SwiftEdit: Lightning Fast Text-Guided Image Editing via One-Step Diffusion

Supplementary Material

In this supplementary material, we first provide a detailed derivation of the regularization loss used in Stage 2, as outlined in Sec. 1. Next, we present several additional ablation studies in Sec. 2. Finally, we include more quantitative and qualitative results in Sec. 3, and Sec. 4. Then we discuss societal impacts in Sec. 5.

1. Derivation of the Regularization Loss in Stage 2

A detailed derivation of the gradient of our proposed regularization loss, as defined in Eq. (8) of the main paper is formulated as follows:

$$\mathcal{L}_{\text{regu}}^{\text{stage2}} = \mathbb{E}_{t,\hat{\boldsymbol{\epsilon}}}\left[w(t)\|\boldsymbol{\epsilon}_\phi(\mathbf{z}_t, t, \mathbf{c}_y) - \hat{\boldsymbol{\epsilon}}\|_2^2\right], \tag{1}$$

where $\boldsymbol{\epsilon}_\phi(.)$ is a teacher denoising UNet, here, we use SD 2.1 in our implementation.

The gradient of the loss w.r.t our inversion network's parameters θ is computed as:

$$\nabla_\theta \mathcal{L}_{\text{regu}}^{\text{stage2}} \triangleq \mathbb{E}_{t,\hat{\boldsymbol{\epsilon}}}\left[w(t)(\boldsymbol{\epsilon}_\phi(\mathbf{z}_t, t, \mathbf{c}_y) - \hat{\boldsymbol{\epsilon}}) \left(\frac{\partial \boldsymbol{\epsilon}_\phi(\mathbf{z}_t, t, \mathbf{c}_y)}{\partial \theta} - \frac{\partial \hat{\boldsymbol{\epsilon}}}{\partial \theta}\right)\right], \tag{2}$$

where we absorb all constants into $w(t)$. Expanding the term $\frac{\partial \boldsymbol{\epsilon}_\phi(\mathbf{z}_t, t, \mathbf{c}_y)}{\partial \theta}$, we have:

$$\frac{\partial \boldsymbol{\epsilon}_\phi(\mathbf{z}_t, t, c_y)}{\partial \theta} = \frac{\partial \boldsymbol{\epsilon}_\phi(\mathbf{z}_t, t, c_y)}{\partial \mathbf{z}_t} \frac{\partial \mathbf{z}_t}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \theta}. \tag{3}$$

Since $\mathbf{z}$ (extracted from real images) and θ are independent, $\frac{\partial \mathbf{z}}{\partial \theta} = 0$, thus, we can turn Eq. (2) into:

$$\nabla_\theta \mathcal{L}_{\text{regu}}^{\text{stage2}} \triangleq \mathbb{E}_{t,\hat{\boldsymbol{\epsilon}}}\left[w(t)(\boldsymbol{\epsilon}_\phi(\mathbf{z}_t, t, \mathbf{c}_y) - \hat{\boldsymbol{\epsilon}})(-\frac{\partial \hat{\boldsymbol{\epsilon}}}{\partial \theta})\right] \tag{4}$$

$$= \mathbb{E}_{t,\hat{\boldsymbol{\epsilon}}}\left[w(t)(\hat{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}_\phi(\mathbf{z}_t, t, \mathbf{c}_y))\frac{\partial \hat{\boldsymbol{\epsilon}}}{\partial \theta}\right], \tag{5}$$

which has the opposite sign of the SDS gradient w.r.t $\mathbf{z}$ loss as discussed in the main paper.

2. Additional Ablation Studies

Compatibility of multi-step inversion with one-step text-to-image model. To showcase the strength of our one-step inversion framework, we test existing inversion techniques on one-step generators. Specifically, we evaluate multi-step methods like DDIM Inversion (DDIMInv) and direct inversion on SBv2. As shown in the first and second row of Tab. 2, these methods yield lower performance and slower inference time, while SwiftEdit excels with superior results and high efficiency.

Figure 1. **Edit images with flexible prompting.** SwiftEdit achieves satisfactory reconstructed and edited results with flexible source and edit prompt input (denoted under each image).

Combined with other one-step text-to-image models. As discussed in the main paper, our inversion framework is not limited to SBv2 and can be seamlessly integrated with

Model	PSNR↑	CLIP-Whole↑	CLIP-Edited↑
Ours + InstaFlow†	24.88	24.03	20.47
Ours + DMD2†	**26.08**	23.35	19.84
Ours + SBv1‡	25.09	23.64	19.96
Ours + SBv2‡ (**SwiftEdit**)	23.33	**25.16**	**21.25**

Table 1. Ablation studies on combining our technique with other one-step text-to-image generation models. † means that these models are based on SD 1.5 while ‡ means that these models are based on SD 2.1.

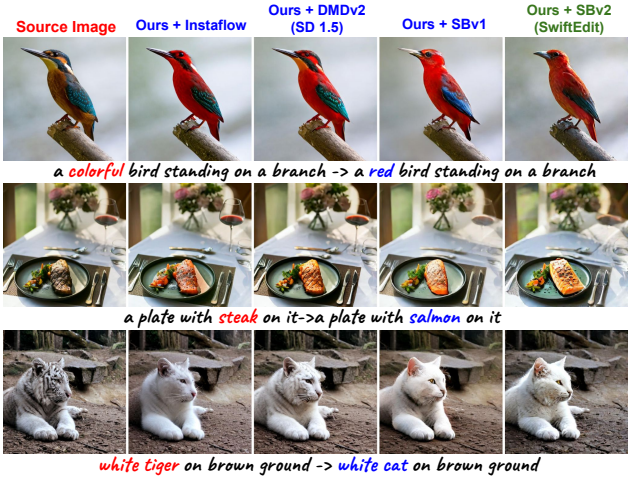
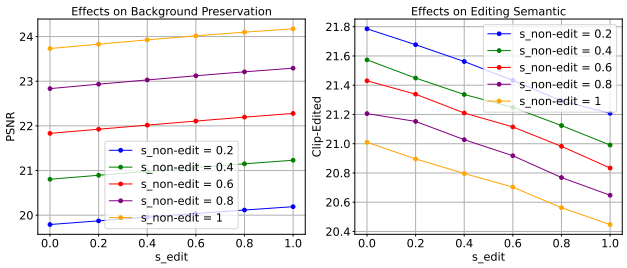


Figure 2. Qualitative results when combining our inversion framework with other one-step text-to-image generation models.

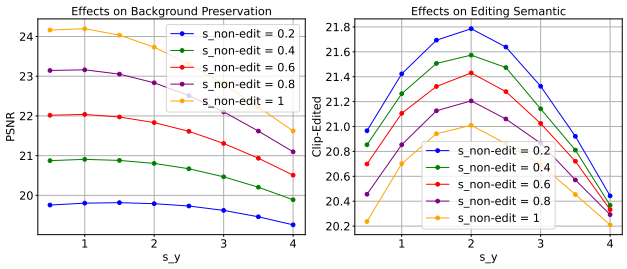
other one-step text-to-image generators. To demonstrate this, we conducted experiments replacing SBv2 with alternative models, including DMD2 [4], InstaFlow [2], and SBv1 [3]. For these experiments, the architecture and pre-trained weights of each generator $\mathbf{G}$ were used to initialize our inversion network in Stage 1. Specifically, DMD2 was implemented using the SD 1.5 backbone, while InstaFlow uses SD 1.5. All training experiments for both stages were conducted on the same dataset, similar to the experiments presented in Tab. 1 of the main paper.

Figure 2 presents edited results obtained by integrating our inversion framework with different one-step image generators. As shown, these one-step models integrate well with our framework, enabling effective edits. Additionally, quantitative results are provided in Tab. 1. The results indicate that our inversion framework combined with SBv2 (SwiftEdit) achieves the best editing performance in terms of CLIP-Whole and CLIP-Edited scores, while DMD2 demonstrates superior background preservation.

Two-stage training rationale. We provide additional ablation study where we train our network in a single stage using a mixed dataset of synthetic and real images. In particular, we construct a mixed training dataset comprised of: 10,000 synthetic image samples (generated by SBv2 using COCOA



(a) Varying s_{edit} scale at different levels of $s_{\text{non-edit}}$ with default $s_y = 2$.



(b) Varying s_y scale at different levels of $s_{\text{non-edit}}$ with default $s_{\text{edit}} = 0$.

Figure 3. Effects on background preservation and editing semantics while varying s_{edit} and s_y at different levels of $s_{\text{non-edit}}$.

prompts), and 10,000 real samples of COCOA dataset. The goal of this experiment is to understand the behavior and advantage of two-stage training compared to single stage training with mixed dataset. As shown in the third row of Tab. 2, the combined training stage resulted in lower performance across all metrics compared to our two-stage strategy. This highlights the effectiveness of our two-stage strategy.

Varying scales. To better understand the effect of varying scales used in Eq. (9) in the main paper, we present two comprehensive plots evaluating the performance of SwiftEdit on 100 random test samples from the PieBench benchmark. Particularly, the plots depict results for varying $s_{\text{edit}} \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ (see Fig. 3a) or $s_{\text{y}} \in \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4\}$ (see Fig. 3b) at different levels of $s_{\text{non-edit}} \in \{0.2, 0.4, 0.6, 0.8, 1\}$. As shown in Fig. 3a, it is evident at different levels of $s_{\text{non-edit}}$ that lower s_{edit} generally improves editing semantics (CLIP-Edited scores) but slightly compromises background preservation (PSNR). Conversely, higher s_{y} can enhance prompt-image alignment (CLIP-Edited scores, Fig. 3b), but excessive values ($s_{\text{y}} > 2$) may harm prompt-alignment result. In all of our experiments, we use default choice of scale parameters setting where we set $s_{\text{edit}} = 0$, $s_{\text{non-edit}} = 1$, and $s_{\text{y}} = 2$.

3. More Quantitative Results

In Tab. 3, we provide full scores on PieBench of comparison results in Tab. 1, with additional scores related to background preservation such as Structure Distance (SDis), LPIPS, and SSIM. We additionally compare with other

Method	SDis↓	PSNR↑	LPIPS↓	MSE↓	SSIM ↑	CLIP-W ↑	CLIP-E↑	Time (s)↓
DirectInv + SBv2	0.050	15.5	0.25	0.003	0.65	24.3	20.3	9.25
DDIMInv + SBv2	0.060	14.4	0.29	0.004	0.63	22.7	19.7	3.85
SwiftEdit (Mixed Training)	0.005	22.5	0.09	0.0008	0.79	23.5	19.3	0.23
SwiftEdit (Ours)	**0.001**	**23.3**	**0.08**	**0.0006**	**0.81**	**25.2**	**21.3**	**0.23**

Table 2. Comparison of SwiftEdit with other settings on PieBench.

Type	Method	SDis$_{\times 10^3}$↓	PSNR↑	LPIPS$_{\times 10^3}$↓	MSE$_{\times 10^4}$↓	SSIM$_{\times 10^2}$ ↑	CLIP-W ↑	CLIP-E↑	Time↓
Multi-step (50 steps)	DDIM + P2P	69.43	17.87	208.80	219.88	71.14	25.01	22.44	25.98
	NT-Inv + P2P	13.44	**27.03**	**60.67**	35.86	**84.11**	24.75	21.86	134.06
	DDIM + MasaCtrl	28.38	22.17	106.62	86.97	79.67	23.96	21.16	23.21
	Direct Inversion + MasaCtrl	24.70	22.64	87.94	81.09	81.33	24.38	21.35	29.68
	DDIM + P2P-Zero	61.68	20.44	172.22	144.12	74.67	22.80	20.54	35.57
	Direct Inversion + P2P-Zero	49.22	21.53	138.98	127.32	77.05	23.31	21.05	35.34
	DDIM + PnP	28.22	22.28	113.46	83.64	79.05	25.41	22.55	12.62
	Direct Inversion + PnP	24.29	22.46	106.06	80.45	79.68	25.41	**22.62**	12.79
	InstructPix2Pix	57.91	20.82	158.63	227.78	76.26	23.61	21.64	3.85
	InstructDiffusion	75.44	20.28	155.66	349.66	75.53	23.26	21.34	7.68
Few-steps (4 steps)	ReNoise (SDXL Turbo)	78.44	20.28	189.77	54.08	70.90	24.30	21.07	5.10
	TurboEdit	16.10	22.43	108.59	9.48	79.68	25.50	21.82	1.31
	ICD (SD 1.5)	**10.21**	26.93	63.61	**3.33**	83.95	22.42	19.07	1.38
One-step	SwiftEdit (Ours)	13.21	23.33	91.04	6.58	81.05	21.16	21.25	**0.23**
	SwiftEdit (Ours with GT masks)	13.25	23.31	93.88	6.19	81.36	**25.56**	21.91	**0.23**

Table 3. Quantitative comparison of SwiftEdit against other editing methods with metrics employed from PieBench [1].

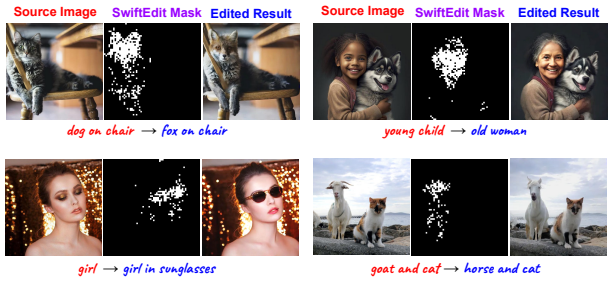


Figure 4. Visualization of our extracted mask along with edited results using guided text described under each image row.

training-based image editing methods such as InstructPix2Pix (InstructP2P), and InstructDiffusion (InstructDiff). Unlike these methods, which require multi-step sampling and paired training data, SwiftEdit trains on source images alone for one-step editing. As shown, SwiftEdit outperforms both in quality and speed, thanks to its efficient one-step inversion and editing framework.

4. More Qualitative Results

Self-guided Editing Mask. In Fig. 4, we show more editing examples along with self-guided editing masks extracted directly from our inversion network.

Flexible Prompting. As shown in Fig. 1, SwiftEdit consistently reconstructs images with high fidelity, even with minimal source prompt input. It operates effectively with just a single keyword (last three rows) or no prompt at all (first two rows). Notably, SwiftEdit performs complex edits with ease, as demonstrated in the last row of Fig. 1, by simply combining keywords in the edit prompt. These results highlight its capabilities as a lightning-fast and user-friendly editing tool.

Facial Identity and Expression Editing. In Fig. 5, given a simple source prompt "man" and a portrait image, SwiftEdit can achieve face identity and facial expression editing via a simple edit prompt by just combining expression word (denoted on each row) and identity word (denoted on each column).

Additional Results on PieBench. In Figs. 6 to 8, we provide extensive editing results compared with other methods

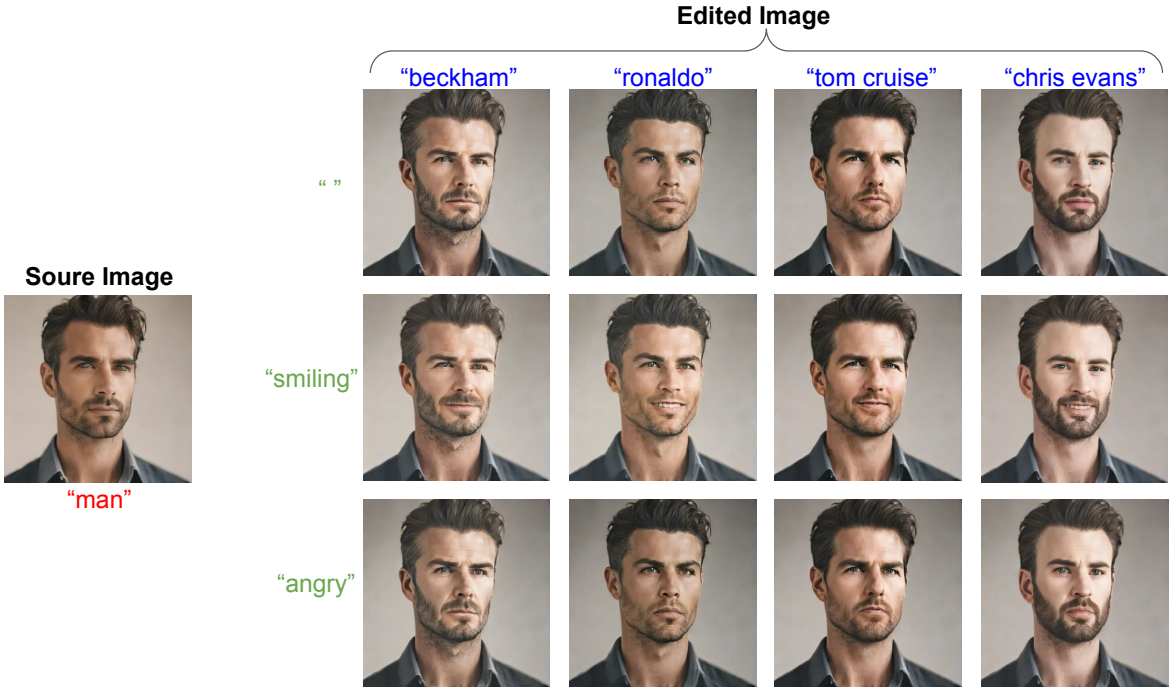


Figure 5. **Face identity and expression editing via simple prompts**. Given a portrait input image, SwiftEdit can perform a variety of facial identities along with expression editing scenarios guided by simple text within just **0.23** seconds.

on the PieBench benchmark.

5. Societal Impacts

As an AI-powered visual generation tool, SwiftEdit delivers lightning-fast, high-quality, and customizable editing capabilities through simple prompt inputs, significantly enhancing the efficiency of various visual creation tasks. However, societal challenges may arise as such tools could be exploited for unethical purposes, including generating sensitive or harmful content to spread disinformation. Addressing these concerns are essential and several ongoing works have been conducted to detect and localize AI-manipulated images to mitigate potential misuse.

References

[1] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. *International Conference on Learning Representations (ICLR)*, 2024. 3

[2] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024. 2

[3] Thuan Hoang Nguyen and Anh Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[4] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024. 2

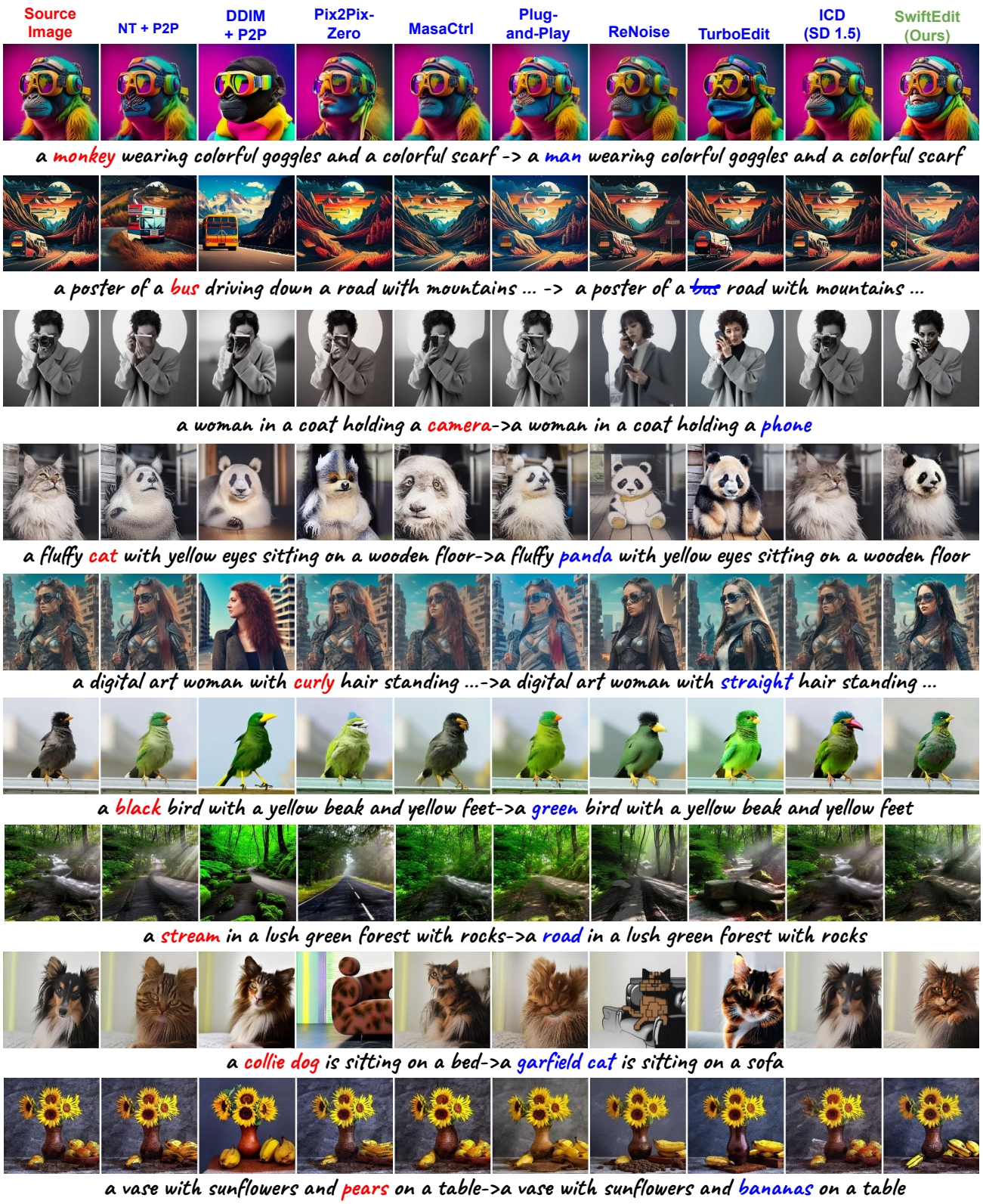


Figure 6. Comparative results on the PieBench benchmark

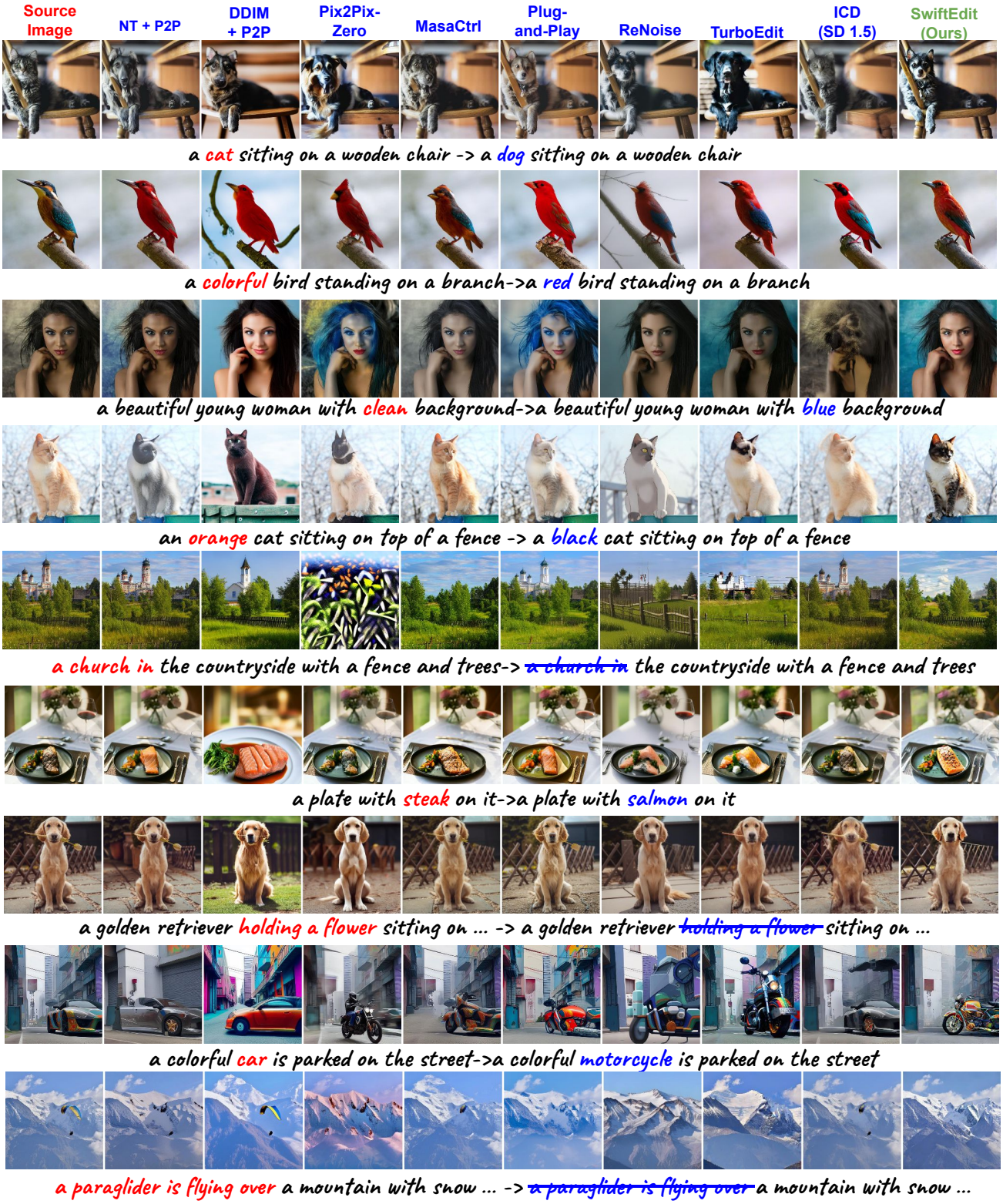


Figure 7. Comparative results on the PieBench benchmark

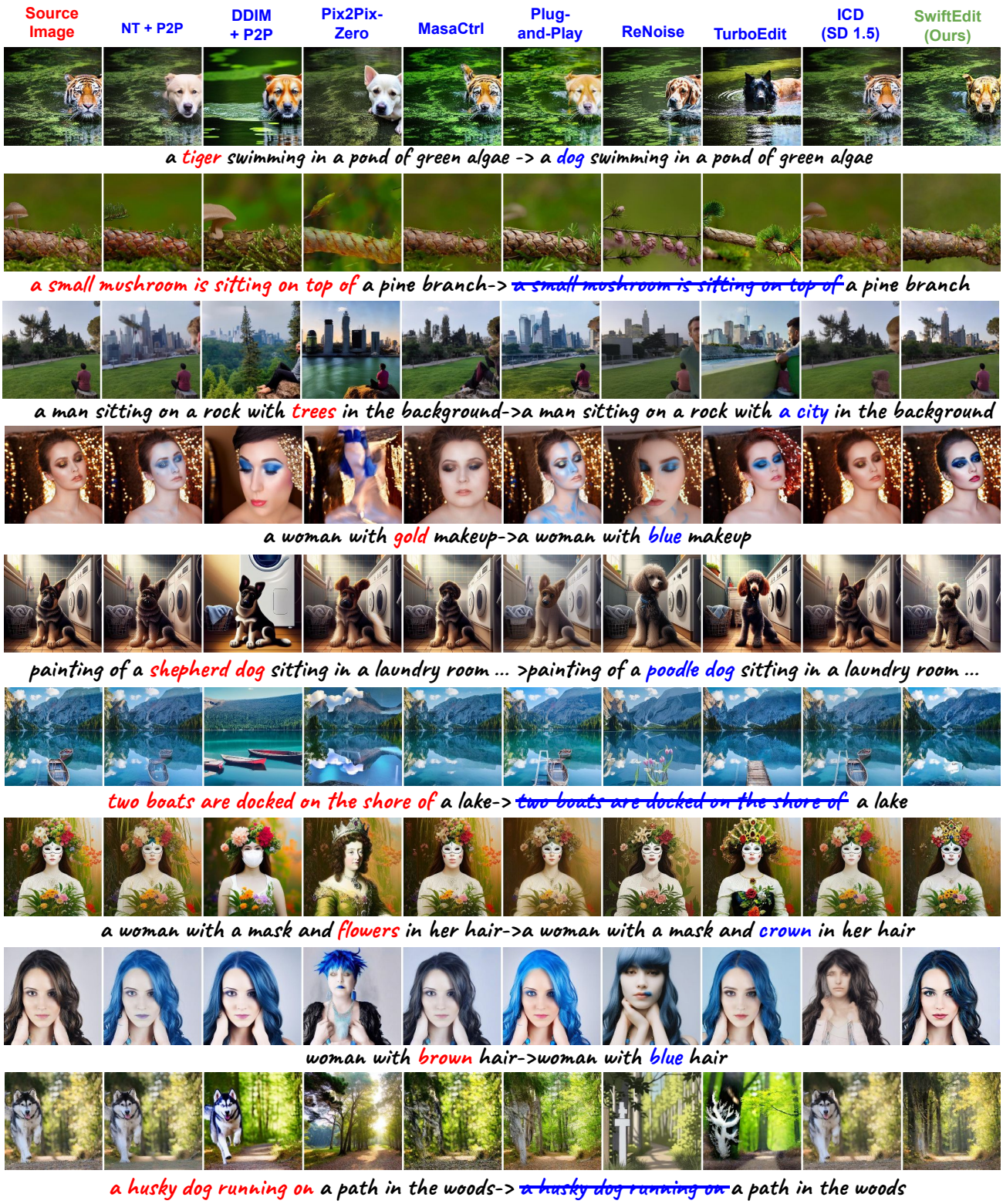


Figure 8. Comparative results on the PieBench benchmark