VeloEdit: A Training-Free Consistent and Continuous Image Editing Method via Velocity Field Decomposition

Zongqing Li[1,2], Zhihui Liu[2], and Songzhi Su[1]

[1] Xiamen University
[2] Truesight

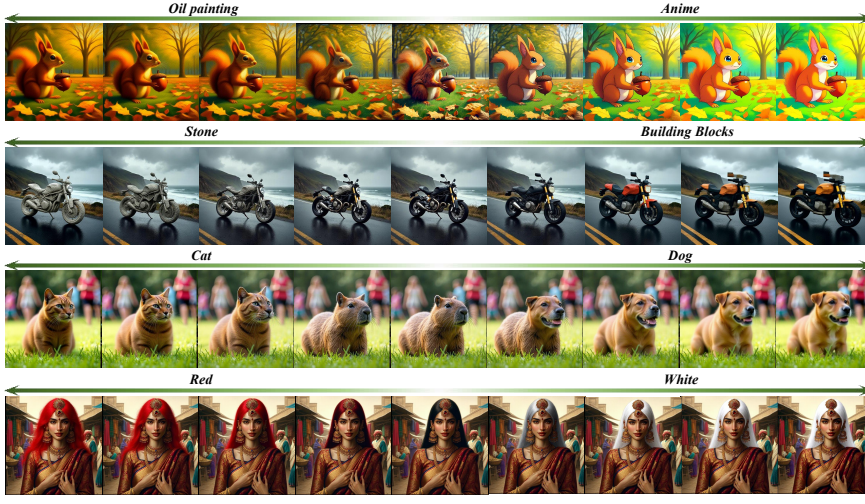

Fig. 1. In the example of continuous editing, our method can generate smooth, multi-intensity editing results across various editing tasks without requiring training.

Abstract. As powerful generative paradigms, diffusion models have garnered significant attention across domains such as image, audio, and video synthesis. With the evolution of generative capabilities, a growing body of research has focused on extending these models' functionalities; in particular, instruction-based image editing has drawn considerable interest due to its potential to modify source images based on specific prompts. However, owing to the stochastic nature of the latent diffusion process and the inherent limitations of current editing models, it remains challenging to preserve visual consistency in non-edited regions. Furthermore, achieving continuous control over the intensity of instruction-based editing proves difficult. In this paper, we propose VeloEdit, a training-free framework designed for consistency-preserving and continuous image

editing. Given a specific editing instruction, VeloEdit automatically delineates the preservation and editing regions by evaluating the similarity between the velocity fields of the source and edited images. Specifically, in the preservation regions, the editing velocity is overridden by the source velocity to ensure consistency. Conversely, within the editing regions, we employ smooth interpolation between the source and editing velocities to modulate the editing intensity, yielding a series of continuously and smoothly edited results. Distinct from prior approaches that manipulate complex internal attention mechanisms or introduce trainable slider-based attribute controllers, our method revisits the fundamental velocity field intrinsic to diffusion models. We apply VeloEdit to state-of-the-art image editing models, including FLUX.1 Kontext and Qwen-Image-Edit-2509, observing significant improvements in both visual consistency and editing controllability.

. . .

Keywords: Diffusion Models, Image Editing, Velocity Field, Consistency, Continuity

1 Introduction

In recent years, diffusion models [1–6] have achieved rapid advancements in generative tasks, demonstrating significant progress across a wide spectrum of domains, including image synthesis [1, 4, 7], video generation [8, 9], 3D generation [10, 11], and audio synthesis [12, 13]. With the emergence of large-scale text-to-image models [7, 14–17], these systems have acquired the capability to comprehend user intent, thereby enhancing output controllability and catalyzing the rapid development of instruction-based image editing methods. These approaches enable precise editing across diverse tasks solely through text instructions, allowing users to generate high-quality results with a minimal learning curve. However, relying exclusively on textual instructions often struggles to preserve the consistency of non-edited regions and fails to achieve continuous editing effects. Consequently, the editing results are confined to a mere subset of the model's latent capabilities, severely restricting the real-world application potential of these models. For instance, given an image of a woman with long hair and the instruction "turn her hair red," current models typically yield an output with a fixed color intensity, often accompanied by unintended alterations to facial features or background drift.

To further unlock the potential of editing models and enhance both consistency and continuity, several approaches have incorporated source feature maps and editing masks to improve editing consistency [18–21]. Meanwhile, other methods introduce trainable neural networks [22–24] to parameterize editing intensity as a controllable slider. However, these approaches typically necessitate extracting feature maps from the source image to derive editing masks and require modifying internal attention layers, or they rely on auxiliary models to generate additional datasets for training. Furthermore, to date, no method

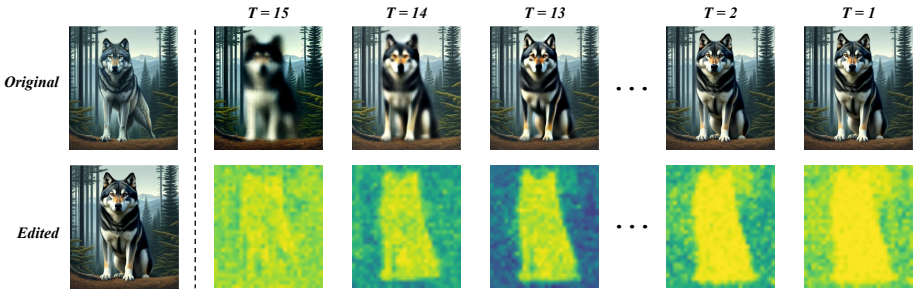

Fig. 2. Mask area calculated using hold speed and edit speed.

has explored how to simultaneously enhance editing continuity while preserving visual consistency.

This raises a fundamental question regarding powerful editing models: what essentially constrains their potential for consistent and continuous editing? Is it truly necessary to manipulate internal attention mechanisms or incorporate additional training modules to unlock these capabilities? We argue that the answer is negative. We posit that the foundational models inherently possess these capabilities, yet they are constrained by the mutual interference of attention maps, feature coupling within the latent space, and the absence of editing instructions that explicitly encode intensity. To further investigate the behavioral dynamics of editing models, we decode the fully denoised representation at each timestep. As illustrated in Fig. 2, we observe that as early as the first step, the model has already localized the editing target, while the non-edited regions are substantially formed. Subsequent steps primarily focus on maintaining consistency in the non-edited areas and refining specific editing details. Subsequently, we replace the editing velocity of the first N steps with the preservation velocity. We find that velocity intervention in merely the first one or two steps is sufficient to completely suppress the editing capability (see Fig. 3), which further corroborates our findings.

Based on these findings, we propose VeloEdit, a training-free framework designed for consistency-preserving and continuous image editing. Specifically, given that full velocity intervention suppresses editing capabilities, we explored the efficacy of partial velocity intervention. By comparing the disparity between the preservation velocity and the editing velocity, we identify regions where similarity exceeds a specific threshold and inject the source velocity into these areas. We observe that this operation not only retains the original editing capability but also substantially improves visual consistency (see Fig. 5). Following this intuition, we postulate that if velocities above the threshold correspond to preservation, then those below the threshold constitute the actual editing velocities driving the image transformation. Consequently, we intervene in the editing velocity by continuously interpolating and extrapolating between the editing and preservation velocities. This approach successfully yields a series of smooth,

fine-grained editing results, effectively extending the model's editing limits, as shown in Fig. 6. In summary, our contributions can be outlined as follows:

- We reveal that the magnitude of editing is predominantly determined by the velocity fields in the initial one or two timesteps, whereas subsequent steps are primarily dedicated to preserving the consistency of non-edited regions and refining specific editing details.
- We propose VeloEdit, the first training-free, velocity-field-based framework for image editing. By leveraging velocity intervention, our method ensures visual consistency while enabling continuous, smooth, and fine-grained control over editing intensity given a single instruction.
- We apply VeloEdit to state-of-the-art image editing models, specifically FLUX.1 Kontext and Qwen-Image-Edit. Extensive experiments demonstrate significant improvements in both editing consistency and continuity, validating the effectiveness of our approach.

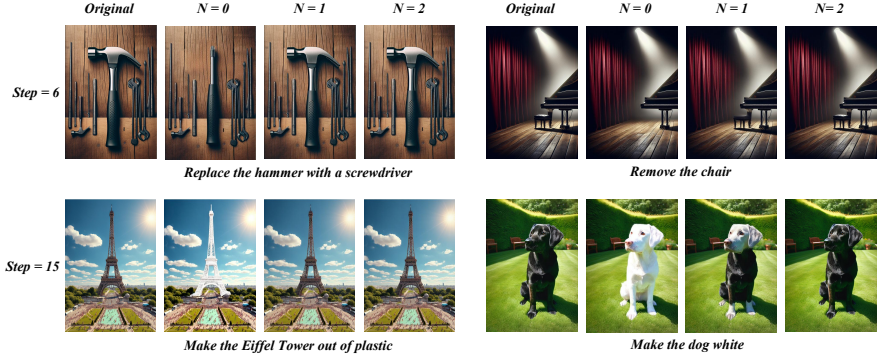

Fig. 3. The editing results after speed-based replacement. Through one or two steps of speed-based replacement, the model's editing capabilities can be completely lost.

2 Related Work

2.1 Instruction-based Image Editing

The growing prowess of text-based image generation models [7, 14, 16, 15, 17] has catalyzed their adaptation and application to more intricate image editing tasks [25–32]. For instance, Prompt-to-Prompt [25] successfully repurposed text-to-image models for editing by manipulating internal cross-attention layers and injecting attention maps from the source image. Subsequently, InstructPix2Pix [31] leveraged Large Language Models (LLMs) and Prompt-to-Prompt [25] to synthesize a triplet dataset comprising source images, editing instructions,

and edited images, thereby training an editing model built upon Stable Diffusion [7]. Inspired by these methodologies, several approaches have explored controllable editing within text-to-image frameworks via inversion and attention injection [33, 34]. Conversely, other works focus on identifying shorter transport paths between the source and target distributions to bypass the inversion process [35, 36]. Furthermore, following the paradigm of InstructPix2Pix [31], certain methods incorporate additional feature channels into text-to-image models to integrate features from guidance images [32, 30, 26], thereby enhancing the controllability of the editing results.

Recently, the emergence of powerful image editing models such as Flux.1 Kontext [27] and Qwen-Image-Edit [29] has further expanded the editing capabilities of diffusion models [37, 28]. These models facilitate precise editing across a variety of tasks solely through textual instructions, enabling users to generate high-quality results with a minimal learning curve. However, relying exclusively on textual instructions presents challenges in maintaining visual consistency and achieving continuous editing effects. This limitation confines editing outcomes to a mere subset of the model's latent capabilities, thereby severely restricting the application potential of large-scale editing models.

2.2 Consistency-preserving Editing

To enhance editing consistency, several approaches employ inversion techniques to extract Key-Value (KV) features from the source image. These features are subsequently injected into the corresponding timesteps of the denoising process, thereby transferring structural and semantic information from the conditional image to the edited result [18–21]. Furthermore, another category of methods introduces mask-based control mechanisms to decouple the editing regions from the preservation regions. By leveraging feature injection, these methods effectively bypass the denoising process within the preservation regions [21, 34, 38, 39].

In contrast, our approach circumvents the complex processes of inversion and KV feature injection. Instead, we intervene directly within the velocity field. This strategy not only reduces computational overhead but also avoids the need for direct manipulation of the model's internal features.

2.3 Continuous Editing

To further unlock the potential of editing models and enhance editing continuity, several approaches introduce trainable LoRA adapters to modulate editing attribute sliders or expand semantic embedding directions [22–24, 40]. By treating editing intensity as a controllable hyperparameter, these methods achieve continuous image editing effects. Alternatively, other studies train encoders to perform fine-grained manipulation within the text embedding space at the token level [41–44], thereby enabling continuous control over editing attributes. However, these methods necessitate additional computational resources for training

and require the generation of training datasets, which hinders their practical utility.

Furthermore, some training-free approaches attempt to identify editing feature directions within the semantic latent space [45] and perform interpolation therein. However, features in the semantic space are not invariably continuous, making it challenging to identify a feature direction that balances both editing accuracy and continuity. Other methods generate continuous editing effects by performing frame interpolation between the source image and the fully edited image [9, 46, 47] or by interpolating within the diffusion feature space [48, 49]. Nevertheless, the generated intermediate states often suffer from discontinuities or exhibit artifacts and blurring [24].

Our approach falls under the category of training-free methods. Distinct from the aforementioned strategies, we neither interpolate within the potentially discontinuous semantic space nor require the pre-generation of fully edited images for frame interpolation. Instead, we leverage the preservation velocity to intervene in the editing velocity, utilizing the robust generative and denoising capabilities of editing models [27, 29] to directly synthesize images with varying editing intensities.

3 Methods

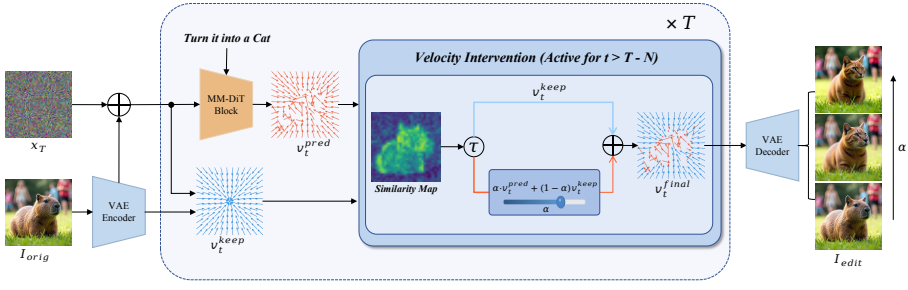

Fig. 4. The overall pipeline of our methods. By calculating the mask of the editing region using the preservation velocity and editing velocity, our method replaces high-similarity regions and fuses low-similarity regions, thus generating a series of smoothly continuous edited images.

3.1 Preliminaries

Flow Matching Flow Matching [5, 6] proposes a generative paradigm based on Continuous Normalizing Flows (CNF), aiming to establish a deterministic mapping between the source distribution (noise) and the target distribution (data).

Flow Matching defines the evolution of the probability density path $p_t(x)$ via Ordinary Differential Equations (ODEs). Given a vector field $v_t(x)$, the generative flow process is described as:

$$\frac{dx_t}{dt} = v_t(x_t). \tag{1}$$

To simplify vector field learning, Rectified Flow [5, 6] constructs straight-line trajectories to approximate the optimal transport path. Given a data sample $x_0 \sim P_{\text{data}}$ and a noise sample $x_1 \sim \mathcal{N}(0, I)$, Rectified Flow defines x_t as a linear interpolation between them:

$$x_t = (1-t)x_0 + tx_1. \tag{2}$$

In this case, the conditional vector field $u_t(x_t|x_0, x_1)$ remains constant as the difference between the target and the source, *i.e.*, $u_t(x_t|x_0, x_1) = \frac{dx_t}{dt} = x_1 - x_0$. Consequently, the objective of Flow Matching is to train a neural network $v_\theta(x_t, t)$ to regress this conditional vector field. The loss function is defined as:

$$\mathbb{E}_{t\sim\mathcal{U}(0,1)}\mathbb{E}_{x_0\sim P_{\text{data}},x_1\sim\mathcal{N}(0,I)}\left\|v_\theta(x_t,t) - (x_1 - x_0)\right\|^2. \tag{3}$$

By minimizing Eq. 3, the network v_θ learns the straight-line velocity field mapping from the data distribution to the noise distribution. By constructing "straight" trajectories connecting noise and data, Rectified Flow avoids curvature in the generation path, thereby enabling efficient and high-quality sampling with minimal steps. This approach has been widely adopted for training text-to-image generation and editing models [14, 27, 29].

3.2 Velocity Field Decomposition

During the inference phase, the model's predicted velocity v_t^{pred} encapsulates tendencies for both preserving the original image content and executing the editing instruction. This observation motivates us to decompose the velocity field. We define the reference velocity v_t^{keep} as the vector directed from the current latent state x_t towards the source image x_{orig}:

$$v_t^{keep} = \frac{x_t - x_{orig}}{t}, \tag{4}$$

Consequently, the effective editing velocity v_{edit}, which drives the actual modification, can be formulated as:

$$v_t^{diff} = v_t^{pred} - v_t^{keep} \tag{5}$$

Considering the premise that editing typically impacts only local regions while the remainder of the image should remain invariant, we posit that v_{pred} and v_{ref} should diverge significantly in editing regions. Conversely, in preservation regions, they should exhibit minimal deviation—or ideally, be identical. Based on this insight, we introduce an element-wise similarity metric S_t to quantify

this discrepancy. Specifically, for a velocity vector at spatial coordinate (i, j), the similarity $S_t[i][j]$ is defined as:

$$S_t[i][j] = \frac{\|v_t^{keep}[i][j]\|}{\|v_t^{keep}[i][j]\| + \|v_t^{diff}[i][j]\|} \tag{6}$$

This similarity metric is bounded within $S_t[i][j] \in (0, 1]$. Here, $S_t[i][j] \to 1$ indicates that the predicted velocity aligns closely with the reference velocity (signifying a preservation region), while $S_t[i][j] \to 0$ implies a substantial deviation between the predicted and reference velocities (signifying an editing region).

3.3 Consistency-preserving Editing

Fig. 5. Editing Results of the High-Similarity Velocity Replacement Strategy. By replacing the predicted velocities with similarity higher than the threshold with the preservation velocity, our method successfully improves the consistency of the unedited regions.

Given a similarity threshold $\tau \in [0, 1]$, we define the velocity replacement strategy as follows:

$$v_t^{replaced}[i][j] = \begin{cases} v_t^{orig}[i][j], & \text{if } S_t[i][j] \geq \tau \\ v_t^{pred}[i][j], & \text{if } S_t[i][j] < \tau \end{cases} \tag{7}$$

For regions exhibiting high similarity ($S_t[i][j] \geq \tau$), we substitute the predicted velocity with the reference velocity to enforce fidelity to the source image. Conversely, for low-similarity regions ($S_t[i][j] < \tau$), we retain the predicted velocity to facilitate the editing process. To prevent the complete suppression of editing capabilities, we restrict the application of this replacement strategy to the initial N timesteps:

$$v_t^{final} = \begin{cases} v_t^{replaced}, & \text{if } t < N \\ v_t^{pred}, & \text{if } t \geq N \end{cases}. \tag{8}$$

where N serves as a configurable hyperparameter governing the duration of intervention. Through this selective velocity replacement, high-similarity regions are constrained to preserve the source content, thereby significantly enhancing editing consistency (ensuring invariance in non-edited regions), while low-similarity regions retain the capacity for modification.

Algorithm 1: Selective Velocity Replacement

Input : Original image I_{orig}, edit prompt P, sampling Steps T, intervention Steps N, intervention Threshold τ

Output: Edited image I_{edit}

Init: $x_T \sim \mathcal{N}(0, I)$, $x_{orig} \leftarrow \text{Encoder}(I_{orig})$

for $t \leftarrow T$ **to** 1 **do**

  $v_t^{pred} \leftarrow \text{Model}(x_t, t, P)$;

  $v_t^{orig} \leftarrow (x_t - x_{orig})/t$;

  if $t > T - N$ **then**

    `// Compute element-wise similarity map`

    $S_t \leftarrow \frac{\|v_t^{keep}\|}{\|v_t^{keep}\| + \|v_t^{keep} - v_t^{pred}\|}$;

    `// Identify regions to replace`

    $M_t \leftarrow \mathbb{I}(S_t \geq \tau)$;

    `// Apply replacement via mask`

    $v_t^{final} \leftarrow M_t \odot v_t^{keep} + (1 - M) \odot v_t^{pred}$;

  $x_{t-1} \leftarrow \text{Step}(x_t, v_t^{final})$;

$I_{edit} \leftarrow \text{Decoder}(x_0)$

return I_{edit};

3.4 Continuous Editing

For low-similarity regions ($S[i][j] < \tau$), we introduce a velocity blending strategy to enable continuous modulation of the editing intensity:

$$v_t^{pred}[i][j] = (1 - \alpha) \cdot v_t^{keep}[i][j] + \alpha \cdot v_t^{pred}[i][j] \tag{9}$$

where $\alpha \in (-\infty, +\infty)$ serves as the blending coefficient. Specifically, when $\alpha \in [0, 1]$, the editing effect smoothly interpolates between the source image and the fully edited output. Conversely, when $\alpha < 0$ or $\alpha > 1$, the editing capability is extrapolated. By integrating selective replacement with velocity blending, the unified intervention formulation is defined as:

$$v_t^{final}[i][j] = \begin{cases} v_t^{keep}[i][j], & \text{if } S_t[i][j] \geq \tau \\ (1 - \alpha) \cdot v_t^{keep}[i][j] + \alpha \cdot v_t^{pred}[i][j], & \text{if } S_t[i][j] < \tau \end{cases} \tag{10}$$

The complete pipeline of VeloEdit is outlined in Algorithm 2.

Algorithm 2: VeloEdit Complete Framework

Input : Original image I_{orig}, edit prompt P, sampling Steps T, intervention Steps N, intervention Threshold τ, mixing weight α

Output: Edited image I_{edit}

Init: $x_T \sim \mathcal{N}(0, I)$, $x_{orig} \leftarrow \text{Encoder}(I_{orig})$

`// Sampling with Velocity Intervention`

for $t \leftarrow T$ **to** 1 **do**

  $v_t^{pred} \leftarrow \text{Model}(x_t, t, P)$;

  $v_t^{keep} \leftarrow (x_t - x_orig)/t$;

  `// Intervention in first` N `steps`

  if $t > T - N$ **then**

    `// Element-wise Similarity Computation`

    $S_t \leftarrow \frac{\|v_t^{keep}\|}{\|v_t^{keep}\| + \|v_t^{keep} - v_t^{pred}\|}$;

    `// High-Similarity Regions: Velocity Replacement`

    $M_t^{high} \leftarrow (S_t \geq \tau)$;

    $v_t^{pred}[M_t^{high}] \leftarrow v_t^{keep}[M_t^{high}]$;

    `// Low-Similarity Regions: Interpolation (if enabled)`

    $M_t^{low} \leftarrow (S_t < \tau)$;

    $v_t^{final}[M_t^{low}] \leftarrow (1 - \alpha) \cdot v_t^{keep}[M_t^{low}] + \alpha \cdot v_t^{pred}[M_t^{low}]$;

  $x_{t-1} \leftarrow \text{Step}(x_t, v_t^{final})$;

$I_{\text{edit}} \leftarrow \text{Decoder}(x_0)$;

return I_{edit}

4 Experiments

4.1 Experiments details

Evaluation Benchmarks: We comprehensively evaluate the editing performance of our method against competing approaches on PIEbench, which encompasses a diverse set of editing tasks including object modification, addition and removal, as well as changes to content, pose, color, material, background and style. Although instructions such as object addition/removal lack explicit continuity, we still conduct tests on these tasks to investigate the failure modes of the methods. The evaluation dataset consists of 700 image-instruction pairs. Additionally, we test VeloEdit on the Subject200K [51] and GPT-Image-Edit [50] datasets to assess the cross-dataset generalization capability of our method.

Experimental Setup: We adopt Flux.1 Kontext and Qwen-Image-Edit-2509 as the base models to validate the effectiveness of VeloEdit. Our default hyperparameter configuration is set as follows: intervention threshold $\tau = 0.8$, sampling steps $T = 6$, intervention steps $N = 1$, and mixing weight $\alpha \in [0.2, 0.4, 0.6, 0.8, 1.0]$.

Evaluation Metrics: We assess all methods in terms of continuity, instruction following and consistency preservation. Specifically, following the protocol in [24], we use the triangular defect δ_{smooth} to quantify the continuity and

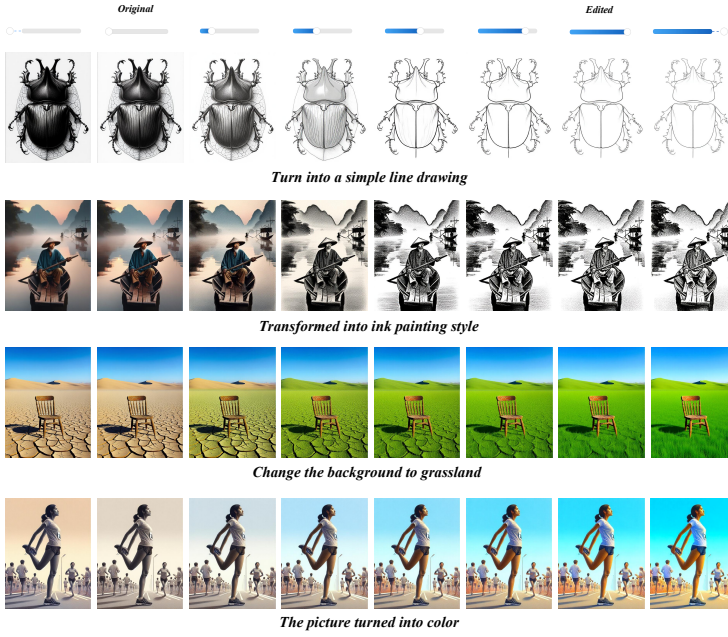

Fig. 6. Visual editing results of VeloEdit on GPT-Image-Edit [50]. Our method achieves smooth control over both local and global editing intensities.

smoothness of editing results, with Dream-Sim [52] as the distance metric. Instruction following capability is evaluated via CLIP directional similarity (CLIP-dir.) [53] aggregated across all editing strengths. For the assessment of consistency preservation, we use the masks in PIEbench to separate the preserved and edited regions, and measure the consistency of the preserved regions by calculating the $L1$ and $L2$ distances within these regions.

4.2 Main results

Qualitative results In this section, we perform a qualitative evaluation of the editing results of VeloEdit to demonstrate its effectiveness across diverse scenarios and editing tasks. Figures 6 and 9 present qualitative examples of VeloEdit applied to the Flux.1 Kontext [27] model. As illustrated in the figures, our method generates smooth and continuous editing trajectories, enabling fine-grained control over editing strength. It can effectively handle a variety of global editing tasks (e.g., style transfer, background modification, image colorization) and local editing tasks (e.g., object replacement, adjustment of object color and attributes).

Quantitative results In this section, we conduct a quantitative evaluation of VeloEdit's performance against multiple baseline methods on PIEbench [54]. We

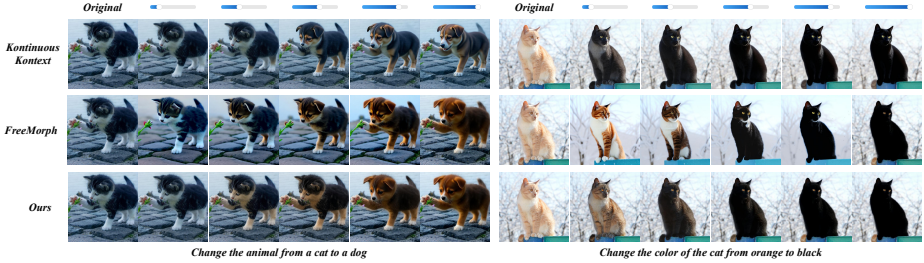

Fig. 7. Qualitative Comparison Results. Our method can generate edited results with excellent consistency and continuity.

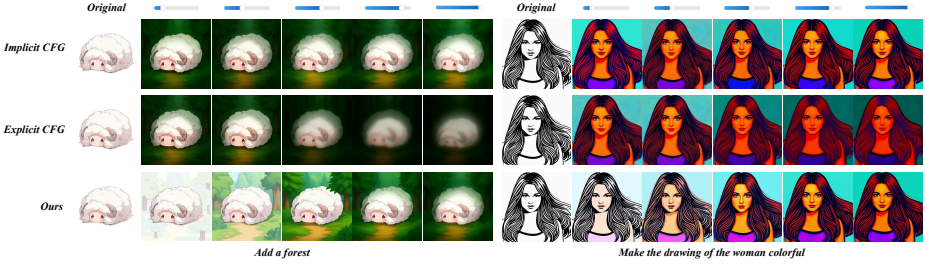

Fig. 8. Qualitative Comparison Results. Our method can generate edited results with excellent consistency and continuity.

assess their capabilities in achieving continuity, instruction following and consistency preservation via quantitative metrics, thereby performing a quantitative analysis of the smoothness and content preservation of the edited trajectories generated by each method. Specifically, we use Flux.1 Kontext [27] to generate fully edited images and pair them with the original images, then leverage Freemorph [48] to produce intermediate editing results. In addition, we evaluate the editing performance of KontinuousKontext [24]—the latest open-source continuous editing model—as well as that of implicit and explicit CFG with different guidance strengths. The results are presented in Table 4.2, which show that our method achieves state-of-the-art or second-best performance across all metrics, fully demonstrating the effectiveness of VeloEdit. Furthermore, we report the metrics of different editing tasks on PIEbench in Table 5, which indicate that our method can yield excellent editing results on a variety of editing tasks. However, like other continuous editing models, it performs poorly on discrete editing tasks, such as object addition/removal and pose modification.

In order to evaluate the cross-model generalization capability of VeloEdit, we apply VeloEdit to Qwen-Image-Edit-2509 [29] with the identical hyperparameter configuration to that of Flux.1 Kontext, and test its performance on PIEbench, the results of which are presented in Table 4.2. The results demonstrate that VeloEdit can be integrated into different editing models to unlock their capabilities for consistent and continuous editing. Furthermore, we compare

the time taken to edit 100 images using the base models alone and with VeloEdit integrated in Table 4.2, demonstrating that our method introduces almost no additional time cost.

Table 1. Quantitative experiments on PIEbench

Method	$\delta_{\text{smooth}} \downarrow$	CLIP-Dir. $\uparrow$	$L_1 \downarrow$	$L_2 \downarrow$
Training-Free				
Freemorph [48]	0.354	0.147	0.142	0.211
Implicit CFG [27, 55]	3.362	0.083	0.140	0.209
Explicit CFG [27, 55]	**0.145**	0.088	0.143	0.214
Training-Based				
KontinuousKontext [24]	0.280	<u>0.219</u>	<u>0.083</u>	<u>0.132</u>
VeloEdit	<u>0.246</u>	**0.294**	**0.074**	**0.116**

Table 2. Quantitative experiments on PIEbench

Method	$\delta_{\text{smooth}} \downarrow$	CLIP-Dir. $\uparrow$	$L_1 \downarrow$	$L_2 \downarrow$
$\text{VeloEdit}_{\text{Flux}}$	0.246	0.294	0.074	0.116
$\text{VeloEdit}_{\text{Qwen}}$	0.286	0.372	0.074	0.110

Table 3. Comparison of time consumption for editing 100 images.

Method	Time (s)
Flux.1 Kontext	246
$\text{VeloEdit}_{\text{Flux}}$	248
Qwen-Image-Edit-2509	286
$\text{VeloEdit}_{\text{Qwen}}$	290

4.3 Ablation Experiments

In this section, we conduct ablation studies on the hyperparameters τ, N, and α of VeloEdit, as illustrated in Fig. 9-14. The results demonstrate that our method exhibits strong robustness across these hyperparameters, with performance degradation—such as loss of continuity or the emergence of artifacts—occurring only at extreme values. We present the ablation study results for the intervention threshold τ in Table 4, which further demonstrate that the hyperparameters of VeloEdit yield robust performance across a wide range of values. Furthermore, while the value of α is theoretically unbounded, empirical observations from

Table 4. Ablation Study on τ

τ	$\delta_{\text{smooth}} \downarrow$	CLIP-Dir. $\uparrow$	$L_1 \downarrow$	$L_2 \downarrow$
0.0	0.038	0.079	0.034	0.052
0.2	0.751	0.168	0.041	0.063
0.4	0.439	0.249	0.055	0.088
0.6	0.303	0.285	0.067	0.107
0.8	0.246	0.294	0.074	0.116
1.0	0.255	0.296	0.075	0.117

Fig. 14 indicate that constraining α within the range of $[-1.0, 2.0]$ yields superior visual quality. Consequently, we recommend this range for optimal performance.

5 Conclusion

In this paper, we present VeloEdit, a generic, training-free framework designed for consistency-preserving and continuous control within instruction-based image editing models. By assessing the disparity between the source and editing velocities to decompose the velocity field, VeloEdit substitutes the editing velocity with the source velocity in high-similarity regions during the early denoising stages to enforce structural consistency. Simultaneously, it employs velocity blending in low-similarity regions to achieve smooth modulation of editing intensity. Upon integration with state-of-the-art models such as FLUX.1 Kontext and Qwen-Image-Edit, our approach significantly enhances both visual consistency and editing continuity without modifying internal attention mechanisms or requiring additional training. This work offers a novel perspective on unlocking the fine-grained editing potential of large-scale diffusion models in a strictly tuning-free manner.

Table 5. Comparison of Image Editing Methods Across Different Edit Types

Edit Type	Method	$\delta_{\text{smooth}} \downarrow$	CLIP-Dir. $\uparrow$	$L_1 \downarrow$	$L_2 \downarrow$
Random	FreeMorph	0.302	0.177	0.150	0.223
	KontinuousKontext	**0.119**	0.251	**0.089**	0.136
	VeloEdit	0.151	**0.321**	**0.089**	**0.135**
Change Object	FreeMorph	0.410	0.208	0.159	0.253
	KontinuousKontext	0.379	0.236	0.091	0.155
	VeloEdit	**0.277**	**0.384**	**0.086**	**0.151**
Add Object	FreeMorph	**0.394**	0.096	0.111	0.174
	KontinuousKontext	0.416	0.158	0.066	0.108
	VeloEdit	0.439	**0.362**	**0.054**	**0.089**
Delete Object	FreeMorph	0.363	0.186	0.147	0.233
	KontinuousKontext	0.661	0.302	0.083	0.138
	VeloEdit	**0.347**	**0.311**	**0.057**	**0.092**
Change Attr. Content	FreeMorph	0.357	0.066	0.142	0.215
	KontinuousKontext	**0.170**	0.146	**0.075**	0.123
	VeloEdit	0.376	**0.151**	0.077	**0.114**
Change Attr. Pose	FreeMorph	**0.335**	0.030	0.119	0.197
	KontinuousKontext	0.748	0.046	0.072	0.131
	VeloEdit	0.577	**0.061**	**0.053**	**0.097**
Change Attr. Color	FreeMorph	0.378	0.295	0.153	0.244
	KontinuousKontext	0.084	0.548	0.104	0.177
	VeloEdit	**0.063**	**0.565**	**0.086**	**0.136**
Change Attr. Material	FreeMorph	0.311	0.087	0.151	0.234
	KontinuousKontext	**0.140**	0.159	0.090	0.153
	VeloEdit	0.183	**0.179**	**0.079**	**0.131**
Change Background	FreeMorph	0.399	0.151	0.254	0.346
	KontinuousKontext	0.203	0.208	0.159	0.224
	VeloEdit	**0.170**	**0.265**	**0.150**	**0.202**
Change Style	FreeMorph	0.315	0.099	0.005	0.008
	KontinuousKontext	**0.017**	0.129	**0.003**	**0.005**
	VeloEdit	0.062	**0.218**	**0.003**	**0.005**

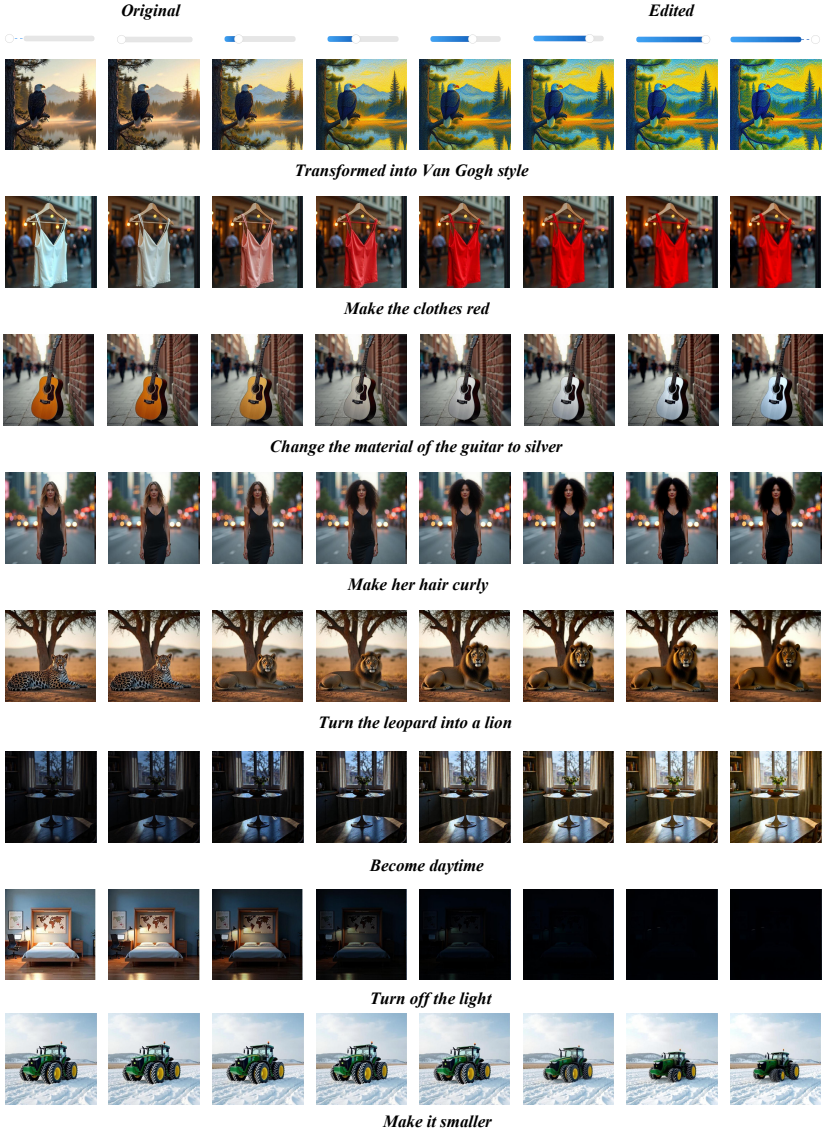

Fig. 9. Additional Visualized Editing Results on Subject200K [51]

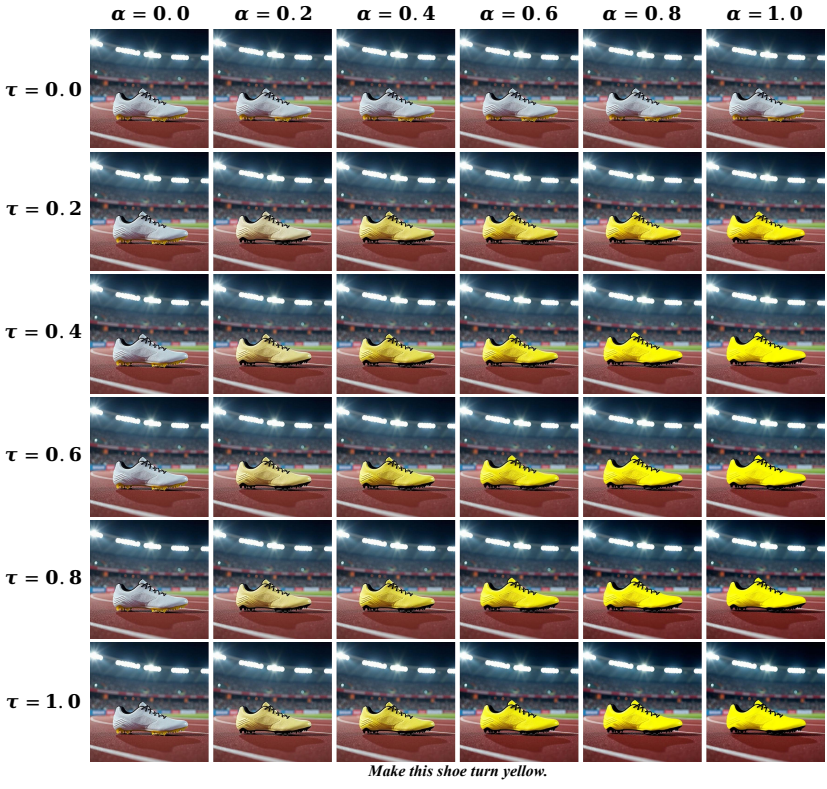

Fig. 10. Visualization results of the ablation study on τ.

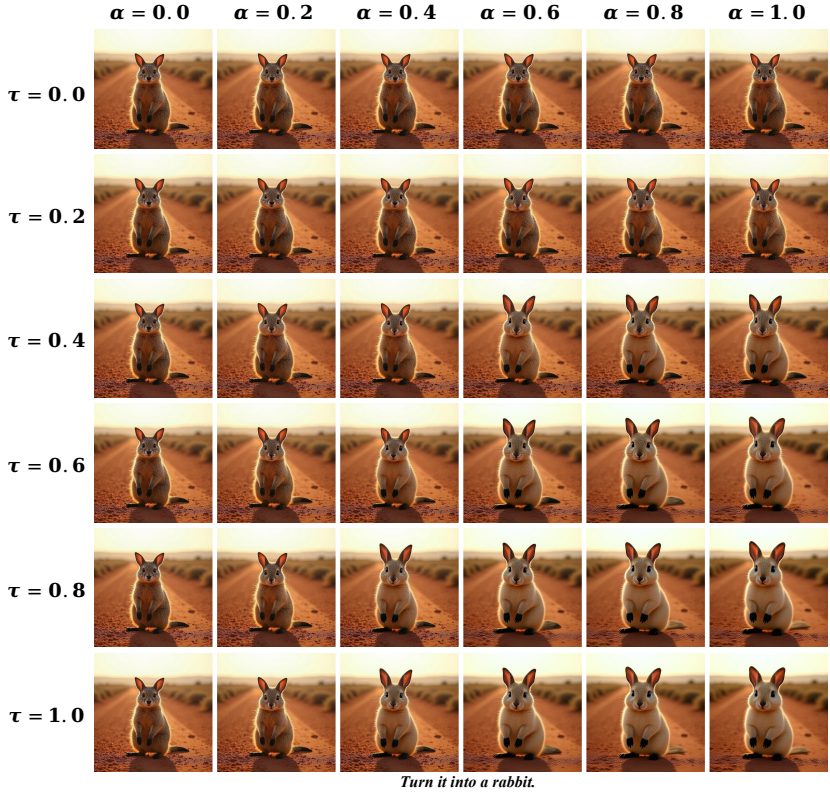

Fig. 11. Visualization results of the ablation study on τ

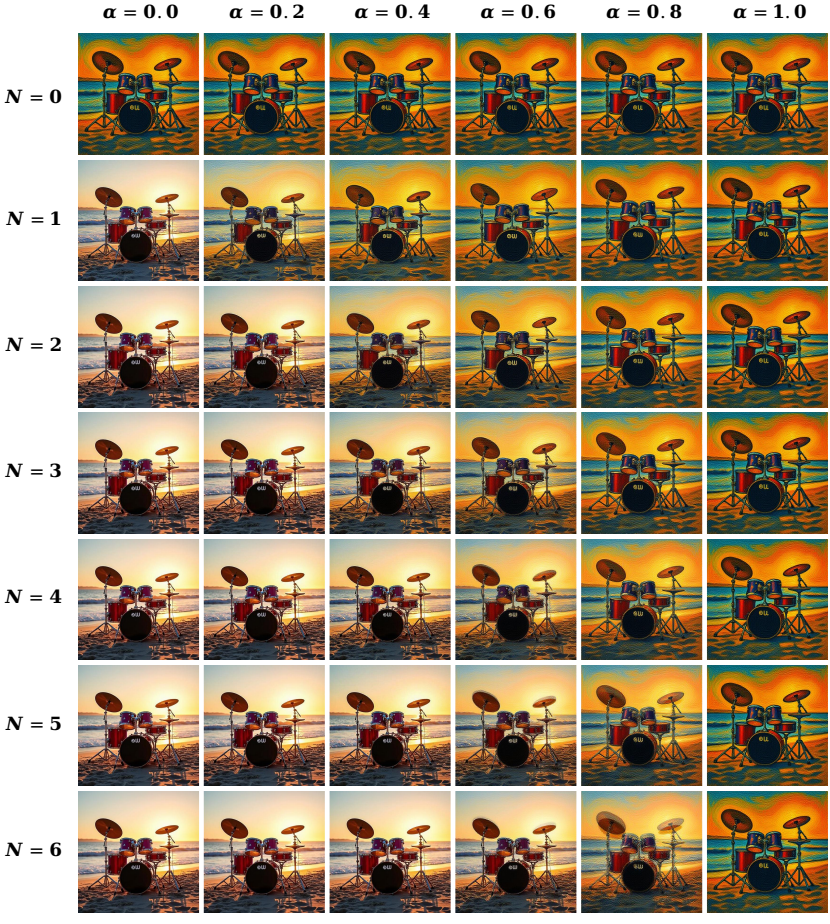

Fig. 12. Visualization results of the ablation study on N

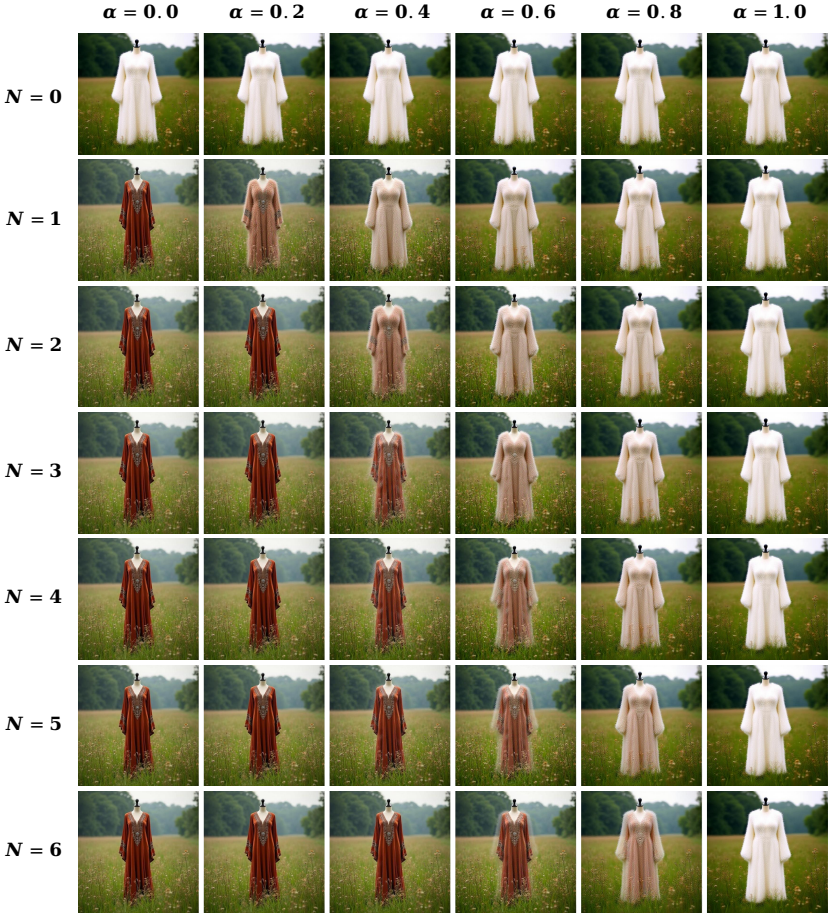

Fig. 13. Visualization results of the ablation study on N

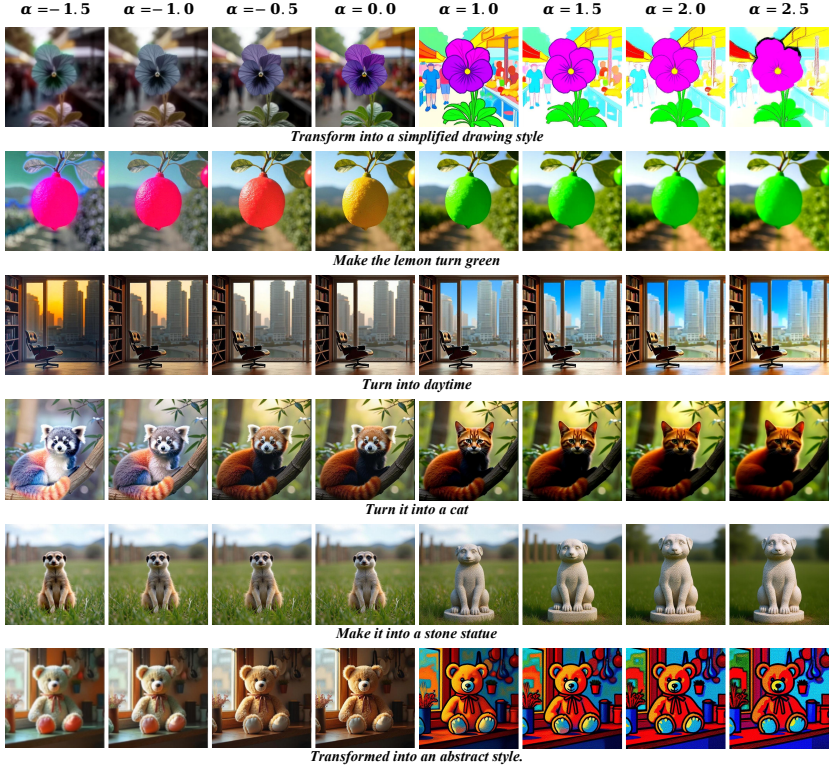

Fig. 14. Visualization results of the ablation study on α

References

1. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
2. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33** (2020) 6840–6851
3. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems **32** (2019)
4. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
5. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
6. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
7. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2022) 10684–10695
8. Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., You, Y.: Open-sora: Democratizing efficient video production for all. arXiv preprint arXiv:2412.20404 (2024)
9. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
10. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
11. Yang, H., Chen, Y., Pan, Y., Yao, T., Chen, Z., Wu, Z., Jiang, Y.G., Mei, T.: Dreammesh: Jointly manipulating and texturing triangle meshes for text-to-3d generation. In: European Conference on Computer Vision, Springer (2024) 162–178
12. Evans, Z., Carr, C., Taylor, J., Hawley, S.H., Pons, J.: Fast timing-conditioned latent audio diffusion. In: Forty-first International Conference on Machine Learning. (2024)
13. Yang, D., Yu, J., Wang, H., Wang, W., Weng, C., Zou, Y., Yu, D.: Diffsound: Discrete diffusion model for text-to-sound generation. IEEE/ACM Transactions on Audio, Speech, and Language Processing **31** (2023) 1720–1733
14. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning. (2024)
15. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
16. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems **35** (2022) 36479–36494
17. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2) (2022) 3

18. Zhu, T., Zhang, S., Shao, J., Tang, Y.: Kv-edit: Training-free image editing for precise background preservation. arXiv preprint arXiv:2502.17363 (2025)
19. Ouyang, Z., Zheng, D., Wu, X.M., Jiang, J.J., Lin, K.Y., Meng, J., Zheng, W.S.: Proedit: Inversion-based editing from prompts done right. arXiv preprint arXiv:2512.22118 (2025)
20. Long, Z., Zheng, M., Feng, K., Zhang, X., Liu, H., Yang, H., Zhang, L., Chen, Q., Ma, Y.: Follow-your-shape: Shape-aware image editing via trajectory-guided region control. arXiv preprint arXiv:2508.08134 (2025)
21. Qin, Z., Tan, Z., Wang, Z., Liu, S., Wang, X.: Spotedit: Selective region editing in diffusion transformers. arXiv preprint arXiv:2512.22323 (2025)
22. Gandikota, R., Materzyńska, J., Zhou, T., Torralba, A., Bau, D.: Concept sliders: Lora adaptors for precise control in diffusion models. In: European Conference on Computer Vision, Springer (2024) 172–188
23. Sharma, P., Jampani, V., Li, Y., Jia, X., Lagun, D., Durand, F., Freeman, B., Matthews, M.: Alchemist: Parametric control of material properties with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2024) 24130–24141
24. Parihar, R., Patashnik, O., Ostashev, D., Babu, R.V., Cohen-Or, D., Wang, K.C.: Kontinuous kontext: Continuous strength control for instruction-based image editing. arXiv preprint arXiv:2510.08532 (2025)
25. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control.(2022). URL https://arxiv. org/abs/2208.01626 **3** (2022)
26. Liu, S., Han, Y., Xing, P., Yin, F., Wang, R., Cheng, W., Liao, J., Wang, Y., Fu, H., Han, C., et al.: Step1x-edit: A practical framework for general image editing. arXiv preprint arXiv:2504.17761 (2025)
27. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., et al.: Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. arXiv preprint arXiv:2506.15742 (2025)
28. Yin, S., Zhang, Z., Tang, Z., Gao, K., Xu, X., Yan, K., Li, J., Chen, Y., Chen, Y., Shum, H.Y., et al.: Qwen-image-layered: Towards inherent editability via layer decomposition. arXiv preprint arXiv:2512.15603 (2025)
29. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
30. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. (2023) 3836–3847
31. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2023) 18392–18402
32. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721 (2023)
33. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2023) 6038–6047
34. Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. arXiv preprint arXiv:2210.11427 (2022)

35. Kulikov, V., Kleiner, M., Huberman-Spiegelglas, I., Michaeli, T.: Flowedit: Inversion-free text-based editing using pre-trained flow models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2025) 19721–19730
36. Beaudouin, G., Li, M., Kim, J., Yoon, S.H., Wang, M.: Delta velocity rectified flow for text-to-image editing. arXiv preprint arXiv:2509.05342 (2025)
37. Wang, P., Shi, Y., Lian, X., Zhai, Z., Xia, X., Xiao, X., Huang, W., Yang, J.: Seededit 3.0: Fast and high-quality generative image editing. arXiv preprint arXiv:2506.05083 (2025)
38. Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: Magicbrush: A manually annotated dataset for instruction-guided image editing. Advances in Neural Information Processing Systems **36** (2023) 31428–31449
39. Tan, Z., Xue, Q., Yang, X., Liu, S., Wang, X.: Ominicontrol2: Efficient conditioning for diffusion transformers. arXiv preprint arXiv:2503.08280 (2025)
40. Zarei, A., Basu, S., Pournemat, M., Nag, S., Rossi, R., Feizi, S.: Slideredit: Continuous image editing with fine-grained instruction control. arXiv preprint arXiv:2511.09715 (2025)
41. Kamenetsky, R., Dorfman, S., Garibi, D., Paiss, R., Patashnik, O., Cohen-Or, D.: Saedit: Token-level control for continuous image editing via sparse autoencoder. arXiv preprint arXiv:2510.05081 (2025)
42. Parihar, R., Agrawal, V., VS, S., Radhakrishnan, V.B.: Compass control: Multi object orientation control for text-to-image generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. (2025) 2791–2801
43. Yang, Y., Chang, D., Fang, Y., SonG, Y.Z., Ma, Z., Guo, J.: Controllable-continuous color editing in diffusion model via color mapping. arXiv preprint arXiv:2509.13756 (2025)
44. Chiu, P.Y., Fang, I., Chen, J.C., et al.: Text slider: Efficient and plug-and-play continuous concept control for image/video synthesis via lora adapters. arXiv preprint arXiv:2509.18831 (2025)
45. Baumann, S.A., Krause, F., Neumayr, M., Stracke, N., Sevi, M., Hu, V.T., Ommer, B.: Continuous, subject-specific attribute control in t2i models by identifying semantic directions. In: Proceedings of the Computer Vision and Pattern Recognition Conference. (2025) 13231–13241
46. Zhu, T., Ren, D., Wang, Q., Wu, X., Zuo, W.: Generative inbetweening through frame-wise conditions-driven video generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. (2025) 27968–27978
47. Guo, Y., Yang, C., Rao, A., Agrawala, M., Lin, D., Dai, B.: Sparsectrl: Adding sparse controls to text-to-video diffusion models. In: European Conference on Computer Vision, Springer (2024) 330–348
48. Cao, Y., Si, C., Wang, J., Liu, Z.: Freemorph: Tuning-free generalized image morphing with diffusion model. (2025)
49. Kabbani, W., Raja, K., Ramachandra, R., Busch, C.: Stablemorph: High-quality face morph generation with stable diffusion. arXiv preprint arXiv:2511.08090 (2025)
50. Wang, Y., Yang, S., Zhao, B., Zhang, L., Liu, Q., Zhou, Y., Xie, C.: Gpt-image-edit-1.5 m: A million-scale, gpt-generated image dataset. arXiv preprint arXiv:2507.21033 (2025)
51. Tan, Z., Liu, S., Yang, X., Xue, Q., Wang, X.: Ominicontrol: Minimal and universal control for diffusion transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2025) 14940–14950

52. Fu, S., Tamir, N., Sundaram, S., Chai, L., Zhang, R., Dekel, T., Isola, P.: Dreamsim: Learning new dimensions of human visual similarity using synthetic data. arXiv preprint arXiv:2306.09344 (2023)
53. Song, K., Han, L., Liu, B., Metaxas, D., Elgammal, A.: Stylegan-fusion: Diffusion guided domain adaptation of image generators. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. (2024) 5453–5463
54. Ju, X., Zeng, A., Bian, Y., Liu, S., Xu, Q.: Direct inversion: Boosting diffusion-based editing with 3 lines of code. arXiv preprint arXiv:2310.01506 (2023)
55. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)