

SliderEdit: Continuous Image Editing with Fine-Grained Instruction Control

Arman Zarei¹, Samyadeep Basu², Mobina Pournemat¹, Sayan Nag², Ryan A. Rossi², Soheil Feizi¹
¹University of Maryland ²Adobe Research

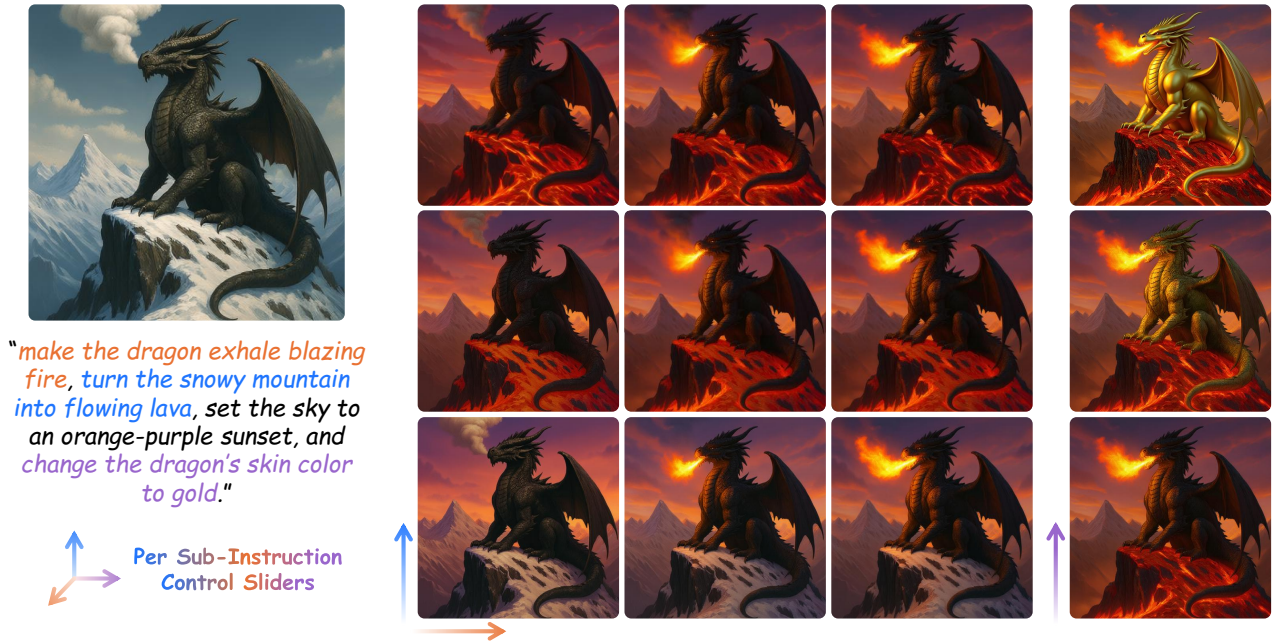


Figure 1. **SliderEdit produces continuous edit trajectories in state-of-the-art instruction-based image editing models.** Our method provides fine-grained and disentangled control over the intensity of edit attributes described in an instruction, allowing continuous transitions between editing strengths. Despite its effectiveness, SliderEdit is extremely lightweight and can be trained efficiently to transform a state-of-the-art instruction-based image editing model into a continuously controllable editing framework.

Abstract

Instruction-based image editing models have recently achieved impressive performance, enabling complex edits to an input image from a multi-instruction prompt. However, these models apply each instruction in the prompt with a fixed strength, limiting the user’s ability to precisely and continuously control the intensity of individual edits. We introduce **SliderEdit**, a framework for continuous image editing with fine-grained, interpretable instruction control. Given a multi-part edit instruction, SliderEdit disentangles the individual instructions and exposes each as a globally trained slider, allowing smooth adjustment of its strength. Unlike prior works that introduced slider-based attribute controls in text-to-image generation, typically requiring separate training or fine-tuning for each attribute or concept, our method learns a single set of low-rank adaptation matrices that generalize across diverse ed-

its, attributes, and compositional instructions. This enables continuous interpolation along individual edit dimensions while preserving both spatial locality and global semantic consistency. We apply SliderEdit to state-of-the-art image editing models, including FLUX-Kontext and Qwen-Image-Edit, and observe substantial improvements in edit controllability, visual consistency, and user steerability. To the best of our knowledge, we are the first to explore and propose a framework for continuous, fine-grained instruction control in instruction-based image editing models. Our results pave the way for interactive, instruction-driven image manipulation with continuous and compositional control.¹

1. Introduction

Recent advances in large-scale diffusion [16, 31, 34] and flow-matching models [9, 27] have revolutionized image synthesis, enabling unprecedented photorealism and seman-

¹Project page is available at: <https://armanzareei.github.io/SliderEdit>

tic fidelity. Building on these foundations, instruction-based image editing has emerged as a powerful paradigm, allowing users to modify images through natural language commands [4, 24, 42, 49]. The latest state-of-the-art models, such as *FLUX-Kontext* [24] and *Qwen-Image-Edit* [42], can perform a wide spectrum of manipulations, from global scene and style transformations to highly localized, fine-grained edits, all within a unified text-driven framework.

Despite these advances, current instruction-based editing models remain inherently *discrete*: they apply edits in an all-or-nothing manner, offering limited control over how strongly each instruction is expressed. For example, given an image of a dragon and a multi-instruction prompt such as “change the skin color to gold and make it exhale fire”, existing models generate a single fixed outcome for a given prompt. While doing multiple generations may yield different variations, it does not allow systematic adjustment of individual edit *strengths*, such as turning the skin slightly gold versus bright metallic gold, or adding a small flame versus a large burst of fire. This lack of fine-grained, continuous control limits both user flexibility and interpretability—the two key properties for truly interactive image editing.

To address this gap, we propose *SliderEdit*, a framework for continuous image editing with fine-grained instruction control. Our goal is to extend state-of-the-art instruction-based editing models into systems that support *continuous, disentangled, and interpretable control* over the effects of individual editing instructions. Specifically, given a multi-instruction prompt, *SliderEdit* assigns each instruction its own slider, allowing smooth adjustment of its influence between suppression, full application, and amplification (See Fig. 1). These sliders provide intuitive and flexible control over complex multi-instruction edits, operating seamlessly without any re-training or per-instruction fine-tuning.

Our key insight is that the latent representations of modern multimodal diffusion transformers (MMDiT) encode instruction semantics within localized token embeddings. By identifying and selectively modulating these tokens, we can gain fine-grained control over how individual instructions affect the output. Building on this observation, *SliderEdit* employs a small set of learnable low-rank adaptation matrices that act directly on instruction-relevant token embeddings. These adapters are trained using a novel and lightweight objective, the *Partial Prompt Suppression (PPS)* loss, which teaches the model how to suppress or neutralize the visual effect of a specific instruction. The loss simply requires that the model’s output, when given the full prompt, matches the output produced when the target instruction is removed, making it intuitive, interpretable, and easy to optimize. Once trained, these low-rank adapters naturally yield continuous sliders by smoothly scaling their learned weights, enabling interpretable adjustment of each instruction’s influence. For single-instruction edits, we fur-

ther extend this idea by applying the adapter across all image and text tokens, resulting in smoother edit trajectories.

SliderEdit integrates seamlessly with existing state-of-the-art instruction-based image editing models such as *FLUX-Kontext* and *Qwen-Image-Edit*, requiring only minimal additional training. Our approach provides a unified framework for continuous and compositional control across diverse editing scenarios—from subtle attribute adjustments and stylistic refinements to complex, multi-object scene manipulations. Through both quantitative evaluation and qualitative analyses, we show that *SliderEdit* delivers superior edit controllability and semantic disentanglement.

In summary, our main contributions are:

- We are the *first to explore and propose a framework for continuous instruction-based image editing*, enabling smooth, fine-grained, and interpretable modulation of edit intensity for individual instructions.
- We propose *Partial Prompt Suppression* loss, which enables efficient training of instruction-aware adapters that learn disentangled, continuous control over edit strengths.
- We demonstrate seamless integration of our framework with state-of-the-art foundation image editing models, achieving substantial improvements in edit consistency and user controllability.

2. Related Works

2.1. Image Editing

Image editing methods have advanced rapidly in recent years. Early approaches built on diffusion priors enabled flexible editing by perturbing and denoising input images [29]. Subsequent methods [2, 14, 20, 22, 40] formulated editing as steering the diffusion trajectory through optimization or conditioning while preserving image fidelity. With the emergence of instruction-based editing [4, 13, 26, 38, 49], models began to directly interpret natural language commands, allowing intuitive user control. More recently, large foundation models for instruction-based image editing [24, 42] have achieved remarkable versatility, performing both local and global modifications within a unified architecture. Despite their impressive capabilities, these models lack *fine-grained controllability*, i.e., the ability to continuously adjust the strength of individual edits. Our work addresses this limitation through a framework that enables continuous and interpretable instruction-level control.

2.2. Continuous Attribute Slider

A growing body of work in image generative modeling has explored continuous attribute control over generated images. Before diffusion models, much of this effort focused on learning structured and manipulable latent spaces in GANs and VAEs [1, 17, 19, 21, 36]. These approaches discovered semantically meaningful directions in the la-

tent space that correspond to interpretable visual attributes. With the emergence of text-to-image diffusion models, recent works [3, 6, 7, 10, 11, 14, 43] have extended continuous attribute control through per-attribute sliders or semantic embedding directions. Methods such as Concept Sliders [10] and Baumann et al. [3] train per-attribute LoRAs or editing directions within the text embedding space to achieve smooth attribute manipulation. While all these approaches mark significant progress toward controllable generation, they face notable limitations: many require training a new LoRA or embedding direction per attribute, suffer from attribute entanglement, or degrade with multiple edits. They also primarily target *text-to-image generation*, offering limited or indirect applicability to real-image editing.

In contrast, our method introduces a novel and unified framework that generalizes slider-based continuous control to *instruction-based image editing*. It eliminates the need for per-attribute retraining, supports multiple simultaneous edits, and remains robust across diverse editing scenarios and unseen attributes, while achieving significantly better performance on real-image editing tasks.

3. SliderEdit: Continuous Image Editing

In this section, we address the problem of enabling fine-grained control over individual editing instructions in a multi-instruction prompt for image editing. Formally, given a prompt $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_K\}$, where each $\mathcal{P}_i$ denotes a distinct edit instruction (e.g., “make her laugh”, “make her hair curly”; see Fig. 2, top row), our goal is to allow the user to modulate the strength of each instruction independently. To this end, we aim to associate each instruction $\mathcal{P}_i$ with a corresponding scaling factor $\beta_i \in [0, 1]$, allowing users to continuously control the strength of that specific edit—ranging from fully suppressing it ($\beta_i = 0$) to fully applying it ($\beta_i = 1$), or even exaggerating it when $\beta_i > 1$.

In Section 3.1, we present the background of the MMDiT architecture and describe how text and image tokens are processed. Section 3.2 then examines how individual instructions $\mathcal{P}_i$ influence the generation process by tracing their effect through the model’s internal representations. This interpretability analysis provides key insights into where and how control can be applied. Building on this, Section 3.3 introduces our method for modulating each instruction’s strength via a continuous scaling mechanism, enabling fine-grained control over multi-instruction prompts.

3.1. MMDiT Architecture

Recent image editing models such as FLUX-Kontext and Qwen-Image-Edit are built upon the MM-DiT architecture, which features a dual-branch structure: one for latent image tokens and one for text embeddings. Specifically, the latent image tokens $x_1, \dots, x_N$ represent both the noisy vectors in the VAE latent space and the encoded latents of the con-



Figure 2. **Instruction-token embedding interpolation for strength control.** Interpolating between instruction and null-token embeddings produces intermediate edit strengths, demonstrating the potential for achieving fine-grained control through direct manipulation of intermediate instruction embeddings.

ditioning source image, while the text tokens $y_1, \dots, y_T$ are obtained by encoding the prompt $\mathcal{P}$ using a pretrained language model (e.g., T5).

The prompt $\mathcal{P}$ is first tokenized into $y'_1, \dots, y'_T$, then padded with the special $\langle \text{pad} \rangle$ token to reach a fixed length T : $\{y'_1, \dots, y'_T, y'_{T+1} = y'_{\langle \text{pad} \rangle}, \dots, y'_T = y'_{\langle \text{pad} \rangle}\}$. These are passed through the T5 encoder to produce final text token embeddings $y_1, \dots, y_T$. The image tokens and text embeddings are then jointly processed by each MM-DiT block, where they interact through shared attention layers that enable cross-modal information exchange between visual and textual representations.

3.2. Instruction-Level Interpretability Analysis

Building on the previous section’s description of token interactions in MMDiT, we next investigate how individual instruction tokens affect generation. Specifically, we analyze the subset $\{y_u, y_{u+1}, \dots, y_{u'}\}$ corresponding to an edit instruction $\mathcal{P}_{\text{target}}$. These embeddings carry the semantic signal responsible for the target edit, and we test whether their influence is localized or diffused through the network via targeted interventions.

Specifically, within each attention block at layer ℓ , we intervene on the target instruction embeddings $\{y_u^\ell, y_{u+1}^\ell, \dots, y_{u'}^\ell\}$, which represent the target instruction tokens input to that block (i.e., the embeddings after processing by the preceding layers). We linearly interpolate

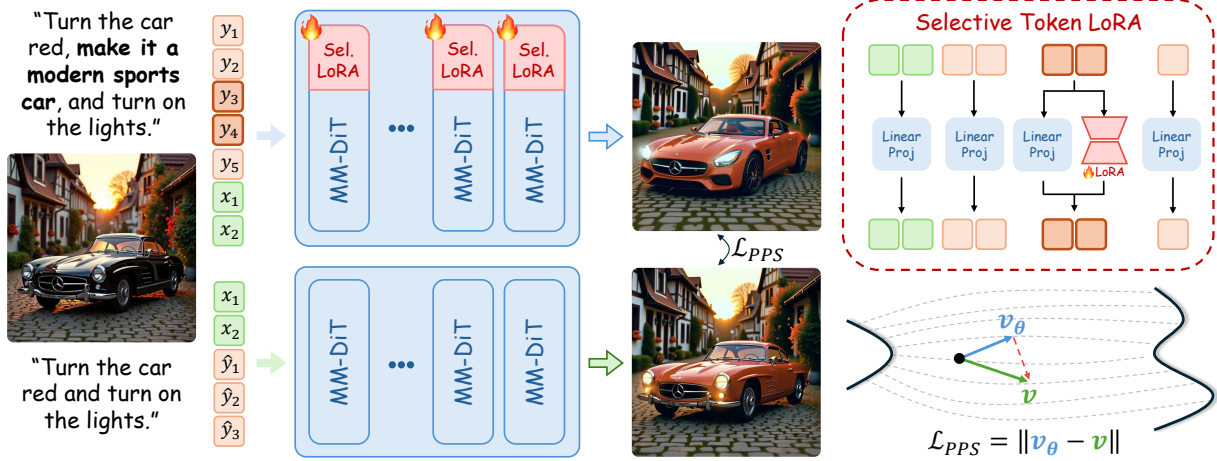


Figure 3. **Overview of the SliderEdit training pipeline.** Learnable low-rank matrices are applied to the intermediate token embeddings corresponding to the target edit instruction. These adapters are trained using the Partial Prompt Suppression (PPS) loss, which encourages the model to suppress or neutralize the visual effect of the selected instruction tokens.

them with the padding token embedding y_{pad}^ℓ :

$$y_j^\ell \leftarrow (1 - \beta) \cdot y_j^\ell + \beta \cdot y_{\text{pad}}^\ell, \quad \text{for } j \in \{u, \dots, u'\}.$$

The interpolation coefficient $\beta \in [0, 1]$ determines how much of the instruction’s information is preserved. Setting $\beta = 1$ effectively removes the instruction by replacing its embeddings with that of the padding token (i.e., no information), while $\beta = 0$ leaves the instruction fully intact.

Figure 2 illustrates the resulting generations. The bottom row corresponds to $\beta = 1$, where the edit is entirely removed, while the middle row shows an intermediate, manually chosen β that partially applies the edit. These results demonstrate that the intermediate token embeddings corresponding to $\mathcal{P}_{\text{target}}$ are highly localized and show strong potential for achieving fine-grained control by directly manipulating their embeddings. While this analysis shows potential for controlling edits via simple embedding interpolation, this approach provides only limited and discontinuous modulation. To achieve stronger and smoother control, we propose a robust method in the next section.

3.3. Fine-Grained Control of Edit Instructions

Building on our interpretability findings, we introduce a mechanism that enables continuous and independent control over each edit instruction in a multi-instruction prompt.

Given an input image X_{orig} and a prompt $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_K\}$ containing K edit instructions, a base image editing model produces an edited output $X_{\text{edited}}^{\mathcal{P}_1, \dots, \mathcal{P}_K}$ where all edits are applied simultaneously. Our objective is to learn a flexible adapter $M_\theta(\mathcal{P}_i)$ capable of suppressing or modulating a specific instruction $\mathcal{P}_i$ within $\mathcal{P}$. When this adapter is activated, the model should generate $X_{\text{edited}}^{\mathcal{P}_1, \dots, \mathcal{P}_{i-1}, \mathcal{P}_{i+1}, \dots, \mathcal{P}_K}$, effectively removing the influence of $\mathcal{P}_i$ while keeping other edits intact.

Partial Prompt Suppression Loss. To train M_θ , we propose the Partial Prompt Suppression (PPS) objective. Using the frozen base model $\epsilon(Z, X, P)$, where Z denotes the noisy latents, X the original image latents, and P the text prompt, we first perform a forward pass with the prompt excluding the i -th instruction $\mathcal{P}_i$. We then require that the adapted model $\epsilon_{M_\theta(\mathcal{P}_i)}$, when given the full prompt, produces an equivalent denoising direction:

$$\mathcal{L}_{\text{PPS}} = \|\epsilon_{M_\theta(\mathcal{P}_i)}(Z, X_{\text{orig}}, \mathcal{P}) - \epsilon(Z, X_{\text{orig}}, \mathcal{P} - \{\mathcal{P}_i\})\|$$

Intuitively, this objective teaches the adapter to neutralize the representation of the tokens corresponding to $\mathcal{P}_i$ throughout the model so that their visual effect disappears. In addition to PPS, we introduce a simplified variant, *Simplified Partial Prompt Suppression (SPPS)*. SPPS treats each edit prompt as a single instruction (i.e., $\mathcal{P} = \mathcal{P}_1$) and applies the same suppression objective directly to $\mathcal{P}_1$ (See Figure 8). Despite its simplicity, SPPS yields highly robust and generalizable adapters, even for multi-instruction editing scenarios. Algorithm 1 outlines the overall training procedure of SliderEdit. Additional details on SPPS and its comparison with PPS are provided in Appendix 7.1.

Selective Token LoRA. We instantiate M_θ as a Selective Token LoRA (STLoRA)—a lightweight, token-aware adapter. STLoRA learns low-rank updates for selected linear projections in the model but applies them *only* to the embeddings of target tokens corresponding to the suppressed instruction $\mathcal{P}_i$. Formally, consider a linear projection at layer ℓ where tokens z (either image or text) are transformed as $z' = W^\ell z$. STLoRA introduces trainable low-rank matrices A^ℓ and B^ℓ with $\Delta W^\ell = B^\ell A^\ell$, updating only the selected target tokens:

$$z'_{\text{target}} = (W^\ell + \Delta W^\ell) z_{\text{target}}, \quad z'_{\text{others}} = W^\ell z_{\text{others}}.$$

Algorithm 1 Training SliderEdit

Require: $\epsilon(Z, X, P)$: Image Editing Model, M_θ : Trainable Adapter, $\{(X_{\text{orig}}^{(i)}, \mathcal{P}^{(i)})\}$: Dataset

- 1: **for** each training step **do**
- 2: $X_{\text{orig}}, \mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_K\} \leftarrow \text{Sample a Data}$
- 3: $\epsilon \sim \mathcal{N}(0, I), t \sim \mathcal{U}[0, 1]$
- 4: **if** use $\mathcal{L}_{\text{SPPS}}$ **then**
- 5: $\mathcal{P} = \{\mathcal{P}'_1\}$ $\triangleright$ Consider whole as a single-instruction prompt
- 6: **end if**
- 7: $Z \leftarrow (1 - t)\epsilon + tX_{\text{orig}}$
- 8: $\mathcal{P}_i \leftarrow \text{Random target instruction from } \mathcal{P} \text{ to suppress}$
- 9: $v^* \leftarrow \epsilon(Z, X_{\text{orig}}, \mathcal{P} \setminus \{\mathcal{P}_i\})$
- 10: $\hat{v} \leftarrow \epsilon_{M_\theta(\mathcal{P}_i)}(Z, X_{\text{orig}}, \mathcal{P})$
- 11: $\mathcal{L}_{\text{PPS}} = \|\hat{v} - v^*\|^2$
- 12: Update θ via gradient descent on $\mathcal{L}_{\text{PPS}}$
- 13: **end for**

Algorithm 2 M_θ^ℓ (STLoRA / GSTLoRA)

Require: W^ℓ : Base Linear Projection, $\{A^\ell, B^\ell\}$: Low-Rank Matrices, $\text{mode} \in \{\text{STLoRA}, \text{GSTLoRA}\}$

Input: $\{x_1, \dots, x_N\}$: Image Tokens, $\{y_1, \dots, y_T\}$: Text Tokens, $\mathcal{P}_i$: Target Instruction

- 1: $\Delta W^\ell = B^\ell A^\ell$
- 2: **if** $\text{mode} = \text{GSTLoRA}$ **then**
- 3: $y_i \leftarrow (W + \Delta W)y_i \quad \forall y_i \in \{y_1, \dots, y_T\}$
- 4: $x_i \leftarrow (W + \Delta W)x_i \quad \forall x_i \in \{x_1, \dots, x_N\}$
- 5: **else if** $\text{mode} = \text{STLoRA}$ **then**
- 6: $\mathcal{T} \leftarrow \text{TokenIndices}(\mathcal{P}_i)$ $\triangleright$ Indices in $\{1, \dots, T\} \mapsto \mathcal{P}_i$
- 7: $y_i \leftarrow (W + \Delta W)y_i \quad \forall y_i \in \mathcal{T}$
- 8: $y_i \leftarrow Wy_i \quad \forall y_i \in \{y_1, \dots, y_T\} \setminus \mathcal{T}$
- 9: $x_i \leftarrow Wx_i \quad \forall x_i \in \{x_1, \dots, x_N\}$
- 10: **end if**
- 11: **return** $\{x_1, \dots, x_N\}, \{y_1, \dots, y_T\}$

This selectivity ensures the adapter modifies only target token embeddings. Figure 3 illustrates the *SliderEdit* training pipeline, and Figure 8 shows the SPPS variant.

Continuous Control via Scaling STLoRA Once trained, the LoRA adapter naturally supports continuous control through its scaling parameter [10, 18, 28, 35]. We denote M_θ^α as the adapter with scaled updates $\alpha \Delta W_\ell$ for each layer. By varying α within a predefined range $[\alpha_{\min}, \alpha_{\max}]$, we obtain a smooth continuum of effects—from complete suppression ($\alpha = 1$) to full application ($\alpha = 0$), and even exaggerated edits for $\alpha < 0$. Note that the scaling parameter α_i follows an inverse range compared to β_i defined earlier. The two scales can be related through $\alpha = 1 - \beta$.

Globally Selective Token LoRA While STLoRA effectively handles both single- and multi-instruction prompts by selectively modulating tokens corresponding to each in-

struction $\mathcal{P}_i$, we introduce Globally Selective Token LoRA (GSTLoRA) for the single-instruction setting. In this variant, all token embeddings (both text and image) are included in the adaptation, allowing LoRA updates to be applied globally across the representation space. This design provides stronger control and often yields higher-fidelity edits when manipulating a single instruction, as the update can leverage global context rather than being limited to a subset of intermediate text token embeddings. Algorithm 2 outlines the operation of STLoRA and GSTLoRA adapters.

4. Experiments

We conduct comprehensive quantitative and qualitative evaluations of SliderEdit, showing that it performs robustly across a wide range of instruction edits. In addition, we compare various baselines and SliderEdit variants, demonstrating that our method achieves superior results, offering continuous and precise control over edits.

4.1. Implementation details

We use FLUX-Kontext and Qwen-Image-Edit as our base models. All models are trained with the ℓ_{SPPS} loss for simplicity and generalization, while $\mathcal{L}_{\text{PPS}}$ provides stronger multi-instruction control for STLoRA (see Appendix 7.1). We set the LoRA rank to 16, keeping the adapters lightweight and efficient. Training uses a small subset (1k–8k samples) of the GPT-Image-Edit dataset [41]. Both STLoRA models are trained for 1,000 iterations, converging around 400 but extended for consistency. GSTLoRA on FLUX-Kontext is trained for 300 iterations. Overall, the training process is computationally very lightweight and data-efficient. Further details are provided in Appendix 8.1.

4.2. Qualitative Results

In this section, we qualitatively evaluate the results of SliderEdit variants, demonstrating their effectiveness across diverse scenarios and editing capabilities.

Figure 4 presents qualitative examples of GSTLoRA applied to the FLUX-Kontext model. As shown, our method produces smooth and continuous edit trajectories, enabling fine-grained control over the strength of edits. It effectively handles both *local edits* (e.g., adding makeup or modifying a car’s age) and *global edits* (e.g., changing the season or adjusting scene lighting). Additional examples, such as edits involving camera view or angle changes, as well as applications in text editing and face editing, are provided in Figures 14, 9, 15, and 16 in the Appendix.

Figures 6 and 17 illustrate qualitative results of STLoRA for instructions containing two edit directions. The resulting 2D intermediate space exhibits smooth and continuous variations, allowing users to precisely control edit strengths along each direction to obtain desired outputs. Figures 1 and 10 provide additional cases with three edit directions.

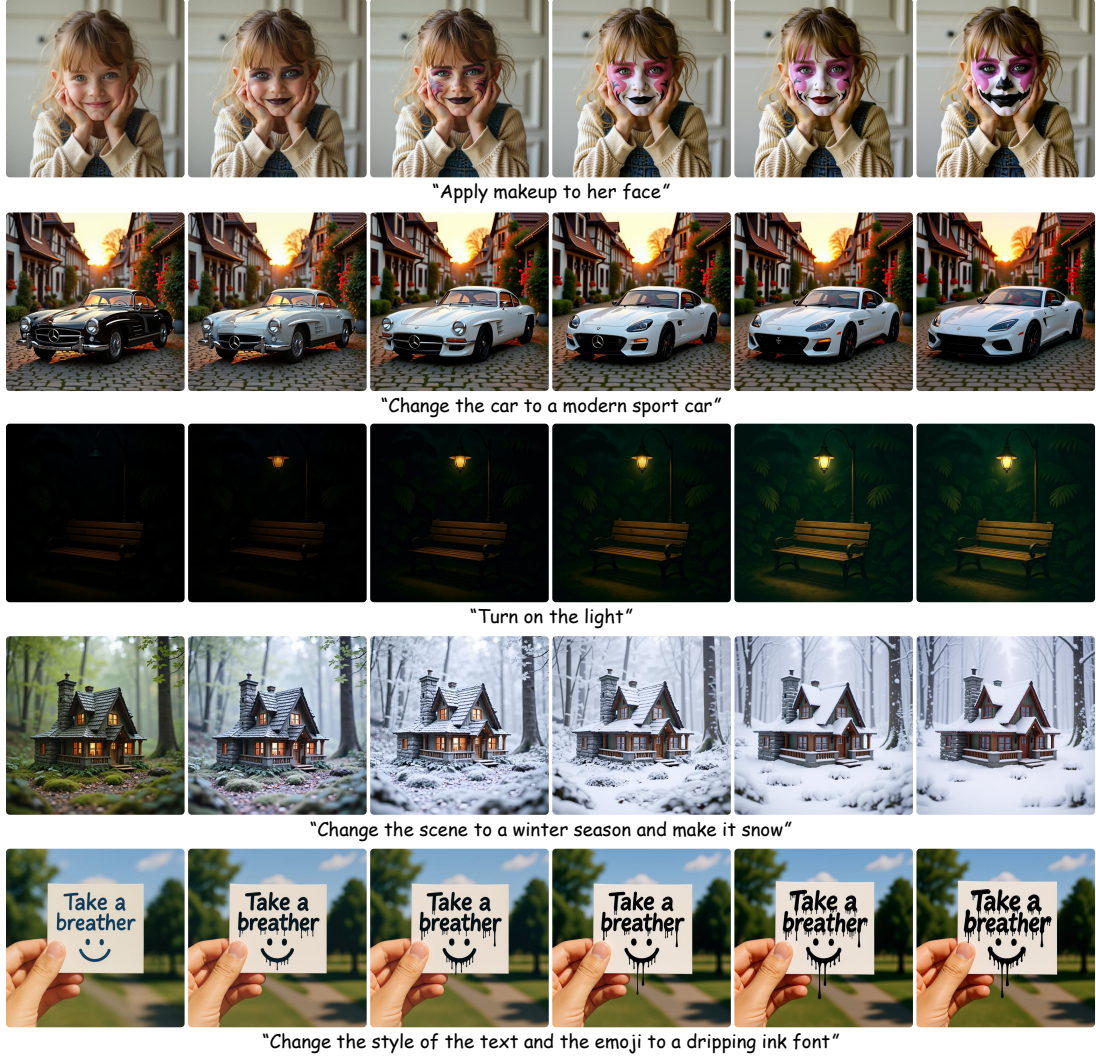


Figure 4. **Qualitative Samples of GSTLoRA.** Demonstrates smooth, continuous control over the strength of both local and global edits.

To further explore the versatility of SliderEdit, we examine its performance on advanced tasks supported by state-of-the-art editing models. One such task is *zero-shot personalization*. Figure 5 shows an example where STLoRA is integrated with Qwen-Image-Edit to perform multi-subject personalization, followed by instruction-based scene editing. Our approach provides users with flexible, fine-grained control—by adjusting sliders, one can generate a coherent series of images that naturally evolve, resembling a narrative. This demonstrates the potential of SliderEdit as a powerful tool for storytelling and creative content generation.

4.3. Quantitative Results

In this section, we quantitatively evaluate the performance of STLoRA and GSTLoRA against multiple baselines. We assess their ability to achieve continuous, extrapolative, and disentangled control through quantitative metrics, providing

an objective analysis of the smoothness and independence of the edit trajectories generated by each method.

4.3.1. Evaluation Set

For quantitative evaluation, we construct a facial editing benchmark with N subjects of diverse genders, ages, and ethnicities, and define M edit directions (e.g., “*make the hair curly*”, “*make the hair long*”). Original images are chosen so that target attributes are absent (e.g., straight, short hair). We evaluate each model under editing configurations containing γ instructions, sampling γ instructions from the M available to form $\binom{M}{\gamma}$ prompts. For each instruction, the edit strength α varies within $[\alpha_{\min}, \alpha_{\max}]$ across δ steps, yielding a γ -dimensional edit space of δ^γ images per prompt. This structured space enables quantitative analysis of *continuity*, *extrapolation*, and *disentanglement*.

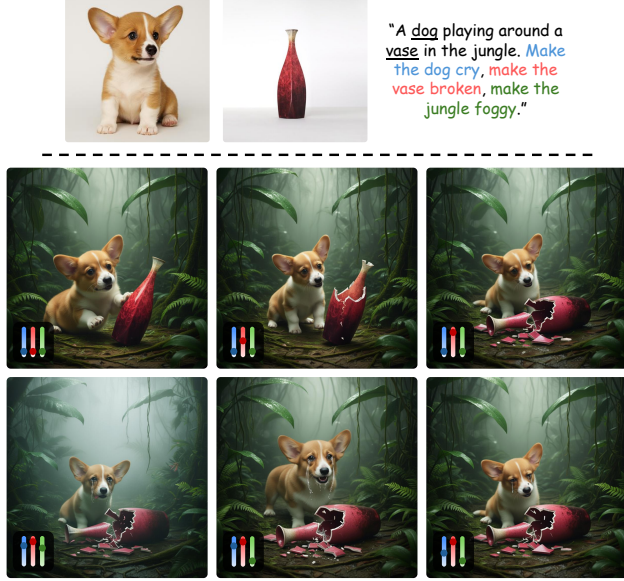


Figure 5. **Controllable zero-shot multi-subject personalization with STLoRA.** STLoRA enables smooth adjustment of each instruction’s strength to generate coherent, evolving image sequences, supporting story-like visual editing. (Best viewed from top-left to top-right, then bottom-right to bottom-left)

4.3.2. Metrics

We employ several quantitative metrics to evaluate different aspects of the editing behavior, including *continuity*, *extrapolation*, and *disentanglement*. For each instruction edit, the model generates a sequence of images at varying edit strengths, which are then analyzed using the following metrics. To measure how strongly each edit is reflected in the generated image, we use vision-language models. For each instruction (e.g., “make the person laugh”), we define a corresponding descriptive prompt (e.g., “a person smiling”) and compute image–text similarity in the embedding space of VLMs such as *CLIP* [33], *SigLIP* [47], and *BLIP* [25]. This score measures how well the edit is executed.

Extrapolation. Extrapolation measures the model’s ability to apply edits beyond the standard range, which is particularly useful when amplifying attributes such as facial expressions. We define the extrapolation score as the maximum VLM’s similarity value, which indicates the strongest expression of the target attribute achieved by the model.

Continuity. Given similarity scores $s_1, \dots, s_\delta$ for increasing α values, we expect them to vary smoothly and uniformly between $\min(s_i)$ and $\max(s_i)$. We quantify this using a chi-squared statistic comparing the observed and expected counts of s_i across bins, where higher $(\chi^2_{\text{agg}}/\text{dof})^{-1}$ indicates smoother edit trajectories. For 2D and 3D edit spaces, we apply the same test to assess the uniformity across the grids. For more details, refer to Appendix 8.2.1

Disentanglement. We evaluate how well the model applies an edit without altering unrelated factors such as iden-

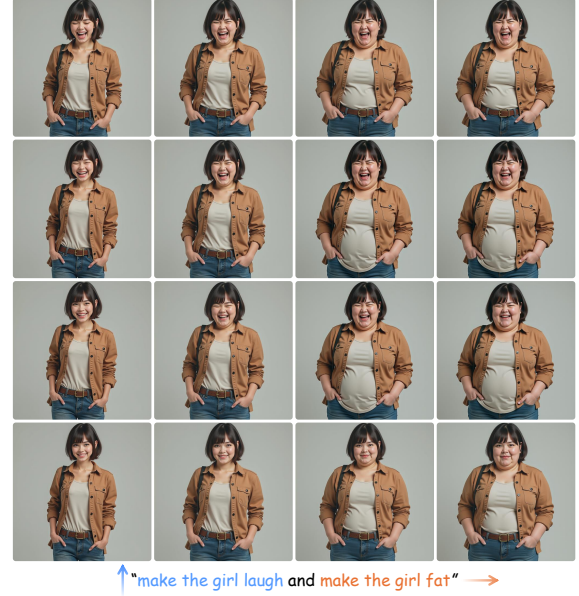


Figure 6. **Qualitative results of STLoRA on 2-instruction edit.** The 2D grid shows smooth, continuous transitions, allowing precise and disentangled control over each instruction’s strength.

tity or background. *Identity preservation* is measured via cosine distance in the *ArcFace* [8] embedding space, where lower values indicate better consistency. We further compute perceptual distances between edited and original images using *LPIPS* [48] (AlexNet [23], VGG [37]) and *DINOv2* [5, 32], capturing both low-level perceptual and high-level semantic changes to assess overall disentanglement.

4.3.3. Baselines

We consider different baselines depending on the number of edit instructions γ used in the prompt.

For the case of a single-instruction setting ($\gamma = 1$), we compare *GSTLoRA* (Ours) and *STLoRA* (Ours) with *Explicit CFG* and *Implicit CFG*, all implemented on top of the *FLUX-Kontext* model, as well as prior methods *ConceptSlider* [10] and *Continuous Attribute Control* [3]. *FLUX-Kontext* is a guidance-distilled model that internally approximates the effect of classifier-free guidance, allowing implicit control over edit strength but offering limited flexibility. To enable explicit control, we reconfigure it to perform guidance externally during inference. *ConceptSlider* and *Continuous Attribute Control* provide fine-grained attribute manipulation but rely on inversion techniques [12, 30], making them less effective for direct image editing.

For multi-instruction edits ($\gamma > 1$), *Explicit CFG*, *Implicit CFG*, and *GSTLoRA* cannot independently control each edit direction, whereas *STLoRA* enables disentangled, per-instruction control. Since *ConceptSlider* and *Continuous Attribute Control* already perform poorly in single-instruction settings, we omit them from this scenario. We evaluate *STLoRA* on both *FLUX-Kontext* and *Qwen-Image*

Model	Extrapolation $\uparrow$			Continuity $\uparrow$			Disentanglement $\downarrow$			
	CLIP	SigLIP	BLIP	CLIP	SigLIP	BLIP	LPIPS _{alex}	LPIPS _{vgg}	DiNO	ID
Concept Slider [10]	0.2581	0.0666	0.4680	0.1803	0.2071	0.1719	0.2174	0.2495	0.2140	0.7091
Cont. Attr. Control [3]	0.2380	0.0484	0.2082	0.1891	0.2167	0.1114	0.1973	0.2485	0.1852	0.5519
Implicit CFG	0.2607	0.0667	0.4658	0.1547	0.1906	0.1513	0.2149	0.2611	0.1405	0.2748
Explicit CFG [15]	0.2731	0.0739	0.5717	0.1993	0.2263	0.1803	0.2465	0.3085	0.1718	0.3415
SliderEdit_{STLoRA}	0.2632	0.0669	0.4762	0.2538	0.2495	0.1830	0.1902	0.2345	0.1071	0.2550
SliderEdit_{GSTLoRA}	0.2648	0.0673	0.4862	0.2998	0.3062	0.2227	0.1868	0.2330	0.1168	0.2675

Table 1. **Quantitative results for single-instruction edits** ($\gamma = 1$). SliderEdit yields smoother trajectories and better identity preservation.

γ	Model	Extrap. _{Avg} $\uparrow$	Cont. _{Avg} $\uparrow$	Dis. _{Avg} $\downarrow$
1	FLUX _{Kontext}	0.2630	0.2709	0.1912
	Qwen _{Image-Edit}	0.3214	0.2319	0.2187
2	FLUX _{Kontext}	0.2776	0.2409	0.2509
	Qwen _{Image-Edit}	0.3160	0.2813	0.3088
3	FLUX _{Kontext}	0.2970	0.3691	0.2762
	Qwen _{Image-Edit}	0.3417	0.4345	0.3630

Table 2. **Quantitative results for multi-instruction edits.** Both models show comparable performance in continuity. FLUX better preserves identity, while Qwen performs better in extrapolation.

Edit. For more details on baselines, refer to Appendix 8.2.2.

4.3.4. Results

Table 1 presents the quantitative comparison for single-instruction prompts ($\gamma = 1$) using $\delta = 15$ across different baselines and metrics, including *continuity*, *extrapolation*, and *disentanglement*. For fair comparison, continuity is calculated based on normalized scores across all methods. As shown, *GSTLoRA* achieves the highest continuity while maintaining strong disentanglement and satisfactory extrapolation performance. Notably, although one might expect *Explicit CFG* to perform comparably, both *STLoRA* and *GSTLoRA* significantly outperform it, demonstrating superior smoothness and control in edit strength.

Figure 7 provides qualitative and quantitative examples for a representative case. *GSTLoRA* produces a remarkably smooth and continuous edit trajectory, in contrast to *Implicit* and *Explicit CFG*, which exhibit abrupt transitions and inconsistent edit intensities. This behavior is well captured by our quantitative metrics—on the left, the aggregated average similarity score across normalized VLM metrics increases gradually for *GSTLoRA*, whereas both CFG variants show sudden jumps. In terms of disentanglement, *GSTLoRA* also achieves lower identity drift and more stable visual consistency compared to the other methods. Refer to Appendix 8.2 for more comparison and other baselines.

Table 2 presents quantitative results for multi-instruction prompts with $\gamma \in 1, 2, 3$ and using $\delta = 7$, comparing *FLUX-Kontext* and *Qwen-Image-Edit*. Both models perform strongly: for single-instruction edits, *STLoRA* on *FLUX* achieves better performance in terms of continuity, whereas in the two- and three-instruction settings, *Qwen* demonstrates stronger results. Moreover, *FLUX* better pre-

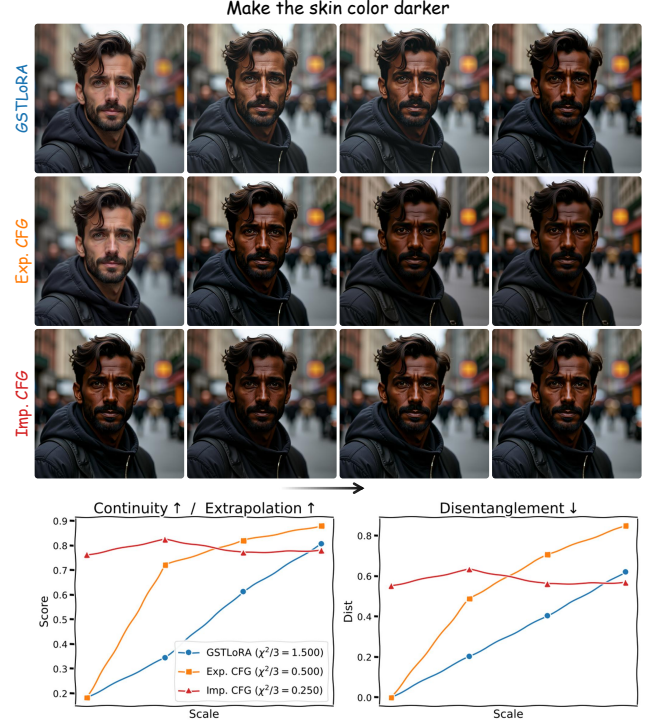


Figure 7. **Qualitative and quantitative comparison of GSTLoRA with CFG baselines.** GSTLoRA shows smooth edit trajectories with gradual similarity changes, unlike Implicit and Explicit CFG, which exhibit abrupt transitions and greater identity drift.

serves identity and disentanglement, whereas *Qwen* performs better in extrapolation. However, as observed across all configurations, there consistently exists a trade-off between continuity, extrapolation, and disentanglement.

5. Conclusion

We introduced *SliderEdit*, a unified framework for continuous, fine-grained instruction control in instruction-based image editing models. By training lightweight low-rank adapters with a novel loss to disentangle and modulate instruction effects, *SliderEdit* enables smooth, interpretable control over edit strength. Integrated with state-of-the-art models like *FLUX-Kontext* and *Qwen-Image-Edit*, it achieves superior controllability, visual coherence, and flexibility, laying the foundation for interactive, instruction-driven editing with continuous and compositional control.

Acknowledgement

This project was supported in part by a grant from an NSF CAREER AWARD 1942230, the ONR PECASE grant N00014-25-1-2378, ARO's Early Career Program Award 310902-00001, Army Grant No. W911NF2120076, the NSF award CCF2212458, NSF Award No. 2229885 (NSF Institute for Trustworthy AI in Law and Society, TRAILS), a MURI grant 14262683, DARPA AIQ grant HR00112590066 and an award from meta 314593-00001.

References

- [1] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics*, 40(3):1–21, 2021. 2
- [2] Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7877–7888, 2025. 2
- [3] Stefan Andreas Baumann, Felix Krause, Michael Neumayr, Nick Stracke, Melvin Sevi, Vincent Tao Hu, and Björn Ommer. Continuous, subject-specific attribute control in t2i models by identifying semantic directions, 2024. 3, 7, 8
- [4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 2
- [5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 7, 3
- [6] Pin-Yen Chiu, I-Sheng Fang, and Jun-Cheng Chen. Text slider: Efficient and plug-and-play continuous concept control for image/video synthesis via lora adapters, 2025. 3
- [7] Yusuf Dalva, Kavana Venkatesh, and Pinar Yanardag. Fluxspace: Disentangled semantic editing in rectified flow transformers, 2024. 3
- [8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 7, 3
- [9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 1, 2
- [10] Rohit Gandikota, Joanna Materzynska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adapters for precise control in diffusion models, 2023. 3, 5, 7, 8
- [11] Rohit Gandikota, Zongze Wu, Richard Zhang, David Bau, Eli Shechtman, and Nick Kolkin. Sliderspace: Decomposing the visual capabilities of diffusion models, 2025. 3
- [12] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. In *European Conference on Computer Vision*, pages 395–413. Springer, 2024. 7, 4
- [13] Qin Guo and Tianwei Lin. 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*, pages 6986–6996, 2024. 2
- [14] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*. 2, 3
- [15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 8
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1
- [17] Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder, 2024. 2
- [18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 5
- [19] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls, 2020. 2
- [20] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations*. 2
- [21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 2
- [22] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6007–6017, 2023. 2
- [23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. 7, 3
- [24] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. 2
- [25] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 7

- [26] Shanglin Li, Bohan Zeng, Yutang Feng, Sicheng Gao, Xiuhui Liu, Jiaming Liu, Lin Li, Xu Tang, Yao Hu, Jianzhuang Liu, et al. Zone: Zero-shot instruction-guided local editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6254–6263, 2024. 2
- [27] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 1
- [28] Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6025–6035, 2025. 5
- [29] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jianjun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*. 2
- [30] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047, 2023. 7, 4
- [31] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 1
- [32] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 7, 3
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 7
- [34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1
- [35] Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora: Any subject in any style by effectively merging loras. In *European Conference on Computer Vision*, pages 422–438. Springer, 2024. 5
- [36] Yujun Shen, Jinjin Gu, Xiaou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing, 2020. 2
- [37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7, 3
- [38] Kunal Swami, Raghu Chittersu, Pranav Adlinge, Rajeev Irny, Shashavali Doodekula, and Alok Shukla. Promptartisan: Multi-instruction image editing in single pass with complete attention control. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 2
- [39] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14940–14950, 2025. 2
- [40] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. In *Forty-second International Conference on Machine Learning*. 2
- [41] Yuhang Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. Gpt-image-edit-1.5 m: A million-scale, gpt-generated image dataset. *arXiv preprint arXiv:2507.21033*, 2025. 5, 2
- [42] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 2
- [43] Yuqi Yang, Dongliang Chang, Yuanchen Fang, Yi-Zhe Song, Zhanyu Ma, and Jun Guo. Controllable-continuous color editing in diffusion model via color mapping, 2025. 3
- [44] Arman Zarei, Keivan Rezaei, Samyadeep Basu, Mehrdad Saberi, Mazda Moayeri, Priyatham Kattakinda, and Soheil Feizi. Improving compositional attribute binding in text-to-image generative models via enhanced text embeddings. *arXiv preprint arXiv:2406.07844*, 2024. 2
- [45] Arman Zarei, Keivan Rezaei, Samyadeep Basu, Mehrdad Saberi, Mazda Moayeri, Priyatham Kattakinda, and Soheil Feizi. Understanding and mitigating compositional issues in text-to-image generative models. *arXiv e-prints*, pages arXiv–2406, 2024. 2
- [46] Arman Zarei, Samyadeep Basu, Keivan Rezaei, Zihao Lin, Sayan Nag, and Soheil Feizi. Localizing knowledge in diffusion transformers. *arXiv preprint arXiv:2505.18832*, 2025. 2
- [47] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 7
- [48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7, 3
- [49] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. In *Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025. 2