

AdaptiveSliders: User-aligned Semantic Slider-based Editing of Text-to-Image Model Output

Rahul Jain*

Department of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA
jain348@purdue.edu

Koichiro Niinuma

Fujitsu Research of America
Pittsburgh, Pennsylvania, USA
kniinuma@fujitsu.com

Amit Goel

Fujitsu Consulting India
Noida, Uttar Pradesh, India
amit.goel@fujitsu.com

Aakar Gupta

Fujitsu Research of America
Redmond, Washington, USA
agupta@fujitsu.com

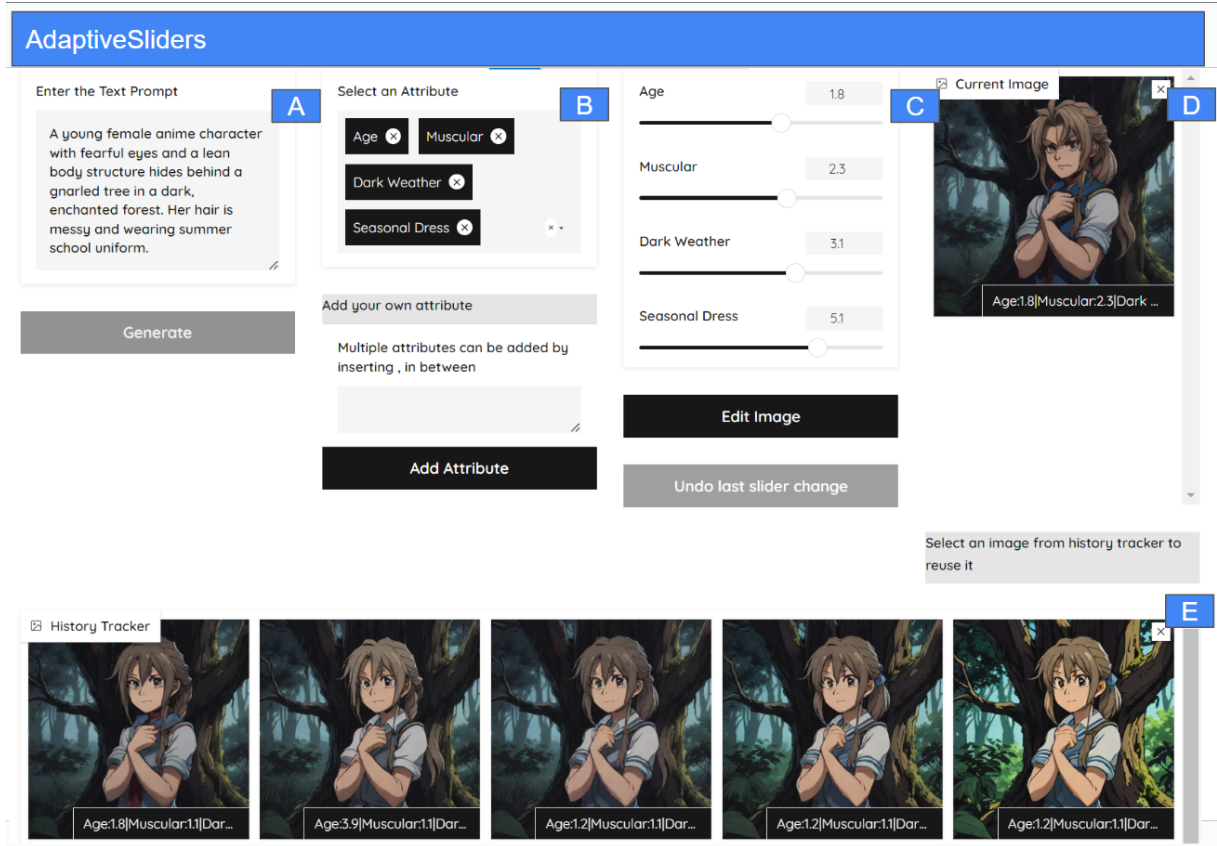


Figure 1: AdaptiveSliders User Interface: A) Text box for inputting prompts to generate images using SDXL, B) Automatic attribute suggestions, with the option to select or remove attributes freely, C) Interactive sliders with adjustable values to manipulate the latent space, D) Image box displaying the initial generated image and ongoing edits, E) History tracker to monitor user progress and changes.

*Work done during Internship at Fujitsu

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI '25, Yokohama, Japan

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1394-1/25/04

<https://doi.org/10.1145/3706598.3714292>

Abstract

Precise editing of text-to-image model outputs remains challenging. Slider-based editing is a recent approach wherein the image's semantic attributes are manipulated via sliders. However, it has significant user-centric issues. First, slider variations are often inconsistent across the sliding range. Second, the optimal slider range is unpredictable, with default values often being too large or small depending on the prompt and attribute. Third, manipulating one attribute can unintentionally alter others due to the complex entanglement of latent spaces. We introduce AdaptiveSliders, a tool that addresses these challenges by adapting to the specific attributes and prompts, generating consistent slider variations and optimal bounds while minimizing unintended changes. AdaptiveSliders also suggests initial attributes and generates initial images more aligned with prompt semantics. Through three validation studies and one end-to-end user study, we demonstrate that AdaptiveSliders significantly improves user control and experience, enabling semantic slider-based editing aligned with user needs and expectations.

CCS Concepts

• **Human-centered computing** → **Text input**; • **Computing methodologies** → **Image manipulation**.

Keywords

Generative AI, Diffusion Model, Sliders, Latent Space Interaction, Large Language Models (LLMs), Multi-Modal Models, Visual Question Answering (VQA) model

ACM Reference Format:

Rahul Jain, Amit Goel, Koichiro Niinuma, and Aakar Gupta. 2025. AdaptiveSliders: User-aligned Semantic Slider-based Editing of Text-to-Image Model Output. In *CHI Conference on Human Factors in Computing Systems (CHI '25)*, April 26–May 01, 2025, Yokohama, Japan. ACM, New York, NY, USA, 27 pages. <https://doi.org/10.1145/3706598.3714292>

1 Introduction

Text-to-Image (T2I) generative models have seen significant progress in recent years, with models like DALL-E [39], Imagen [44], Dream booth [42], and Stable Diffusion [40] demonstrating impressive capabilities in generating high-quality images. However, the outputs often struggle to fully align with human intentions [12, 47], necessitating the need for precise editing of output images at the semantic level [19].

While multiple prompt-based editing approaches have been proposed [22, 30], they are not well suited for performing nuanced modulations of semantic attributes within the image (e.g. manipulating the age of a character in the image). To solve this, a recent approach involves generating sliders for semantic attributes which enable the user to make precise adjustments to the attributes in a continuous space without altering other parts [3, 17, 18, 36]. The sliders are mapped to control different latent directions that represent different semantic attributes.

However, as we observed in our trials, there are significant user-centric challenges when we try to use them in end-applications as observed in Figure 2. First, the default slider bound values that are fixed across all images and attributes are often sub-optimal, ending up being too large or small. Small bounds might fail to

capture the range of changes desired in the attribute, while large bounds could result in unrealistic images beyond a certain value. Second, slider variations are often inconsistent across the sliding range. For equal adjustments of the slider value, the change in the corresponding attribute in the image can be a lot or little due to the complexity of the latent space. Third, manipulating one attribute can unintentionally alter others due to the complex entanglement of latent spaces [17]. These challenges pose a significant impediment to the usability of semantic editing through sliders and to their adoption in real-world applications.

In this paper, we introduce AdaptiveSliders, a slider-based semantic editing tool that addresses these challenges by adapting to the specific attributes and prompts, and aligns slider manipulations with user expectations. Upon receiving a prompt from the user, AdaptiveSliders analyzes the prompt's semantics and generates potential attribute suggestions. It then aligns the zero value of the sliders to the prompt description providing an appropriate starting point that enables maximum flexibility for the user's manipulations. For slider manipulations, it generates adaptive slider bounds so that the sliders do not go under or over an attribute's logical range of manipulation. It ensures slider variations are perceptually consistent for the user by modifying how the images along the specific semantic direction in the latent space map to the sliding range. It minimizes unintended alterations to parts of the image that are unrelated to the attribute being manipulated.

We conduct three validation experiments that evaluate the accuracy and validate the effectiveness of specific components of AdaptiveSliders. The first experiment demonstrates that the initial zero-value aligned images generated by AdaptiveSliders match the prompt more closely than the default outputs from stable diffusion. In the second and third experiments, human assessors rated the slider bounds and slider variations of AdaptiveSliders as being more preferred and convenient compared to the default baseline. Finally, we conducted a user study that demonstrates how AdaptiveSliders outperforms the baseline semantic sliders (without our adaptive components) on the task completion time, number of slider manipulations, predictable task progression, and on the subjective metrics of mental demand, effort, and frustration.

Our primary contribution in this paper is the *AdaptiveSliders tool that solves the multiple user-centric challenges* pertaining to semantic slider editing of images that relies on interacting with the latent space of the diffusion model. To the best of our knowledge, this is the first tool that attempts to solve these user issues in this context. To this end, we make multiple sub-contributions: a) The design of multiple components, each of which solves a user-centric challenge (such as sub-optimal slider bounds or inconsistent variation), and how they work together in the end-to-end system. b) The validation of the main components of the system through three experiments that demonstrate how effectively they solve the challenges. We further contribute a validation dataset which can be used for future comparative investigations into these challenges. c) The user evaluation that demonstrates the significant impact on user performance as a result of using AdaptiveSliders over a baseline system that does not solve the user-centric issues.

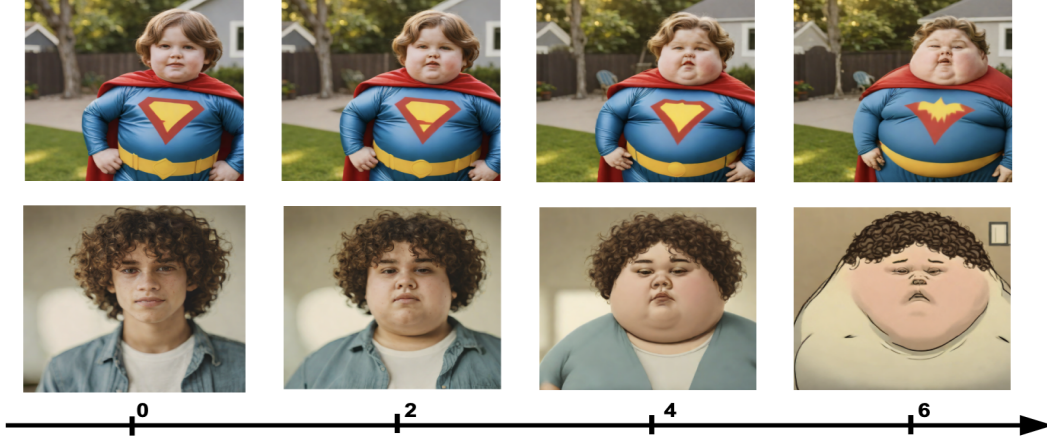


Figure 2: Illustrating user-centric challenges with semantic sliders. As the 'Chubby' attribute increases in value, it causes other unrelated attributes to change and shows images that should not be within slider bounds.

2 Related Work

2.1 Latent Space Exploration Techniques

Generative models offer unique capabilities for interacting with the latent space, enabling exploration of image attributes. They construct an entangled latent space with semantic directions corresponding to various attributes which can be manipulated. Current state of the art model suggest that generation from stable diffusion model are of high fidelity as compared to any other models [35, 39, 44]. While Diffusion model is being widely used, the semantic structure of its latent space is less explicitly defined, making it more challenging to find clear attribute-specific directions. Despite these challenges, there are still ways by which attribute editing in diffusion model can be achieved by Textual Inversion[15], Prompt Engineering[2], fine-tuning [5]. Fine tuning helps in adding new attributes to the original models[29] and finding semantic directions in the latent space[17]. Another way to add new attributes to the model is by Low Rank Adaptation [23]. LoRA helps in adding new concept without affecting the base diffusion model. Another advantage of using LoRA models is finding meaningful semantic directions in latent space of stable diffusion to achieve fine grained control[16, 17]. Concept Sliders[17] uses LoRA to find meaningful editing directions of the concepts in stable diffusion model. Once the direction is identified, α parameter is used to control the strength of edits. Specifically, α determines the extent to which the original Stable Diffusion weights (W) are modified in the identified direction (d), which corresponds to the LoRA weights. Below is the equation where W_e represents the updated weights incorporating the desired edits, W are the original Stable Diffusion weights and d is the direction (LoRA weights). We used Concept Sliders as our backend architecture providing control to the user for manipulating meaningful directions for semantically editing the image.

$$W_e = W + \alpha d \quad (1)$$

2.2 Semantic Image Editing

Interacting with the latent space and steering the model to get intended output has been explored a lot. Prompt engineering is one way to explore the images[2, 4, 53] by providing users with better support for text-based search. However, smooth control over continuous attributes is difficult by just using text [36]. Prompt based exploration are also less useful for guiding the intended output because of sensitivity of the diffusion model to prompt-seed pair. Even a slight change in the text prompt will lead to a new image. Image galleries are one way to explore the latent space images[13, 14, 48, 54]. However these approaches involve finding the relevant attribute directions in latent space and then manipulate to get various images[13, 54]. Another way to get the desired output is by providing additional context in the form of conditions to the model[34, 55]. ControlNet[55] and T2I-Adapter[34] add extra conditions in the form text, image, sketch and depth to guide the output. Alternatively, editing the output is another way to explore and get the desired output[7, 20]. Humans are likely be unsatisfied with certain aspects of the initial image generated, which they will attempt to improve over multiple iterations. Consequently, fine-grained semantic control over the generation process is useful and should be easy to use similar to initial generation. Various editing techniques like prompt-prompt[22], image inpainting[50], dragging the object[46] are some of the ways to iteratively improve the output. Method like Promptcharm[50] allow users to first generate the image and then refine using image inpainting methods. As interacting with the latent space allows users to explore and precisely edit the images, we use sliders for interacting with the latent space. AdaptiveSliders contributes to this area by offering tools supporting users to explore variations and semantically edit the image consistent with their intention in the domain of text-to-image generation.

2.3 Slider Based UIs for Semantic Editing

Slider-based UIs have become a popular choice for interacting with the latent space in generative models, as sliders effectively map to specific latent dimensions, allowing users to manipulate the strength of these dimensions numerically [9, 10, 13, 21, 37, 41]. Sliders provide users with control over the generated output by adjusting the strength of edits, enabling both global and local image modifications [9, 10, 28, 38, 41, 49]. For instance, Dang et al. [9] used sliders to globally edit image face attributes, or to fine-tune local edits after highlighting or inpainting [13, 14]. Additionally, sliders have been applied to control the model's attention to text, refining the generated content based on user inputs [50]. Furthermore, sliders are widely used for design space exploration [10]. For example, Davis et al. [10] explored fashion creativity by manipulating the latent space of GANs using sliders to traverse and experiment within the design space. In this work, we focus on using sliders as the means of semantic editing/exploration by interacting with GenAI models since sliders are very common in practical scenarios. While sliders are effective for precise editing, determining the optimal strength for accurate edits remains challenging. Due to model randomness, small adjustments may not sufficiently alter the desired attribute, while larger adjustments can lead to issues such as disentanglement, poor image quality, and even absurd outputs. Similar difficulty has been identified in other GenAI models such as GANs [25]. This difficulty arises when latent codes are pushed out of the optimal latent space. Additionally, the required strength for edits can vary depending on the input. AdaptiveSliders addresses these challenges by incorporating adaptive bounds for each attribute slider, dynamically adjusting based on the specific prompt and seed. This adaptive approach aims to improve the precision and quality of edits, ensuring that users can achieve their desired outcomes more consistently. We also evaluate the impact of these adaptive bounds by comparing them to fixed bounds by conducting user study.

3 Design Goals for a UI for Semantic Sliders

To understand the requirements for designing a slider based interactive UI which allows users to explore the latent space of the diffusion model while minimizing unintended output, we reviewed prior works which involved latent space exploration using sliders [8–10, 13, 14, 17, 25, 28, 31, 33, 48, 51]. We summarize five design goals for efficient exploration of semantic direction in diffusion model using sliders.

3.1 D1: Attribute Suggestions based on prompt

Users may have a general idea of the target image they want to achieve but may lack clarity on which attributes they can or should manipulate. Thus, there is a need for suggesting the appropriate attributes based on the user's input prompt. Another practical challenge is that semantic sliders require atleast a 30-minute pre-training for the desired attributes [17], which implies that sliders cannot be generated in real time for attributes that are not pre-trained. While the app developer can store thousands of pre-trained attributes in a library, the user may ask to manipulate an attribute that is not pre-trained (e.g. mood) even though a closely related pre-trained attribute might be present (e.g. emotion). This again

points to the need for suggesting pre-trained attributes to the user that are relevant to the user's intentions.

3.2 D2: Initial Image to Align with Slider Value Zero

The slider value for each attribute is initialized at zero for the initial image generated by the diffusion model. However, if the attribute in the initial image is misaligned with the prompt description (e.g. the image shows an obese body structure for a prompt that says 'muscular'), the slider may end up not offering enough relevant nuance on either side of zero for the user to try out different variations adjacent to their description. Since the default diffusion model outputs often contain such misalignments [31, 51], it is important to produce an initial image that aligns well with the prompt descriptions so that it can serve well as the image that aligns with slider value zero.

3.3 D3: Adaptive Mapping of Slider Bounds to Latent Space

When a user moves the slider, they navigate the latent space along a specific semantic direction. However, beyond a certain range, the points in the latent space do not correspond well to the attribute being manipulated due to entanglement with other semantic attributes. This can cause unintended or meaningless alterations to the image and can be confusing to a user. We thus require the left and right slider bounds to map to the latent space such that they cover enough range to enable a sensible exploration of the attribute in the vicinity of the initial image, without devolving into unintended or meaningless alterations (Figure 3). However, the challenge here is that this sensible mapping range in the latent space would be different for different attributes and different initial images (Figure 2). Thus, to yield sensible bounds, the mapping of the latent space to the slider bounds needs to adapt in real-time based on the initial image and the attribute.

3.4 D4: Consistent Variation upon Slider Manipulation

The latent space can be highly inconsistent leading to another problem wherein the amount of variation in the image does not map linearly to the distance moved on the slider. As Figure 6 shows, the user may see minimal changes from 0 to 3 and then suddenly see a huge change at 4. This again leads to an expectation mismatch for the user and makes it hard to predict what's going to happen in the next manipulation. It results in a more trial-and-error-behavior as opposed to a methodical navigation. Thus, our goal is to enable a more consistent image variation when a slider is manipulated.

3.5 D5: Composing Images for Multiple Attribute Changes

So far we have discussed problems pertaining to individual slider manipulation. However, in a real-world application, the user would want to manipulate multiple sliders at once and then observe their combined output together. This becomes more important because the generation of the edited image for a new slider value is not instant, taking up 6-8s. Given this latency, it makes more sense for the



Figure 3: Images representing the bounds problem for the 'chubby' slider where the left most and right most images should not be part of the bounds.

user to request the generation of a new edited image after manipulating multiple sliders instead of sending multiple requests with single edits. However, manipulating several latent attribute vectors simultaneously can lead to unintended interference between these control dimensions. While existing work has proposed solutions to minimize such interference problems [43, 57], they have not been tested in the context of semantic sliders. Our goal is to adapt these approaches for sliders and enable less noisy image compositions for multiple attribute changes.

4 AdaptiveSliders: Design and Implementation

To address the design goals identified in section 3, we developed AdaptiveSliders, a tool for enabling user-aligned editing of semantic sliders. In this section, we first describe the user interface design of our tool, then detail its software implementation, followed by the system description of how we attain the design goals.

4.1 User Interface Design

The user begins by entering a prompt in the text box, as shown in Figure 1(A). AdaptiveSliders analyzes the prompt and recommends relevant attributes for sliders, each representing a specific semantic direction (D1). Users can further add more attributes from a drop-down list which contains pre-trained attributes (Figure 1(B)). AdaptiveSliders then analyzes the attributes and produces an initial image that best aligns with the prompt Figure 1(D)(D2). This image is the user's starting point to make necessary edits with all slider values being zero. In the process of generating the sliders for each attribute, the system generates adaptive bounds for each attribute enabling a sensible range (D3), applies the consistent variation mapping for each slider that ensures smooth and predictable changes in response to slider manipulation (D4). Users can freely manipulate multiple sliders Figure 1(C)(D5) and then press 'Edit Image' to see the changed image (Figure 1(D)). The prior image gets stored and displayed in the History Tracker along with its corresponding slider values (Figure 1 (E)). Users can choose to go back to any past image and resume editing from that point.

4.2 Software Implementation

AdaptiveSliders was implemented on the Gradio platform, with all machine learning algorithms running on a server equipped with

four A100 GPUs, each with 82 GB of memory. We used the SDXL Turbo model [45] for text-to-image generation.

We trained the semantic direction for each attribute in the latent space which would then map to the sliders. For this, we used LoRA-based approach in Concept Sliders [17]. We used GPT-4 [1] to generate words that described the extremes of an attribute. For instance, to train the "age" attribute, one extreme would be "old, highly wrinkled, grey hair" and the other extreme would be "young, smooth skin, no wrinkles". Such descriptors were used for training the LoRA-based attributes, each of which took about 30 minutes.

The LoRA based edits depend on a weight value [23] where lower weights reduce the effect of LoRA and consequently the strength of the edit, and higher weights increase its effect. For application purposes, these values can be tuned by the users based on their editing needs [43]. For a LoRA-based slider, the zero value is mapped to the default SDXL output image and the selected LoRA weight range is mapped to the slider bounds. Thus, for the age attribute, the left bound would represent young, while the right bound would represent old with the intermediate values representing intermediate editing strengths. Current works [17, 27, 43] use a fixed default range of -1 to 1. This fixed range and mapping causes multiple user-centric challenges as we detail in section 3. We will now describe how our AdaptiveSliders system addresses these challenges and attains our design goals.

4.3 System Description: Attaining the Design Goals

AdaptiveSliders suggests prompt-specific attributes utilizing the Attribute Suggestion module (Section 4.3.1) with the help of large language models (LLMs). The system also provides the best aligned image with prompt (Section 4.3.2), allowing users to explore the attribute space within defined bounds (Section 4.3.3) while maintaining consistent variations (Section 4.3.4) and finally composing multiple sliders to edit multiple attributes simultaneously (Section 4.3.5)

4.3.1 Attribute Suggestion. This module retrieves contextually relevant attributes based on the user's prompt. The module consists of two steps: (1) Attribute Extractor, (2) Attribute Mapper.

- (1) **Attribute Extractor:** This step uses GPT-4 to identify attributes in the prompt that could have a continuous range of intensities that can be represented visually. For instance,

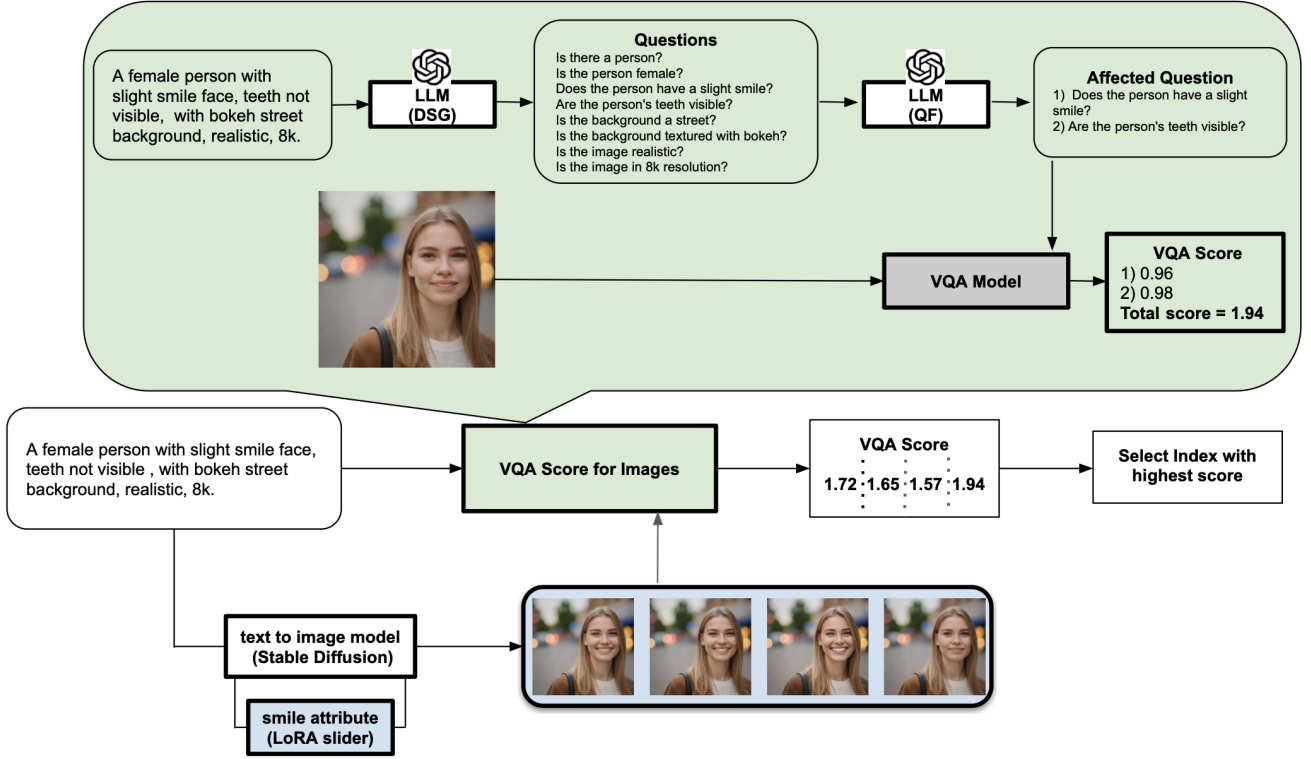


Figure 4: Workflow for Initial Image Alignment: A prompt is used to generate 16 images and parallelly is used to generate questions by DSG. Then, Question Filter(QF) LLM identifies smile-related questions, which are then evaluated by the VQA model to compute a total score for each image. The image with the highest score is selected as the initial image.

for the prompt "A muscular smiling male person in a tropical weather garden", the extracted attributes are *muscular*, *smile*, and *tropical weather*. To guide GPT-4 in identifying visual continuous attributes, we adopted a few-shot fine tuning method, providing a few examples to help it understand the idea of visual continuous attributes. Our prompt to GPT-4 encourages a step-by-step reasoning approach that ensures accurate detection and extraction of attributes within the prompt. See Figure 22 in Appendix.

- (2) **Attribute Mapper:** As mentioned earlier, any pre-trained library of attributes would be finite. Even if an application focuses on a specific domain of images (e.g. fitness) and pre-trains exhaustively for the relevant attributes, it will still not be able to train on all possible synonyms of similar concepts (e.g. thin and lean point to the same concept). Therefore, in this step, we map the extracted attributes from the previous step to their best matches in our existing pretrained library of attribute sliders. We again use GPT-4 to find this match. The mapping can be one-to-one or one-to-many, depending on the nature of the continuous attributes (e.g. grin attribute present in the prompt will be mapped to Smile Slider). To guide accurate mapping, we employed a similar few-shot learning approach, providing instructions and examples as an initial prompt for ChatGPT, as detailed in the appendix Figure 23. The module also assesses whether new sliders are

needed to better capture specific attributes, in which case, the user can choose to wait as the training of the new sliders becomes complete.

Algorithm 1 Initial Image Alignment

INPUT: prompt, attribute ▷ prompt and attribute
OUTPUT: V ▷ Image with highest Total VQA Score

- 1: questions = DSG(prompt)
- 2: affected_questions = Question Filter LLM(questions, attribute)
- 3: **for** slider_value = $-n, -n-1, \dots, -1, 0, 1, \dots, n-1, n$ **do**
- 4: $image_i = model(prompt, attribute, slider_value_i)$
- 5: ▷ Generate image for all slider value
- 6: $Total_VQA_Score_i = \sum_{q=1}^k P(Yes|image_i, affected_questions_q)$
- 7: ▷ Calculate Total VQA score for all the images
- 8: **end for**
- 9: $V = \arg_max(Total_VQA_Score)$ ▷ Select image with the highest VQA score

4.3.2 Initial Image Alignment. The default scenario is where the original SDXL image is mapped to a slider value of zero. To produce an initial image for a slider value of zero that better aligns with the prompt descriptions, AdaptiveSliders explores 16 images spanning a large, continuous latent space of the attribute, including the original default image from SDXL. It then selects the best-matching image

that is most faithful to the description of each attribute in the prompt and consequently to the overall prompt. We generate the 16 images by selecting equidistant images within the LoRA weight range of -2 to 2. In our investigations, we found that -1 to 1 often does not cover enough range of the attribute.

To find the best-match image among these, we devise a Visual Question Answering (VQA) approach. Recent large multimodal models are capable of performing the VQA task wherein the model can provide detailed answers to questions about an image [24, 31, 47]. Leveraging similar capabilities, GenAssist [26] used a VQA-based approach to summarize image descriptions. In contrast, our approach focuses on extracting probability scores from the VQA model to determine the likelihood of specific features being present in an image. For instance, instead of obtaining detailed answers, we query the model to estimate the probability that a person in the image has curly hair. This unique use of VQA probability scores forms a key component of our approach. Our approach progresses as follows: (i) We first generate the questions using Dynamic Scene Graph (DSG) [6] to provide full semantic coverage of the prompt. DSG uses ChatGPT to generate questions. (ii) We then pass these questions to the custom Question Filter (QF) LLM (see appendix Figure 24) which identifies questions influenced by the suggested attributes. For example, as shown in Figure 4, QF LLM identifies two questions that are related to the smile attribute. (iii) We then calculate the VQA score[32] for each such question for each of the 16 images.

$$P(\text{Yes}|\text{Image}, \text{Question}) \quad (2)$$

The total VQA score is calculated for each image as is shown below:

$$\text{Total_VQA_Score}_i = \sum_{q=1}^k P(\text{Yes}|\text{Image}_i, \text{Question}_q) \quad (3)$$

(iv) The image with the highest score is considered to be aligned with the prompt and becomes the initial image shown to the user with its weight value corresponding to the new slider value zero as shown in Algorithm 1. Note that if there are multiple suggested attributes, then multiple such sets of 16 images are generated corresponding to each attribute and the highest score image is found out for every attribute. A new image is generated using their corresponding weight values (see section 4.3.5) which serves as the initial image and their weight values serve as the new zero value for the respective sliders.

4.3.3 Adaptive Slider Bounds Mapping. As mentioned earlier, existing work maps a fixed LoRA weight range to slider bounds. However, this range does not work well across different attributes and initial images (Figure 3). Our solution in AdaptiveSliders avoids using single fixed range, but determines the optimal range mapping for a particular attribute and initial image scenario. To this end, we again employed a similar VQA approach as shown below in Figure 5. We use the same 16 images as before. For VQA, for a particular attribute, this time we pick all questions that do *not* pertain to that attribute. This is because to establish sensible bounds, we want to

Algorithm 2 Adaptive Slider Bounds

INPUT: prompt, attribute ▷ prompt and attribute
OUTPUT: l_bound, u_bound ▷ bounds

```

1: questions = DSG(prompt)
2: unaffected_questions = Question Filter LLM(questions, attribute)
3: for slider_value = -n, -n - 1, .. - 1, 0, 1, ..., n - 1, n do
4:   imagei = model(prompt, attribute, slider_valuei)
5:   ▷ Generate images for all slider values
6: end for
7: u_bound, l_bound ▷ Upper and Lower Bound
8: for slider_value = -n, -n - 1, .. - 1, 1, ..., n - 1, n do
9:   ▷ Check for each slider value
10:  for unaffected_questions = 0, 1, ..., q - 1, q do
11:    ▷ Check for each question
12:    vqa_score = P(yes/unaffected_questionsq, imagei)
13:    if vqa_score < 0.5 then
14:      if slider_value > 0 then
15:        u_bound = slider_value - 1
16:      else
17:        l_bound = -slider_value + 1
18:      end if
19:    end if
20:  end for
21: end for
```

find out where the images of a particular attribute start showing changes unrelated to the attribute ('entanglements'). If the VQA score for any question falls below a threshold of 0.5 for the image, that image is considered out of bounds. The bounds for an attribute in the context are thus chosen based on the lowest and highest slider values among the images that are not out of bounds as shown in Algorithm 2. These are then mapped to the slider bounds on the UI (Figure 1).

4.3.4 Consistent Slider Variation. Mapping the slider values within the range linearly to the corresponding LoRA weight range causes inconsistent variations. Instead, in AdaptiveSliders, we observe the image variations and generate a dynamic remapping specific to the attribute and the initial image (Figure 6). AdaptiveSliders first characterizes the original variations by calculating the LPIPS score at multiple values within the bounds found in Adaptive sliders bounds mapping. LPIPS score is an image similarity metric considered to be closely aligned with human perception. The score is calculated in an incremental manner where the current image's similarity is calculated relative to the previous one when progressing through images at equidistant points over the range. The LPIPS curve (Figure 7) is not linear which causes the inconsistent variation. We remap the UI slider values to the LoRA weight range such that the resultant LPIPS curve becomes linear (see Algorithm 3). Note that since LPIPS is not a perfect proxy for human visual perception, we may still see inconsistencies, however the problem is minimized to a large extent.

4.3.5 Composability for Multiple Attributes. Existing work has proposed multiple methods to combine and use multiple LoRA models at the same time, such as LoRA Switch, LoRA Compose and

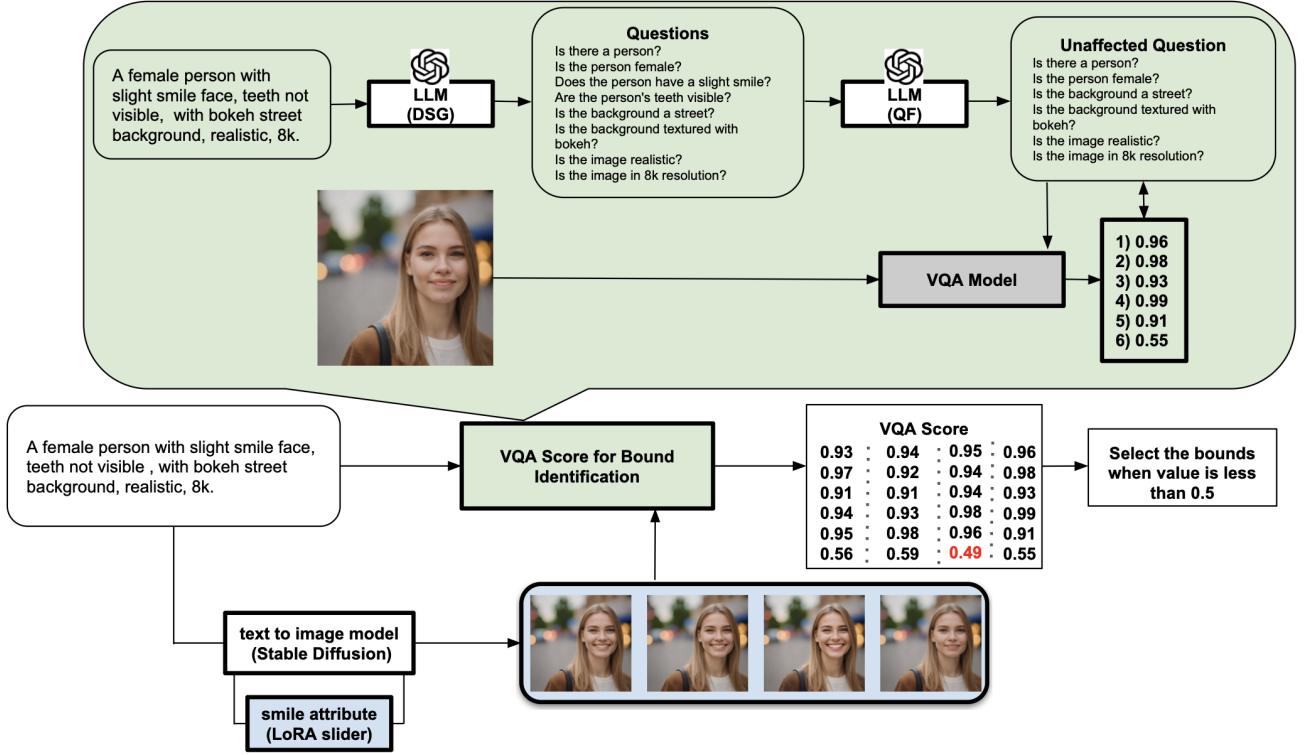


Figure 5: Workflow for Adaptive Slider Bounds: A prompt is used to generate 16 images. Parallely, prompt is used to generate questions by DSG. A prompt is used to generate 16 images and parallely is used to generate questions by DSG. Then, Question Filter(QF) LLM finds questions not related to smile. These questions are then sent to VQA for scores. The index of the image with the score < 0.5 for any question becomes out of bounds and the previous index becomes the bound.

Algorithm 3 Slider Consistency

INPUT: prompt, attribute ▷ prompt and attribute
OUTPUT: *Intervals* ▷ Unit scale for each unit increment

```

1: for slider_value = -n, -n - 1, ..., -1, 0, 1, ..., n - 1, n do
2:   imagei = model(prompt, attribute, slider_valuei)
3:   ▷ New scale
4: end for
5: for slider_value = -n, -n - 1, ..., -1, 0, 1, ..., n - 1 do
6:   lpips_scorei = LPIPS(imagei, imagei+1)
7:   ▷ Get LPIPS score for all images
8: end for
9: total_grad_positive =  $\sum_{i=1}^{n-1} lpips\_score_i$ 
10: total_grad_negative =  $\sum_{i=-n}^0 lpips\_score_i$ 
11:
12: unit_scale(i,i+1) = lpips_scorei * n / total_grad_positive
13:
14: unit_scale(-i-1,-i) = lpips_scorei * n / total_grad_negative
15: ▷ Previous unit scale changes to this new scale.

```

LoRA Merge [43, 57]. However, none of these approaches have been applied towards semantic sliders. We tried all three and found LoRA Merge to work best for composing outputs with multiple LoRA-based sliders. We incorporate this solution into AdaptiveSliders as

well as into the Baseline condition we use in our user study since the system would not be usable without it.

$$LoRA_merge = \sum_{i=1}^S LoRA_i \quad (4)$$

5 Validation Experiments

We conduct three experiments to validate the performance of our proposed approaches for 1) Initial Image Alignment, 2) Adaptive Slider Bounds Mapping, and 3) Consistent Slider Variation. We first created a dataset of 100 prompts that we subsequently used in all three validations.

5.1 Validation Dataset

We used GPT-4 to generate the 100 prompts providing our pre-trained attributes information (appendix Table 5) along with five hand-crafted example prompts to guide GPT-4 to create similar prompts that include our pre-trained attributes. Additionally, we specified the desired prompt length, ranging from small (1 attribute) to large (5 attributes). We perform first iteration asking GPT to give 20 prompts at temperature value 0. Then we vary the temperature of GPT-4 from 0.1 to 0.4 four times with 0.1 unit change to get additional 80 prompts. We further verified that these prompts did

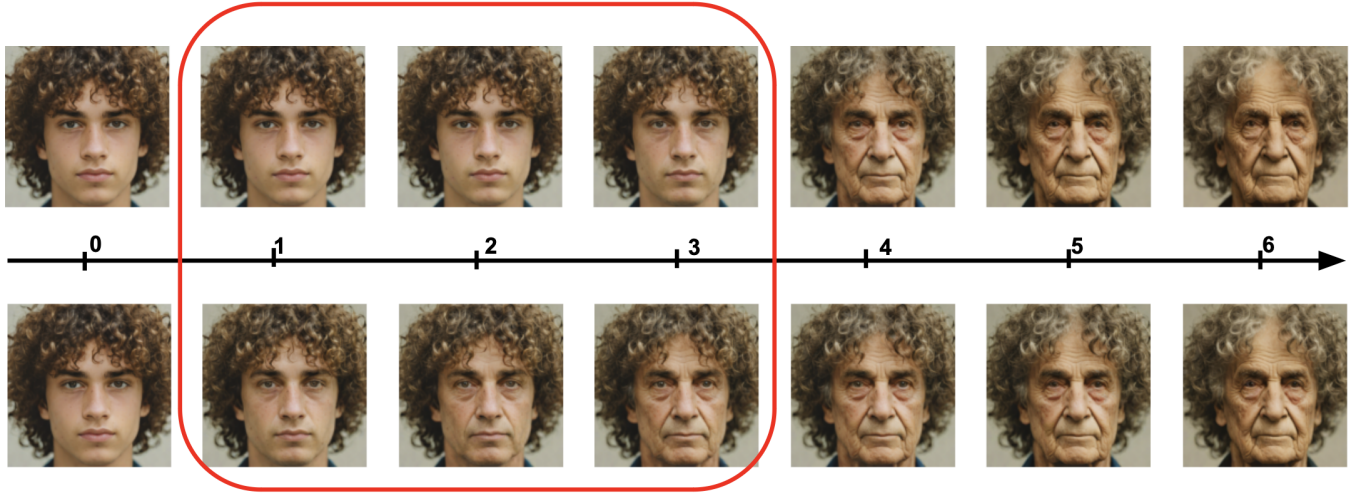


Figure 6: Slider Variation: Top is baseline which shows the inconsistent age variation where there is little age variation from 1 to 3 and suddenly a huge jump from 3 to 4. Bottom is AdaptiveSliders where the variation is more consistent.

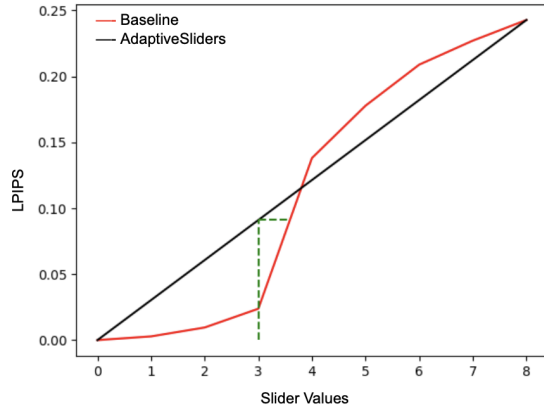


Figure 7: The Red line denotes the baseline lpips curve for Figure 6 while black curve denotes the AdaptiveSlider consistent slider variation. The green line denotes correspondence LPIPS values for slider value at 3 in both baseline and AdaptiveSliders.

not contain attributes that we did not pre-train and asked GPT-4 to modify those prompts to remove such attributes and regenerate the prompt. The regenerated prompts were again verified before being finalized.

5.2 Experiment 1: Initial Image Alignment

We evaluated how well the initial images generated by our method aligned with the prompt descriptions compared to the default image from SDXL. We use two metrics to evaluate this image-text alignment: CLIP score [39] and ImageReward [51]. While CLIP score

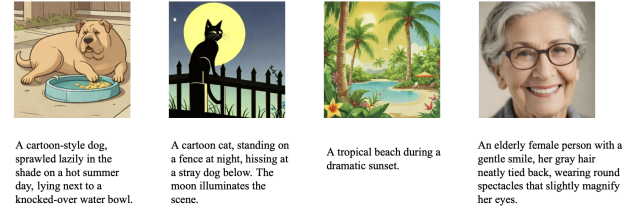


Figure 8: Data samples for evaluating our proposed approaches

is a highly popular metric, ImageReward is a recent metric that has been shown to outperform CLIP score significantly in terms of understanding human preference. CLIP scores lie between 0 and 100, the higher the better. ImageReward scores roughly follow a standard normal distribution, with a mean of 0 and a variance of 1. Scores above 0 are generally considered good, with higher positive values indicating better alignment.

We randomly sampled 100 attribute-prompt pairs from our dataset. For each sample, we retrieved two images: 1) the original image generated by SDXL by default and 2) the initial aligned image suggested by our AdaptiveSliders system.

5.2.1 Results. As Table 1 shows, AdaptiveSliders has a higher mean ImageReward score and similar CLIP scores. A Wilcoxon signed test found significant differences in the ImageReward score. This shows that AdaptiveSliders was indeed effective in finding a better prompt-aligned image ($p = 0.012$) as shown in Figure 9.

Prompt: “A female person with slight smile face, teeth not visible, with bokeh street background, realistic, 8k.”



Figure 9: Left Image is generated by SDXL and right image is generated by AdaptiveSliders by exploring the Smile attribute

5.3 Experiment 2: Adaptive Slider Bounds Mapping

We conducted a human assessment of how well AdaptiveSliders’ slider bounds compared to the fixed default bounds (LoRA weights: -1 to 1) that have been used in prior work [17, 43]. We randomly selected 100 prompt-attribute pairs from our dataset. For each prompt-attribute sample, we generated the pair of images corresponding to the AdaptiveSliders slider bounds and the pair of images corresponding to the default bounds.

Each sample was evaluated by three human assessors who were invited from the technical university where they were shown the AdaptiveSliders and baseline pairs side by side as shown in Figure 25 (Appendix). The assessors were explained the rationale behind the task and were asked to select the pair that depicts the largest sensible range for the specified attribute while other parts of the image remain unaltered. If the two pairs looked similar, the assessors were asked to select ‘Can’t Decide’.

5.3.1 Results. We analyzed the evaluation data and used majority voting to determine the final decision for each sample. Users preferred AdaptiveSliders in 71 samples, while the default method was preferred in 18 samples. For 11 samples, no preference was indicated. To assess the inter-rater agreement among the three annotators, we used Fleiss’ Kappa measure. The agreement score of 0.78 indicates substantial agreement. This demonstrates the effectiveness of AdaptiveSliders in choosing highly relevant slider bounds as shown in Figure 10.

5.4 Experiment 3: Consistent Slider Variation

We again performed a human assessment to evaluate the variations depicted by AdaptiveSliders vs. the baseline with the original variations. As done previously, we randomly selected 100 prompt-attribute samples. We then identify the adaptive bounds for each sample to avoid images with unintended alterations. We then generate two sets of 9 images within these bounds for each sample, one with the original variation and other with our AdaptiveSliders variation. The assessors were then shown the two sets and asked which

one represents a more consistent variation as shown in Figure 26 (Appendix).

5.4.1 Results. We used majority voting as before. Users preferred AdaptiveSliders in 66 samples, while only 11 samples were favored in the baseline, and 23 samples had no preference. The inter-rater agreement score was 0.71 indicating substantial agreement. This demonstrates that AdaptiveSliders is highly effective in making the image variations for slider manipulations more consistent.

6 Slider Manipulation User Study

The three validation experiments prove the individual effectiveness of our proposed components in minimizing the user-centric challenges pertaining to semantic slider based image editing. Next, we conducted a user study to evaluate how our AdaptiveSliders tool, that brings these components together, impacts user performance and experience when performing slider manipulations.

6.1 Study Design

6.1.1 Baseline. Our goal was to do an objective comparison of user speed, accuracy, and effort when performing slider manipulations with and without our proposed solutions of adaptive bounds and consistent variation. We thus designed a goal-directed task where the participants had to reconstruct a target image starting from the same initial image [9]. To our knowledge, there are no prior works solving the user-centric challenges in semantic sliders for diffusion-based images. Our work is the first to tackle these issues comprehensively. Consequently, we defined the baseline tool as one that shared the same UI as AdaptiveSliders but without the consistent variation and adaptive bounds components. This enables a fair comparison where any observed performance differences could be attributed to our AdaptiveSliders components.

The foundational works on LoRA [23] and QLoRA [11] have explored weight values from 0.25 to 2 (e.g. 0.25 denotes a range from -0.25 to +0.25). For our baseline scenario, we used a weight range of -2 to 2 (which mapped to -8 to 8 on the sliders in the UI), so that the range could fully capture the potentially large spectrum of different semantic attributes (See Figure 19 in Appendix for a distribution of slider bounds required to capture the entire range for our validation dataset). This is important because the target image that the participant needs to reach should be an achievable target in the Baseline. Since the target images are selected by random (see Table 2 for the LoRA weights of the target images), selecting a smaller range would imply that the participants would *never* be able to reach the target image in certain cases in the Baseline which would be unfair to the baseline scenario. Thus, the weight range of -2 to 2 served as the appropriate default baseline.

6.1.2 Task Design. Participants did 9 reconstruction tasks each with the AdaptiveSliders and Baseline tools. The order of tools was counterbalanced across participants.

We selected 9 prompts from our dataset for the 9 tasks. For each prompt, we used a fixed seed to ensure the same initial image across all participants. Further, as soon as participants typed in the prompt, the exact sliders needed to reach the target image were displayed along with the initial image. The original image generated by SDXL served as the starting image, corresponding to

Table 1: Metrics for Comparison. Higher the better

	Clip	ImageReward
Original	31.52 ± 1.98	0.73 ± 1.06
AdaptiveSliders	32.91 ± 1.43	0.88 ± 1.01

Table 2: The 9 tasks used in the slider manipulation user study. 9 prompts for each task are used for generating original image by stable diffusion. The table also details the attributes manipulated by users, along with the weight values applied to generate the target images. Additionally, it includes the perceptual similarity scores (LPIPS) between the original and target images.

Task	Prompt	Attribute (Weights)	Original Image	Target Image	LPIPS
1	A teenage female person in anime style, her face showing intense anger with sharply drawn eyebrows and a slight blush on her cheeks. Her hair is a vibrant pink, styled into spiky pigtails, contrasting with her dark, Gothic Lolita outfit.	Age (1.25)			0.152
2	A realistic male person wearing summer light fabrics clothes, walking on the street	Seasonal Dress (0.45)			0.095
3	A muscular male person wearing white t-shirt standing on the beach	Muscular (1.45)			0.105
4	An elderly man with a slight smile and medium length hair, in a modern office.	Age (0.75), Smile (1), Hair length (0.5)			0.108
5	A surprised young woman with straight hair, wearing a winter coat, standing in the park, in front of the house.	Surprise (0.5), Age(1.25), Curly Hair(1.75)			0.049
6	A pixar style smiling male person in tropical weather garden.	Real Person (-1.25), Smile (0.9), Tropical Weather(-1.6)			0.244
7	In a whimsical, animated park, a happy child with short, spiky hair and oversized glasses, joyfully leaps from one floating lily pad to another, beneath a candy-colored festival sky, rendered in a playful 2D style.	Age (0.75), Smile(0.75), Hair length(0.75), Weather(1), festive (0.75)			0.164
8	An anime character, a young girl with large expressive eyes filled with fear, hiding behind a tree in a dark, enchanted forest. Her hair is short and messy, in summer school uniform.	Age (1.1), Eye Size (0.75), Hair Length (0.87), Winter Dress (1.25), Dark Weather (0.26)			0.263
9	A male white fashion blogger confidently posing in a old clothes with sparse patterns, summer outfit. He is smiling and has medium length hair.	Smile (0.9), Hair Length (0.8), Seasonal Dress (1.26), Modern Dress (0.37), Pattern Frequency (1.6)			0.247

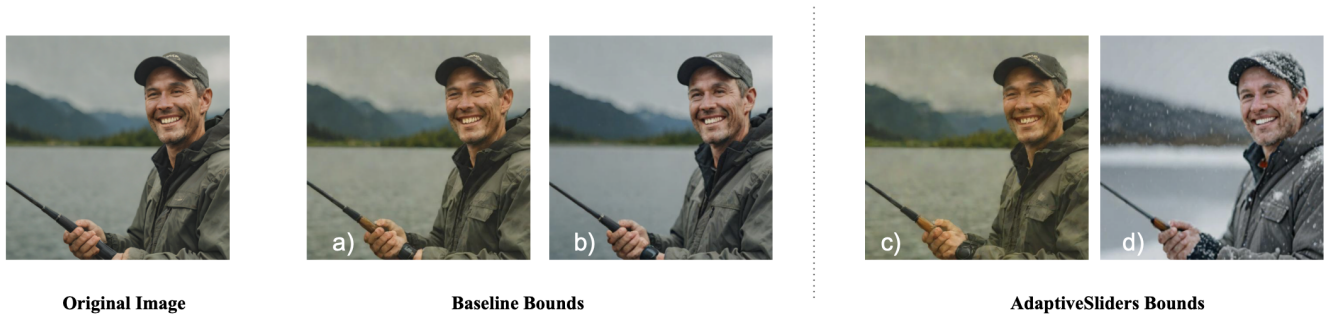


Figure 10: Left side image is original image from SDXL model for the prompt - "A man smiling contentedly while fishing at a lake, surrounded by mountains during a cloudy day". The remaining images show slider bounds for weather (a) minimum bound using baseline (b) maximum bounds using baseline (c) minimum bounds using AdaptiveSliders (d) maximum bounds using Adaptive Sliders

a slider value of 0 in both tools. Thus, for a particular prompt, all participants performed the task of slider manipulation starting from the same initial point towards the goal of reaching the same target image. To maintain consistency in the final target image generation and prevent varying task difficulty due to different images, the same 9 images were used across both tools. This is similar to the approach followed by GANSlider [9]. Since the mapping of slider values to image changes is very different for AdaptiveSliders and Baseline, the possibility of learning effects impacting the results was low. We found no order effects in our results.

The 9 tasks were split into 3 levels of complexity - requiring 1 slider, 3 sliders, and 5 sliders respectively. Please see Table 2 for the exact tasks. To generate the target images for each task, we randomly selected offsets between -2 and 2 for each slider.

6.1.3 Participants. We ran a within-subjects study with 12 participants (9 male, 3 female, age: 21-29 years), all of whom had some experience with image editing tools. 6 participants had prior experience with text-to-image tools like DALL-E and Stable Diffusion. The overall design was 12 participants x 2 Tools x 9 Tasks (3 task Complexities x 3 tasks).

6.2 Procedure

Participants were given an initial introduction to the study. Since the interface was the same across both tools, participants were provided with a brief tutorial and a period of 15 mins to familiarize themselves with the interface. There was no time limit imposed during the actual tasks, and the users were instructed to end the task whenever they felt they had successfully reached the target image. After completing all tasks with one tool, participants were asked to fill out a 7-point Likert scale questionnaire and the NASA-TLX survey. Upon completing the tasks with both tools, we conducted a 20-minute post-session interview to gather subjective feedback and gain further insights. The entire study lasted between 90 and 120 minutes.

6.3 Measures

In addition to subjective measures, we analyzed the objective measures of task completion time, task completion accuracy, number

of slider interactions, and task completion progress. For accuracy, we used the LPIPS score to measure the distance from the target image. LPIPS [56] is an image similarity metric that is considered to align well with human perception.

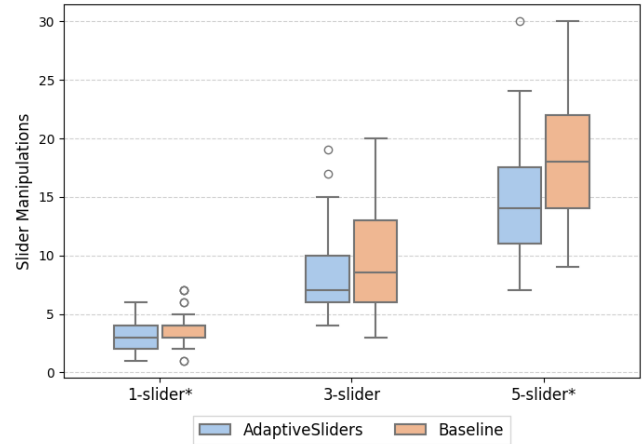


Figure 11: Slider Manipulations Comparison (*) = ($p < 0.05$)

7 Results

7.1 Task Completion Time

We conducted separate analyses for each of the task complexities. The Shapiro-Wilk tests showed that normality cannot be assumed. We therefore performed Wilcoxon signed-rank tests to analyze the effect of Tool on completion time. As Figure 12 shows, completion time was significantly lower with AdaptiveSliders (AS) compared to Baseline (B) for all 3 task complexities, 1-slider ($p = 0.006$, $Z = -2.686$, $M_{AS} = 37.74$, $SD_{AS} = 13.46$, $M_B = 55.13$, $SD_B = 25.33$), 3-slider ($p = 0.028$, $Z = -2.190$, $M_{AS} = 94.41$, $SD_{AS} = 39.92$, $M_B = 121.71$, $SD_B = 68.99$), 5-slider ($p = 0.035$, $Z = -2.105$, $M_{AS} = 213.00$, $SD_{AS} = 84.07$, $M_B = 260.93$, $SD_B = 84.44$).

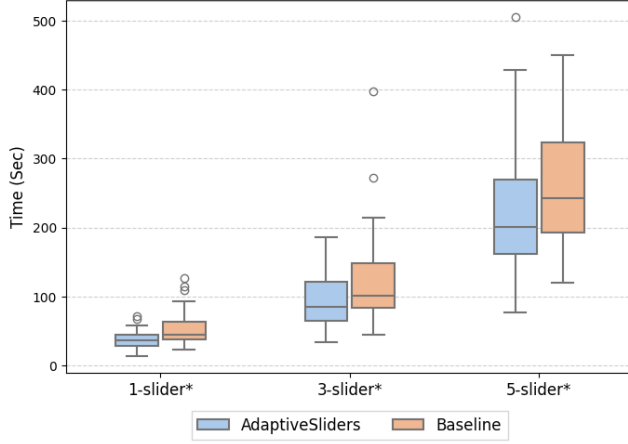


Figure 12: Time Performance Comparison (*) = ($p < 0.05$)

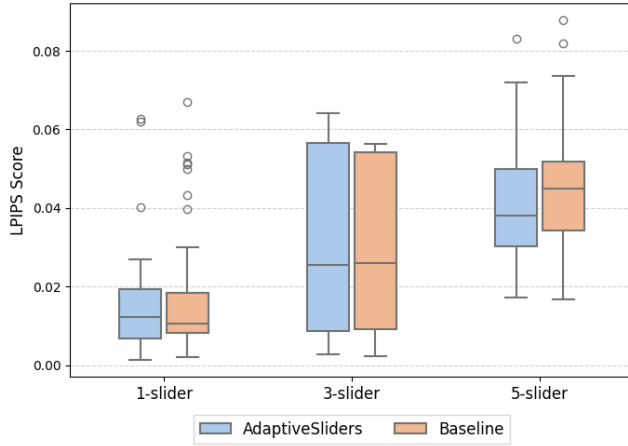


Figure 13: Final LPIPS Score Comparison. Lower the better

Table 3: Metrics: LPIPS score. lower the better

		LPIPS
1-slider	AdaptiveSliders	0.015 \pm 0.014
	Baseline	0.018 \pm 0.017
3-slider	AdaptiveSliders	0.030 \pm 0.022
	Baseline	0.029 \pm 0.020
5-slider	AdaptiveSliders	0.040 \pm 0.015
	Baseline	0.046 \pm 0.021

7.2 Task Completion Accuracy

We did not find any significant differences in the LPIPS score for AdaptiveSliders vs Baseline for the three task complexities (Figure 13). Note that a lower LPIPS score implies a better match with the target image. We did find a significant effect of task complexity on the overall LPIPS score which indicates that as the sliders increased, participants found it harder to reach the target image.

7.3 Task Completion Progress

To track how consistently the participants were able to progress towards the target image, we looked at the LPIPS score of the intermediate images generated by the participants as they progressed in the task. As Figure 14 shows, AdaptiveSliders appears to have a more consistent decrease in scores over time compared to the Baseline. In a few cases, the baseline shows an initial upward variation, notably in Tasks 2 and 7. Upon analyzing individual logs, we found instances of users employing different initial approaches. One approach is when users engaged in exploratory behaviors where they initially moved the sliders to extreme positions to understand the available range. For the Baseline, this can result in absurd images at the extremes, causing the sharp upward variations in LPIPS. Figure 20 shows such an instance for Task 7. Such variations are much lower for AdaptiveSliders due to the adaptive bounds. Another approach that users tried initially is to guess the target slider values directly without exploring the extremes. Figure 21 depicts this behavior for Task 2 where the participant's initial guesses deviated quite a bit more from the target image in the Baseline compared to AdaptiveSliders since the variation in AdaptiveSliders is more predictable.

7.4 Slider Manipulations

We analyzed the number of times in a task participants manipulated the sliders by clicking or dragging. The Wilcoxon signed-rank test showed that participants performed a significantly lower number of slider manipulations in AdaptiveSliders than in the Baseline for the 1-slider ($M_{AS} : 3.0, SD_{AS} : 1.26, p = 0.028, Z = -2.195, M_B = 3.75, SD_B = 1.5$) and 5-slider ($M_{AS} : 15.22, SD_{AS} : 6.26, M_B : 18.51, SD_B : 5.94, p = 0.018, Z = -2.352$) task complexities. No significant differences were found for the 3-slider tasks ($M_{AS} : 8.47, SD_{AS} : 3.54, p = 0.375, Z = -0.886, M_B = 9.5, SD_B = 4.22$) (Figure 11).

7.5 Cognitive Load

We conducted a Wilcoxon signed-rank test to analyze the effect of AdaptiveSliders on the NASA-TLX scores. As shown in Figure 15, participants reported AdaptiveSliders to have significantly lower mental demand ($p = 0.006, Z = -2.738$) and effort ($p = 0.006, Z = -2.724$). Participants echoed this sentiment in their interviews with P4 stating "first one is more easy to adjust the image". Participants reported significantly higher frustration with Baseline ($p = 0.038, Z = -2.071$). P12: "sometime the images generated suddenly are so bad and it took me for a while to think which made me kinda demotivated". Further, participants felt more successful in accomplishing the task with AdaptiveSliders ($p = 0.037, Z = -2.081$), suggesting that the tool is better aligned with their intentions and helped them achieve their desired output better. P3: "the first interface (AdaptiveSliders) is much fast and I think I completed all the task in less time to the second one".

7.6 Subjective Scores on Specific Features

We collected subjective scores on the effectiveness of the UI features as shown in Figure 16. Overall, the user felt the images generated were consistent and there were significantly less sudden changes(Q1). P7: "second system is more precise when I change the

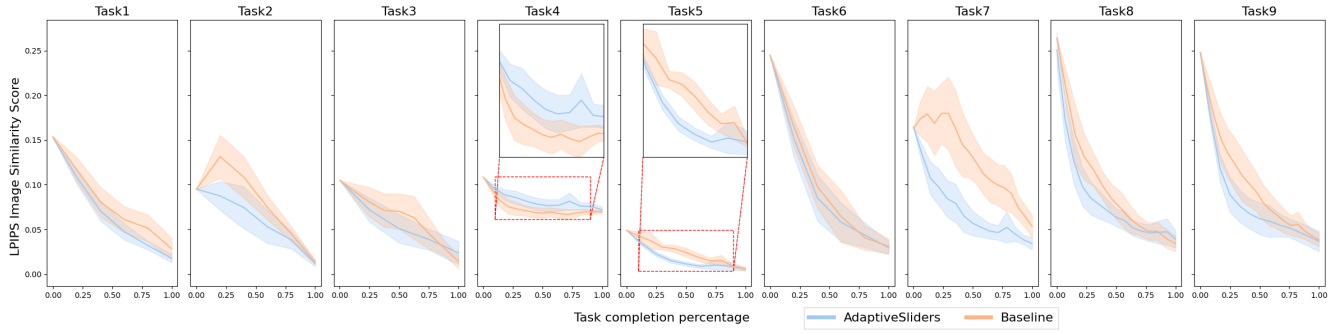


Figure 14: Task Completion Progress. X-axis denotes the normalized task duration. Y-axis denotes the curves obtained by extrapolating the points depicting the mean LPIPS scores of the intermediate images. Solid lines denote means, shaded regions denote standard deviation. Task 4 and Task 5 also includes magnified image for showing enlarged view.

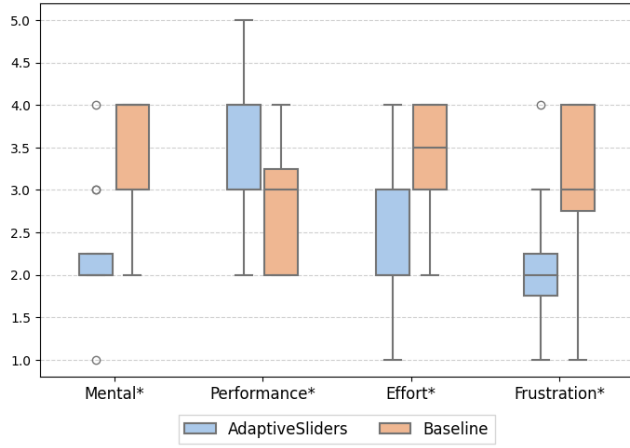


Figure 15: NASA-TLX Scores. Note, for Performance, Higher is better (*) = ($p < 0.05$)

slider to change the attributes and the change is linear". It was easy to determine how far to move the sliders to achieve the desired image (Q4). P4: "When I saw the slider values in the history then it was matching my expectation with the image in first (AdaptiveSliders). Then I guess the number to reach the target. In second I had little clue and was doing hit and trial." The user found that the baseline UI images were generating absurd images at the extremes(Q2). P11: "Intuitively I like to move slider to the extreme to identify the range but it was pretty bad experience like winter case (winter slider) where the human disappear) but second UI it (human in the image) was good and understandable." AdaptiveSliders was generating images specifically to other and restricting other parts to change as compared to baseline (Q3). P2: "In the first system(baseline) changing one attribute unexpected inference on other. Attributes are more independent in the second system". However, two users pointed out that there is some disentanglement in the semantic direction in AdaptiveSliders which we discuss in section 8.2.

8 Discussion

In this section, we discuss the results of our study, offering key insights and implications. Additionally, we discuss potential opportunities for future research.

8.1 Results Discussion: Impact of Consistent Variation and Adaptive Slider Bounds

Our user study results indicate that AdaptiveSliders was successful in reducing the total time taken and slider manipulations compared to the Baseline which did not have consistent variation and adaptive slider bounds. As we can see in the mean values, the difference between AdaptiveSliders and Baseline is highest in the 5-slider case indicating that with the higher complexity of managing multiple sliders, the impact of AdaptiveSliders becomes more evident. Further, participants reported lower mental demand, effort, and frustration with AdaptiveSliders. Participants also felt more confident about their performance with AdaptiveSliders owing to consistent variation and adaptive bounds enabling manipulations that matched their expectations.

8.1.1 Interaction Progress and Overshooting. AdaptiveSliders effectively reduces the sharp initial overshooting observed in the Baseline, which occurred due to both exploratory behavior and initial guessing strategies. Similar exploratory behavior has been noted in previous studies involving slider-based user interfaces[9]. In the Baseline condition, users struggled to understand how the sliders affected the image, especially since they could move them to extreme values, leading to sudden and unexpected changes. AdaptiveSliders helped by dynamically adjusting the slider limits, allowing users to explore changes more smoothly without absurd image generation. The system maintains interaction consistency by restricting the bounds and consistent variations together ensuring predictability and overall interaction quality. By limiting excessive deviations and providing a structured editing experience, AdaptiveSliders improves both usability and precision in slider-based image manipulation tasks.

8.1.2 Degree of Manipulation and Performance. In addition to the number of sliders manipulated, the degree of manipulation determined by the perceptual difference (LPIPS) between the source and

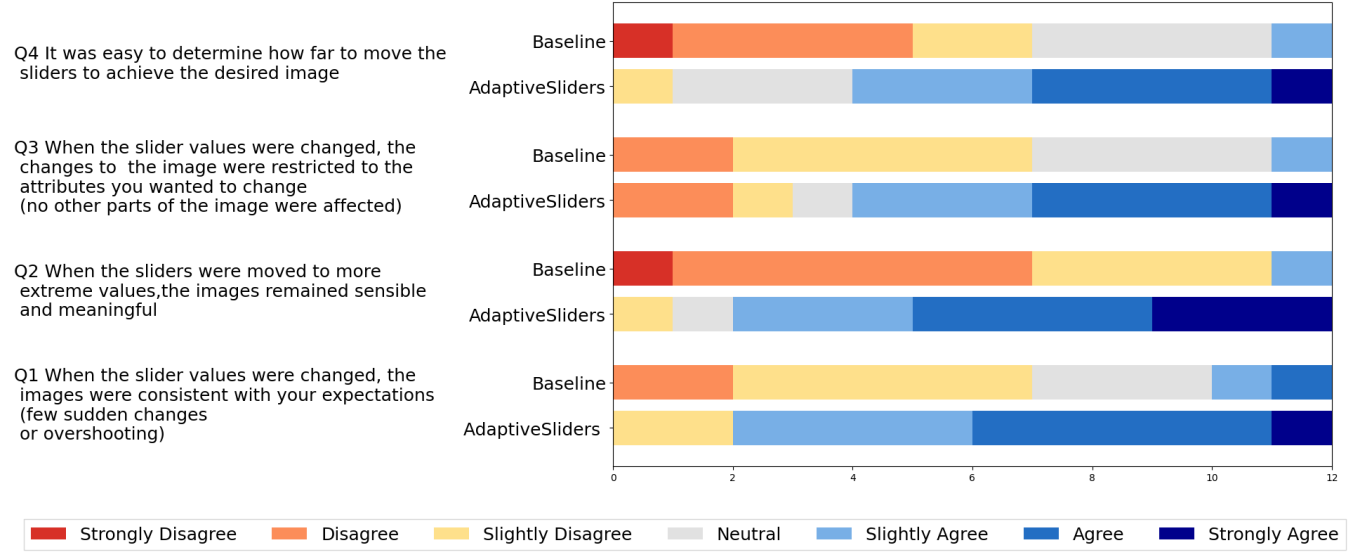


Figure 16: 7-point Likert scale questionnaire

target image may also affect task difficulty. An analysis of LPIPS scores, time taken, and slider manipulations (Figure 17 and Figure 18) revealed that while there were no significant differences for tasks with the same slider count, tasks with lower LPIPS scores (Tasks 2, 5, and 7) tended to have a slightly higher number of manipulations. Additionally, the performance gap between the progression of AdaptiveSliders and Baseline as shown in Figure 14 is biggest for these three tasks. We may therefore speculate that the finer adjustments required initially by low LPIPS scores make it more difficult for the user, especially in the Baseline condition increasing the risk of overshooting and necessitating re-edits. The issue with lower LPIPS scores seems to be present, but appears less pronounced in AdaptiveSliders where the consistent variation and adaptive bounds allow users to make a more steady, incremental progress. However, a deeper study of usage behavior is needed to conclusively establish the impact of the degree of manipulation. Overall, our findings suggest that AdaptiveSliders consistently outperformed the Baseline across all slider counts, regardless of task difficulty. The advantage was particularly pronounced in tasks with lower LPIPS scores, where users in the Baseline struggled with fine-tuned adjustments. This suggests that adaptive constraints play a crucial role in improving usability in slider-based interfaces, particularly for precision-based tasks.

8.2 Entanglement and Effect of Editing Multiple Attributes

While our approach tries to minimize the entanglement problem by applying slider bounds beyond which other attributes start to get impacted, this is a complex problem for editing T2I model images. This becomes even more of an issue when dealing with multiple attribute-edits at the same time. In Table 3, we observe an increase in the distance between the target image and the final user-generated image as the number of sliders increases, both in the baseline and AdaptiveSliders conditions. Based on our study

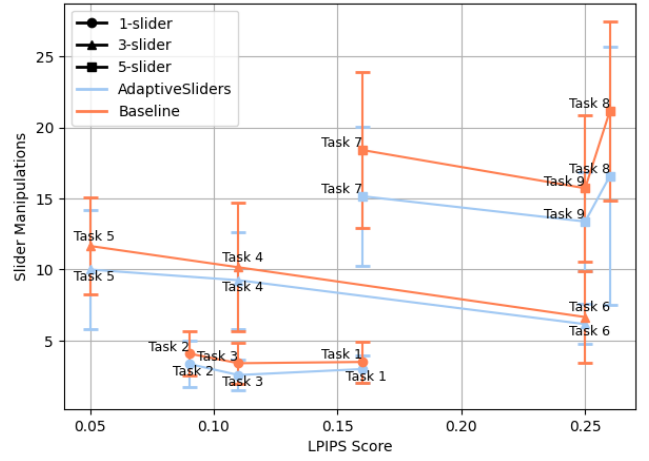


Figure 17: Slider Manipulations Comparison with LPIPS

results and user feedback, users reported feeling overwhelmed as more sliders were introduced. A key observation is that while slider adjustments often align with the prompt, there is still some degree of disentanglement between the latent space directions. This issue becomes more pronounced as the number of sliders increases. Despite using a state-of-the-art model for composing the sliders, we still observed this disentanglement, which in turn made it more difficult for users to achieve their final goal. The similar findings were found in previous works [9, 41], where it was found that too many adjustable dimensions can overwhelm users. Dang et al.[9] specifically suggest that the optimal range is between 5 to 10 sliders. There is a growing body of work aimed at improving the composability of semantic directions in LoRA[52, 57], and we believe that with further research, composing multiple directions

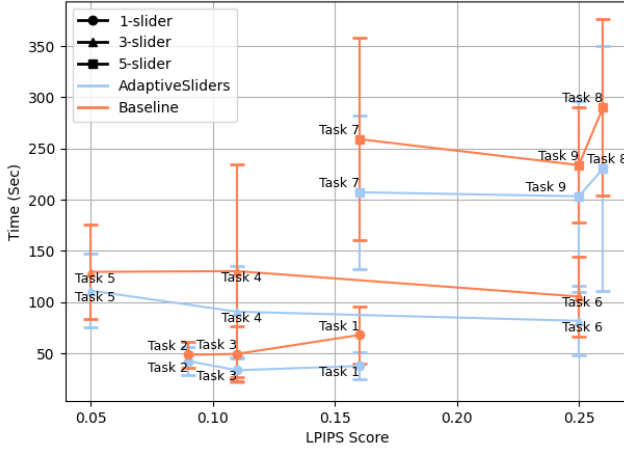


Figure 18: Time Performance Comparison with LPIPS

could become much more accurate, reducing the difficulty and cognitive load on users.

8.3 Processing Time and Computation Cost

AdaptiveSliders includes multiple large machine learning components such as Large Language Models (LLMs) [1], Vision-Language Models (VLMs) [32], and Stable Diffusion [45] for generating and manipulating images based on user input. While these models provide high-quality results, their computational demands pose challenges in enabling real time performance. We parallelized our code in multiple places to minimize this latency.

8.3.1 Latency in Image Rendering. However, the LoRA-based image generation for a new slider value still had a rendering latency which is a result of several factors: 1) SDXL Model Size: The stable diffusion model (SDXL) is large, requiring significant computational resources. 2) Multiple Sliders: Composing multiple sliders adds complexity to the generation process. 3) Image Resolution: Generating 512x512 pixel images demands more processing power. The models inherently face a trade-off between speed and accuracy. Real-time image generation is possible, but it often comes at the cost of reduced image quality. Alternative approaches to mitigate latency includes solutions such as reducing image resolution (smaller images, e.g. for preview purposes) and utilizing multiple higher-performance GPUs along with parallelization across those. With a growing interest in the computer vision community for making these models lightweight and real-time, we anticipate future models will enable real-time rendering without significant quality compromises or compute costs.

8.4 Baseline Slider Bounds

In our baseline condition, we used the slider range of -2 to 2. As we explain in section 6.1.1, this decision was motivated by the fact that our dataset of 100 image-attribute pairs showed that the sensible bounds for most image-attribute pairs were within the -2, 2 range and the -1, 1 range often failed to fully capture the spectrum of a semantic attribute (Appendix A). Selecting a -1, 1 range for the

baseline would have meant that we could only consider target images that were also within this range since we want the participants to be able to reach the target images successfully in the Baseline as well. While this may have resulted in participants seeing less absurd images in the Baseline, the task would have been severely constrained with regards to the richness and range of the attributes and a smaller LPIPS range. Further, this would not be representative of a real-world scenario where despite the possibility of absurd images, one would want the user to have access to the whole spectrum if the fixed-range Baseline system is deployed. We therefore considered the -2,2 range as more representative of a real-world Baseline. While it may be worth investigating a smaller Baseline range in the future, so as to observe how the user performance varies as a result, in the end, the problem lies with the fixed nature of the range in the Baseline and so any choice of range will have imperfections.

8.5 Integration with Existing Methods

8.5.1 Integrating with Editing Tools. Multiple editing techniques can enhance the overall quality of output generated by text to image generation model[50]. Slider-based editing is a technique for precise editing of continuous semantic attributes [7, 9]. It would be highly valuable to explore integration of slider-based editing with other complementary techniques, such as prompt-based editing, inpainting, and outpainting. Combining these methods could provide users with a more flexible and comprehensive editing tool for generating desired image.

8.5.2 Generalization of our Workflow to other T2I Models. The proposed workflow in the paper primarily focuses on working with diffusion models and utilizes input prompts provided by users to generate and edit images via slider-based control of continuous semantic attributes. While this approach is effective in manipulating the latent space of diffusion models, it is essential to test the workflow’s versatility and applicability with other text to image generative models such as GANs (Generative Adversarial Networks) and VAEs (Variational Autoencoders). Testing with these models would help determine the generalizability of the proposed slider-based editing framework across different types of generative models, each with its own unique latent space structure and generation mechanisms.

9 Conclusion

In this work, we presented AdaptiveSliders, a novel tool for editing of semantic attributes in text-to-image generation models. AdaptiveSliders suggests relevant attributes for sliders based on the user prompt. It uses a VQA-based approach to yield an initial image that aligns better with the prompt in order to serve as the image at slider value zero. It further uses VQA to provide adaptive slider bounds that enable a sensible slider range for attribute exploration that is not too big or too small. It uses an LPIPS-based approach to improve consistency in how the images vary across the sliding range of an attribute. We performed three validation experiments that showed AdaptiveSliders clearly improved initial image alignment, slider bounds, and consistent variation. Our user study demonstrated that AdaptiveSliders significantly improves user efficiency and effort for goal-based editing. We anticipate that our slider based interactive

tool will open up novel opportunities in creating efficient tools in creative image generation, and has applications across various domains such as graphic design, AI-assisted creativity, and beyond.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
- [2] Shm Garanganao Almeda, JD Zamfirescu-Pereira, Kyu Won Kim, Pradeep Mani Rathnam, and Bjoern Hartmann. 2024. Prompting for Discovery: Flexible Sense-Making for AI Art-Making with DreamSheets. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–17.
- [3] Francesco Bodria, Salvatore Rinzivillo, Daniele Fadda, Riccardo Guidotti, Fosca Giannotti, and Dino Pedreschi. 2022. Explaining Black Box with Visual Exploration of Latent Space. In *EuroVis (Short Papers)*. 85–89.
- [4] Stephen Brade, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. 2023. Promptify: Text-to-image generation through interactive prompt exploration with large language models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–14.
- [5] Pu Cao, Lu Yang, Feng Zhou, Tianrui Huang, and Qing Song. 2023. Concept-centric personalization with large-scale diffusion priors. *arXiv preprint arXiv:2312.08195* (2023).
- [6] Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. 2023. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-image generation. *arXiv preprint arXiv:2310.18235* (2023).
- [7] John Joon Young Chung and Eytan Adar. 2023. Promptpaint: Steering text-to-image generation through paint medium-like interactions. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–17.
- [8] Yusuf Dalva, Hidir Yesiltepe, and Pinar Yanardag. 2024. GANTASTIC: GAN-based Transfer of Interpretable Directions for Disentangled Image Editing in Text-to-Image Diffusion Models. *arXiv preprint arXiv:2403.19645* (2024).
- [9] 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*. 1–15.
- [10] Richard Lee Davis, Thiemo Wambgsanss, Wei Jiang, Kevin Gonyop Kim, Tanja Käser, and Pierre Dillenbourg. 2024. Fashioning Creative Expertise with Generative AI: Graphical Interfaces for Design Space Exploration Better Support Ideation Than Text Prompts. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–26.
- [11] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: efficient finetuning of quantized LLMs (2023). *arXiv preprint arXiv:2305.14314* 52 (2023), 3982–3992.
- [12] Omar Elharrouss, Noor Almaadeed, Somaya Al-Maadeed, and Younes Akbari. 2020. Image inpainting: A review. *Neural Processing Letters* 51 (2020), 2007–2028.
- [13] Noyan Evirgen and Xiang'Anthony' Chen. 2022. Ganzilla: User-driven direction discovery in generative adversarial networks. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–10.
- [14] Noyan Evirgen and Xiang'Anthony' Chen. 2023. Ganravel: User-driven direction disentanglement in generative adversarial networks. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–15.
- [15] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618* (2022).
- [16] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. 2023. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2426–2436.
- [17] Rohit Gandikota, Joanna Materzynska, Tingrui Zhou, Antonio Torralba, and David Bau. 2023. Concept sliders: Lora adaptors for precise control in diffusion models. *arXiv preprint arXiv:2311.12092* (2023).
- [18] Julia Guerrero-Viu, Milos Hasan, Arthur Roullier, Midhun Harikumar, Yiwei Hu, Paul Guerrero, Diego Gutierrez, Belen Masia, and Valentin Deschaintre. 2024. Textsliders: Diffusion-based texture editing in clip space. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.
- [19] Jiayi Guo, Xingqian Xu, Yifan Pu, Zanlin Ni, Chaofei Wang, Manushree Vasu, Shiji Song, Gao Huang, and Humphrey Shi. 2024. Smooth diffusion: Crafting smooth latent spaces in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7548–7558.
- [20] Qin Guo and Tianwei Lin. 2024. Focus on your instruction: Fine-grained and multi-instruction image editing by attention modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6986–6996.
- [21] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. Ganspace: Discovering interpretable gan controls. *Advances in neural information processing systems* 33 (2020), 9841–9850.
- [22] 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). URL <https://arxiv.org/abs/2208.01626> (2022).
- [23] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).
- [24] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. 2023. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 20406–20417.
- [25] Zhizhong Huang, Siteng Ma, Junping Zhang, and Hongming Shan. 2023. Adaptive nonlinear latent transformation for conditional face editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 21022–21031.
- [26] Mina Huh, Yi-Hao Peng, and Amy Pavel. 2023. GenAssist: Making image generation accessible. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–17.
- [27] Maxwell Jones, Sheng-Yu Wang, Nupur Kumari, David Bau, and Jun-Yan Zhu. 2024. Customizing Text-to-Image Models with a Single Image Pair. *arXiv preprint arXiv:2405.01536* (2024).
- [28] Hyung-Kwon Ko, Subin An, Gwanmo Park, Seung Kwon Kim, Daesik Kim, Bohyoung Kim, Jaemin Jo, and Jinwook Seo. 2022. We-toon: A Communication Support System between Writers and Artists in Collaborative Webtoon Sketch Revision. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–14.
- [29] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. 2023. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1931–1941.
- [30] Dongxu Li, Junning Li, and Steven Hoi. 2024. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *Advances in Neural Information Processing Systems* 36 (2024).
- [31] Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, et al. 2024. Rich human feedback for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19401–19411.
- [32] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. 2024. Evaluating text-to-visual generation with image-to-text generation. *arXiv preprint arXiv:2404.01291* (2024).
- [33] 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*. 1–23.
- [34] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. 2024. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 4296–4304.
- [35] 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).
- [36] Rishubh Parihar, Sachidanand VS, Sabariswaran Mani, Tejan Karmali, and R Venkatesh Babu. 2024. PreciseControl: Enhancing Text-To-Image Diffusion Models with Fine-Grained Attribute Control. *arXiv preprint arXiv:2408.05083* (2024).
- [37] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. 2020. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems* 33 (2020), 7198–7211.
- [38] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2085–2094.
- [39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.
- [40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
- [41] Andrew Ross, Nina Chen, Elisa Zhao Hang, Elena L Glassman, and Finale Doshi-Velez. 2021. Evaluating the interpretability of generative models by interactive reconstruction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.
- [42] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 22500–22510.
- [43] Ryu, S. 2023. Merging loras. <https://github.com/cloudfans/simolora>.
- [44] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep

- language understanding. *Advances in neural information processing systems* 35 (2022), 36479–36494.
- [45] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. 2023. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042* (2023).
 - [46] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. 2024. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8839–8849.
 - [47] Jiao Sun, Deqing Fu, Yushi Hu, Su Wang, Royi Rassin, Da-Cheng Juan, Dana Alon, Charles Herrmann, Sjoerd van Steenkiste, Ranjay Krishna, et al. 2023. Dreamsync: Aligning text-to-image generation with image understanding feedback. In *Synthetic Data for Computer Vision Workshop@ CVPR 2024*.
 - [48] Qian Wan and Zhicong Lu. 2023. Gancollage: A gan-driven digital mood board to facilitate ideation in creativity support. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. 136–146.
 - [49] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. 2022. High-fidelity gan inversion for image attribute editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11379–11388.
 - [50] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. 2024. PromptCharm: Text-to-Image Generation through Multi-modal Prompting and Refinement. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–21.
 - [51] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. 2024. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems* 36 (2024).
 - [52] Yang Yang, Wen Wang, Liang Peng, Chaotian Song, Yao Chen, Hengjia Li, Xi-aolong Yang, Qinglin Lu, Deng Cai, Boxi Wu, et al. 2024. LoRA-Composer: Leveraging Low-Rank Adaptation for Multi-Concept Customization in Training-Free Diffusion Models. *arXiv preprint arXiv:2403.11627* (2024).
 - [53] Xingchen Zeng, Ziyao Gao, Yilin Ye, and Wei Zeng. 2024. IntentTuner: An Interactive Framework for Integrating Human Intentions in Fine-tuning Text-to-Image Generative Models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–18.
 - [54] 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*. 1–15.
 - [55] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.
 - [56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.
 - [57] Ming Zhong, Yelong Shen, Shuohang Wang, Yadong Lu, Yizhu Jiao, Siru Ouyang, Donghan Yu, Jiawei Han, and Weizhu Chen. 2024. Multi-lora composition for image generation. *arXiv preprint arXiv:2402.16843* (2024).

A Slider Bound Analysis

The distribution of slider bounds as shown in Figure 19, provides key insights into the range of values required to capture the semantic variations. We analyze 100 prompt-attribute pair using AdaptiveSliders bound method. The graph highlights a significant concentration of bounds from middle to end for both negative and positive values. To optimize usability and exploration of the design space, selecting a baseline range of -2 to 2 mapped to -8 to 8 is a practical and efficient choice. It captures broader spectrum allowing user to explore semantic attribute space.

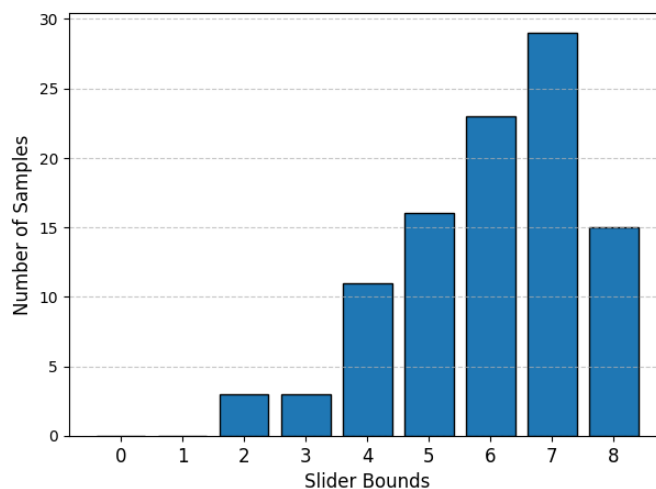


Figure 19: Distribution of samples across different slider bounds from AdaptiveSliders. The x-axis represents the slider bounds and y-axis indicates the number of samples corresponding to each bound. More than 80 samples are beyond 4 (weight 1).

B Slider bounds using AdaptiveSliders

Table 4: Slider bounds by AdaptiveSliders for the attributes used in the study for all 9 tasks. The last column describes the attributes used in the study along with slider bounds from AdaptiveSliders.

Task	Initial Image	Target Image	Attributes (Bounds)
1			Age [-8, 7]
2			Seasonal Dress [-4, 7]
3			Muscular [-8,7]
4			Age [-4, 5], Smile [-8,8], Hair length [-8,7]
5			Surprise [-8,8], Age[-8,7], Curly Hair[-6,8]
6			Real Person [-7,8], Smile [-8,7], Tropical Weather[-8,8]
7			Age [-8,8], Smile[-8,8], Hair length[-5,7], Weather[-8,5], festive [-8,6]
8			Age [-4,6], Eye Size [-8,7], Hair Length [-8,7], Winter Dress [-5,6], Dark Weather (-5,5)
9			Smile [-8,7], Hair Length [-8,8], Seasonal Dress [-6,7], Modern Dress ([-7,7], Pattern Frequency [-4,8]

C User Study results

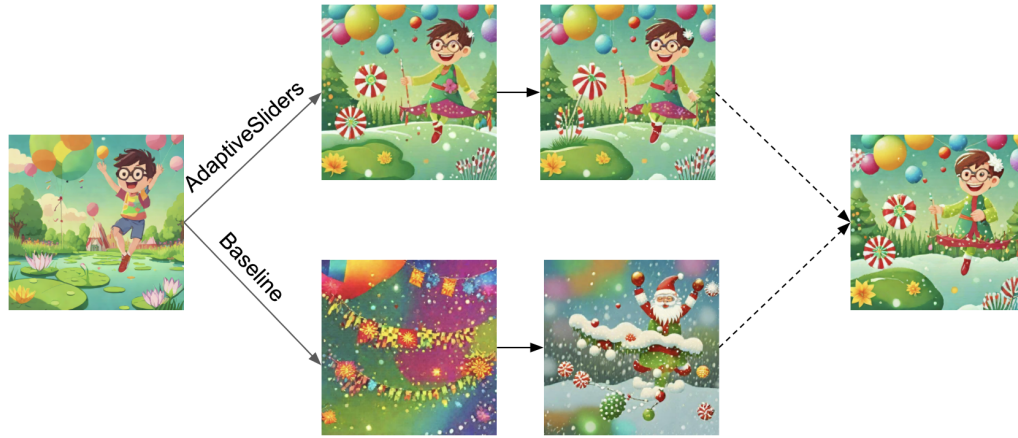


Figure 20: Exploratory Behavior at the extremes: Comparison of a participant’s performance in Task 7 using AdaptiveSliders (Top row) and Baseline (Bottom row). User starts with original image (Left most) and manipulates sliders to reconstruct target image (Rightmost). Here, the user manipulates the sliders near their extremes (values not shown due to multiple sliders). While the AdaptiveSliders images are still meaningful, those generated by baseline are absurd. Solid lines indicate consecutively generated images, while dashed line implies that there were more intermediate slider manipulations to reach the target.

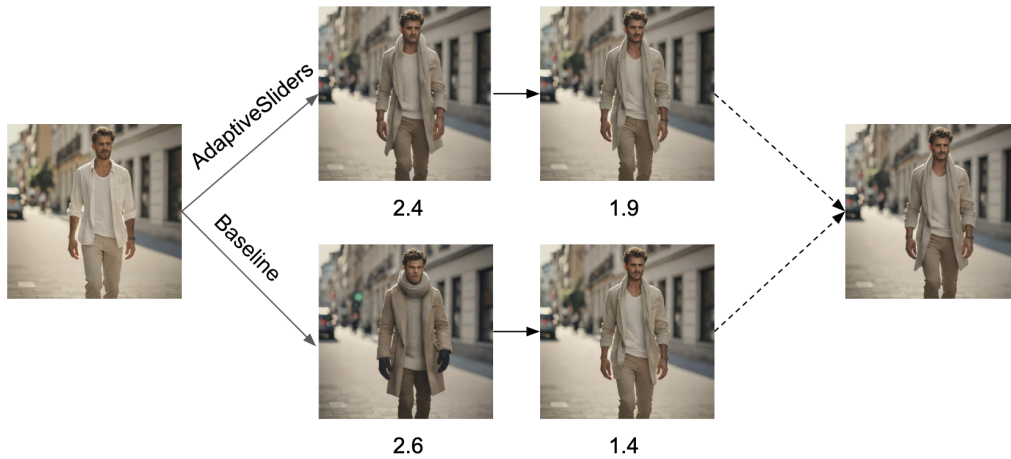


Figure 21: Directly guessing the target value behavior: Comparison of a participant’s performance in Task 2 using AdaptiveSliders (top row) and the Baseline method (bottom row). User starts with original image (Left most) and manipulates the Seasonal Dress slider to reconstruct target image (Right most). The values below each image denote the exact slider values used to generate the image by participants. Here, the user tries to guess the target value but gets an image that is highly different from the target in the Baseline. The participant made more slider manipulations (5) to reach the target than AdaptiveSliders (3). Solid lines indicate consecutively generated images, while dashed line implies that there were more intermediate slider manipulations to reach the target.

D Trained Sliders

Table 5: The Pretrained sliders used for our experiments and user study. The description for each slider provides the information about how each slider changes the attributes

	Sliders	Description
Attributes	Age: Chubby: Beard: Hair length: Hair curly: Glasses slider: Muscular: Obese: Eye Size:	change age of the person change the chubbiness of the person add/remove the beard from the face change the hair length of the person change the curlyness of the hair put the glasses on the person change the intensity of muscle on the person change the intensity of obese in the person change the eye size from bigger to small
Style	Pixar style: Cartoon style: Clay style: Sculpture style: Anime style: Metal style:	change the person to pixar style change the person to cartoon style change the person to clay style change the person to sculpture style change the person to anime style change the person to metal style
Emotions	Anger: Surprise: Happy: Fear: Disgust: Smile:	change the intensity of the anger of the person change the intensity of the surprise of the person change the intensity of the happiness of the person change the intensity of the fear of the person change the intensity of the disgust of the person change the intensity of the smile of the person
Background	Tropical: Weather: Night/Day: Chaotic Dark sky:	change the background to tropical change the weather to snow change the background to night or day change the background to chaotic dark sky
Fashion	Modern dress: Formal dress: Seasonal dress: Increasing Pattern dress: Emotive intensity sliders: Color variations clothing:	changing the dress from old to modern dress changing the dress from casual to formal dress changing the dress from summer to winter dress changing the dress from plain to pattern dress change the intensity of the emotion of the person change the color of the muted colors to vibrant colors

You are an expert prompt analyzer for text-to-image generation models. Text-to-image generation models take a text prompt as input and generate images. You analyze the prompt in detail and identify key concepts/attributes in prompts that user will be willing to modify or edit in the image. Focus on attributes that can have a range of values/intensity, such as age, light levels, human expression and so forth. Avoid discrete or categorical attributes. Your attributes should be from the prompt itself. Avoid density variation in attributes. no need to give explanation.

The concepts in the prompts are defined as attributes that can vary continuously that means attributes that have intensity. Often these attributes are abstract/vague and cannot be quantified. So, make sure you find the attributes which you think user can edit or change continuously through a slider in the image to get the desired output. Do not include spatial relation, counting, activity as they are categorical.

The following are defined as:

Activity: Some of the example activities are playing cooking, running etc.

Spatial relation: Some of the examples for spatial relation are under, above, below

counting: Some of the examples for counting are numbers-two, three, several

The prompt will be used in text to image generation model. Such that the attributes can be changed by the sliders and the change can be seen in the image.

Your continuous attributes should present in the prompt.

Your output should be in this format.

Attribute1

Attribute2

Attribute3

Attribute4

For example,

Prompt: A young person with slight smile

Young

slight smile

Prompt: A serene landscape with a vibrant sunset over calm waters, and a gentle breeze rustling through the trees.

.....

.....

Figure 22: Prompt to GPT for retrieving continuous sliders attribute from users prompt

You are an expert prompt analyzer for text-to-image models. Text-to-image models take a text prompt as an input and generate images. You take **{continuous attribute}** list from the user and the **{prompt}** and pretrained **{sliders}** list. Your task is to identify which sliders can be used to modify the continuous attributes. Remember, you need to analyze both the attribute and the prompt to correctly map the attributes to the corresponding pretrained sliders. If you are not able to find correct correspondence, then try to break the attribute into smaller parts and then see if pretrained sliders can be mapped to those parts. If you cannot find the correct correspondence, suggest the user to train their custom sliders. You must find sliders for suggested attribute list only.

{sliders}

Your response should follow below format strictly.

Attribute1:slider1,slider2

Attribute2:slider1,slider2

Attribute3:slider1,slider2

Attributes need custom sliders:Custom slider1,Custom slider2

Here are some of the examples below.

prompt: A chubby penguin

continuous attributes: chubby

chubby:

Attributes need custom sliders: chubby for penguin

.....

.....

.....

Figure 23: Prompt to GPT for mapping continuous sliders attribute from users prompt

You are question answer analyzer. You are provided with the list of questions from the user. Your job is to analyze these questions as thoroughly as possible. These questions are made in such a way to check image prompt faithful. Image prompt faithfulness is defined as how much prompt information is present in the image. If the answer for every question is yes in an image, then image and prompt are faithful.

Given a **{attribute}** to change using a slider, the slider will change the certain part of image keeping rest of the image same. Your job is to remove the questions from the original list which will be affected by the change in the **{attribute}** slider.

Further modify the remaining questions based on the **{attribute}**. The final question list will be passed through the Vision Question Answer model to check the other feature whether they are changed or not. Analyze questions in detail and no need to provide explanation.

Your output should be in this format.

removed question: question 1, question 2, question 3

modified question: question 1, question 2, question 3

Example:

question list = ["Is the person young?", "Is the person hair curly?", "Is the person a child?"]

attribute = "age"

removed question: Is the person young?

modified question: Is the person hair curly? Is the person a child?

.....

.....

.....

Figure 24: Prompt for GPT to filter the questions

Slider Bound - Evaluation UI

Instructions:
For every attribute, we have two methods to generate their most extreme values.
For example, for a prompt that specifies "young", the left bound image shows a really young face and the right bound image shows a really old person.
The goal is to capture "as much range of the attribute as possible" without

- 1) Materially changing the other parts of the image present in the prompt,
- 2) Absurd images
- 3) Without making changes that are unaligned with the attribute.

Prompt:
A chubby child with medium-length hair, wearing a superhero costume, in a backyard
Attribute : Chubby

Annotation# 2 of 100


Which image pair is better?

☐ Left side image pair ☐ Right side image pair ☐ Can't Decide


Submit

Left side image pair

Lower Bound




Upper Bound



Right side image pair

Lower Bound



Upper Bound

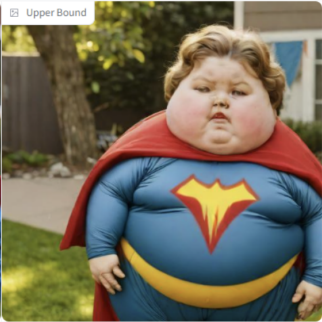


Figure 25: UI for experiment 2

Slider Consistency- Evaluation UI

Instructions:
For every attribute, we have two methods to generate their continuous variation.
The goal is to capture as smooth as possible variation from left to right without any sudden jumps between two images.

Attribute Variation: Age

Annotation# 3 of 100


Which images has better continuity?

☐ Top Images ☐ Bottom Images ☐ Can't Decide

Submit

Top Images

Image



Bottom Images

Image




Figure 26: UI for experiment 3