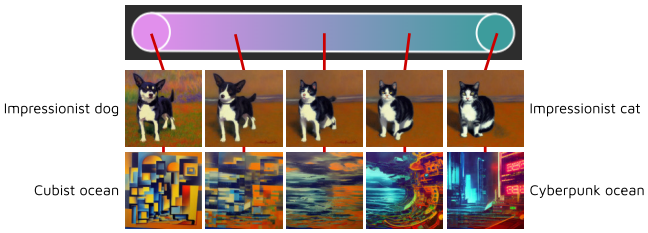

Figure 6: Example results of prompt mixing.

4.2 T2I Generation Functions

Through diffusion-based T2I functions, the user can generate images on the canvas. The user first specifies the prompts to guide the generation (*prompt mixing* and *directional prompt*). The user can then start the generation by specifying the area to which generation results should be applied (*prompt stencil*). During generation, the user can also change the guiding prompts (*prompt intervention*).

4.2.1 Prompt List. The user can add prompts in the Prompt List (Figure 1b). They can add a new prompt with the + button. Each added prompt has its own color (editable through a color picker) and editable prompt text. The user can delete the prompt with the X button.

4.2.2 Prompt Mixing. PromptPaint renders each prompt as a circle in a palette area (Figure 1c). The user can move and organize these prompts by dragging them (just as they could decide where to place their paints on a regular palette). The user can 'blend' the prompts though *prompt mixing* to explore the vector space between specified "discrete" prompts [44]. To do this, the user can directly manipulate prompt color circles with an interaction similar to how we mix paint mediums [76]. As in Figure 5, the user can touch one of the prompts on another to mix two prompts. If the user wants to add a third prompt to the mix, they can touch the already mixed gradient with the third one. While the current version of PromptPaint allows mixing a maximum of three prompts, future versions could allow mixing more. If the user wants to detach a prompt from the mix,

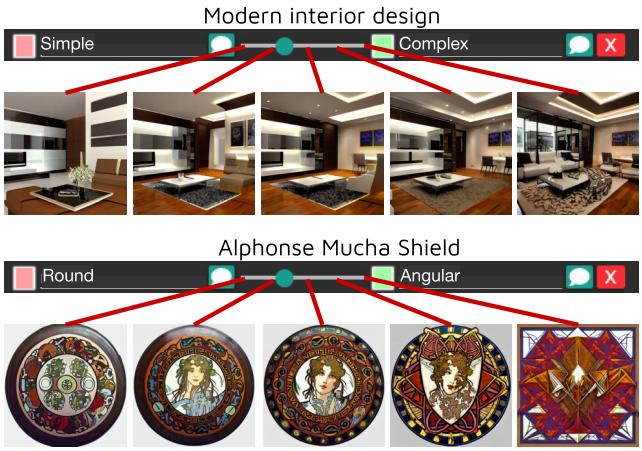

Figure 7: Example results of directional prompts. The center is the result without a directional prompt and the left and the right are results of applying directional prompts. The rightmost and leftmost results applied the full vector difference between the two end prompts.

they can touch the prompt with another prompt within the mix. For a generation input, the user can select one of the prompts or a point from a mixed gradient of prompts. The selected will be rendered as a circle (Figure 1c-2). The Prompt List will highlight the prompts mixed in the selection in green (Figure 1b). PromptPaint would interpolate these prompts to guide the diffusion model's generation. Figure 6 shows examples of mixing two different prompts with prompt mixing.

4.2.3 Directional Prompt. Directional prompts allow users to shift the prompts by introducing additional attributes [61, 71], with interactions similar to adding other colors (Figure 1d). The user can add a new directional prompt with the + button. With it, they can set two ends with prompts and decide the direction of the attribute to add. Each end has a unique, user-definable, color. After setting the two ends, the user can toggle the slider to set the intensity of the attribute to add. In Figure 1d, for example, the user can add a slight amount of the "matte" attribute by moving the slider closer to "matte." As the user moves the slider to one end in the directional prompt, the background color of the Prompt Palette gradually changes to the unique color of the end. With multiple directional prompts, this color changes to a mix of colors from the ends, with weights according to the slider values. Figure 7 shows examples of using directional prompts.

4.2.4 Prompt Stencil. After setting the prompt to use, the user can gradually build the visual images with a prompt stencil. The user can specify the area of the generation with brushing [7, 58, 69] while DRAW is selected (Figure 1e). As the user completes brushing, PromptPaint starts to generate an image, with intermediate generation results shown to the user in real-time (Figure 8). The progress bar in Figure 1g shows how much of the generation is done. The user can repeat this process to fill in other canvas areas. Note that the user can adjust the intensity of the guidance and the number of intermediate steps with GUIDE SCALE and STEPS in Figure 1e.

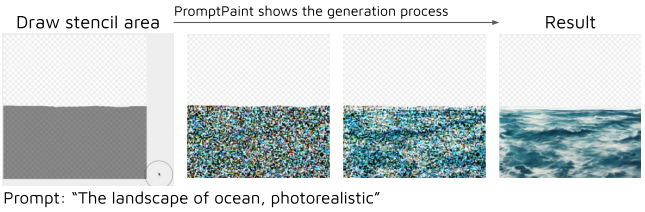

Figure 8: With a prompt stencil, the user can specify the area of generation with brushing (dark grey). When the user completes brushing, the tool starts generating a part of the image while showing the process to the user.

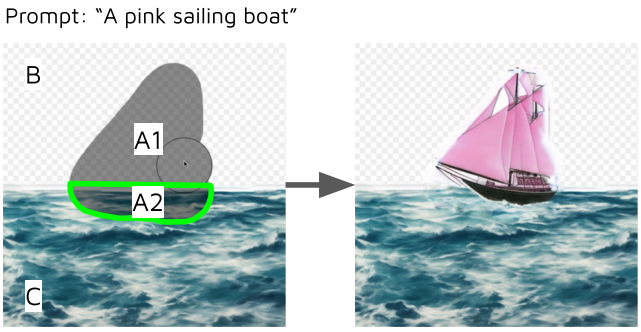

Figure 9: When applying prompt stencil upon the existing image, for the area where the stencil is overlapping with the existing image (A2), PromptPaint generates a new image that is similar to the existing image based on the overcoat value (higher, less similar).

Figure 10: Example results of overcoating generation. Images are generated again from the far left image with varying overcoat values (more right, higher overcoat values). Images generated with higher overcoat values tend to be less similar to the original image.

When the canvas already has images on the layer, PromptPaint considers those existing content to generate new content. For example, as in Figure 9A2, when the user's new stencil overlaps with the existing images, based on the OVERCOAT value (Figure 1e), PromptPaint tries to generate a new image that is similar to the already generated images (the higher the overcoat value, the less similar).

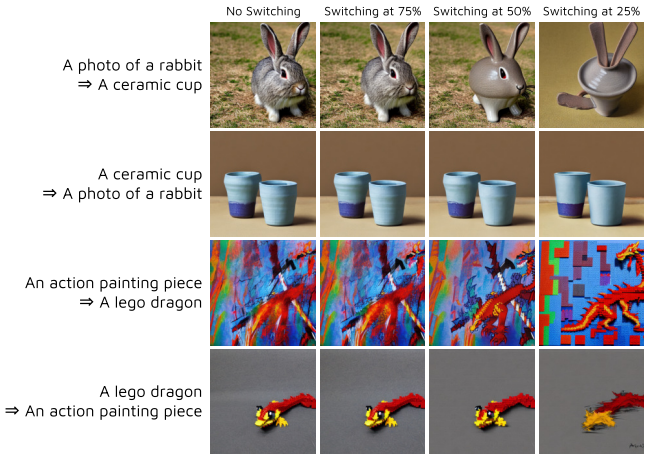

Figure 11: Example results of prompt intervention. Except for the first column which did not change the prompt during the generation, each column switched the guiding prompt at a different point of the generation process.

Figure 10 shows the impact of varying overcoat values when generating an image again with the same prompt. We can also use overcoating to change an image by using different prompts from those used to generate the existing image.

4.2.5 Prompt Intervention. With *prompt intervention*, PromptPaint allows interactions *during* the generation process. The user can change either 1) the selection of the prompt in the Prompt Palette or 2) the slider values for directional prompts. With the change of prompts, as in Figure 11, prompts used in the earlier stage of the generation tend to decide the overall form and color, while those used in the later stage decide details. Hence, this technique can maintain the visual form of the generation with iterations. However, as in the second row of Figure 11, sometimes the generation result does not change much even with early prompt intervention.

As the user can change the prompts during generation, PromptPaint allows for control over the generation process. First, PromptPaint visualizes the prompts used in previous generations as paths in the palette interface as in Figure 1c-1. Note that the used directional prompts decide the colors of the dots. This visualization of past paths helps users understand what they have tried and eases iteration on different combinations of prompts. Furthermore, they can stop and restart the generation (the button in Figure 1g changes to either STOP or START). When the user has stopped generation, they can roll back the generation to a specific step, either by selecting the past point in the progress bar (Figure 1g) or undoing with CTRL and Z keys. If the user wants to switch to past versions of generations, they can click one of the dots on the past paths. PromptPaint highlights the dot for the current generation step with a green border. When restarting the generation, the user can also set the number of steps processed in a single "round" of generation, with SINGLE STROKE in Figure 1e.

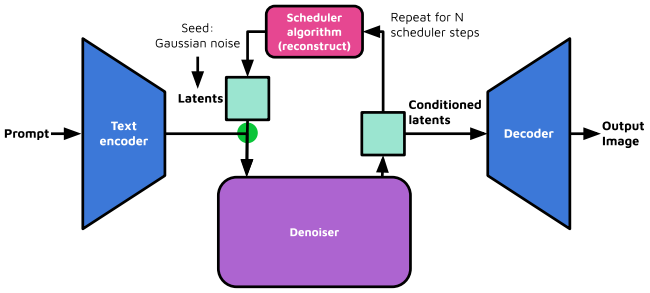

Figure 12: The pipeline of diffusion-based text-to-image models. It shows a specific version, which processes the diffusion process in latent representation. The technical manipulation of PromptPaint happens in the green dot, either by manipulating vector-embedded prompts or intermediate latent.

5 PROMPTPAINT: TECHNICAL DETAILS

We describe the technical details of PromptPaint. First, we give an overview of diffusion-based T2I generation models. Then, we explain the technical approaches for each function and the implementation details.

5.1 Background: Diffusion-based T2I

While there are many variants of diffusion-based T2I generation models [66, 70], we focus our explanation on the latent diffusion models [68] we used. Latent diffusion models [66, 68, 70] can largely be characterized by (Figure 12): 1) a text encoder that converts text prompts into vectors used to guide diffusion, 2) a denoiser and scheduler, which gradually process image generation by reducing noise in the latent vectors of the image, and 3) a decoder, which turns latent vectors into a higher resolution output image. Note that the decoder is not the universal feature of diffusion-based T2I models, but helps with computational efficiency by processing images in lower-dimension latent representations. Often the geometry of the latent representation corresponds to the output image, whereas the output image tends to have higher dimensions. When the image on the canvas needs to be used for diffusion (e.g., overcoating), it requires an encoder that encodes the image into a latent representation. All technical manipulations of PromptPaint occur in the green dot of Figure 12, either by manipulating vector-encoded text prompts (prompt mixing, directional prompts, prompt intervention) or latent representations from the denoiser and scheduler (prompt stencil).

5.2 Prompt Vector Manipulations

Prompt mixing and *directional prompt* manipulate the prompts in the vector space embedded by the text encoder. Prompt mixing interpolates different prompt vectors with weights [66] from the user's Prompt Palette (with proximity to each prompt):

$$v_{p_m} = \sum^{N} w_{p_i} v_{p_i} \quad (1)$$

v_{p_m}, w_{p_i}, and v_{p_i} represent the interpolated vector, the weight of each prompt, and the embedded vector of each prompt, respectively. The directional prompt first calculates the directional vector

between two different prompts:

$$v_{d_j} = v_{d_j1} - v_{d_j2} \tag{2}$$

where v_{d_j} indicates the directional vector and v_{d_j1} and v_{d_j2} stand for the embedding of the prompts at both ends. Then, with weights on the slider interfaces (w_{d_j}), PromptPaint calculates the final input of the prompt vector:

$$v_f = v_{p_m} + \sum^{M} w_{d_j} v_{d_j} \tag{3}$$

Then, PromptPaint would use v_f as input to the denoiser model to guide the denoising process. Users can change v_f during generation, which is how *prompt intervention* is technically done.

5.3 Latent Representation Manipulation

Prompt stencil and overcoated generation require manipulation in latent representations before the denoiser process. Specifically, PromptPaint manipulates latent representations considering different areas with or without stencils and existing image content. This approach is similar to image-to-image diffusion in previous work [7]. For the area that is stenciled but does not have any image content (Figure 10A1), no manipulation is performed. However, for the stenciled area with image content (Figure 10A2), latent representation is manipulated as follows:

$$l_{m,k}(x,y) = \begin{cases} v(l_I(x,y),k), & \text{if } k < (1-o/100) * K. \\ l_k(x,y), & \text{if } k >= (1-o/100) * K. \end{cases} \tag{4}$$

As in the above equation, when the diffusion step (k) is smaller than the threshold $(1-o/100) * K$ (o is the overcoat ratio and K is the number of all diffusion steps), $l_{m,k}(x,y)$ (the latent representation after manipulation at the position of (x,y) and the step of k) is replaced by $v(l_I(x,y),k)$ (where l_I is the latent representation of the existing image and $v(l,k)$ is a function that adds noise to the latent representation to the amount adequate to step k). Otherwise, the latent representation would not be manipulated and PromptPaint would use the reconstructed latent representation from the scheduler algorithm. For unstenciled areas, the area of Figure 9B is considered to have a white background. Then, the area of Figure 9B and C would be replaced as follows:

$$l_{m,k}(x,y) = v(l_I(x,y),k) \tag{5}$$

Therefore, these unstenciled areas are replaced by the latent representation of existing images with noise added for each step. Note that we adopted this approach instead of inpainting models [83] to simulate "overcoating" effects where an already-filled area needs to maintain visual similarity with the additional generation on the area.

5.4 Implementation

We implemented PromptPaint as a web app using HTML, CSS, JavaScript, and React. For deploying a diffusion-based T2I model, we built a WebSocket-based Flask server, as PromptPaint shows intermediate generation results in real-time. For the T2I model, we used Stable Diffusion [68], which uses UNet for the denoiser and the variational autoencoder for the decoder and encoder. For the UNet and the variational autoencoder models, we used `runwayml/stable-diffusion-v1-5`[1]. For the text encoder, we used the `openai/clip-vit-large-patch14` checkpoint[2] of CLIP [64]. For the scheduler, we used the DDIM scheduler [79] with a beta start of 8.5e-4, a beta end of 1.2e-2, and a scaled linear beta schedule.

Table 1: Attributes used in characterization study.

Type	Attributes
Objects	tree, river, man, woman, dog, cat, love, hate
Styles	cubist, surrealism, action painting, high renaissance, impressionism, cyberpunk, unreal engine, VSCO
Specific visual attributes	vivid color, subtle color, rough texture, smooth texture, fine line, thick line, curvy shape, angular shape

6 CHARACTERIZATION STUDY

Through a crowdsourced study, we characterize PromptPaint functions in terms of how they allow users to iterate on the already generated images by adding another attribute.

6.1 Conditions

We considered 1) **prompt mixing**, 2) **directional prompt**, 3) **prompt stencil**, 4) **prompt intervention**, and 5) **prompt concatenation**. While the prompt stencil's fundamental purpose is not to add another attribute to existing images, we can repurpose this to "overcoat" other visual elements on existing images. The last condition, prompt concatenation, is textually adding another attribute to the existing prompt (e.g., "mix of impressionism and cubist").

6.2 Dataset

To characterize each condition, we generated an experimental dataset. When adding a new attribute, we considered three attribute types: 1) **objects**, 2) **styles**, and 3) **specific visual attributes**. We considered **objects** and **styles**, as they are often used as the most basic attributes in T2I generation [53]. We additionally considered specific visual attributes such as colors, textures, lines, and shapes, as they are widely used to describe visual attributes in the practice of visual arts [4]. Table 1 shows the descriptors we used for each type. We used a subset of object and style descriptors from Liu and Chilton [53]. We sample specific visual descriptors from art learning materials [4].

With these attributes, we systematically generated pairs of images: an image generated with the initial prompt and an iterated version that added another attribute. First, we chose a "target attribute set" from objects, styles, and specific visual attributes, which would be the type of attribute added in the iterated image. Then, we sampled two attributes from the target attribute set, the "original attribute," to be included in the initial prompt, and the "additional attribute," to be added in the iterated image. Note that some pairs of attributes can be semantically more relevant to each other than

[1] https://huggingface.co/runwayml/stable-diffusion-v1-5
[2] https://huggingface.co/openai/clip-vit-large-patch14

other pairs (e.g., man and woman). Below, we show how the generation can be different between more and less relevant pairs. To generate images, we needed attributes other than the target attribute and the seed to initialize the noisy latent representation. For example, when we use objects as target attributes, we would need to have style attributes in the prompts. On the other hand, when we pick specific visual attributes as target attributes, both style and object attributes would be required. For a pair of target attributes, we randomly sampled two sets of other attributes and seeds.

For each set of attributes (original, additional, and non-target) and seed, we generated images with varying weights and conditions. For **prompt mixing**, with varying weights, we interpolated vector embeddings of two prompts (one with an original attribute and non-target attributes and the other with an additional attribute and non-target attributes). Note that when we composed the text prompt with different attributes, we combined them with commas, in the order of object, style, and the specific attribute (if considered). For **directional prompt**, we calculated a directional vector between the original and additional attributes and added it to the prompt composed of the original and non-target attributes with different weights. For **prompt stencil**, we first generated an image with the prompt of the original and non-target attributes and did an overcoat with the prompt of the additional and non-target attributes. Here, we varied the level of the overcoat with the weight. When we added noise to the overcoating, we used a seed that was different from the seed we sampled before (as using the same seed resulted in a low-quality image). We fixed this seed across different overcoat weights. For **prompt intervention**, we first started the generation with the prompt of the original and non-target attributes. Then, at a specific moment of the generation, we changed it to the prompt of the additional and non-target attributes. In this case, with higher weights, we changed the prompt earlier. For these approaches, we used three weights, 0.25, 0.5, and 0.75, on a scale of 0 to 1. For **prompt concatenation**, we combined all of the original, additional, and non-target attributes in the prompt (e.g., "the mix of cat and dog, impressionism"). With these approaches, we generated 4368 image pairs for all target attribute types, using 50 diffusion steps with the guide scale set to 7.5.

6.3 Metrics

We considered four metrics: 1) how clearly the new attribute is added (`addition`), 2) how clearly the original attribute persists (`remain`), 3) how similar the newly generated image is to the original image (`similarity`), and 4) the specific way in which the new attribute is added (`addition approach`). For metrics other than `addition approach`, we used a 7-level Likert scale to gather answers. For the `addition approach` question, options varied depending on the types of attributes added, as in Table 2.

6.4 Procedure

For generated image pairs, we conducted a characterization study on Amazon Mechanical Turk. We showed each crowd worker ten pairs of randomly sampled images. For pairs generated with the target attribute of styles, we included examples of each style so that those who do not know the style terms can see examples. For each pair, we asked the worker four questions about all metrics. One pair in each set showed identical images and was used as an attention check. Crowd works were filtered if they answered the `similarity` metric at lower than 6 (of 7) or their answer to the `addition approach` metric was something other than NoMixNoChange (Table 2). We also filtered out a worker's answers when they answered in streaks of the same value for `addition`, `remain`, and `similarity`. Specifically, we filtered out the worker's answer if the ratio of the same value is higher than 70%. We recruited workers with an acceptance rate higher than 98% and an accepted HIT number greater than 10,000. We only recruited workers in the US and paid them $1.50 (9-minute task, $10/hr payment rate).

Table 2: Options used for `addition approach` questions. Bolded text indicates the option name.

	Options of `addition approach` questions
For styles and specific visual attributes	ORIGINAL ATTRIBUTE and ADDITIONAL ATTRIBUTE are both applied in the new image, but largely to different places/things in the image. (**Separate**)
For objects	ORIGINAL ATTRIBUTE and ADDITIONAL ATTRIBUTE are placed together in the new image as separate objects. (**Separate**)
	The new image is ORIGINAL ATTRIBUTE-shaped ADDITIONAL ATTRIBUTE. (**O-Shaped A**)
	The new image is ADDITIONAL ATTRIBUTE-shaped ORIGINAL ATTRIBUTE. (**A-Shaped O**)
Common	ADDITIONAL ATTRIBUTE is added to the new image, but not in the ways described above. (**Mixed**)
	ADDITIONAL ATTRIBUTE is not added to the new image, but the image changed. (**NoMixChange**)
	ADDITIONAL ATTRIBUTE is not added to the new image, and the image did not change. (**NoMixNoChange**)

6.5 Result

Figure 13 shows the `addition`, `remain`, and `similarity` results for each target attribute. Figure 14 presents how mixtures of prompts with different conditions and weights resulted in different `addition approaches`. For all attributes and conditions that can be weighted, the increase of weights resulted in higher `addition` while decreasing `remain` and `similarity`, with more `addition approaches` other than NoMixChange and NoMixNoChange. There were clear trade-offs between conditions: those approaches that more clearly add new attributes tend to lose the original attribute and the similarity to the original image. Overall, prompt intervention induced the minimal addition of the new attribute while highly maintaining the original attribute and the similarity to the original image. Other conditions followed in manifesting such patterns: prompt stencil, directional prompt, and prompt mixing (in that order). The specific trend varied between different target attributes. With the target attribute of objects and styles, prompt stencil and prompt intervention had similar `remain` and `similarity` scores, where prompt mixing and directional prompting formed another group. For specific visual attributes, `remain` and `similarity` gradually changed with weight change, which would be because the added attribute is a smaller part of the whole image.

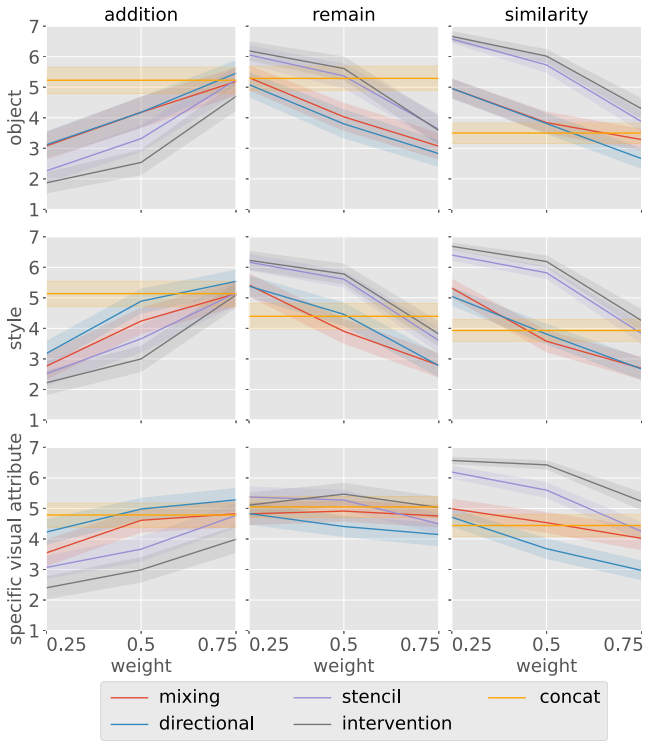

Figure 13: `addition`, `remain`, and `similarity` of different conditions and target attributes. The shaded areas indicate 95% confidence intervals.

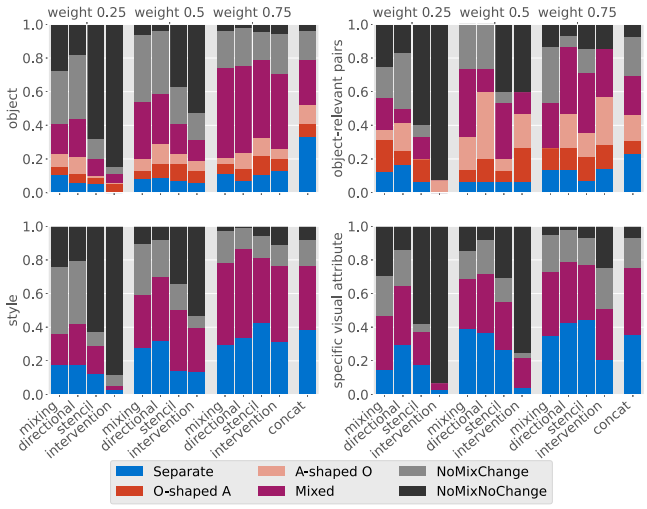

Figure 14: `addition` approach of different conditions and target attributes. For objects, we also show the results that only consider pairs consisting of closely relevant objects.

With `addition` approaches, prompt intervention had a low rate of changing images with low weights (i.e., high NoMixNoChange), followed by prompt stencil. For the target attributes of objects and styles, as weights increase, these rates for prompt intervention and prompt stencil increase to those of prompt mixing and directional prompt. Only for the target attribute of specific visual attributes, prompt intervention changes images in a lower rate than other conditions even with increased weights.

Concatenation could add the new attribute while remaining the original attribute, but it produced images not very similar to the original image. Moreover, concatenation of objects more frequently placed separate objects rather than mixing them.

For objects, the distribution of `addition` approaches could be different when only considering pairs that are closely relevant (i.e., tree-river, woman-man, dog-cat, and love-hate). With relevant pairs, there were more O-shaped-A or A-shaped-O, potentially as two mixed objects were semantically relevant (sometimes, even visually). At the weight of 0.5, directional prompt and prompt intervention has high rates of O-shaped-A and A-shaped-O, while those rates for prompt stencil were low with the weight of 0.5. With high weights (0.75), prompt intervention had the highest rate of O-shaped-A and A-shaped-O.

Table 3: User study participants, with their expertise in visual arts, the domain of interest, and experience in T2I models.

	Visual art	Year	Domain	T2I
P1	Hobbyist	5	Vector arts	Yes
P2	Hobbyist	10	Paintings, cartoons, graphic arts	No
P3	Hobbyist	20	Sketches, paintings	No
P4	Hobbyist	30	Simple drawings, paintings	Yes
P5	Novice	N/A	N/A	No
P6	Novice	N/A	N/A	Yes
P7	Hobbyist	3	Sketches	No
P8	Novice	N/A	N/A	No

7 USER STUDY

We conducted a user study to understand how PromptPaint extends the use of diffusion-based T2I models with interactions inspired by how we handle paint mediums. We focussed on how interactions of PromptPaint could affect the user's experience in exploring and steering T2I generations to "create" visual artifacts. Therefore, we conducted an observational study with qualitative analysis.

7.1 Participants

We recruited eight participants (five females and three males, ages 22-51, M=28, SD=9.53) through university mailing lists. We asked participants to do a prescreening survey, checking if they can participate, as the study requires participants to see and hear. We recruited hobbyists or novices in visual arts, as experts would be less likely to use automated generation tools in their practice (i.e., they have the expertise to create visual arts by themselves). During the study, we asked them to complete a prescreening survey that asked about their experience in visual arts and T2I models, whose results are summarized in Table 3. We gave each participant an Amazon gift card worth $20.

7.2 Procedure

We conducted an in-person lab study. We first asked participants to complete a pre-survey. Then, we showed the participants a video with an overview of the study (5 minutes). As we asked participants to think aloud during the study, this video instructed participants about the concept and an example of think-aloud. The video also introduced the basic functions of PromptPaint, which are raster image editing functions other than image generation (e.g., brushing, erasing). Then, the video explained how to generate images with a single prompt. After the first video, we asked the participants to try the functions in PromptPaint. The participants then went through four rounds of task sessions for four functions, in the order of prompt mixing, directional prompt, prompt intervention, and prompt stencil. The participants went through the fixed order since the latter functions require knowledge of the previous ones. For each task session, participants went through four steps: 1) watching an instruction video, 2) trying out the function as a tutorial with the researcher's guidance, 3) freely creating visual artifacts as they want, while thinking aloud, and 4) completing a post-task survey. Each instruction video took 1-2 minutes. Each tutorial took about 5 minutes. We gave 10 minutes for each creation task and asked the participants to actively try the newly learned function. Post-task surveys asked participants if the function they had just tried facilitated 1) control of image generation or 2) exploration of good surprises. After all functions, participants were asked to complete an exit survey, which asks about the general usage of the tool. This survey asked questions in the creativity support index [14], except those that questioned whether the tool helped collaboration. The post-survey also asked about the participant's sense of ownership and contribution, and if they felt they were collaborating with the system. After the post-survey, we conducted a short interview. In the interview, we asked about their strategies for using PromptPaint, how they felt about the ownership of the artifacts, and their impression of the four functions. The entire study took no more than 100 minutes.

7.3 Results

We report on the results of surveys, observations of the task with think-aloud, and interviews.

7.3.1 Survey results. Figure 15 shows the results on the creativity support indexes. Participants were generally positive about PromptPaint, perceiving that it facilitated enjoyment, exploration, expressiveness, and immersion, while the results were worth their effort. However, there was one participant who responded neutrally or negatively to these questions. In this case, the participant had very concrete expectations of what they wanted. For immersion, there was one participant who answered negatively about the immersive aspect of the tool.

Figure 16 shows how participants felt about ownership of the generated images, how much contribution they made (compared to AI), and whether they collaborated with PromptPaint in creating the artifact. Interestingly, participants felt that AI contributed more, but many still answered that they have some ownership of the generated artifact. At the same time, participants tended to answer that they 'collaborated' with PromptPaint.

Figure 17 shows the participants' perceptions on how each function supported 1) the control of generation and 2) the exploration of interesting and good surprises. Overall, participants perceived all functions positively. While it is difficult to learn significant differences between functions due to the small size of the data, participants tend to perceive that the prompt stencil helped the most with controlling and exploring generation, while intervention prompts helped the least.

7.3.2 Qualitative Results. For qualitative results, we analyzed think-aloud, screen recording, and interviews by iterative coding with inductive analysis. We present findings on four functions, trade-offs in designs, the complexity of AI, and ownership issues.

Four functions. As seen in Figure 18, **prompt mixing** allowed participants to explore the image space that is difficult to describe verbally (N=7). Some participants mentioned that visualization and interactions on the Prompt Palette interface helped them explore and manipulate the prompt semantics (N=2). One interesting thing that one of our participants (P1) discovered was that when two prompts are semantically far, mixing concepts does not change the image linearly, but more in drastic "steps." For example, in Figure 19, P1 tried to mix two prompts, "Colorful 8k photograph of a man's face" and "Satellite image in North America." Here, at a certain boundary, a small increase of weights on one prompt could drastically change the image, indicating that the interpolation did not impact the result linearly.

Directional prompts could help participants add attributes that do not exist within the first prompt they have tried (N=6, Figure 20). P3 mentioned that the function helped explore and do fine-grained controls between two opposite concepts: *"I think the strength of this is like you can see how opposed things are, and how it is seen in between . . . Basically this function allows you to explores something that is in-between. Sometimes it's very difficult to imagine things, and this would have been very helpful in that."* One challenge in using directional prompts was deciding on two semantically opposite prompts (N=5). The participants expected that PromptPaint could have recommended the options for opposite prompts after the user inputs one prompt. In other cases, the way the participants interpret the prompts did not align with PromptPaint's. We return to this below.

Prompt stencil allowed users to perform fine-grained controls with localized image generation (N=7). As in Figure 21, participants could gradually create the image by adding and changing visual elements in the scene. Participants could also adjust the overcoat level to generate partial images that are more or less similar to the existing ones. However, the prompt stencil also had limitations. For example, as in "a mystical elf druid" in Figure 21, newly generated parts could be incomplete. Furthermore, as in "a face of a happy woman" of Figure 21, generated images could be mismatched with the existing ones. In some cases, the style of the newly generated images did not match the existing ones.

Participants thought that **prompt intervention** allows them to generate interesting mixes of prompts (N=6). P2, who created the image in Figure 22, mentioned: *"I think the strength of that is making something completely ridiculous and fun and changing an aspect of something to match something else."* However, prompt intervention was the most difficult to use (N=5). Some participants were unable

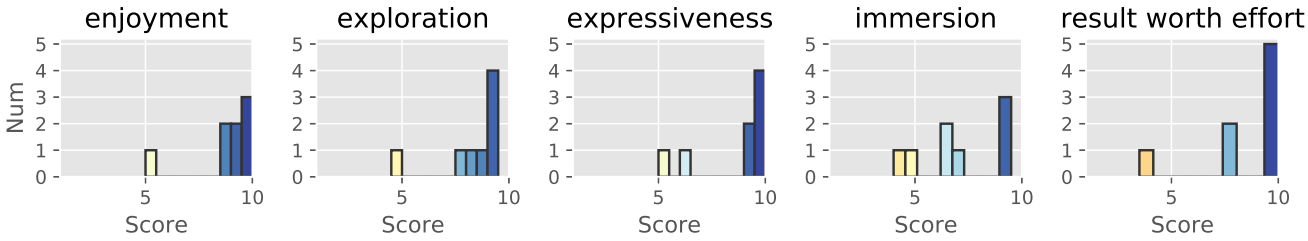

Figure 15: The histogram of responses on the creativity support index questions. The high score indicates that the participant perceived that PromptPaint supports the criteria.

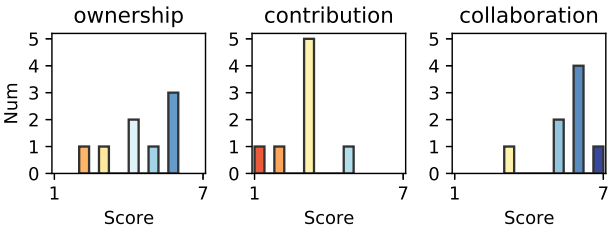

Figure 16: The histogram of responses to the question about the sense of ownership, contribution, and collaboration. The higher the scores, the participant felt that they have more ownership than PromptPaint, they contributed more than AI, and they collaborated with AI functions.

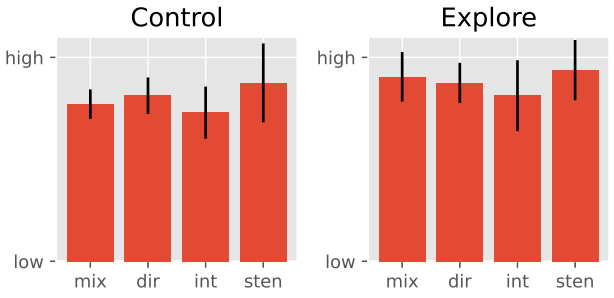

Figure 17: Comparison of four different functions on if they helped with 1) controlling or 2) exploring generation. `mix`, `dir`, `int`, and `sten` stand for *prompt mixing*, *directional prompt*, *prompt intervention*, and *prompt stencil*, respectively. The error bars indicate 95% confidence intervals.

to create a satisfactory image. It was due to the difficulty of deciding when to switch prompts as it was hard to guess the result only by seeing intermediate generation results (i.e., noisy images during the diffusion process). The interface showing all previous generations (in the prompt palette in Figure 1c) could help users understand the previous generations they have tried (N=3) and iterate on the prompt intervention. However, it was easy to clutter the interface with multiple rounds of iteration.

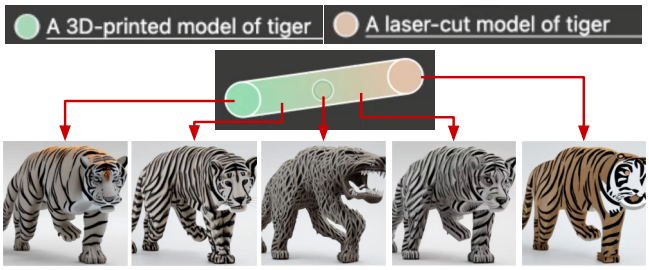

Figure 18: Prompt mixing from P5. By mixing semantically close prompts, the user can control the generation to explore the image space in between.

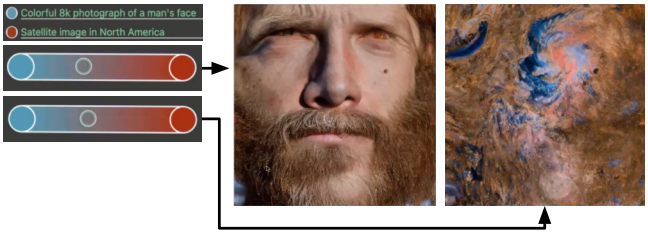

Figure 19: Prompt mixing from P1. For semantically far prompts, interpolation of vector-embedded prompts sometimes did not result in images with a mixture of concepts.

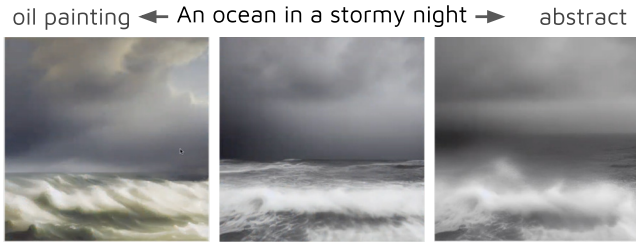

Figure 20: Directional prompt from P6.

Design Trade-offs. Participants mentioned two potential trade-offs in the design of T2I generation tools. The first was the design trade-off between focusing on one canvas versus curating many results (N=2). We designed PromptPaint to allow users to iterate

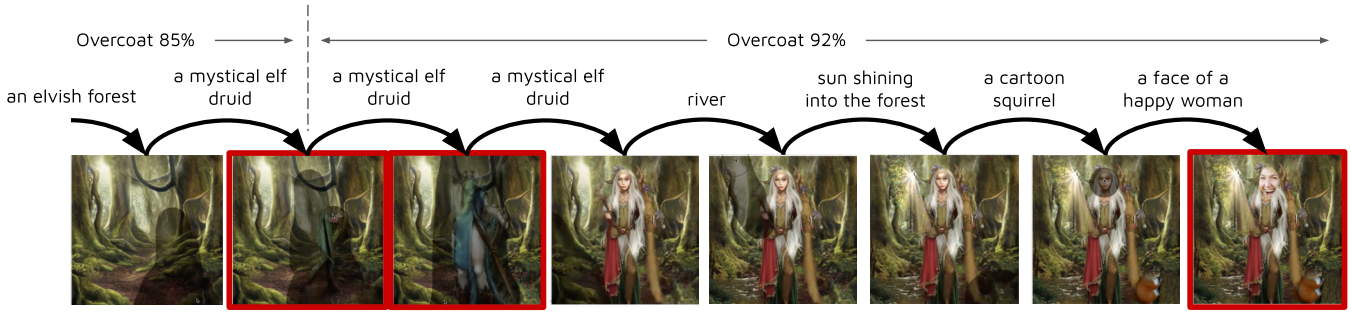

Figure 21: Prompt stencil from P2.

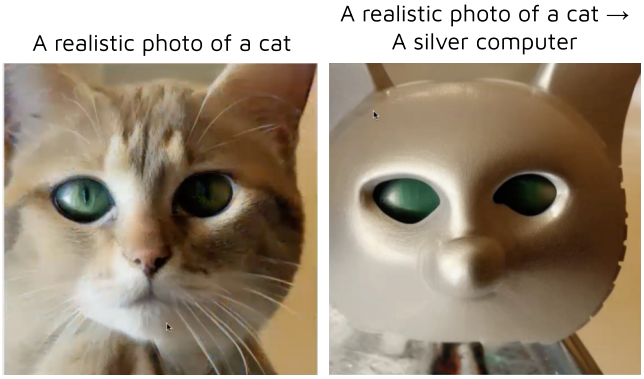

Figure 22: Prompt intervention from P2.

on a single canvas, giving users the experience closer to "gradually creating an image." However, due to stochasticity in the diffusion model (e.g., randomness from different seeds), participants found that seeing multiple results would be helpful in some cases. Furthermore, we designed PromptPaint to allow users to have more controls and interventions. For example, the tool allows users to have a high degree of freedom to change the prompts during generation. Although such designs open up new interactions in using diffusion-based T2I tools, some users found such designs too manual (N=2). Participants mentioned that the balance between automation and manual interventions would help.

High Complexity and Randomness of AI. Participants mentioned that the high complexity and randomness of AI behaviors were limitations of PromptPaint (N=7). Such complexity and randomness could make the generation result misaligned from the user's intention. Dissatisfaction tends to be more intense when the user has a more concrete picture of what they want. For example, P7 thought that prompt stencils often failed to generate image parts with consistent perspectives and was most dissatisfied with the prompt stencil. To facilitate generative AI even with such barriers, participants adopted some strategies. Some tried to understand how AI works in simple settings (e.g., using a single prompt) and then applied more complex functions (e.g., prompt mixing) based on their understanding (N=2). Some tried to understand how the model "interpret" prompts by using functions that interpolate or shift the semantics of the prompts (N=2). For example, P1 interpolated the prompts of an apple and a pear to learn how the machine interprets the attributes of each fruit.

Ownership and Contributions. Participants felt some ownership of the resulting visual images (N=8), with varying degrees between participants. They mentioned that they contributed high-level ideas, while AI contributed low-level ideas and implementations. P1 mentioned that they became like "Steve Jobs" and AI would be "an Apple employee," and P2 thought that using PromptPaint felt like doing an art commission with more control. P5 felt less ownership of the resulting piece because they were a novice in visual arts. P5 mentioned: *"I think AI contributes more than me and it's because I'm a novice. I did not paint at all, and I don't use any drawing software as well. So I think all the beautiful images are created by AI instead of me. I just specify the position, and it's just parameters."* Some participants also mentioned that they felt more ownership in the resulting artifacts if they align with what they expected (N=2). One participant mentioned the potential issues in the legal ownership of the generated artifacts, showing concerns about the copyright.

8 DISCUSSION

Based on the suggested design of generative tools and PromptPaint, we discuss 1) the generalizability of paint-medium-like interactions in generative tools, 2) the characterization of different approaches to mixing prompts, 3) in-generation interactions for T2I models, 4) design trade-offs in generative tools, 5) ownership issues, and 6) limitations.

8.1 Paint-like Interactions for Generative Tools

We can apply the idea of paint-medium-like interactions beyond PromptPaint. For mixing discrete semantics, we can easily replace prompts with other inputs, such as examples. On the other hand, we would need to redesign modularized generation specifications for each content modality. For example, for the generation of 3D models [41, 63], users would need to be able to select 3D parts to iterate. Similarly, for content with sequential axes, such as text, video, or music, the generation specification would need to consider the sequential dimension [18]. In the interface, they can be instantiated in sketches of different semantics along the sequential axis. Interactions for modularized specifications would likely be more complex for mediums with both spatial and sequential dimensions

(e.g., videos). Still, the design pattern of applying different semantics (analogically, colors) to different parts of the artifact would generally hold across modalities.

For in-generation interventions, in PromptPaint, diffusion-based T2I models used the earlier prompts to decide the overall form and colors while using the later ones to render details. Similarly to diffusion-based T2I models, models for other mediums can be designed to gradually generate from "high-level characteristics" to "details" to allow user interventions in a generation. For example, music generation algorithms can generate, in the order of song structures, bars with chords, notes in each bar, and then embellishments such as legato or staccato.

8.2 Characterizing Approaches to Mix Prompts

Our characterization study revealed the pros and cons of approaches to mix prompts. Prompt intervention and prompt stencil tend to maintain the original attribute and similarity to the original image, while prompt mixing and directional prompt tend to add the new attribute with higher chances. All these approaches also have benefits over concatenating prompts, as the user can adjust how much of the new attribute to add. This characterization would guide us in deciding the mixing approach that would best achieve a user's specific purpose. We argue that researchers need to conduct this type of characterization for emerging T2I techniques, as with many different approaches, we do not yet have a good understanding of which would best fulfill a user's specific intention.

8.3 Interaction for T2I models

PromptPaint allows users to interact with generative models *during the generation process* by changing the prompts, with earlier prompts forming the overall composition and later prompts deciding on details. While participants found this function interesting, they struggled to learn the best way to use it (specifically, for finding the right moment to change the prompt). Seeing and interpreting intermediate representations might help overcome the limitations. For example, if the intermediate noise-added image has quite a concrete object, it might indicate that changing the prompt would not induce changes. However, not many users are familiar with such noise-added images. Therefore, users would need to *learn* to interpret noisy images, which places more load on users. Making intermediate results more understandable to human users would be an approach to facilitate in-generation interactions for diffusion-based T2I models [8]. For example, diffusion models that gradually concretize images from more pixelated ones would be more understandable to the users, allowing them to grasp what the model might generate from the current intermediate step. Moreover, such representations can allow users to edit the intermediate results. For example, with pixelated intermediate images, if the user is generating a human face and spots a "blonde" color in the area of hair, they would be able to change the color of the hair by changing the region to other colors.

We also emphasize that T2I models are quickly evolving, and PromptPaint can be extended to new models. For example, with models designed specifically for overcoating and inpainting [83], prompt stencil interaction can be improved, and the user reaction might likely change. In PromptPaint, we did not include negative prompts [6]. This can be adopted into our interface, as either a text box that applies to all of the user's generation or as another palette that allows users to flexibly define the negative semantics. PromptPaint also does not include approaches that add structural conditions, such as ControlNet [93]. Again, this also can be included either by allowing the specification of those conditions during prompt stencil or by training a ControlNet model that can condition the generation with the rough stencils.

8.4 Design Trade-Offs in Generative Tools

From the user study, we found design trade-offs for generative tools. First, while "creation tools" often assume a single artifact to be created (e.g., a single canvas for image editors), due to the complexity and randomness in generative models, generative tools would require some "curation" of multiple results. Providing both features would allow steering experiences while addressing some issues with the randomness of algorithms [12, 46]. For example, a generative tool can have multiple rounds of interactions that first receive user specifications, generate a set of candidates, allow users to select one of them, and then iterate. For effective steering, other control approaches, such as giving structural information of image renditions [93], can be adopted. For effective curation, it would be valuable to learn the user's preferences during the interactions to better align curated results with the user preferences. The second trade-off is the balance between automation and manual controls. With this trade-off, simple and automated interactions can be a "low threshold" way to steer the generation, while more manual steering interactions can be a "high ceiling" option [67].

8.5 Ownership of Generated Artifacts

Users of PromptPaint had some sense of ownership of the artifacts generated, as they contributed high-level ideas. At the same time, as PromptPaint contributed ideas and implementations of lower levels, they would have felt less ownership than creating artifacts themselves. For generative creation tools to secure the user's sense of ownership regarding the final artifacts, it would be important to understand which aspects of artifact creation contribute to the sense of ownership. For users who do not put a lot of value on manual labor, automating some parts of the artifact creation would not hurt the user's sense of ownership much. However, if the user values the skills and efforts involved in the creation of artifacts, then automation would hurt the sense of ownership. Hence, the tool would need to understand the user's values and allow users to select for which part they want to use generative AI. Ultimately, we need to incorporate generative functions into existing workflows so that we can preserve user values in their workflow [91].

Ownership is also a legal issue, such as not hurting copyrights. As existing diffusion models could copy content from the training dataset [78], it would be crucial to carefully curate the training dataset so that users do not infringe the legal ownership during their use. Although researchers have started to exclude images if the original owners do not want their images in the dataset[3], it is still opt-out. Moreover, there could be some trade-offs between

[3] https://haveibeentrained.com/

preserving legal ownership and having a large-scale dataset. Potentially, the transformation of image data can be a way to balance the preservation of ownership and the scale of the data.

8.6 Limitations

We did not compare interactions of PromptPaint to those of other existing tools, as our studies focused on 1) comparing different functions in combining semantics in prompts (Section 6) and 2) qualitatively studying usage patterns with paint medium-like interactions (Section 7). As many relevant T2I tools keep arising [1, 5, 80], doing systematic analysis on different interaction modes would be necessary for future work. Our qualitative study results can inform specific future study designs. For example, users perceived the value of iterative interactions in our tool while acknowledging the benefit of seeing many results at once. Based on this, future comparative studies can consider two dimensions, one being "facilitation of iteration" and the other being "showing multiple results."

9 CONCLUSION

In this paper, we introduce an approach for interacting with generative models as if prompts were paint colors. This design approach allows users to explore the semantic vector space in a way similar to how we mix colors. They can also gradually build the artifact with different semantics in a way similar to how we apply colors to varying parts of the painting process and the canvas. We are motivated by a desire to make end-to-end use of generation models more flexible and gradual with iterative steering. We apply the design approach in diffusion-based T2I models and introduce PromptPaint. PromptPaint adopts four steering approaches, prompt mixing, directional prompts, prompt intervention, and prompt stencil. Through user studies, we characterize these approaches and identify how people use the suggested interactions. Based on the findings, we draw insights into how we should design and build future generative tools.

ACKNOWLEDGMENTS

We thank Nikola Banovic and Anhong Guo for their valuable feedback on this work.

REFERENCES

[1] Adobe. 2023. Adobe Firefly (Beta). https://firefly.adobe.com/ Accessed: July, 2023.

[2] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. MusicLM: Generating Music From Text. https://doi.org/10.48550/ARXIV.2301.11325

[3] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. 2021. HyperStyle: StyleGAN Inversion with HyperNetworks for Real Image Editing. *CoRR* abs/2111.15666 (2021). arXiv:2111.15666 https://arxiv.org/abs/2111.15666

[4] Atlee Arts. 2023. ELEMENTS OF ART and PRINCIPLES OF DESIGN. https://atleearts.weebly.com/uploads/5/0/4/9/50491891/3.elements_and_principles_of_art-descriptive_words__1_.pdf

[5] AUTOMATIC1111. 2023. Stable-diffusion-webui: Features. https://github.com/AUTOMATIC1111/stable-diffusion-webui/wiki/Features Accessed: March, 2023.

[6] AUTOMATIC1111. 2023. Stable-diffusion-webui: Negative prompt. https://github.com/AUTOMATIC1111/stable-diffusion-webui/wiki/Negative-prompt Accessed: March, 2023.

[7] Omri Avrahami, Dani Lischinski, and Ohad Fried. 2022. Blended Diffusion for Text-driven Editing of Natural Images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 18187–18197. https://doi.org/10.1109/CVPR52688.2022.01767

[8] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S. Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2022. Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise. https://doi.org/10.48550/ARXIV.2208.09392

[9] Luca Benedetti, Holger Winnemöller, Massimiliano Corsini, and Roberto Scopigno. 2014. Painting with Bob: Assisted Creativity for Novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 419–428. https://doi.org/10.1145/2642918.2647415

[10] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. 2022. InstructPix2Pix: Learning to Follow Image Editing Instructions. https://doi.org/10.48550/ARXIV.2211.09800

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[12] Hsiang-Ting Chen, Li-Yi Wei, and Chun-Fa Chang. 2011. Nonlinear Revision Control for Images. *ACM Trans. Graph.* 30, 4, Article 105 (jul 2011), 10 pages. https://doi.org/10.1145/2010324.1965000

[13] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. https://doi.org/10.48550/ARXIV.2107.03374

[14] Erin Cherry and Celine Latulipe. 2014. Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Trans. Comput.-Hum. Interact.* 21, 4, Article 21 (jun 2014), 25 pages. https://doi.org/10.1145/2617588

[15] Chia-Hsing Chiu, Yuki Koyama, Yu-Chi Lai, Takeo Igarashi, and Yonghao Yue. 2020. Human-in-the-Loop Differential Subspace Search in High-Dimensional Latent Space. *ACM Trans. Graph.* 39, 4, Article 85 (aug 2020), 15 pages. https://doi.org/10.1145/3386569.3392409

[16] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[17] John Joon Young Chung, , and Eytan Adar. 2023. Artinter: AI-powered Boundary Objects for Commissioning Visual Arts. In *Designing Interactive Systems Conference* (Pittsburgh, PA) *(DIS '23)*. Association for Computing Machinery, New York, NY, USA.

[18] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. *TaleBrush: Sketching Stories with Generative Pretrained Language Models.* Association for Computing Machinery, New York, NY, USA.

[19] Colaboratory. 2023. Colaboratory. https://colab.research.google.com/ Accessed: July, 2023.

[20] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. 2022. DiffEdit: Diffusion-based semantic image editing with mask guidance. https://doi.org/10.48550/ARXIV.2210.11427

[21] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. 2022. VQGAN-CLIP: Open Domain Image Generation and Editing with Natural Language Guidance. https://doi.org/10.48550/ARXIV.2204.08583

[22] Hai Dang, Lukas Mecke, and Daniel Buschek. 2022. GANSlider: How Users Control Generative Models for Images Using Multiple Sliders with and without Feedforward Information. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 569, 15 pages. https://doi.org/10.1145/3491102.3502141

[23] Nicholas Davis, Chih-PIn Hsiao, Kunwar Yashraj Singh, Lisa Li, and Brian Magerko. 2016. Empirically Studying Participatory Sense-Making in Abstract Drawing with a Co-Creative Cognitive Agent. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (Sonoma, California, USA) *(IUI '16)*. Association for Computing Machinery, New York, NY, USA, 196–207. https://doi.org/10.1145/2856767.2856795

[24] Richard De Charms. 1970. *Personal causation: The international affective determinants of behavior.* Acad. Press.

[25] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 8780–8794. https://proceedings.neurips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf

[26] Daniel Dixon, Manoj Prasad, and Tracy Hammond. 2010. ICanDraw: Using Sketch Recognition and Corrective Feedback to Assist a User in Drawing Human Faces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 897–906. https://doi.org/10.1145/1753326.1753459

[27] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629* (2016).

[28] Elliot W Eisner. 1978. What do children learn when they paint? *Art Education* 31, 3 (1978), 6–11.

[29] Yuki Endo. 2022. User-Controllable Latent Transformer for StyleGAN Image Layout Editing. *Computer Graphics Forum* 41, 7 (2022), 395–406. https://doi.org/10.1111/cgf.14686

[30] Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. 2011. Sketch-Sketch Revolution: An Engaging Tutorial System for Guided Sketching and Application Learning. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 373–382. https://doi.org/10.1145/2047196.2047245

[31] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A Neural Algorithm of Artistic Style. *CoRR* abs/1508.06576 (2015). http://dblp.uni-trier.de/db/journals/corr/corr1508.html#GatysEB15a

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[33] Gradio. 2023. Gradio. https://gradio.app/ Accessed: July, 2023.

[34] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2022. Prompt-to-prompt image editing with cross attention control. (2022).

[35] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 6840–6851. https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[36] Jonathan Ho and Tim Salimans. 2022. Classifier-Free Diffusion Guidance. https://doi.org/10.48550/ARXIV.2207.12598

[37] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. 2023. Composer: Creative and Controllable Image Synthesis with Composable Conditions. arXiv:2302.09778 [cs.CV]

[38] H.D. Hume. 2010. *The Art Teacher's Book of Lists.* Wiley. https://books.google.com/books?id=D4GwOqhPQ88C

[39] Emmanuel Iarussi, Adrien Bousseau, and Theophanis Tsandilas. 2013. The Drawing Assistant: Automated Drawing Guidance and Feedback from Photographs. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 183–192. https://doi.org/10.1145/2501988.2501997

[40] Jennifer Jacobs, Joel Brandt, Radomír Mech, and Mitchel Resnick. 2018. Extending Manual Drawing Practices with Artist-Centric Programming Tools. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3174164

[41] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. 2021. Zero-Shot Text-Guided Object Generation with Dream Fields. (December 2021).

[42] Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4396–4405. https://doi.org/10.1109/CVPR.2019.00453

[43] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. 2022. Imagic: Text-Based Real Image Editing with Diffusion Models. https://doi.org/10.48550/ARXIV.2210.09276

[44] Kevin Gonyop Kim, Richard Lee Davis, Alessia Eletta Coppi, Alberto Cattaneo, and Pierre Dillenbourg. 2022. Mixplorer: Scaffolding Design Space Exploration through Genetic Recombination of Multiple Peoples' Designs to Support Novices' Creativity. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 308, 13 pages. https://doi.org/10.1145/3491102.3501854

[45] Tae Soo Kim, DaEun Choi, Yoonseo Choi, and Juho Kim. 2022. Stylette: Styling the Web with Natural Language. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 5, 17 pages. https://doi.org/10.1145/3491102.3501931

[46] Yuki Koyama, Issei Sato, and Masataka Goto. 2020. Sequential Gallery for Interactive Visual Design Optimization. *ACM Trans. Graph.* 39, 4, Article 88 (aug 2020), 12 pages. https://doi.org/10.1145/3386569.3392444

[47] Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. GeDi: Generative Discriminator Guided Sequence Generation. arXiv:2009.06367 [cs.CL]

[48] Gihyun Kwon and Jong Chul Ye. 2021. CLIPstyler: Image Style Transfer with a Single Text Condition. arXiv:2112.00374 [cs.CV]

[49] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. 2023. GLIGEN: Open-Set Grounded Text-to-Image Generation. https://doi.org/10.48550/ARXIV.2301.07093

[50] Jun Hao Liew, Hanshu Yan, Daquan Zhou, and Jiashi Feng. 2022. MagicMix: Semantic Mixing with Diffusion Models. *arXiv preprint arXiv:2210.16056* (2022).

[51] Alex Limpaecher, Nicolas Feltman, Adrien Treuille, and Michael Cohen. 2013. Real-Time Drawing Assistance through Crowdsourcing. *ACM Trans. Graph.* 32, 4, Article 54 (July 2013), 8 pages. https://doi.org/10.1145/2461912.2462016

[52] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. 2022. Compositional Visual Generation with Composable Diffusion Models. https://doi.org/10.48550/ARXIV.2206.01714

[53] Vivian Liu and Lydia B Chilton. 2022. Design Guidelines for Prompt Engineering Text-to-Image Generative Models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 384, 23 pages. https://doi.org/10.1145/3491102.3501825

[54] Vivian Liu, Han Qiao, and Lydia Chilton. 2022. Opal: Multimodal Image Generation for News Illustration. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 73, 17 pages. https://doi.org/10.1145/3526113.3545621

[55] Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J. Cai. 2020. Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376739

[56] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2022. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=aBsCjcPu_tE

[57] Midjourney. 2023. Midjourney. https://www.midjourney.com/ Accessed: July, 2023.

[58] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).

[59] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. 2018. I Lead, You Help but Only with Enough Details: Understanding User Experience of Co-Creation with Artificial Intelligence. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3174223

[60] D. Y. Park and K. H. Lee. 2019. Arbitrary Style Transfer With Style-Attentional Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5873–5881. https://doi.org/10.1109/CVPR.2019.00603

[61] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. 2020. Swapping Autoencoder for Deep Image Manipulation. In *Advances in Neural Information Processing Systems*.

[62] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. 2023. Zero-shot Image-to-Image Translation. https://doi.org/10.48550/ARXIV.2302.03027

[63] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv* (2022).

[64] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV]

[65] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1511.06434

[66] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. https://doi.org/10.48550/ARXIV.2204.06125

[67] Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. 2005. *Design Principles for Tools to Support Creative Thinking*. Technical Report.

[68] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs.CV]

[69] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. 2022. Palette: Image-to-Image Diffusion Models. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 15, 10 pages. https://doi.org/10.1145/3528233.3530757

[70] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. https://doi.org/10.48550/ARXIV.2205.11487

[71] Sarah Schwettmann, Evan Hernandez, David Bau, Samuel Klein, Jacob Andreas, and Antonio Torralba. 2021. Toward a Visual Concept Vocabulary for GAN Latent Space. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 6784–6792. https://doi.org/10.1109/ICCV48922.2021.00673

[72] Ticha Sethapakdi and James McCann. 2019. Painting with CATS: Camera-Aided Texture Synthesis. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3290605.3300287

[73] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. 2018. Avatar-Net: Multi-scale Zero-Shot Style Transfer by Feature Decoration. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 8242–8250.

[74] Ben Shneiderman and Pattie Maes. 1997. Direct Manipulation vs. Interface Agents. *Interactions* 4, 6 (Nov. 1997), 42–61. https://doi.org/10.1145/267505.267514

[75] Maria Shugrina, Jingwan Lu, and Stephen Diverdi. 2017. Playful Palette: An Interactive Parametric Color Mixer for Artists. *ACM Trans. Graph.* 36, 4, Article 61 (July 2017), 10 pages. https://doi.org/10.1145/3072959.3073690

[76] Maria Shugrina, Wenjia Zhang, Fanny Chevalier, Sanja Fidler, and Karan Singh. 2019. Color Builder: A Direct Manipulation Interface for Versatile Color Theme Authoring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300686

[77] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. 2022. Make-A-Video: Text-to-Video Generation without Text-Video Data. https://doi.org/10.48550/ARXIV.2209.14792

[78] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2022. Diffusion Art or Digital Forgery? Investigating Data Replication in Diffusion Models. https://doi.org/10.48550/ARXIV.2212.03860

[79] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=St1giarCHLP

[80] Stability.ai. 2023. DreamStudio. https://beta.dreamstudio.ai/ Accessed: July, 2023.

[81] Hendrik Strobelt, Albert Webson, Victor Sanh, Benjamin Hoover, Johanna Beyer, Hanspeter Pfister, and Alexander M. Rush. 2022. Interactive and Visual Prompt Engineering for Ad-hoc Task Adaptation With Large Language Models. *IEEE Transactions on Visualization and Computer Graphics* (2022), 1–11. https://doi.org/10.1109/TVCG.2022.3209479

[82] Qingkun Su, Wing Ho Andy Li, Jue Wang, and Hongbo Fu. 2014. EZ-Sketching: Three-Level Optimization for Error-Tolerant Image Tracing. *ACM Trans. Graph.* 33, 4, Article 54 (July 2014), 9 pages. https://doi.org/10.1145/2601097.2601202

[83] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. 2021. Resolution-robust Large Mask Inpainting with Fourier Convolutions. *arXiv preprint arXiv:2109.07161* (2021).

[84] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. 2022. UniTune: Text-Driven Image Editing by Fine Tuning an Image Generation Model on a Single Image. https://doi.org/10.48550/ARXIV.2210.09477

[85] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. 2022. Phenaki: Variable Length Video Generation From Open Domain Textual Description. https://doi.org/10.48550/ARXIV.2210.02399

[86] Yunlong Wang, Shuyuan Shen, and Brian Y Lim. 2023. RePrompt: Automatic Prompt Editing to Refine AI-Generative Art Towards Precise Expressions. *arXiv preprint arXiv:2302.09466* (2023).

[87] Blake Williford, Abhay Doke, Michel Pahud, Ken Hinckley, and Tracy Hammond. 2019. DrawMyPhoto: Assisting Novices in Drawing from Photographs. In *Proceedings of the 2019 on Creativity and Cognition* (San Diego, CA, USA) *(C&C '19)*. Association for Computing Machinery, New York, NY, USA, 198–209. https://doi.org/10.1145/3325480.3325507

[88] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. 2021. Wav2CLIP: Learning Robust Audio Representations From CLIP. *arXiv preprint arXiv:2110.11499* (2021).

[89] Tongshuang Wu, Michael Terry, and Carrie J Cai. 2021. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. *arXiv preprint arXiv:2110.01691* (2021).

[90] Jun Xie, Aaron Hertzmann, Wilmot Li, and Holger Winnemöller. 2014. PortraitSketch: Face Sketching Assistance for Novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 407–417. https://doi.org/10.1145/2642918.2647399

[91] Chuan Yan, John Joon Young Chung, Kiheon Yoon, Yotam Gingold, Eytan Adar, and Sungsoo Ray Hong. 2022. *FlatMagic: Improving Flat Colorization through AI-driven Design for DigitalComic Professionals*. Association for Computing Machinery, New York, NY, USA.

[92] Enhao Zhang and Nikola Banovic. 2021. 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* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 76, 15 pages. https://doi.org/10.1145/3411764.3445714

[93] Lvmin Zhang and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. https://doi.org/10.48550/ARXIV.2302.05543