LATENTSWITCH: ONE-STEP TEXT-GUIDED IMAGE EDITING VIA SWITCHABLE NOISY LATENTS

Anonymous authors
Paper under double-blind review



Figure 1: **Given a source image, a target editing instruction, and a mask specifying the region to be modified, our method generates a high-quality edited image in a *single step*.**

ABSTRACT

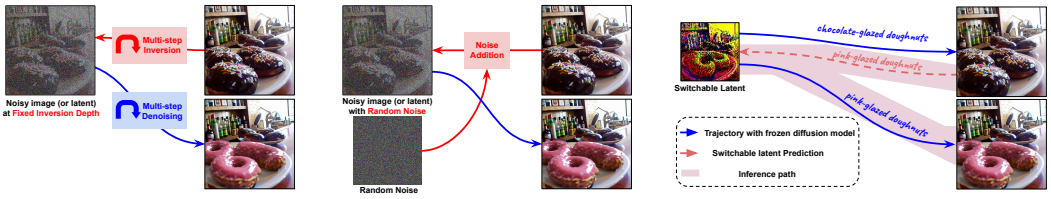
Text-guided image editing aims to modify an input image according to a target prompt while preserving its structure and layout. While diffusion models provide strong generative priors for this task, most prior methods rely on iterative denoising, resulting in high inference cost and unintended changes to unedited regions. To address this, we propose LatentSwitch, a one-step diffusion framework based on the observation that the same latent can produce different semantic contents under different text conditions. Specifically, given a source image and a target prompt, LatentSwitch predicts a shared (switchable) latent that preserves source-specific visual information. Under the source prompt, this latent reconstructs the input image; by switching only the text condition to the target prompt, the pretrained diffusion denoiser directly produces the desired edit in a single step. The latent noise level is adaptively chosen based on the required editing strength, enabling low-noise latents to preserve source details for localized edits and higher-noise latents to support larger semantic changes. Experiments show superior editing quality with substantially lower inference cost than prior image editing methods.

1 INTRODUCTION

Text-guided image editing Hertz et al. (2022); Brooks et al. (2023); Cao et al. (2023); Deutch et al. (2024) aims to modify an input image according to a target instruction while preserving its original structure and visual identity. This task enables a wide range of applications, including object replacement, attribute editing, and appearance transfer Meng et al. (2022); Couairon et al. (2023); Tumanyan et al. (2023). Recent diffusion models Ho et al. (2020); Rombach et al. (2022); Song et al. (2021) have become a dominant foundation for image editing due to their strong generative priors and remarkable ability to synthesize realistic, diverse, and text-aligned visual content. Given an input image and a target prompt, diffusion-based editors can generate visually compelling edits that would be difficult to obtain with conventional image-to-image translation methods Meng et al. (2022); Parmar et al. (2023); Tumanyan et al. (2023); Wang et al. (2023).

However, achieving both faithful and efficient image editing remains challenging for two main reasons. First, most inversion Kadosh et al. (2025); Wallace et al. (2023)-based diffusion editing methods reconstruct an input image by estimating an inverted latent and then repeatedly applying

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Inversion-based (b) Inversion-free (c) **LatentSwitch (Ours)**

Figure 2: **Overview of LatentSwitch.** (a) Inversion-based editing requires costly multi-step inversion and denoising. (b) Inversion-free editing uses random noise, risking loss of source information. (c) In contrast, LatentSwitch predicts a switchable latent directly from the source image and target text (pink arrow). The predicted latent can follow different trajectories under different text guidance through a frozen diffusion model (blue arrow), enabling efficient and source-faithful editing.

a large denoising network over multiple diffusion steps Mokady et al. (2023); Dong et al. (2023); Brack et al. (2024), as shown in Figure 2a. This process incurs substantial computational cost during both inversion and generation, while typically using a fixed, heuristically chosen inversion depth regardless of the required editing strength. Moreover, the inverted noisy sample is derived solely from the source image and therefore does not explicitly encode the target content required for the desired edit.

Second, inversion-free methods Lugmayr et al. (2022); Brooks et al. (2023); Xu et al. (2024); Xie et al. (2023); Ju et al. (2024) often follow a standard generative process initialized from pure Gaussian noise, which can discard source-specific structural and semantic information needed for faithful editing, as shown in Figure 2b. However, many editing tasks require preserving most of the source image while modifying only a specific attribute. Starting from pure noise discards useful structural and semantic cues already present in the source image, forcing the model to reconstruct them from scratch. This can introduce semantic inconsistencies between the edited region and the preserved content, while also causing structural drift and unintended modifications to unrelated regions Hertz et al. (2022); Tumanyan et al. (2023); Cao et al. (2023).

To address these challenges, we revisit the role of noisy latents in diffusion-based image generation Rombach et al. (2022). Our key observation is that the same noisy latent can yield different images when paired with different text conditions Ho & Salimans (2021); Hertz et al. (2022); Parmar et al. (2023). As shown in Figure 2c, the same noisy latent can produce multiple plausible outputs under different text conditions (*i.e.*, it can be denoised into either chocolate or pink-glazed doughnuts depending on the input prompt). Rather than treating the noisy latent merely as an intermediate state along a denoising trajectory, we therefore interpret it as a shared, switchable representation whose semantic realization can be *switched* by changing the conditioning signal.

Based on this observation, we propose LatentSwitch, a one-step diffusion framework for efficient and faithful text-guided image editing. Given a source image (*i.e.*, the chocolate doughnuts in Figure 2) and a target instruction (*i.e.*, converting them into pink-glazed doughnuts in Figure 2), LatentSwitch first predicts a switchable latent representation that is compatible with both the source content and the desired edit. It then switches the text condition from the source prompt to the target prompt and directly generates the edited image with a single diffusion prediction, avoiding a long iterative denoising trajectory. Here, the pretrained diffusion denoiser is used without per-image optimization or task-specific retraining, allowing LatentSwitch to retain the generative prior of the pretrained model while substantially reducing the computational cost of image editing.

Moreover, because the switchable latent is inferred jointly from the source image and the target editing prompt, it preserves source structural and semantic cues instead of discarding them through initialization from pure Gaussian noise. Its noise level is adaptively determined according to the extent of the desired edit. For attribute-level modifications, such as changing only the color, a lower noise level retains more information from the source image, assisting preserve object structure. In contrast, edits requiring larger semantic changes can use a higher noise level, providing sufficient flexibility for the target condition to alter the relevant content. By adapting the noise level to the editing strength, the switchable latent balances faithful preservation of the original image with the flexibility required to realize the target edit, thereby reducing structural drift and unintended changes to unedited regions. Our contributions are summarized as follows:

- We propose LatentSwitch, a novel image editing framework that predicts a switchable latent compatible with both the source image and the target instruction, and directly generates the edited result through a single diffusion step.
- In LatentSwitch, the switchable latent adaptively adjusts its noise level according to the required editing strength, preserving source structure for small edits while allowing greater flexibility for larger semantic modifications.
- Extensive experiments show that LatentSwitch achieves superior editing quality, target-prompt alignment, and source-content preservation compared with existing methods.

2 Related Works

Diffusion-Based Image Editing. Diffusion models enable flexible text-guided editing but must balance semantic modification with preservation of the source image Ho et al. (2020); Song et al. (2021); Rombach et al. (2022). Specifically, SDEdit Meng et al. (2022) edits by partially noising and denoising the input, while DiffEdit Couairon et al. (2023) localizes changes with semantic masks. Other methods preserve structure through cross-attention transfer Hertz et al. (2022), feature injection Tumanyan et al. (2023), mutual self-attention Cao et al. (2023), or attention-constrained editing directions Parmar et al. (2023). Training-based approaches such as InstructPix2Pix Brooks et al. (2023), InstructEdit Wang et al. (2023), and FlexEdit Nguyen et al. (2024) directly condition on the source image and editing instruction. Despite their effectiveness, prior diffusion editors still require iterative denoising and must balance semantic flexibility against source preservation.

Diffusion Inversion-Based Image Editing. Diffusion inversion-based methods map a real image to a latent state or trajectory from which it can be reconstructed and edited using a pretrained diffusion model. DDIM inversion Song et al. (2021) approximately reverses the deterministic sampling process, but accumulated errors can degrade reconstruction under strong text guidance. Later, Null-text Inversion Mokady et al. (2023) improves fidelity by optimizing timestep-specific unconditional embeddings. AIDI Pan et al. (2023) iteratively corrects inversion errors, and Tight Inversion Kadosh et al. (2025) introduces image conditioning to better balance reconstruction and editability. The resulting source trajectory can also support attention- and feature-based manipulation methods such as Prompt-to-Prompt Hertz et al. (2022), Plug-and-Play Diffusion Features Tumanyan et al. (2023), and MasaCtrl Cao et al. (2023). Although these approaches preserve source details effectively, they rely on image-specific inversion and iterative denoising with manually predetermined inversion schedules and terminal noise levels, resulting in considerable inference cost and limited adaptability to the magnitude of the requested edit.

Few-Step Diffusion Image Editing. Recent methods reduce editing latency by employing shortened diffusion trajectories or distilled generative models Yin et al. (2024). Specifically, InfEdit Xu et al. (2024) avoids explicit inversion through virtual inversion, jointly propagating source and target branches over a reduced sampling trajectory. TurboEdit Deutch et al. (2024) adapts few-step distilled diffusion models to text-guided editing while addressing the loss of source information caused by compact denoising trajectories. TurboEdit Wu et al. (2024) similarly combines accelerated inversion and generation to achieve efficient editing. Later, SwiftEdit Nguyen et al. (2025) further enables one-step editing by learning an image-conditioned representation compatible with a one-step diffusion model. While these approaches substantially reduce inference time, they may still require dedicated inversion procedures, auxiliary conditioning networks, or task-specific training. Moreover, most still employ a fixed corruption level regardless of the magnitude and spatial extent of the requested edit.

3 Proposed Method

Figure 3 shows the overall pipeline of our LatentSwitch framework. Our goal is to predict a shared, switchable noisy latent $\hat{\mathbf{Z}}_{\hat{\mathrm{t}}}$ that can be transformed into the desired edited image $\mathbf{I}_{\mathrm{edt}}$ through a single evaluation of a frozen diffusion model. Given a source image $\mathbf{I}_{\mathrm{src}} \in \mathbb{R}^{H\times W\times 3}$, a target prompt $\mathrm{p}_{\mathrm{tgt}}$, and a binary mask $\mathbf{M} \in [0,1]^{H\times W\times 1}$, our latent predictor $\mathcal{P}$ estimates both the switchable noisy latent and its corresponding noise level (*i.e.*, diffusion timestep). Here, H and W denote the image height and width, respectively. The predicted latent $\hat{\mathbf{Z}}_{\hat{\mathrm{t}}}$ is subsequently denoised once by a frozen diffusion denoiser to obtain the edited image.

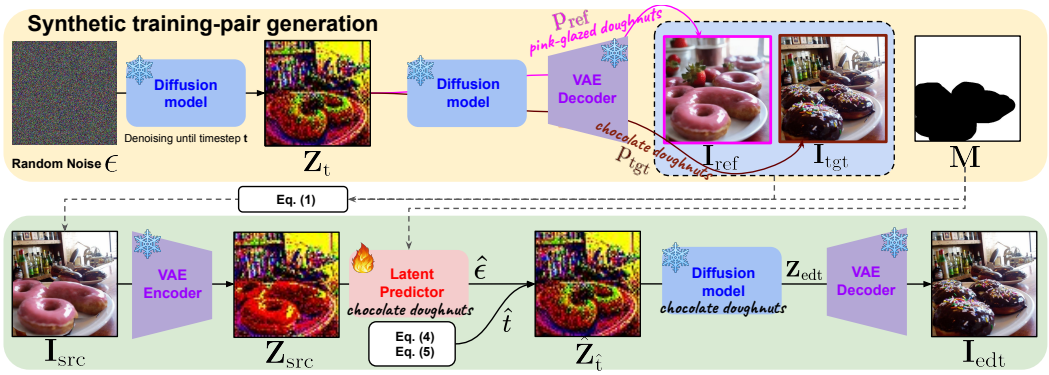


Figure 3: **Overall pipeline of LatentSwitch**. *Top:* Training pairs are synthesized from a shared noisy latent $\mathbf{Z}_t$ using reference and target prompts, and combined by mask $\mathbf{M}$ to form $\mathbf{I}_{\text{src}}$. *Bottom:* A latent predictor estimates the noise component and timestep from $\mathbf{Z}_{\text{src}}$ to construct $\hat{\mathbf{Z}}_{\hat{t}}$, which is edited by a single frozen denoiser evaluation and VAE decoding to produce $\mathbf{I}_{\text{edt}}$.

3.1 Synthetic Training-Pair Generation.

Since our training requires not only paired source and target images but also their corresponding switchable latent, we construct synthetic training pairs using a pretrained diffusion model. We first sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and generate a latent $\mathbf{Z}_{\text{t}}$ at a randomly sampled diffusion timestep $\text{t} \in [300, 999]$. Using the same noisy latent $\mathbf{Z}_{\text{t}}$ with two different text conditions—a reference prompt p_{ref} and a target prompt p_{tgt}—we generate a reference image $\mathbf{I}_{\text{ref}}$ and a target image $\mathbf{I}_{\text{tgt}}$, respectively. Here, the text descriptions are obtained either through BLIP-based image captioning Li et al. (2022) or from a predefined collection of local descriptions; further details are provided in the appendix.

In parallel, we sample a binary mask $\mathbf{M}$ by combining free-form masks (*e.g.*, strokes, lines, and circular regions) generated following LaMa Suvorov et al. (2022) with randomly selected semantic masks obtained using FastSAM Zhao et al. (2023). We then construct the source image by spatially combining the target and reference images as:

$$\mathbf{I}_{\text{src}} = \mathbf{M} \odot \mathbf{I}_{\text{tgt}} + (1 - \mathbf{M}) \odot \mathbf{I}_{\text{ref}}, \tag{1}$$

where $\odot$ denotes element-wise multiplication. Accordingly, regions where $\mathbf{M} = 1$ are taken from the target image and treated as content to be preserved, whereas regions where $\mathbf{M} = 0$ are taken from the reference image and designated for editing toward the target semantics. The resulting source image $\mathbf{I}_{\text{src}}$, target image $\mathbf{I}_{\text{tgt}}$, target prompt p_{tgt}, mask $\mathbf{M}$, switchable noisy latent $\mathbf{Z}_{\text{t}}$, and its corresponding timestep t are then used to train the latent predictor $\mathcal{P}$.

3.2 Switchable Latent Prediction

From the source image $\mathbf{I}_{\text{src}}$, the latent predictor $\mathcal{P}$ estimates a noisy latent $\hat{\mathbf{Z}}_{\hat{\text{t}}}$ that can be transformed into the desired edited image through a single evaluation of the pretrained diffusion denoiser. We first encode the source image $\mathbf{I}_{\text{src}}$ using the frozen VAE encoder, resulting in source latent $\mathbf{Z}_{\text{src}}$.

In the standard forward diffusion process, a clean latent $\mathbf{Z}_0$ is perturbed at timestep t as:

$$\mathbf{Z}_{\text{t}} = \sqrt{\bar{\alpha}_{\text{t}}}\mathbf{Z}_0 + \sqrt{1 - \bar{\alpha}_{\text{t}}}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

where $\bar{\alpha} = \{\bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_T\}$ is the cumulative product of the predefined variance schedule $\{\beta_*\}_{*=1}^{T}$.

Unlike the conventional forward process, the appropriate timestep and Gaussian noise are unknown for a given editing request. We therefore condition our UNet-based latent predictor $\mathcal{P}$ on the source latent $\mathbf{Z}_{\text{src}}$, mask $\mathbf{M}$, and the target prompt p_{tgt}, and directly predict the *scaled noise component*:

$$\hat{\epsilon} = \mathcal{P}\left(\mathbf{Z}_{\text{src}}, \mathbf{M}, \text{p}_{\text{tgt}}\right), \tag{3}$$

where $\hat{\epsilon}$ is trained to represent the complete noise term $\sigma_{\text{t}}\epsilon$ ($\sigma_{\text{t}} = \sqrt{1 - \bar{\alpha}_{\text{t}}}$), rather than unit Gaussian noise ϵ alone. Consequently, its standard deviation indicates the corresponding diffusion noise level.

We estimate the timestep by matching the standard deviation of the predicted noise to the scheduler:

$$\hat{\mathrm{t}} = \arg\min_{*\in 1,\dots,T} \left|\mathrm{Std}(\hat{\epsilon}) - \sigma_*\right|. \tag{4}$$

Using the predicted noise $\hat{\epsilon}$ and timestep $\hat{\mathrm{t}}$, we construct a source-consistent noisy latent as:

$$\hat{\mathbf{Z}}_{\hat{\mathrm{t}}} = \sqrt{\bar{\alpha}_{\hat{\mathrm{t}}}}\mathbf{Z}_{\mathrm{src}} + \hat{\epsilon}. \tag{5}$$

Finally, the predicted switchable latent is passed to the pretrained denoiser together with its corresponding timestep $\hat{\mathrm{t}}$ and target prompt $\mathrm{p}_{\mathrm{tgt}}$, producing the edited latent $\mathbf{Z}_{\mathrm{edt}}$. The edited latent is then decoded by the frozen VAE decoder to obtain the final edited image $\mathbf{I}_{\mathrm{edt}}$.

3.3 Adaptive timestep prediction.

Different editing requests require different levels of latent corruption (*i.e.*, timestep). Specifically, a minor localized edit should preserve most of the source information, whereas a substantial semantic modification requires a noisier latent that provides greater generative flexibility. As described in Section 3.1, the reference image $\mathbf{I}_{\mathrm{ref}}$ and target image $\mathbf{I}_{\mathrm{tgt}}$ are generated by branching from the same shared latent $\mathbf{Z}_{\mathrm{t}}$ at timestep t under different text conditions. Consequently, the semantic discrepancy between the source and target images is governed by $\mathbf{t}$: larger timesteps generally produce more substantial changes, while smaller timesteps yield more conservative edits. Based on this relationship, we supervise the timestep predictor $\mathcal{P}$ using the noise scale associated with the timestep employed during synthetic pair generation:

$$\mathcal{L}_{\mathrm{t}} = \left\|\mathrm{Std}(\hat{\boldsymbol{\epsilon}}) - \sigma_t\right\|_2^2, \tag{6}$$

encouraging the magnitude of the predicted noise to reflect the degree of semantic variation required within the editable region.

3.4 Training objectives.

Together with the timestep loss $\mathcal{L}_{\mathrm{t}}$, which enables the latent predictor to estimate a timestep $\hat{\mathrm{t}}$ adapted to the required editing strength, we employ standard image-level reconstruction objectives. These include L1 reconstruction loss $\mathcal{L}_1$, perceptual loss $\mathcal{L}_{\mathrm{per}}$, and adversarial loss $\mathcal{L}_{\mathrm{adv}}$ between edited image $\mathbf{I}_{\mathrm{edt}}$ and target image $\mathbf{I}_{\mathrm{tgt}}$. Furthermore, we introduce a distribution-matching loss $\mathcal{L}_{\mathrm{dm}}$ to encourage the generated content within the edited region to follow the target-image distribution. Given the complementary mask $\mathbf{M}^\dagger = (1 - \mathbf{M})$, we construct an auxiliary source image by combining the current edited output with the reference image:

$$\mathbf{I}^\dagger_{\mathrm{src}} = \mathbf{M}^\dagger \odot \mathbf{I}_{\mathrm{edt}} + \left(1 - \mathbf{M}^\dagger\right) \odot \mathbf{I}_{\mathrm{ref}}. \tag{7}$$

We then process $\mathbf{I}^\dagger_{\mathrm{src}}$ using the same editing pipeline while stopping gradient propagation, obtaining an auxiliary edited latent $\mathbf{Z}^\dagger_{\mathrm{edt}}$. Based on this auxiliary prediction, the distribution-matching loss $\mathcal{L}_{\mathrm{dm}}$ is then defined as:

$$\mathcal{L}_{\mathrm{dm}} = \left|\mathbf{Z}_{\mathrm{edt}} - \mathrm{sg}\left(\mathbf{Z}_{\mathrm{edt}} - \Delta\mathbf{Z}\right)\right|_2^2, \tag{8}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operation and the latent-space update direction $\Delta\mathbf{Z}$ is defined:

$$\Delta\mathbf{Z} = \frac{\mathbf{Z}_{\mathrm{edt}} - \mathbf{Z}^\dagger_{\mathrm{edt}}}{\left\|\mathbf{Z}_{\mathrm{tgt}} - \mathbf{Z}_{\mathrm{edt}}\right\|_{1,\mathrm{mean}}}, \tag{9}$$

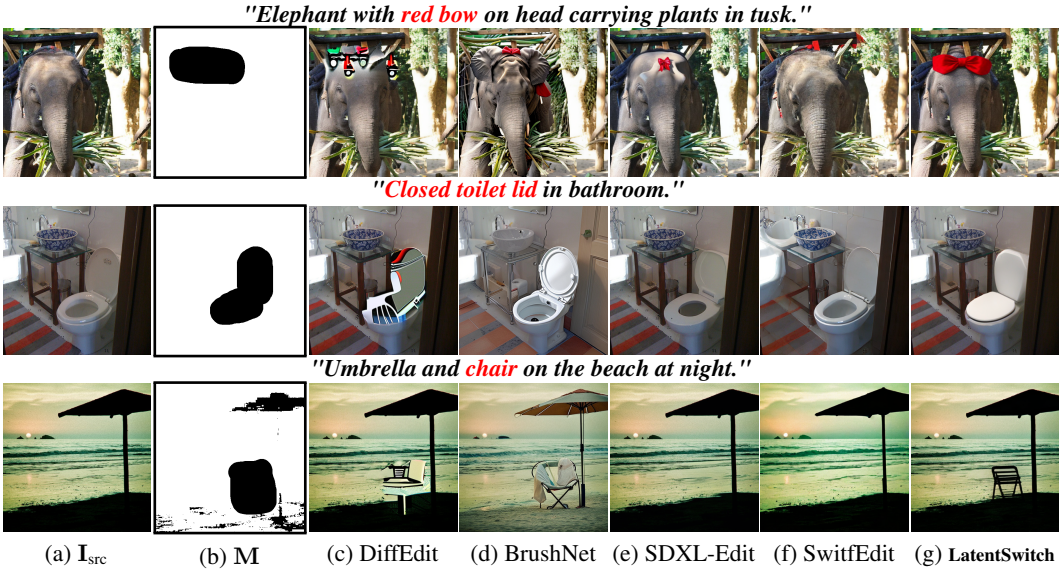
where $\mathbf{Z}_{\mathrm{tgt}}$ denotes the target latent obtained by encoding the target image $\mathbf{I}_{\mathrm{tgt}}$ with the frozen VAE encoder. Finally, the overall training objective is defined as a weighted combination of the $\mathcal{L}_{\mathrm{t}}$, $\mathcal{L}_1$, $\mathcal{L}_{\mathrm{per}}$, $\mathcal{L}_{\mathrm{adv}}$, and $\mathcal{L}_{\mathrm{dm}}$.

4 Experiments

Dataset. Since the training pairs are generated synthetically as described in Section 3.1, LatentSwitch does not require an external training dataset. For evaluation, we use the MagicBrush benchmark Ju et al. (2024) and HumanEdit Bai et al. (2024) benchmark, which provides source images, editing masks, and target text prompts, offering a diverse and challenging testbed for image editing.

Table 1: **Quantitative comparison on image editing benchmarks.** The **best** and **second-best** results are highlighted in **red** and **blue**, respectively. Latency is measured on an NVIDIA A100 40GB GPU.

Methods	MagicBrush					HumanEdit					User study (%)	# of steps$_\downarrow$	Latency (FPS)$_\uparrow$
	Distortion			Text-alignment		Distortion			Text-alignment				
	PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$	HPS$_\uparrow$	CLIP$_\uparrow$	PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$	HPS$_\uparrow$	CLIP$_\uparrow$			
DiffEdit	27.2388	0.8526	0.0832	0.2040	0.2751	27.9063	0.8807	0.0637	0.2103	0.2591	0.00	50	0.2175
BrushNet	16.9024	0.6721	0.2972	**0.2488**	**0.3103**	19.1790	0.7137	0.2539	**0.2420**	0.2818	**12.83**	50	0.2940
SDXL-Edit	**29.5925**	**0.9536**	**0.0483**	0.2305	0.3001	**28.7462**	**0.9332**	**0.0589**	0.2317	0.2717	11.55	20	0.3720
SwiftEdit	21.4399	0.7556	0.1923	0.2272	0.3040	21.5090	0.7740	0.1868	0.2380	**0.2819**	7.70	2	4.0849
LatentSwitch	**27.6732**	**0.8579**	**0.0726**	**0.2536**	**0.3113**	**26.9845**	**0.8367**	**0.0829**	**0.2519**	**0.2850**	**67.95**	1	1.9694



(a) $\mathbf{I}_{\text{src}}$ (b) $\mathbf{M}$ (c) DiffEdit (d) BrushNet (e) SDXL-Edit (f) SwitfEdit (g) **LatentSwitch**

Figure 4: **Visual comparison on various image editing benchmark dataset.** The text prompt shown above each image is used as the corresponding editing instruction. (Zoom-in for best view)

Metrics. We evaluate the edited results from two perspectives: image distortion and text alignment. To assess image distortion in accordance with the objective of image editing, we measure distortion in the unmasked region, which is expected to remain unchanged, using PSNR, SSIM, and LPIPS Zhang et al. (2018). To evaluate semantic alignment between the edited image and the target text prompt, we use HPS v2 Wu et al. (2023) and CLIP score Hessel et al. (2021).

Model configurations. We initialize the noise predictor $\mathcal{P}$ using the pretrained SD-Turbo model Sauer et al. (2024). During training, we use source-target image pairs generated according to the protocol described in Section 3.1, with a batch size of 4. The model is trained for 300k iterations using the AdamW Loshchilov & Hutter (2019) optimizer with an initial learning rate of 1×10^{-4}, which is reduced by half every 100k iterations.

4.1 Comparison with State-Of-the-Art Methods

Table 1 compares LatentSwitch with recent diffusion-based image editing methods under their original evaluation settings, including the prescribed number of sampling steps. As shown in the table, LatentSwitch achieves strong performance across distortion and text alignment, despite requiring only a single diffusion step. The efficiency results on the right further show that multi-step methods incur substantially higher latency. Specifically, DiffEdit is particularly slow because it additionally requires an inversion process before editing. Although LatentSwitch is slightly slower than SwiftEdit due to its UNet-based noise predictor $\mathcal{P}$, it retains comparable efficiency while delivering substantially better performance across all evaluation metrics. Furthermore, Table 1 reports the results of a user study in which 13 participants were asked to select the best edited image among five candidate outputs, given the source image and target text prompt. As shown, LatentSwitch achieves the highest user preference among the compared methods.

Moreover, Figure 4 provides the corresponding qualitative comparisons. BrushNet and SwiftEdit, which are not explicitly guided by the editing mask, often alter unedited regions and disrupt the structure of the source image, as illustrated by the unintended person generated in the second row of

Table 2: **Quantitative comparison under different number of iterations.**

# of iterations	Distortion PSNR↑	SSIM↑	LPIPS↓	Text-alignment HPS↑	CLIP↑	Latency (FPS)↑
1	27.6732	0.8579	0.0726	0.2536	0.3113	1.9694
2	27.6945	0.8586	0.0717	0.2600	0.3135	1.7707
5	27.7501	0.8598	0.0711	0.2732	0.3157	1.2230
10	27.6797	0.8591	0.0724	0.2636	0.3142	0.8982

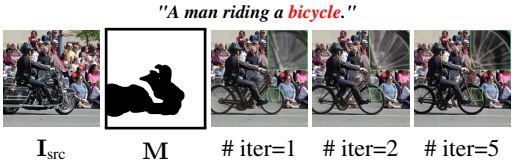


Figure 5: **Visual comparison under different number of iterations.** (Zoom-in for best view)

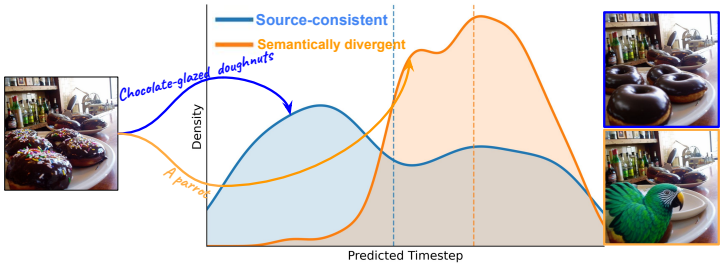


Figure 6: **Predicted timestep $\hat{\mathbf{t}}$ for source-consistent and semantically divergent prompts.** LatentSwitch assigns larger timesteps to edits requiring greater semantic changes.

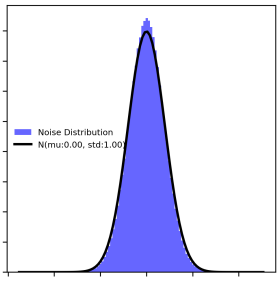


Figure 7: **Distribution of $\hat{\epsilon}$ after normalization.**

Figure 4. In contrast, mask-guided SDXL-Edit better preserves the source image, consistent with its favorable distortion scores, but frequently fails to follow the target prompt and produces generic inpainting results. LatentSwitch achieves a better balance between source preservation and text alignment, accurately reflecting the target instruction while generating coherent edited content.

4.2 Number of Iterations

Table 2 compares LatentSwift using different numbers of refinement iterations. For multi-step inference, we update the input to the latent predictor $\mathcal{P}$ by combining the original source latent $\mathbf{Z}$src with the previously edited latent $\mathbf{Z}$edt according to the downsampled mask as $\mathbf{M}$ as $\mathbf{M}_{\downarrow} \odot \mathbf{Z}_{\text{src}} + (1 - \mathbf{M}_{\downarrow}) \odot \mathbf{Z}_{\text{edt}}$, where $_{\downarrow}$ denotes nearest downsampling. As the number of iterations increases, the editable region becomes better aligned with the target prompt p_{tgt}, resulting in improved text-alignment scores. Interestingly, this stronger semantic alignment also improves the distortion metrics by providing more target-relevant information within the editable region. However, performance slightly degrades when too many iterations (*i.e.*, 10 iterations) are used, which is likely due to error accumulation in the frozen SD-Turbo denoiser, whose generation quality also declines under excessive iterative application. Furthermore, Figure 5 provides the corresponding qualitative comparison. With additional iterations, the edited regions become progressively sharper and more detailed; for example, the spokes of the bicycle wheel are reconstructed more clearly. This improvement arises because each iteration reuses a better-aligned edited output as the input to the next refinement step.

4.3 Predicted Timestep Analysis

Figure 6 compares the predicted timestep distributions for two types of target prompts: a source-consistent prompt and a semantically divergent prompt. For example, given an image of chocolate-glazed doughnuts, we use 'Chocolate-glazed doughnuts' as the source-consistent prompt and 'A parrot' as the semantically divergent prompt. Since LatentSwitch adaptively determines the noise level (*i.e.*, timestep) according to the required editing strength, it assigns smaller timesteps to source-consistent prompts, for which little or no modification is needed, and larger timesteps to semantically divergent prompts, which require substantial changes. These results demonstrate that LatentSwitch adjusts the noise level according to the desired degree of editing, thereby reducing structural drift and unintended modifications in unedited regions.

4.4 Predicted Noise Analysis

Since the latent predictor estimates the scaled noise component $\hat{\epsilon}$, Figure 7 further analyzes its distribution. To recover the corresponding unit-noise estimate, we first normalize the prediction

Table 3: **Source-consistent recovery using the source prompt.** The metrics are computed over the entire image.

PSNR	SSIM	LPIPS
18.7159	0.5508	0.2030



Figure 8: **Visual results of source-consistent recovery using the source prompt.**

Table 4: **Quantitative comparison with and without shared latent training.** Without shared latent training, the reference and target images are obtained from independent noise samples at the same timestep, preventing them from sharing a common latent.

Latent	Distortion PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$	Text-alignment HPS$_\uparrow$	CLIP$_\uparrow$
Independent	20.1392	0.7312	0.1890	0.2258	0.3059
Shared (Ours)	**27.6732**	**0.8579**	**0.0726**	**0.2536**	**0.3113**

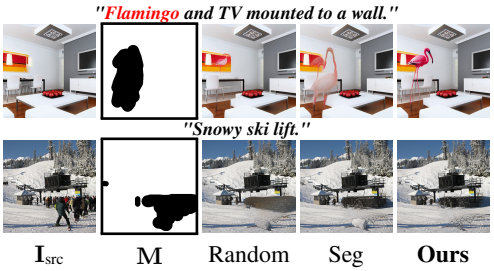


Figure 9: **LatentSwitch with different mask-generation strategies.**

by the noise scale associated with the predicted timestep (*i.e.*, $\sqrt{1-\bar{\alpha}_{\hat{t}}}$). As shown in Figure 7, the normalized prediction approximately follows a Gaussian distribution, consistent with the noise distribution expected by the pretrained diffusion denoiser. This indicates that the latent predictor produces noise-like inputs that remain compatible with the pretrained denoising process, without requiring explicit supervision to enforce a Gaussian distribution on $\hat{\epsilon}$. For visualizations of the predicted $\hat{\epsilon}$ and the resulting noisy latent $\hat{\mathbf{Z}}_{\hat{t}}$ with more details, please refer to the Appendix.

4.5 Switchable Latent Analysis

Our goal is to identify a switchable latent that can be denoised into either the source or the edited content, depending on the input text prompt. To evaluate its ability to recover the source content, we provide the source-consistent prompt during inference. Table 3 reports the resulting distortion metrics, while Figure 8 presents the corresponding outputs. As shown, the recovered images still exhibit minor structural changes, leading to suboptimal distortion scores. For example, the car changes from an angular shape to a more rounded one. This is likely caused by restricting the timestep range to $[300, 999]$; even the smallest timestep (*i.e.*, $t = 300$) introduces a non-negligible level of corruption and may therefore cause unavoidable content changes. Nevertheless, while the source reconstruction is not perfect, these results indicate that the predicted switchable latent retains sufficient information to recover content close to the original image.

4.6 Ablation Studies

Switchable Latent Prediction. The key to LatentSwitch is predicting a shared, switchable latent representation of the source and target images, enabling target-guided editing without training the pretrained denoiser. To validate this design, Table 4 considers a variant where the source and target are generated from different latents, forcing the model to predict the target latent independently of the source. This consistently degrades performance, as predicting a source-irrelevant latent is substantially harder, evidenced by the large timestep prediction error. Moreover, removing the switchable latent causes a significant drop in distortion metrics, confirming that the switchable latent is crucial for preserving source fidelity.

Mask generation. As we construct the binary mask $\mathbf{M}$ by combining free-form random masks Suvorov et al. (2022), which are independent of image semantics, with segmentation-based semantic masks, we conduct ablation studies on different mask-generation strategies during training. Figure 9 compares LatentSwitch trained using only random masks or only segmentation-based masks. When trained with random masks, LatentSwitch tends to remove existing content and often fails to generate new content that is well aligned with the target prompt Conversely, using only segmentation-based masks encourages generation of target-relevant content but often fails to account for the surrounding

Table 5: **Quantitative comparison with and without the proposed distribution-matching loss $\mathcal{L}_{\text{dm}}$.** We provide diverse comparisons using different numbers of refinement iterations.

$\mathcal{L}_{\text{dm}}$	# of iterations	Distortion PSNR$_\uparrow$	Distortion SSIM$_\uparrow$	Distortion LPIPS$_\downarrow$	Text-alignment HPS$_\uparrow$	Text-alignment CLIP$_\uparrow$
✗	1	26.3057	0.8235	0.0856	0.2510	0.3098
✓		**27.6732**	**0.8579**	**0.0726**	**0.2536**	**0.2413**
✗	2	26.5091	0.8256	0.0851	0.2561	0.3097
✓		**27.6945**	**0.8586**	**0.0717**	**0.2600**	**0.2435**
✗	5	26.6130	0.8285	0.0848	0.2580	0.3095
✓		**27.7501**	**0.8598**	**0.0711**	**0.2732**	**0.2457**

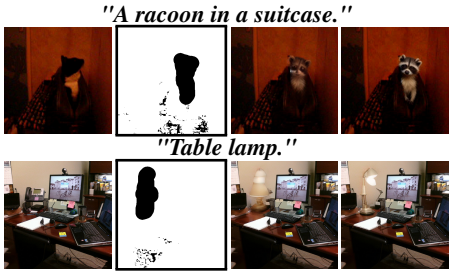


Figure 10: **Visual comparison with and without the proposed $\mathcal{L}_{\text{dm}}$**



Figure 11: **Single-step visual results on real-world images.** The images are from public sources with manually annotated masks; the text below each figure serves as the editing instruction.

context. Our strategy, which combines both mask types, produces edited content that better reflects the target prompt while remaining coherent with the preserved surrounding regions.

Use of distribution-matching loss $\mathcal{L}_{\text{dm}}$. During training, we use the distribution-matching loss $\mathcal{L}_{\text{dm}}$ to encourage the generated regions to follow the input distribution of the latent predictor $\mathcal{P}$. Table 5 shows the effect of $\mathcal{L}_{\text{dm}}$. With a single iteration, introducing distribution-matching loss $\mathcal{L}_{\text{dm}}$ consistently improves both distortion- and text-alignment-based metrics by aligning the generated outputs with the real-image distribution. The benefit becomes more pronounced with additional iterations, as maintaining distributional consistency between successive outputs and the predictor input helps mitigate error accumulation across refinement steps. Furthermore, Figure 10 compares results with and without the distribution-matching loss $\mathcal{L}_{\text{dm}}$. Using $\mathcal{L}_{\text{dm}}$ produces more realistic edited images.

5 Real-World Cases

Figure 11 presents visual results on real-world images collected from publicly available sources, demonstrating the effectiveness of LatentSwitch. When the editing mask targets the man's clothing and the Yoda character on the man's back in the second row, LatentSwitch successfully modifies the specified regions according to the corresponding target prompts while preserving the surrounding content.

6 Conclusion

In this work, we introduce LatentSwitch, a single-step diffusion framework for efficient and faithful text-guided image editing. LatentSwitch predicts a switchable latent that preserves source-specific information while enabling semantic changes through text-condition switching. By adaptively adjusting the noise level according to the required editing strength, it balances source preservation and target-prompt alignment. Extensive experiments show that LatentSwitch achieves strong editing quality with substantially lower cost than prior diffusion-based editing methods.

REFERENCES

Jinbin Bai, Wei Chow, Ling Yang, Xiangtai Li, Juncheng Li, Hanwang Zhang, and Shuicheng Yan. Humanedit: A high-quality human-rewarded dataset for instruction-based image editing. *arXiv preprint arXiv:2412.04280*, 2024.

Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *CVPR*, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *ICCV*, 2023.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *ICLR*, 2023.

Gilad Deutch, Rinon Gal, Daniel Garibi, Or Patashnik, and Daniel Cohen-Or. Turboedit: Text-based image editing using few-step diffusion models. In *SIGGRAPH Asia*, 2024.

Wenkai Dong, Song Xue, Xiaoyue Duan, and Shumin Han. Prompt tuning inversion for text-driven image editing using diffusion models. In *ICCV*, 2023.

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

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *NeurIPS*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *ECCV*, 2024.

Edo Kadosh, Nir Goren, Or Patashnik, Daniel Garibi, and Daniel Cohen-Or. Tight inversion: Image conditioned inversion for real image editing. In *ICCV*, 2025.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *ICLR*, 2022.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023.

Trong-Tung Nguyen, Duc-Anh Nguyen, Anh Tran, and Cuong Pham. Flexedit: Flexible and controllable diffusion-based object-centric image editing. *arXiv preprint arXiv:2403.18605*, 2024.

Trong-Tung Nguyen, Quang Nguyen, Khoi Nguyen, Anh Tran, and Cuong Pham. Swiftedit: Lightning fast text-guided image editing via one-step diffusion. In *CVPR*, 2025.

Zhihong Pan, Riccardo Gherardi, Xiufeng Xie, and Stephen Huang. Effective real image editing with accelerated iterative diffusion inversion. In *ICCV*, 2023.

Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *arXiv preprint arXiv:2403.12015*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, 2022.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023.

Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *CVPR*, 2023.

Qian Wang, Biao Zhang, Michael Birsak, and Peter Wonka. Instructedit: Improving automatic masks for diffusion-based image editing with user instructions. *arXiv preprint arXiv:2305.18047*, 2023.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Zongze Wu, Nicholas Kolkin, Jonathan Brandt, Richard Zhang, and Eli Shechtman. Turboedit: Instant text-based image editing. In *ECCV*, 2024.

Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *CVPR*, 2023.

Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. *CVPR*, 2024.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

A Appendix

B Limitations

Although LatentSwitch achieves strong image-editing performance, as demonstrated quantitatively in Table 1 and qualitatively in Figures 4 and 11, its use of a diffusion-based backbone introduces an inherent limitation with respect to input resolution. Specifically, the diffusion model is typically trained and optimized for a fixed spatial resolution, making it difficult to directly process images whose dimensions differ substantially from those encountered during training. Consequently, images with arbitrary resolutions generally need to be manually cropped, resized, or otherwise adjusted to match the expected input size before being passed to the editing pipeline.

C Feature Visualization

Figure 12 further visualizes $\hat{\epsilon}$ and the resulting noisy latent $\hat{\mathbf{Z}}_{\hat{\mathrm{t}}}$ together with the target prompt and edited output. Although the overall value distribution remains approximately Gaussian as shown in Figure 7, the predicted noise exhibits spatial structures related to the target semantics, such as pizza- or raccoon-like patterns. This suggests that LatentSwitch preserves the expected diffusion-noise statistics while embedding target-specific information into its spatial organization.

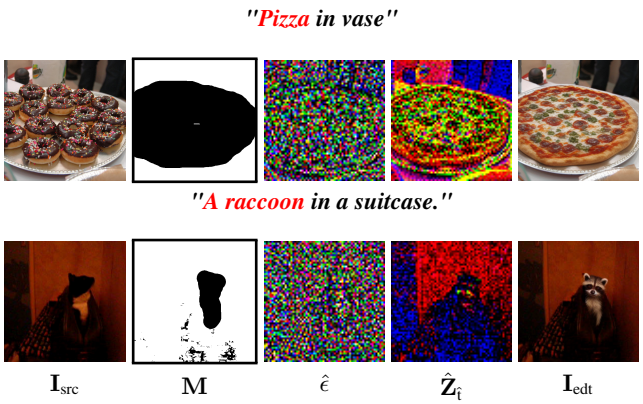


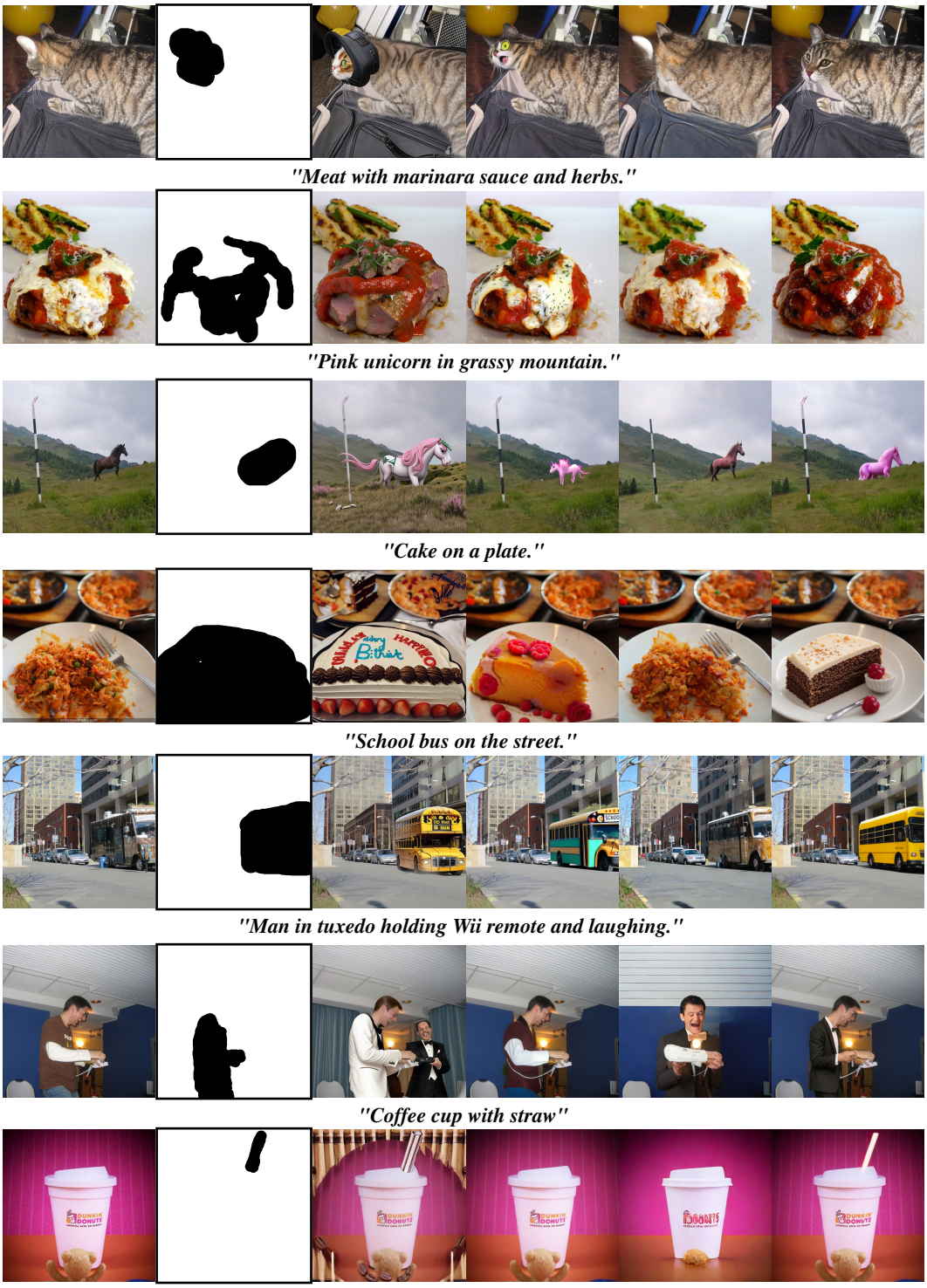
Figure 12: **Visualizations of $\hat{\epsilon}$ and $\hat{\mathbf{Z}}_{\hat{\mathbf{t}}}$.** While $\hat{\epsilon}$ follows an Gaussian, it encodes target-specific semantic information.

D Text descriptions for Synthetic Training Pair Generation

In Section 3.1, we generate the reference image $\mathbf{I}_{\mathrm{ref}}$ and target image $\mathbf{I}_{\mathrm{tgt}}$ using different text conditions, namely the reference prompt $\mathrm{p}_{\mathrm{ref}}$ and target prompt $\mathrm{p}_{\mathrm{tgt}}$, respectively. For prompt construction, we sample sentences from a predefined dictionary of 100,000 prompts generated by ChatGPT, including imaginative descriptions (*e.g.*, "A woman skateboarding beneath a UFO in the sky."). Since such LLM-generated prompts can be biased toward unrealistic scenarios, we additionally use captions extracted from real images (*i.e.*, ImageNet) using BLIP to improve prompt diversity and realism.

E Additional Visual Comparison

Figure 13 presents additional visual comparisons between our LatentSwitch and previous image editing methods. The results clearly demonstrate that our LatentSwitch consistently generates superior outputs compared to existing approaches.



(a) $\mathbf{I}_{\text{src}}$ (b) $\mathbf{M}$ (c) BrushNet (d) SDXL-Edit (e) SwitfEdit (f) **LatentSwitch**

Figure 13: **Visual comparison on various image editing benchmark dataset.** The text prompt shown above each image is used as the corresponding editing instruction. (Zoom-in for best view)