ReFlex: Text-Guided Editing of Real Images in Rectified Flow via Mid-Step Feature Extraction and Attention Adaptation

Supplementary Material

A. Additional preliminaries

A.1. Rectified flow

Rectified Flow models a transport map between two distributions, π_0 (real data) and π_1 (typically $N(0, I)$), by constructing straight-line paths between samples. The transition along these paths is governed by an ODE with a time-dependent velocity field $V(Z_t, t)$:

$$dZ_t = V(Z_t, t)dt, \quad t \in [0, 1]. \tag{1}$$

To define these straight paths, the forward process is formulated as a linear interpolation:

$$X_t = tX_1 + (1-t)X_0, \quad X_0 \sim \pi_0, \quad X_1 \sim \pi_1. \tag{2}$$

The velocity field $V(X_t, t)$ is then trained to approximate the dynamics of X_t, given by $dX_t = (X_1 - X_0)dt$, by minimizing the following least squares objective:

$$\min_v \int_0^1 \mathbb{E}\left[\|(X_1 - X_0) - v(X_t, t)\|^2\right] dt. \tag{3}$$

Once trained, sampling is performed by discretizing time steps $\{t_i\}_{i=0}^T$, where $t_T = 1$ and $t_0 = 0$, and solving the ODE iteratively. Starting from a noise sample $Z_{t_T} \sim \mathcal{N}(0, I)$, the velocity field updates Z_t at each step:

$$Z_{t_{i-1}} = Z_{t_i} + (t_{i-1} - t_i)V(Z_{t_i}, t_i), \quad i = T, \ldots, 1. \tag{4}$$

This process gradually transforms the initial noise sample Z_{t_T} into a structured sample following the learned data distribution π_0, effectively generating an image from noise.

A.2. FLUX

FLUX [17] is one of the state-of-the-art open-source text-to-image models based on rectified flow and MM-DiT. It extends MM-DiT by introducing two specialized block types: Double-Stream Block and Single-Stream Block. The Double-Stream Block uses separate Q, K, V projection matrices and modulation layers for text and image tokens, whereas the Single-Stream Block shares these layers across both modalities. FLUX employs Double-Stream Blocks in the first 19 layers and Single-Stream Blocks in the remaining 38 layers.

B. Implementation Details

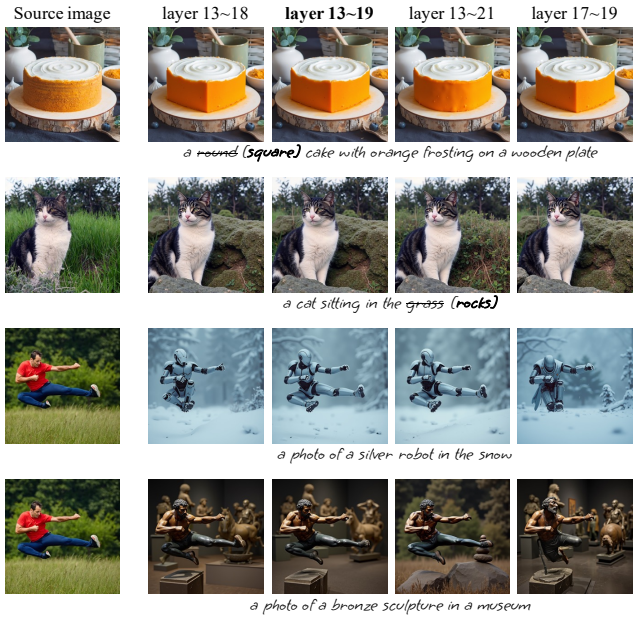
B.1. Implementation details for feature analysis

In Sec. 4.1, we conducted an analysis of the intermediate features within the MM-DiT block. Our analysis involves extracting each intermediate feature during the source image generation process and injecting it into the target image generation process to examine the resulting outputs. For the attention components, we use each component from the first 25 single-stream blocks, as they span slightly more than the middle third of the model. This setting is motivated by previous works in DM [4, 11, 36], which found that injecting attention from the middle or later layers is effective. For the residual components, we use the features from the last six double-stream blocks and the first four single-stream blocks.

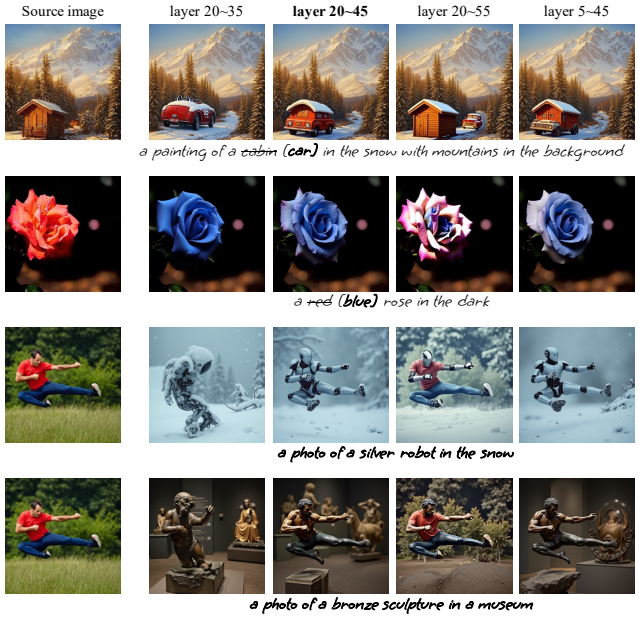
B.2. Implementation details for our method

Injection layers In Fig. S.1, we conduct an ablation study to determine the suitable layers for extracting residual and attention features in our experimental setting, where three key features are injected, and two attention adaptation methods are applied. Fig. S.1(a) presents the ablation results for residual feature layers. Our findings indicate that extracting residual features from the last six double-stream blocks and the first single-stream block (13∼19) achieves a balanced trade-off between structure preservation and editability. We found that increasing the number of single-stream blocks restricts editability, whereas using too few double-stream blocks or omitting the single-stream block may results in insufficient structural preservation. Fig. S.1(b) presents the ablation results for attention layers. We found that using attention from layers 20 to 45 achieved the best balance between structure preservation and editability, whereas extending attention injection to layer 35 weakened structure preservation, and further extending it to layer 55 limit the editability. Unlike U-Net, FLUX does not follow an encoder-decoder structure and consists of 57 layers at the same resolution, making it challenging to analyze the specific roles of individual layers. Therefore, we recognize that our findings have room for further refinement and expect that the current setting we identified will serve as a building block for future advancements.

Noised inversion The inversion process is typically performed by reversing the sampling process described in Eq. (4), starting from z_0. However, we observed that directly using z_0 for inversion can lead to an unnatural z_T. We attribute this to the model perceiving z_0, which corresponds to a clean latent, as an out-of-distribution sample, because the model primarily processes noised samples as input, except at $t = 0$. To mitigate this issue, we introduce a



(a) Ablation examples for assessing the impact of layer selection for residual feature injection.



(b) Ablation for attention layers

Figure S.1. Ablation examples for determining the layers for spatial feature and attention injection in (a) and (b), respectively.

noising step before reversing the sampling process. Specifically, we first sample random noise $\epsilon \sim N(0, I)$ from a standard Gaussian distribution. Then, using Eq. (2), we perform n forward steps to generate a noised latent z_n, defined as: $z_n = t_n \cdot \epsilon + (1 - t_n) \cdot z_0$. Finally, we reverse Eq. (4) for $T - n$ steps to obtain z_T. In Fig. S.2, we present the results of generating images using z_T obtained with and without the noising step. We observed that the latent obtained with the noising step produced images well aligned with the target

Figure S.2. We compare two approaches for obtaining z_T: one where z_T is obtained by directly inverting z_0, and another where z_T is obtained by applying n noising steps to reach z_n before performing inversion.

prompt, whereas z_T obtained without the noising step failed to achieve full alignment. Therefore, we introduce noised inversion, where a noising step is applied to the clean latent before reversing the sampling process, setting $n = 7$ in our experiment. However, it also introduces a degree of stochasticity, as the added noise affects the inversion process. The implications of this limitation are further discussed in Appendix F.

I2I-SA adaptation We observed that during the early stages of image generation, when abstract structural information is being formed, using smaller k values can be helpful for overall source structure preservation. Therefore, we apply I2I-SA adaptation after a few initial steps. When I2T-CA injection is not used, adaptation starts from the 4th step, whereas when I2T-CA injection is applied, it starts from the 2nd step.

B.3. Baseline implementation

In this section, we describe the implementation details of the baselines we used.

For **SD-based baselines**, we primarily use results provided by PnPInversion [14] when available. For **SDEdit** [21] and **P2P-Zero** [25], we use their Diffusers implementation [38]. We set the noising level to 0.75 for SDEdit and generate the source prompt for P2P-Zero using BLIP [18], following the offical example.

For **RF-Inversion** [32], we used the official Diffusers implementation [38]. The stopping time, τ, was set to 6, and the strength, η, was set to 0.9.

For **RF-Solver [39]**, we used the official GitHub repository [42]. Following the 'boy' example in the official GitHub repository, the `num_steps` was set to 15, and the `inject` was set to 2.

For **FireFlow [7]**, we used the official GitHub repositories [12]. Following the 'boy' example in the official GitHub repository, the `num_steps` was set to 8, and the `inject` was set to 1.

For **Flowedit** [15] we used the official Github repositories [9] and used their default setting for FLUX.

C. Additional qualitative evaluation

We conducted additional qualitative comparisons with baseline methods. Extended comparison results on PIE-Bench

and Wild-TI2I-Real are presented in Fig. S.3 and Fig. S.4, respectively. We also provided our additional editing results in Fig. S.5 and Fig. S.6. For the comparisons in Fig. S.3 and Fig. S.4, we set the I2T-SA adaptation parameter k to 20. In Fig. S.5 and Fig. S.6, k was set to 20 for the first, second, and last rows, 40 for the third row, and 80 for the fourth row. In the last row, where no source prompt was provided, the method was applied without I2T-CA injection.

D. Additional quantitative comparison

D.1. PIE-Bench

We report the full results of the quantitative evaluation on PIE-Bench [14] in Tab. S.2. We measure Structure Distance [35], PSNR, LPIPS [43], MSE, and SSIM [41] to assess source preservation. For text-alignment, we compute CLIP text similarity [27] for both the entire image and within the editing mask, referred to as Whole Image Clip Similarity and Edit Region Clip Similarity, respectively.

For hyperparameters, we report results using $k = 20, 40$ and $m = 0.7T, T$, where k is the number of replaced keys in I2I-SA adaptation, and m is the number of steps for applying latent blending. Our method outperforms the baselines across various metrics. Our method not only effectively generates images that align with target prompts, but also well preserves the original information of the source images. It is noteworthy that FLUX-based methods tend to lose background information of the source image, and SD-based methods struggle to accurately follow the target prompts, while our method addresses both challenges. We found that increasing m improves background preservation of the source image. However, setting $m = T$ creates an unnatural border between masked and unmasked regions, so we set $m = 0.7T$ as the default.

D.2. Limitation of existing structure distance metric

Previous studies [14, 36] have used the difference in the self-similarity of DINO-ViT [5]'s value features as a measure of structure distance [35] to quantify structural similarity between images. However, as shown in Fig. S.7, we found that structure distance often fails to provide reliable measurements when significant semantic changes occur, such as large differences in color distribution. Therefore, in Sec. 5.2, we conduct a quantitative analysis using alternative metrics to measure source preservation. Specifically, we use background PSNR in PIE-Bench and compute the IoU of subject segmentation masks in Wild-TI2I-Real.

D.3. Details on User Studies

For the two user studies presented in Sec. 5.3, we used Amazon Mechanical Turk (MTurk) to collect responses, requiring participants to have over 500 HIT approvals, an approval rate above 98%, and US residency. Each participant was presented with five images–generated from the same source image and target prompt but using different methods–and asked: 'Which edited image best aligns with the target description while preserving most of the structural details (e.g., pose, shape, or position of subjects) from the source image?' We excluded responses from participants who did not follow the survey instructions. In total, we collected 1410 answers from 97 valid participants for the FLUX-based comparison and 1530 answers from 102 valid participants for the SD-based comparison, with each participant answering 15 questions. An illustration of our user study is provided in Figure S.8.

E. Additional examples of ablation studies on each technique

In this section, we provide additional examples for the ablation study conducted in Sec. 5.4.

Ablation study on the role of key techniques is provided in Fig. S.9. we conducted an ablation study on four key techniques of our method: mid-step feature extraction, I2T-CA adaptation, I2I-SA adaptation, and latent blending.

Effect of varing t' in mid-step feature extraction is provided in Fig. S.10.

Effect of varing k in I2I-SA adaptation is provided in Fig. S.10.

Effect of varing α in I2T-CA adaptation is provided in Fig. S.12. With $\alpha = 1$, the model fails to accurately align the generated images with the text prompt, whereas increasing α improves alignment. However, too large α compromises source structure information, making it crucial to choose an appropriate α. We set $\alpha = 4$ for all following experiments.

F. Limitations

Fig. S.13 illustrates the limitations of our method. (a) When the edited region overlaps with the subject, it may unintentionally change other features of the subject. (b) If the editing mask generated from I2T-CA does not perfectly localize the editing region, it may result in ineffective editing. Using a ground-truth mask can effectively address this issue. (c) Since our method produces different editing results depending on the random seed, it can sometimes lead to editing failures. (d) Our method shows slower inference speed compared to other baselines, primarily because it cannot leverage FlashAttention, which provides significant acceleration. This limitation arises from our need to store attention maps—a feature not supported by FlashAttention. We provide inference time comparison of FLUX-based approaches in Tab. S.1.

Method	Ours	RF-Inversion	RF-Edit	FireFlow	FlowEdit
Time Cost (Sec.)	90.8s	54.3s	93.3s	18.2s	15.6s

Table S.1. Inference time comparison of FLUX-based approaches.

G. Societal Impact

Our work introduces a new Rectified Flow-based real-image editing method that significantly enhances text alignment. This approach allows users to easily edit real images using text prompts and obtain high-quality results. However, like most real-image editing methods, it carries the risk of misuse by malicious users. Fortunately, extensive research has been conducted to prevent the generation of ethically problematic content, such as violent imagery. We believe that our analysis of Rectified Flow features can contribute to ongoing efforts to restrict the creation of such harmful content.

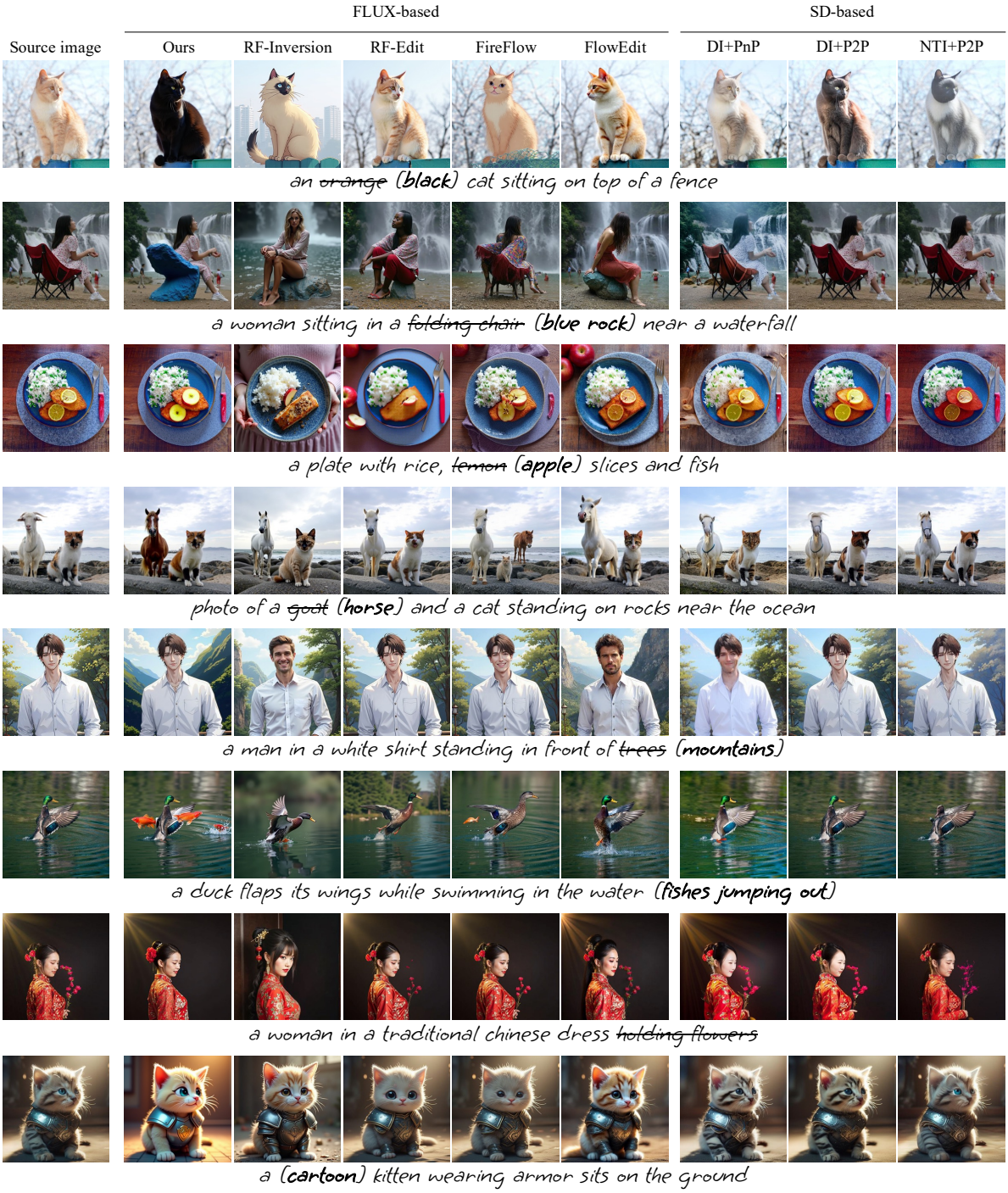


Figure S.3. Additional evaluation comparisons on PIE-bench

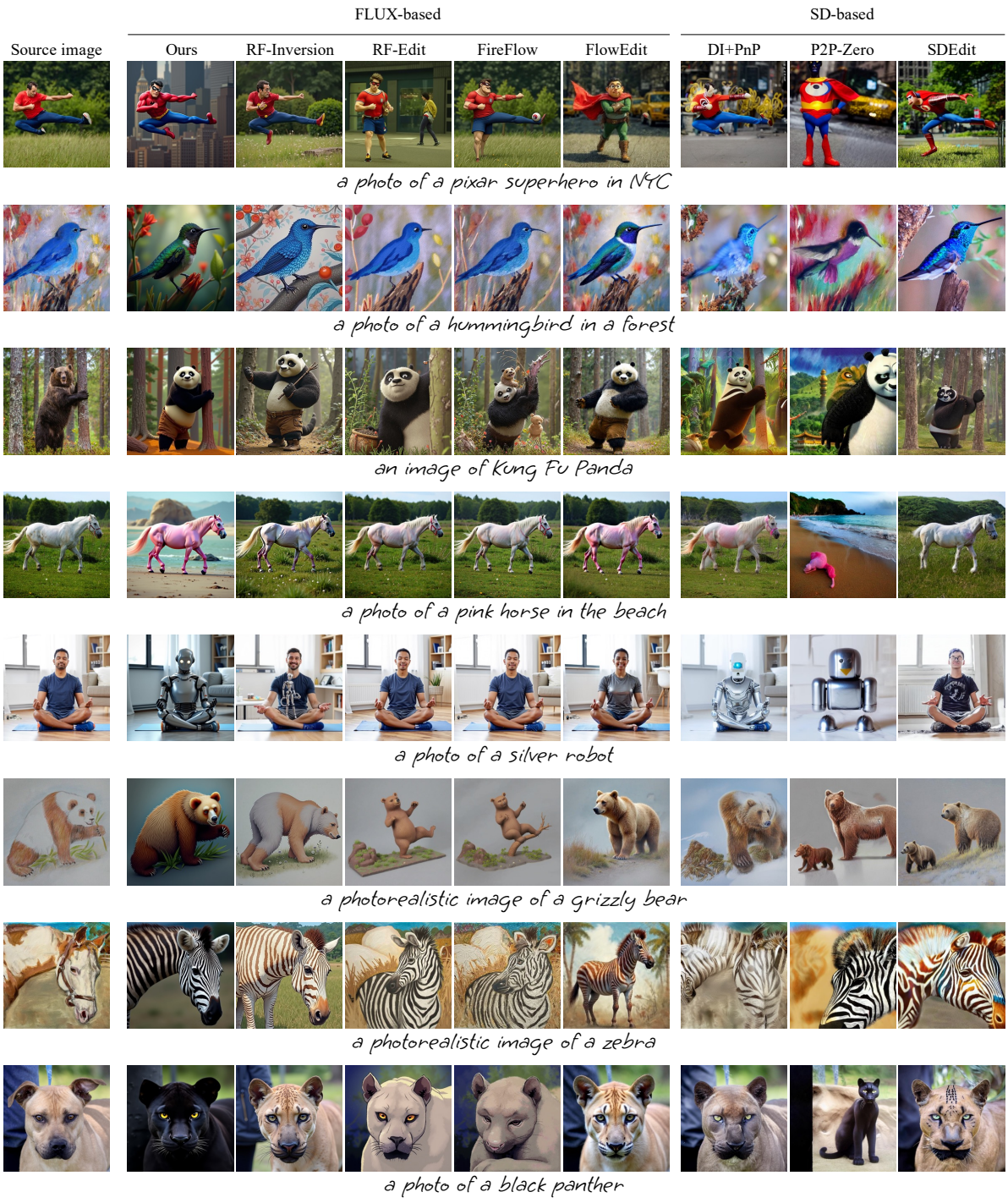


Figure S.4. Additional qulitative evaluation on Wild-TI2I-Real

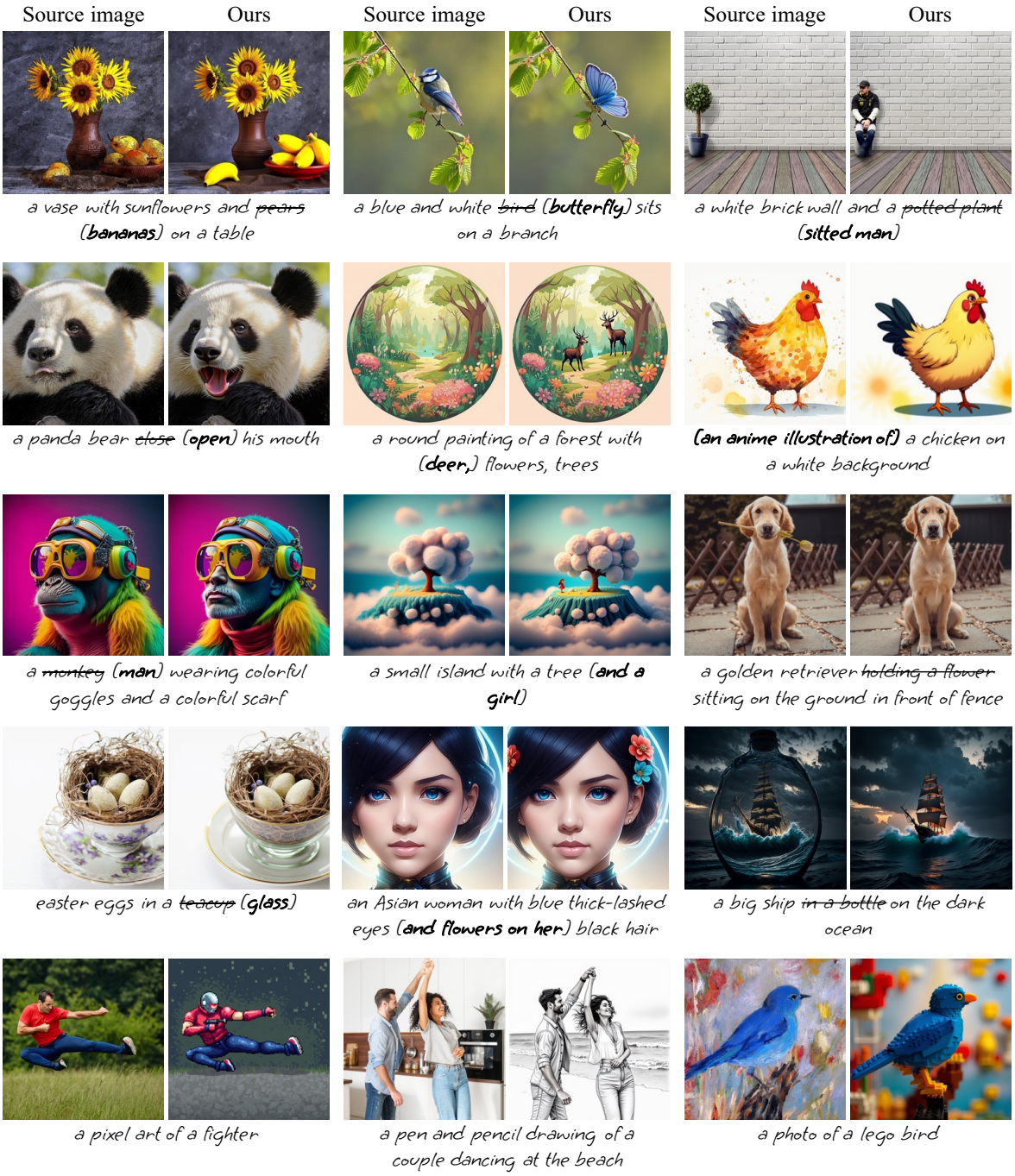


Figure S.5. Diverse edited results of our method.

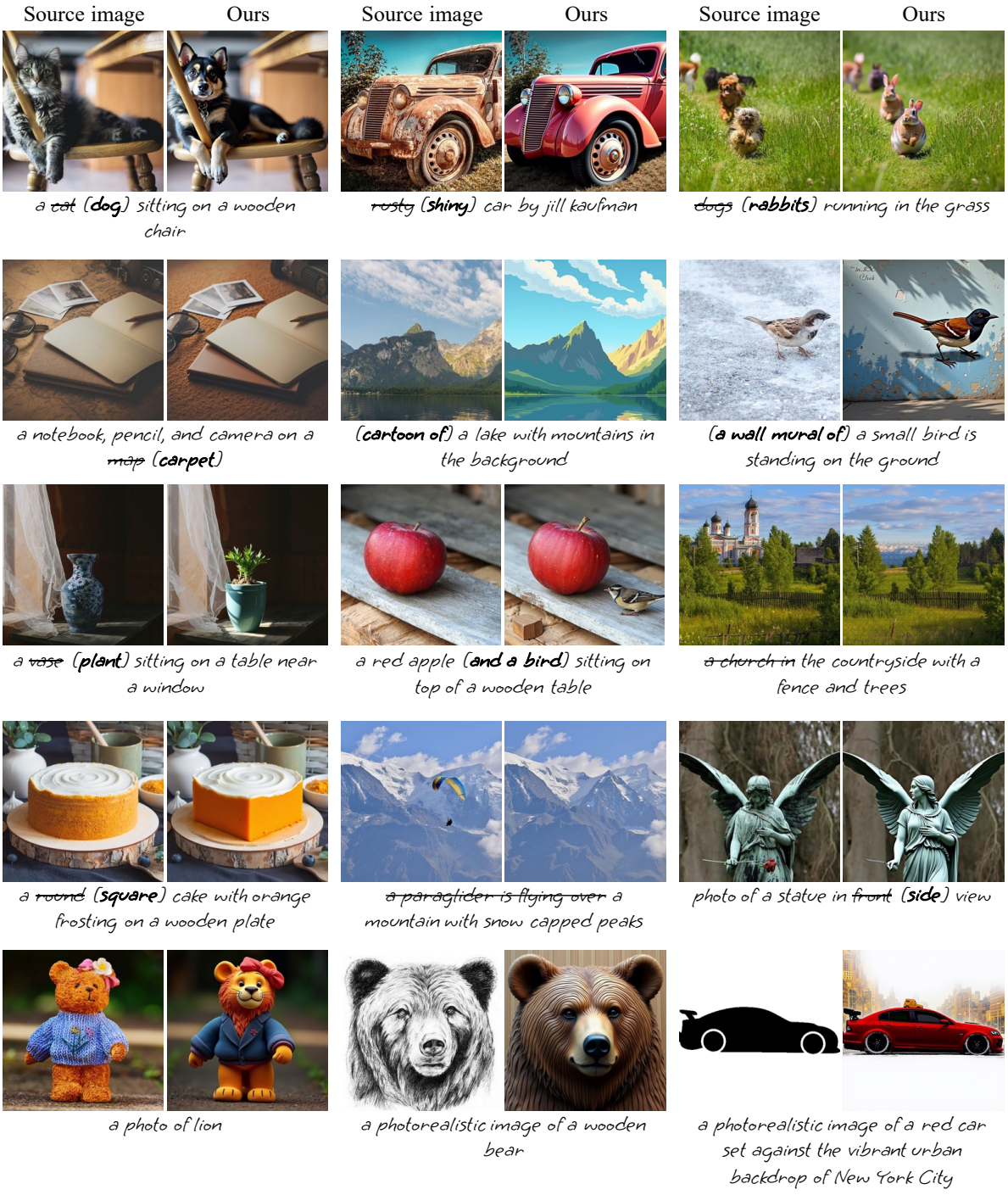


Figure S.6. Diverse edited results of our method.

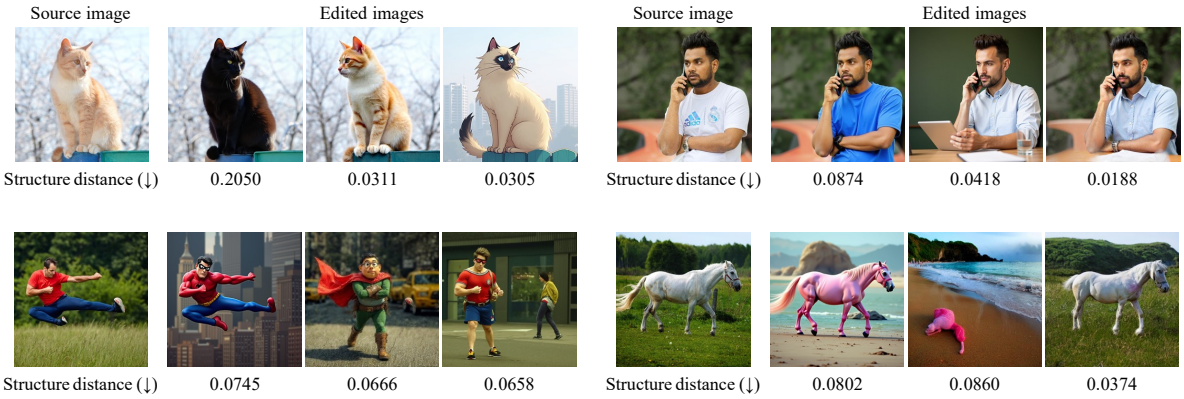


Figure S.7. **Limitation of existing structure distance metric.** The structure distance between each edited image and the source image is displayed below the corresponding edited image.

Method	Model	Structure Distance$_{\times 10^3}$ ↓	Background Preservation PSNR ↑	LPIPS$_{\times 10^3}$ ↓	MSE$_{\times 10^4}$ ↓	SSIM$_{\times 10^2}$ ↑	Clip Similarity Whole ↑	Edited ↑
DDIM + P2P	SD	69.43	17.87	208.80	219.88	71.14	25.01	22.44
NT + P2P	SD	<u>13.44</u>	27.03	60.67	<u>35.86</u>	84.11	24.75	21.86
DirectInversion + P2P	SD	**11.65**	27.22	**54.55**	**32.86**	84.76	25.02	22.10
DirectInversion + PnP	SD	24.29	22.46	106.06	80.45	79.68	25.41	22.62
RF-Inversion	FLUX	47.26	20.14	203.82	139.08	70.42	25.84	22.90
RF-Edit	FLUX	25.86	25.35	128.85	48.25	85.28	25.41	22.22
Fire Flow	FLUX	21.13	25.98	113.18	42.82	86.73	25.48	22.23
Flow Edit	FLUX	27.43	22.03	106.47	93.23	84.39	26.07	22.79
Ours (k=20, m=0.7T)	FLUX	40.30	24.21	112.55	76.74	83.01	<u>26.51</u>	23.17
Ours (k=40, m=0.7T)	FLUX	41.63	24.03	113.91	79.05	82.88	**26.62**	**23.37**
Ours (k=20, m=T)	FLUX	35.42	**28.05**	<u>57.87</u>	57.30	**92.08**	26.30	23.05
Ours (k=40, m=T)	FLUX	36.65	<u>27.83</u>	58.94	59.10	<u>91.99</u>	26.41	<u>23.23</u>

Table S.2. **Quantitative evaluation on PIE-Bench.** The best score is highlighted in **bold**, and the second-best score is <u>underlined</u>.

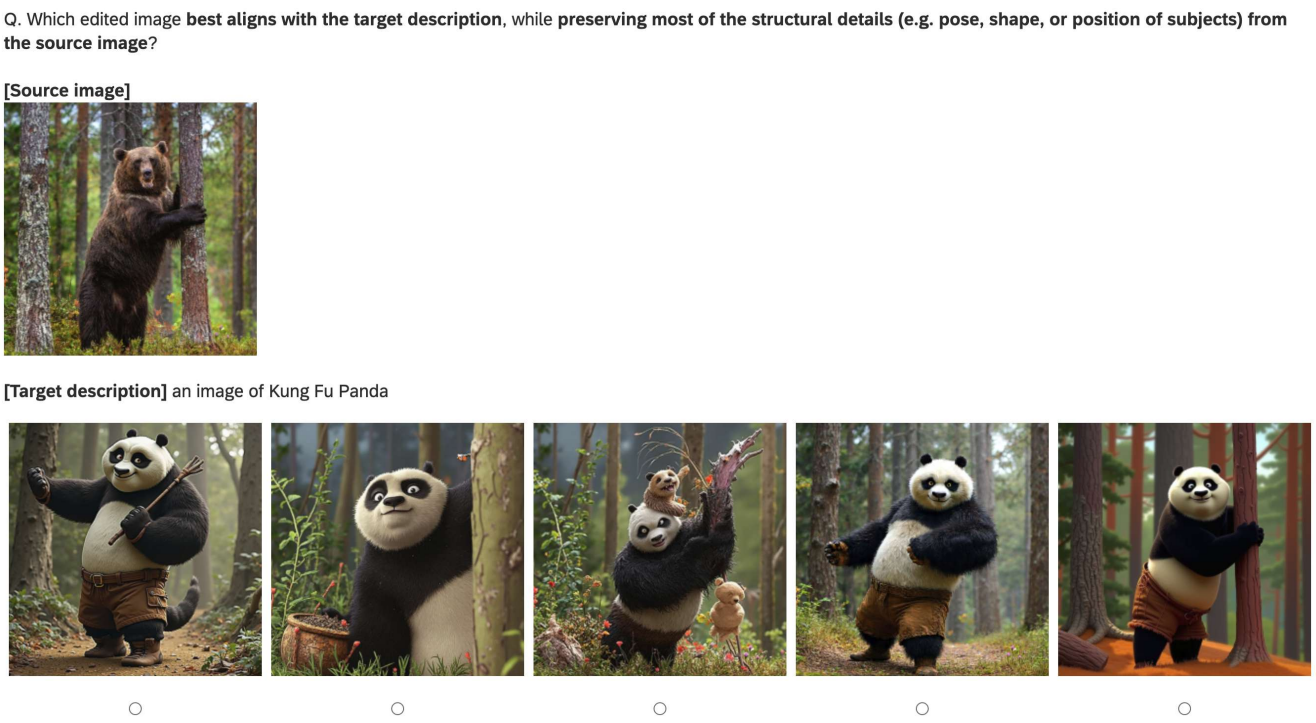


Figure S.8. Example screenshot from the user study, displaying images generated using different methods, where participants selected the one that best represents the intended edit.

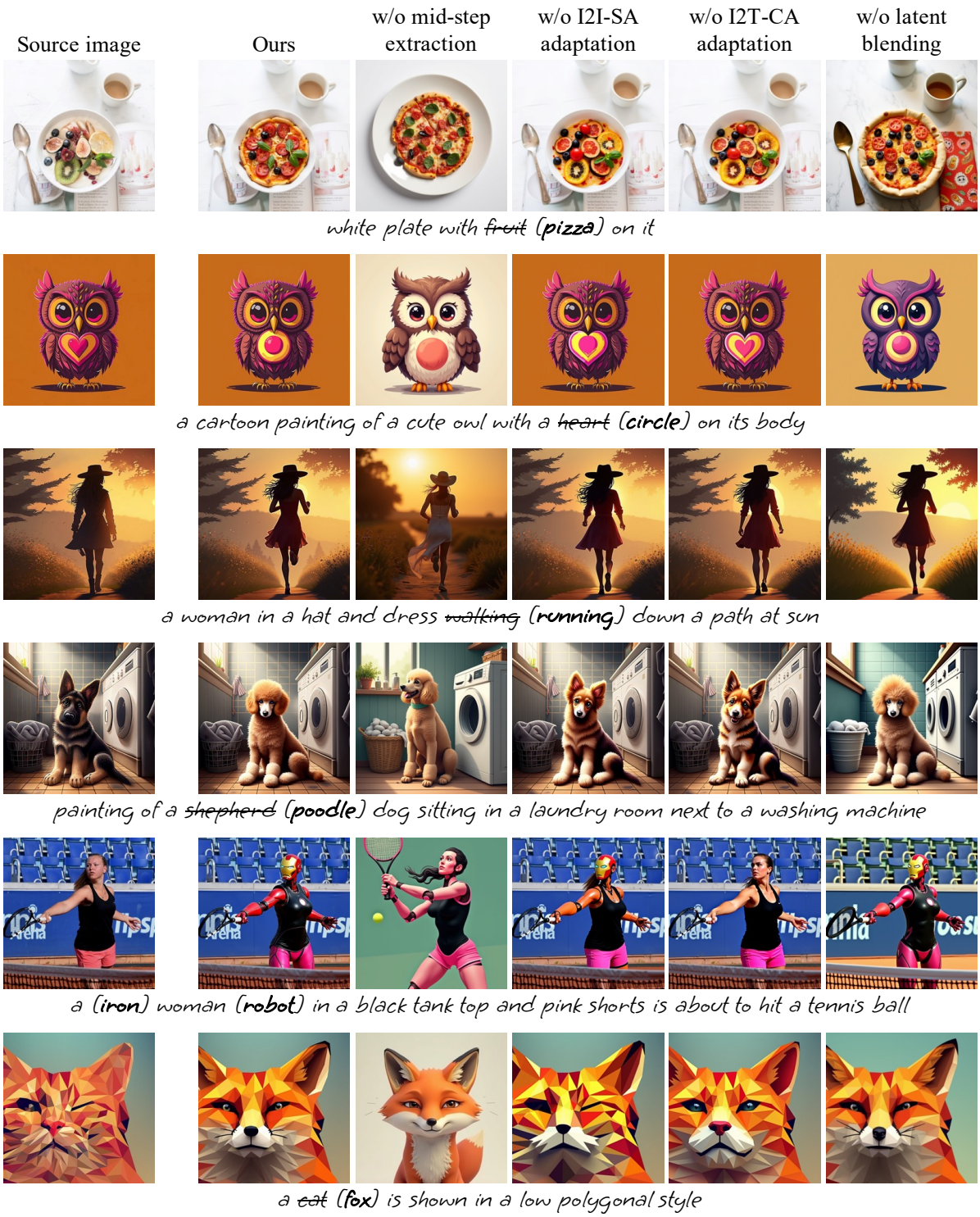


Figure S.9. Ablation examples for assessing the impact of each technique in our method, including latent blending, which is explained in Sec. 4.4.

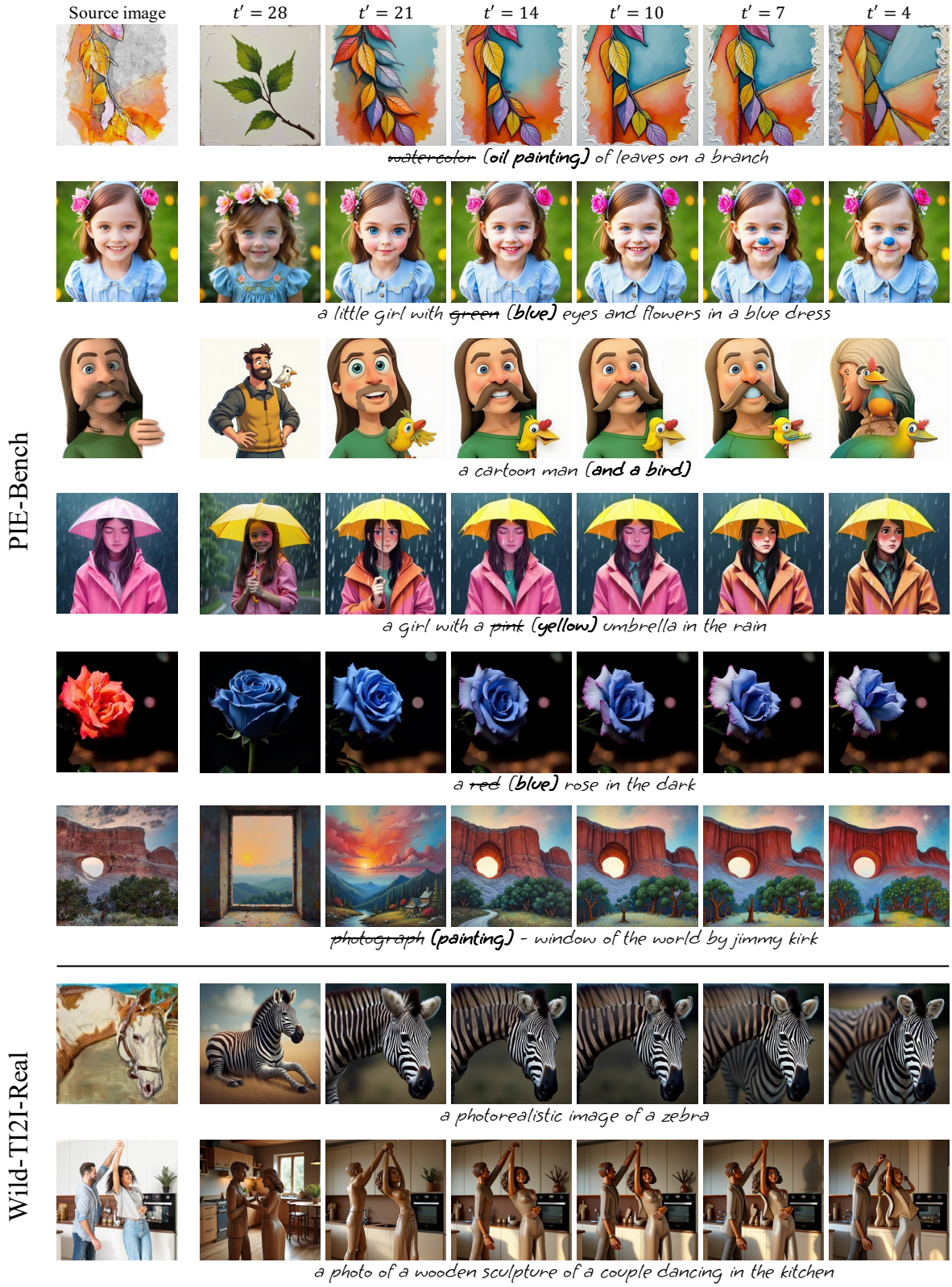


Figure S.10. Effect of the t' selection in mid-step feature extraction, where t' is the timestep of the latent from which features are extracted.

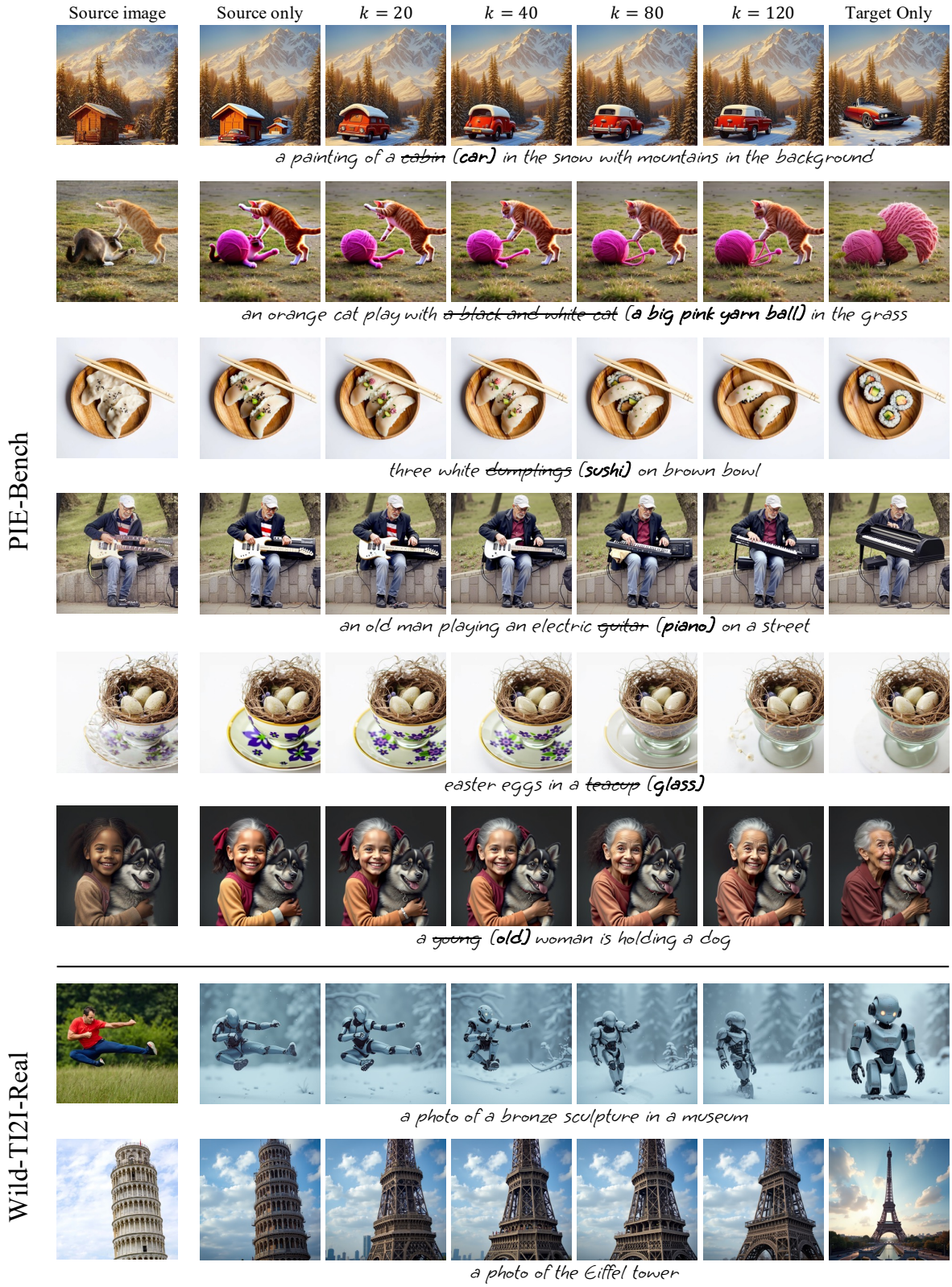


Figure S.11. Effect of varying k in I2I-SA adaptation, where k denotes the number of top attention values replaced.

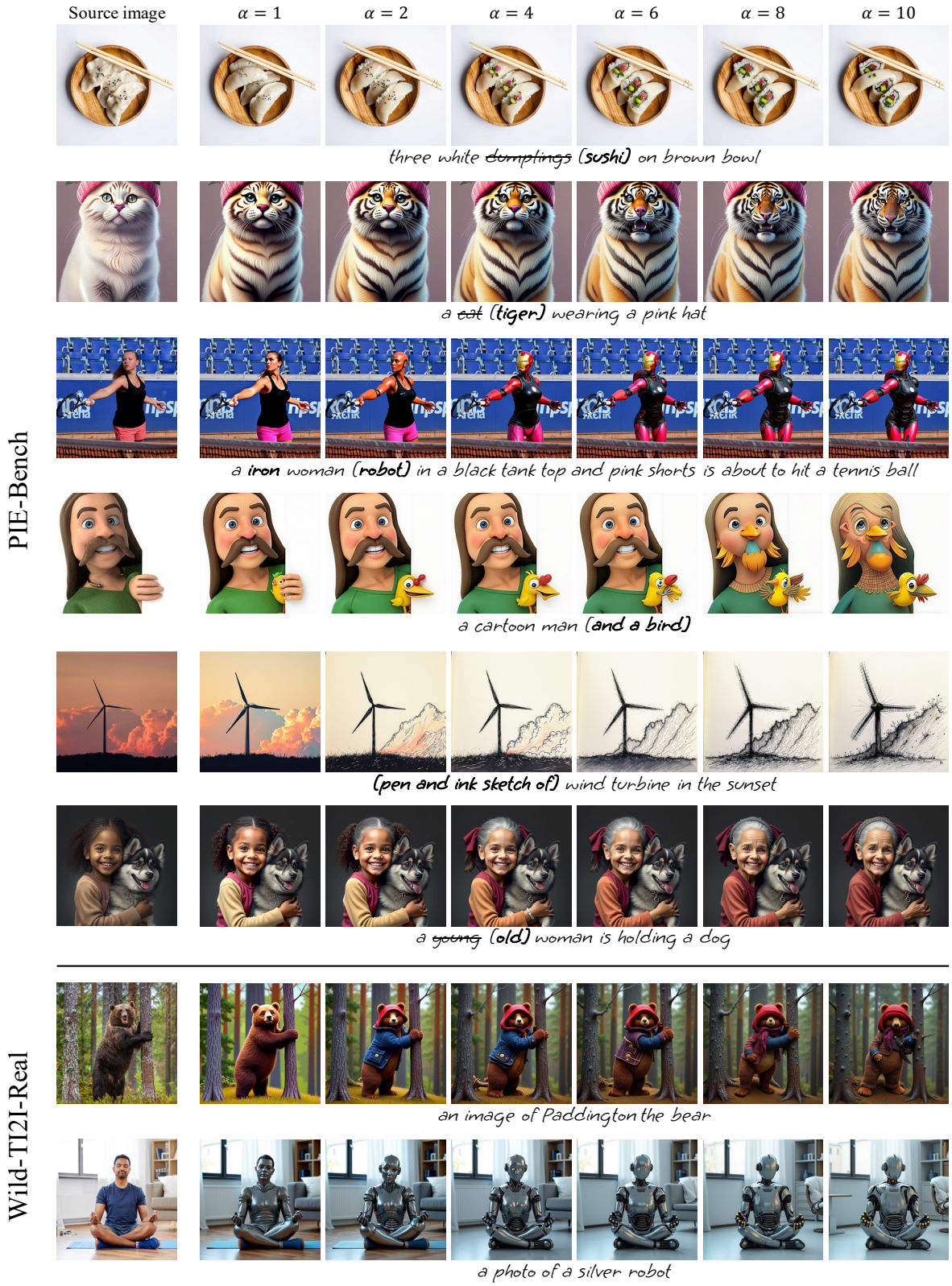


Figure S.12. Effect of varying α in I2I-SA adaptation, where α denotes the scale factor of the I2I-SA adaptation.

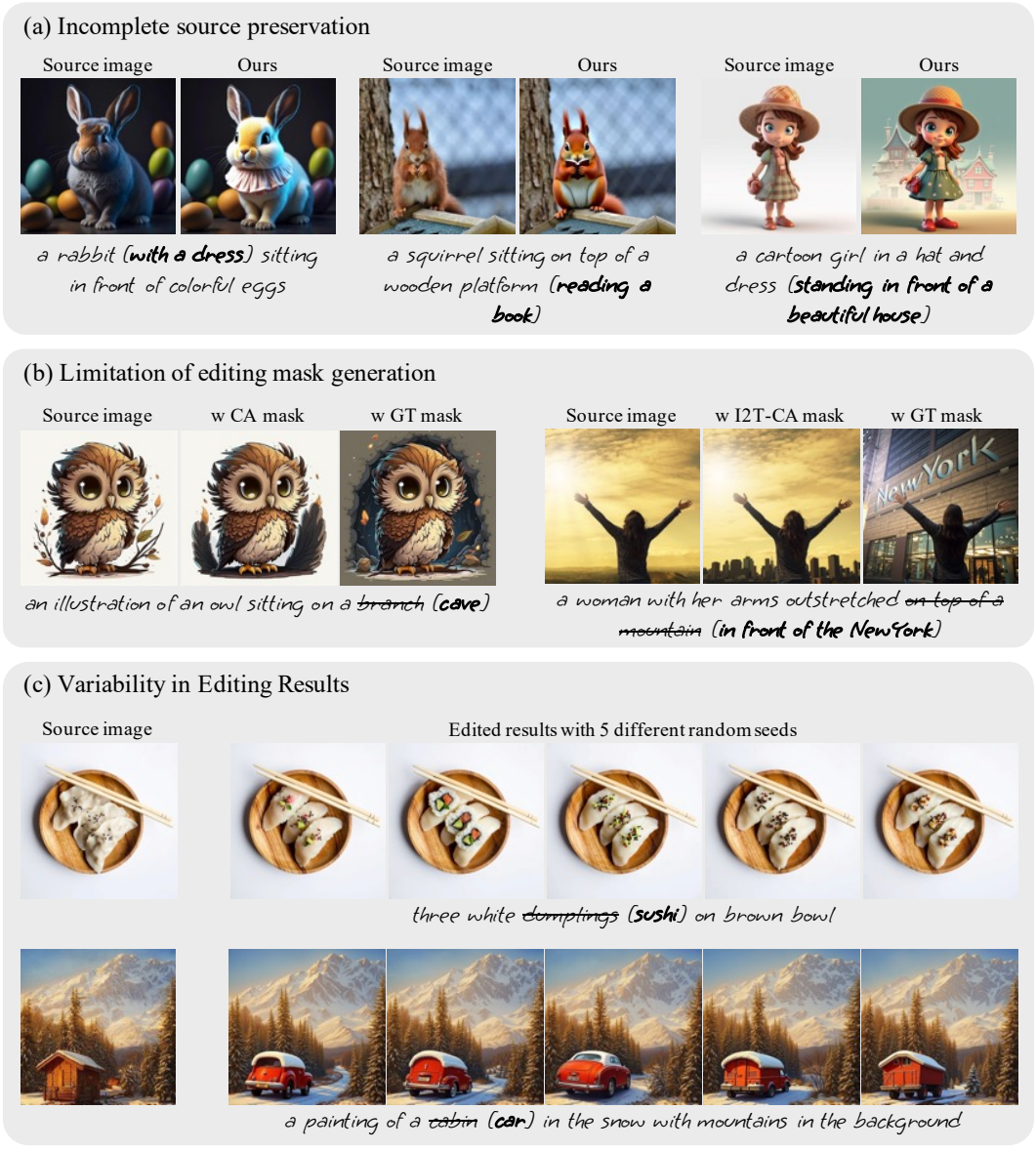


Figure S.13. **Limitations of our method.** (a) Incomplete preservation of source image details, when the edited region overlaps with the subject; (b) Limitation of editing mask generation; (c) Variability in editing results arises from the random seed.