Supplementary Material

Ruibin Li[1], Ruihuang Li[1], Song Guo[2], and Lei Zhang[1]*

[1] The Hong Kong Polytechnic University
[2] The Hong Kong University of Science and Technology
{ruibin.li,21039075r,cslzhang}@connect.polyu.hk,songguo@cse.ust.hk

In this supplementary file, we provide the following materials:

- Noise gap D_{noi} reduction by our SPDInv (referring to Sec. 1 and Fig. 2 in the main paper);
- Derivation of Eq. (1) (referring to Eq. (1) in Sec. 3.1 of the main paper);
- Experimental results on TDE-Bench (referring to Sec. 4.2 in the main paper;
- More visual comparisons between different methods using P2P, MasaCtrl and PNP editing engines (referring to Sec. 4.2 in the main paper);
- More localised editing results with customized generation methods (referring to Sec. 4.3 in the main paper);
- Failure cases (referring to Sec. 5 in the main paper).

1 Reduction of Noise Gap by SPDInv

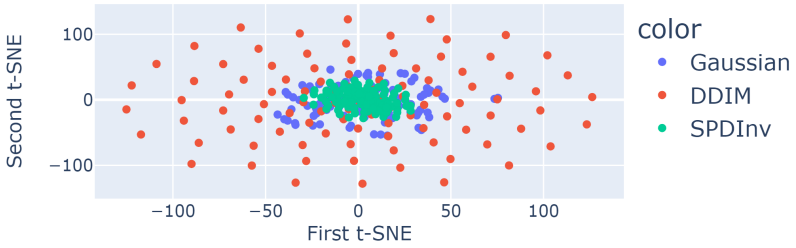

Fig. 1: Visualization of inverted noise codes.

We conduct experiments to evaluate the noise gap reduction by the proposed SPDInv (please refer to Fig. 2 in the main paper and the analysis in the introduction section for details of noise gap D_{noi}). We first utilize the captions extracted from the COCO2017 evaluation dataset [4] to generate 100 images by Stable Diffusion V1.4. We then record the initial noises z_T and all the latent features $z_t, t \in [0, T)$, which can be regarded as the ground-truth features of the generated images (please refer to the generation path in Fig. 2 of our main paper). Since previous text-driven image editing methods mostly employ

* Corresponding author.

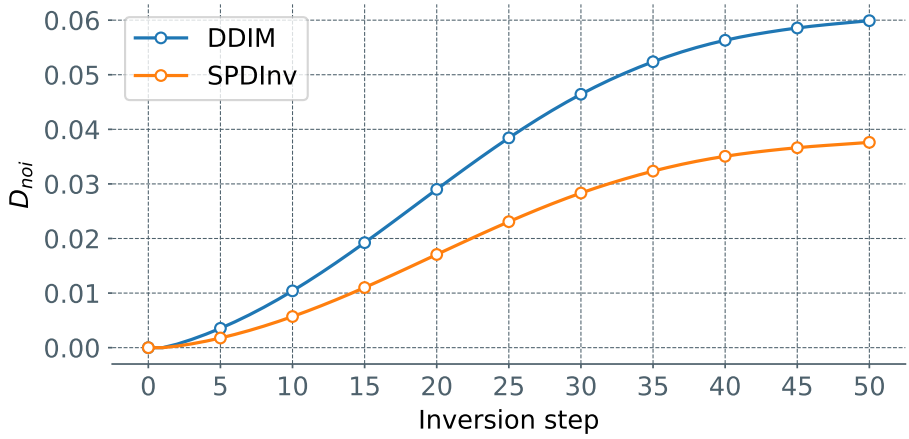

Fig. 2: The noise gap D_{noi} of inverted noise codes by DDIM inversion and our SPDInv using 100 generated images with captions extracted from COCO2017.

Table 1: Clip score between noise codes and source prompt.

Metrics	Gaussian Noise	DDIM	SPDInv
Clip Score	12.66	14.55	13.18

DDIM inversion to compute the inverted noise code, we thus compare DDIM inversion with our SPDInv on their inversion performance using the generated ground-truth features and conduct qualitative and quantitive experiments.

For qualitative evaluation, we utilize t-SNE to reduce the dimension of inverted noise codes and visualize them in Fig. 1. One can clearly see that the noise codes acquired through SPDInv exhibit a distribution more akin to Gaussian noise than DDIM inversion.

For quantitive evaluation, we first record all inverted latent features during the inversion process. Then for each inversion step t, we can calculate the mean square error (MSE) between the ground-truth and inverted latent features. Finally, the MSEs of all the 100 images are averaged as the D_{noi}, which is illustrated in Fig. 2. Compared with DDIM inversion, our SPDInv reduces the noise gap from 0.06 to 0.04 (36% $\downarrow$) after 50 inversion steps. Based on the same setting, we further calculate the clip score between the Gaussian noise, DDIM inverted noise, SPDInv inverted noise and the source prompt. The results are shown in Tab. 1. We see that SPDInv can reduce the clip score by 10% compared with DDIM inversion, and it is closer to the clip score calculated by pure Gaussian noise. Owe to the reduced noise gap, our SPDInv can effectively reduce the effect of source prompt on the inverted noise code, and consequently reduce the editing artifacts.

2 Derivation of Eq. (1) in the Main Paper

According to DDIM [5], the deterministic sampling equation (please refer to the proof of DDIM [5]) is:

$$z_{t-1} = \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}} z_t + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \epsilon_\theta(z_t, t, c), \tag{1}$$

where z_t is used as the input of the neural network ϵ_θ. $\alpha_t, \alpha_{t-1}, c$ can be regarded as constants. z_{t-1} is the output less noisy code by Eq. (1). If we want to put z_t on the left side of the equation as the output, we have the following step by step derivations:

$$\begin{aligned}
z_{t-1} &= \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}} z_t + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \epsilon_\theta(z_t, t, c), \\
\frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}} z_t &= z_{t-1} - \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \epsilon_\theta(z_t, t, c), \\
\frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}} z_t &= z_{t-1} + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_t} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1} \right) \epsilon_\theta(z_t, t, c), \\
z_t &= \frac{\sqrt{\alpha_t}}{\sqrt{\alpha_{t-1}}} z_{t-1} + \sqrt{\alpha_t} \left(\sqrt{\frac{1}{\alpha_t} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1} \right) \epsilon_\theta(z_t, t, c),
\end{aligned} \tag{2}$$

where z_{t-1} is used to calculate z_t to achieve the goal of image inversion. In Eq. (2), (z_t, t, c) are supposed to be the input of neural network ϵ_θ. However, previous methods mostly utilized $(z_{t-1}, t-1, c)$ as the input of ϵ_θ, coupling the inverted noise code with source prompt (please refer to Sec. 3.1 in main paper for our detail analysis).

3 Results on TDE-Bench Dataset

Tab. 2 presents the quantitative results of the competing inversion-based methods on the TDE-Bench. The parameter settings are the same as the Tab. 1 in the main paper. When using the P2P editing engine, compared with the second-best methods (DirectINV or DDIM), SPDInv achieves visible improvements in DINO score (12% ↑), PSNR (1.7% ↑), LPIPS (10% ↓), MSE (7.7% ↓), SSIM (0.41 ↑), and CLIP (0.03 ↑). When the MasaCtrl and PNP editing engines are used, greater improvement on most metrics can be achieved by using SPDInv. Fig. 3 visualizes some editing results, including changing texture (fruit, lemon, sandwich), content (cat, dog, zebra, airplane), color (bird, truck, airplane), and style (room) on TDE-Bench.

Table 2: Performance comparison of different inversion methods under the Prompt-to-Prompt (P2P) [3], Mutual self control (MasaCtrl) [1] and Plug-and-Play (PNP) [6] editing engines on TDE-Bench. Best and second best metrics are highlighted in red and blue colors, respectively.

Inversion	Editing Engine	DINO↓ $\times 10^3$	PSNR↑	LPIPS↓ $\times 10^3$	MSE↓ $\times 10^4$	SSIM↑ $\times 10^2$	CLIP↑	Inversion times(s)
DDIM	P2P	77.50	23.23	101.23	115.33	84.83	25.73	11.55
NTI	P2P	17.32	28.54	58.14	44.11	88.69	25.70	137.54
NPI	P2P	20.74	28.10	63.67	48.91	88.42	25.59	11.75
AIDI	P2P	15.20	28.85	59.14	46.54	88.83	25.57	87.21
NMG	P2P	19.18	29.08	52.78	37.99	89.29	25.23	16.71
DirectINV	P2P	12.75	29.17	49.41	37.39	89.48	25.57	19.94
ProxEdit	P2P	18.67	28.60	56.54	42.89	88.91	25.60	11.75
SPDInv	P2P	11.23	29.69	44.25	34.50	89.89	25.76	27.04
DDIM	MasaCtrl	82.68	22.16	105.83	137.55	84.33	26.75	11.55
NMG	MasaCtrl	42.84	24.75	70.28	81.67	87.37	25.45	16.71
DirectINV	MasaCtrl	59.29	24.46	70.47	82.74	87.33	26.13	19.94
AIDI	MasaCtrl	80.55	21.97	106.30	140.01	84.35	26.30	87.21
SPDInv	MasaCtrl	22.93	27.96	45.88	42.33	89.32	26.32	27.04
DDIM	PNP	30.66	26.14	71.41	64.03	87.91	26.79	11.55
DirectINV	PNP	27.79	26.17	69.07	63.13	88.01	26.61	19.94
AIDI	PNP	28.16	26.75	69.57	59.56	88.18	26.68	87.21
SPDInv	PNP	20.03	29.73	55.94	25.55	89.07	26.72	27.04

4 More Visual Comparisons between Different Methods using P2P, MasaCtrl and PNP Editing Engines

Results with P2P: In Fig. 4, we provide more visual comparisons of competing text-driven image editing methods with the P2P engine. Similar to the results in the main paper, DDIM struggles to preserve the details in most cases. NTI encounters the problem of content inconsistency (women in row 3) and collapse (boy in row 6). NPI and ProxEdit collapse in editing boy (row 6) and cat (row 7). They may also change details in some cases (room layout in row 1, wings in row 2, face in row 8). AIDI, NMG and DirectINV fail in editing rose (row 4), tie (row 5), mushroom (row 10), while NMG and DirectINV change the human face in row 7. On the contrary, SPDInv achieves successful editing in most cases.

Results with MasaCtrl: Fig. 5 provides the visual results with MasaCtrl editing engine. Different from P2P, MasaCtrl prefers editing nonrigid objects, such as changing pose or removing items, because it can drastically change the structure of the main subject. All competing methods show good editing performance on removing mushroom (row 2), boat (row 3). However, DDIM, AIDI

and DirectINV lose details of mountain (row 3) or bear (row 6). NMG shows competitive performance in most cases. Nevertheless, SPDInv exhibits better detail preservation on branch in row 1, and reflection in row 3.

Results with PNP: Fig. 6 provides the results with PNP engine. DDIM and DirectINV do not perform very well on images such as dog (row 1) and mountain (row 4), and show inconsistency during editing the duck (row 3) and bunny (row 6). AIDI fails in removing the flower in row 1. SPDInv achieves successful editing in all these cases.

5 Localized Editing with Customized Generation Methods

Except for ELITE [7], we integrate our SPDInv into another customized generation method, *e.g.*, Custom-Diff [2]. Leveraging its pre-trained model on two distinct concepts, pot (row 1) and cat (row 2), we perform localized editing and present the corresponding visual results in Fig. 7. One can see that the embedding of SPDInv endows Custom-Diff the ability to maintain consistent layout and pose, while the original Custom-Diff exhibits very different layout and pose from the input images.

6 Failure Cases

As we stated in the conclusion of the main paper, although SPDInv achieves great improvement on the overall editing performance, it may fail in cases such as adding, dropping items and editing portrait with current editing engines (*i.e.*, P2P [3], MasaCtrl [1], PNP [6]). Some failure cases are depicted in Fig. 8. Future work will be conducted for improving the stability and robustness of SPDInv on these editing scenarios.

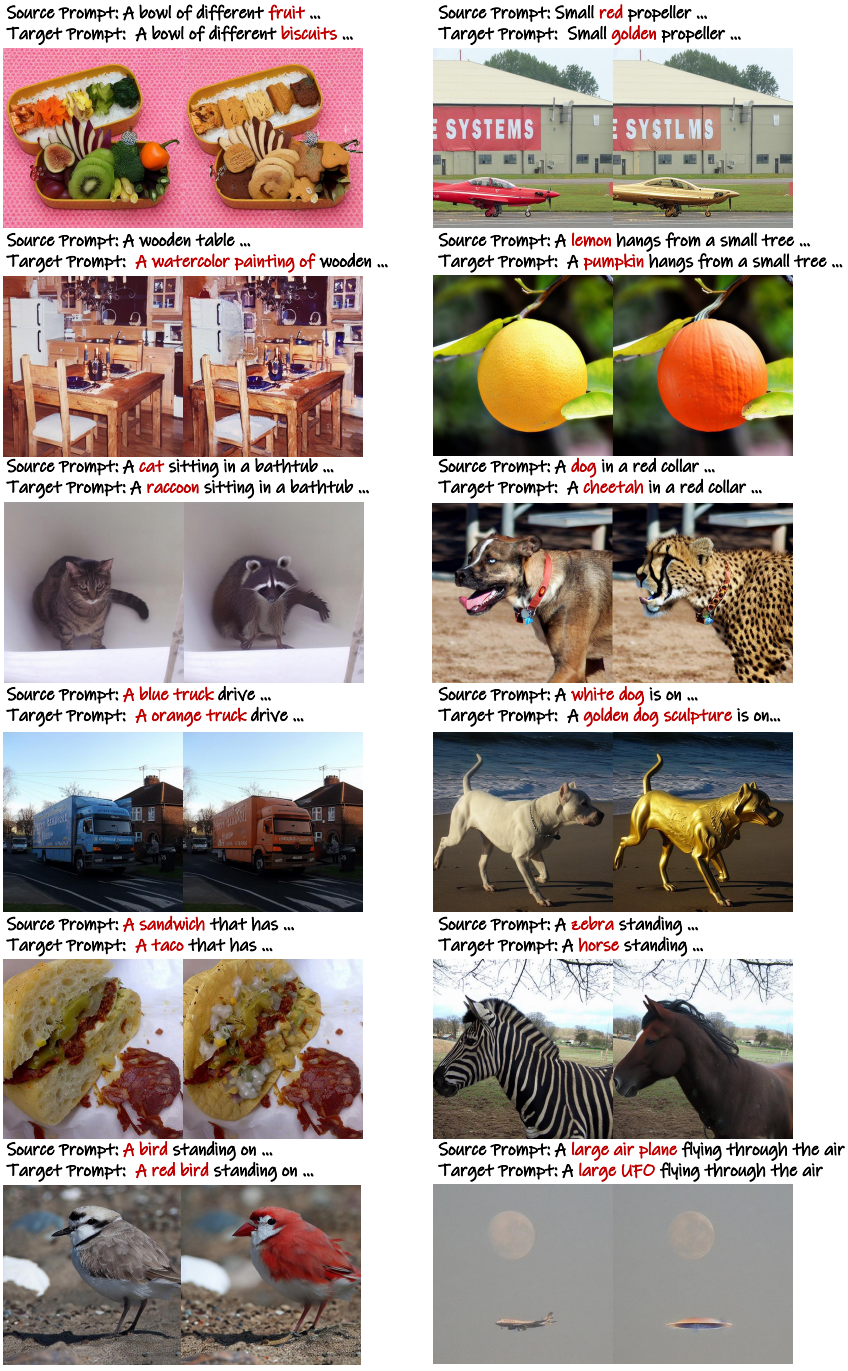

Fig. 3: Results of SPDInv on TDE-Bench.

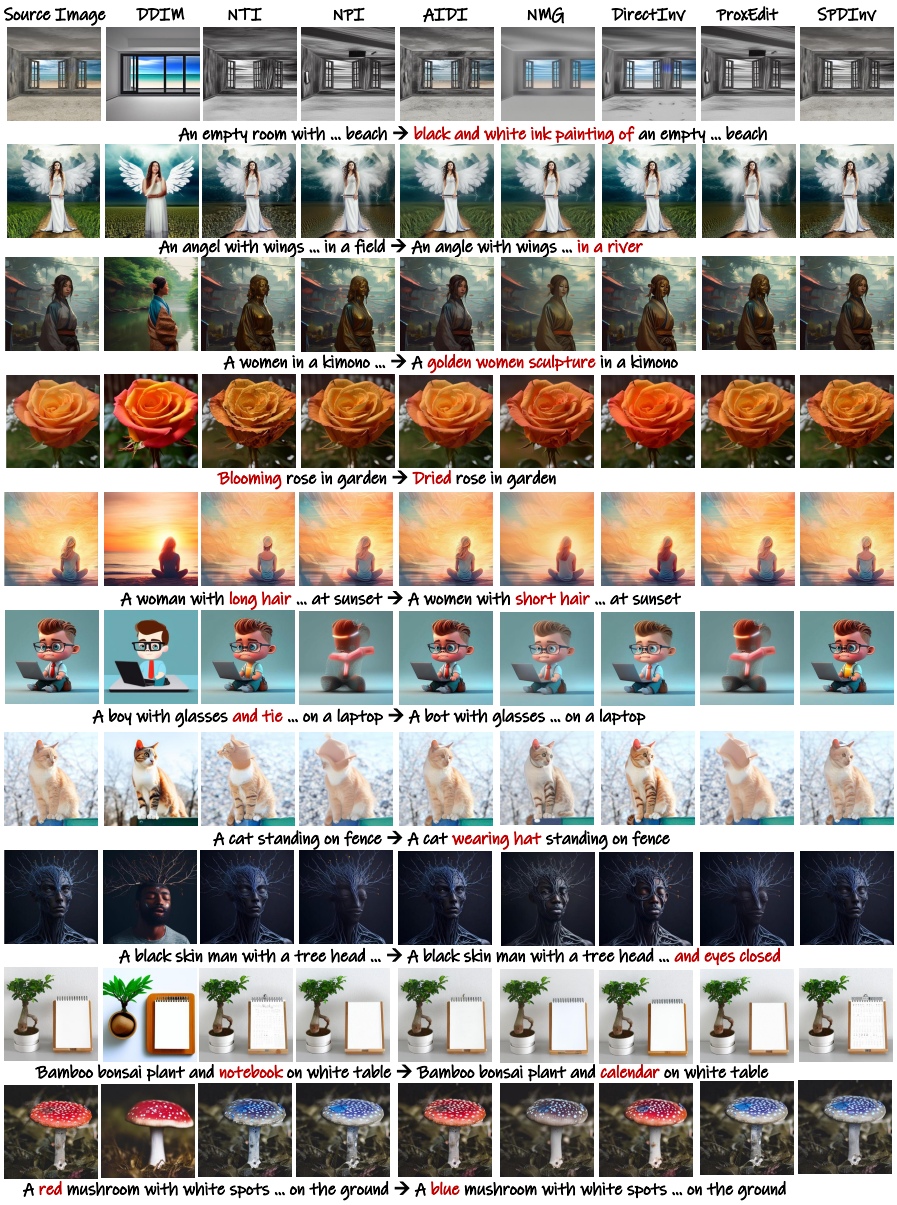

Fig. 4: More Visual comparisons of different editing methods with P2P.

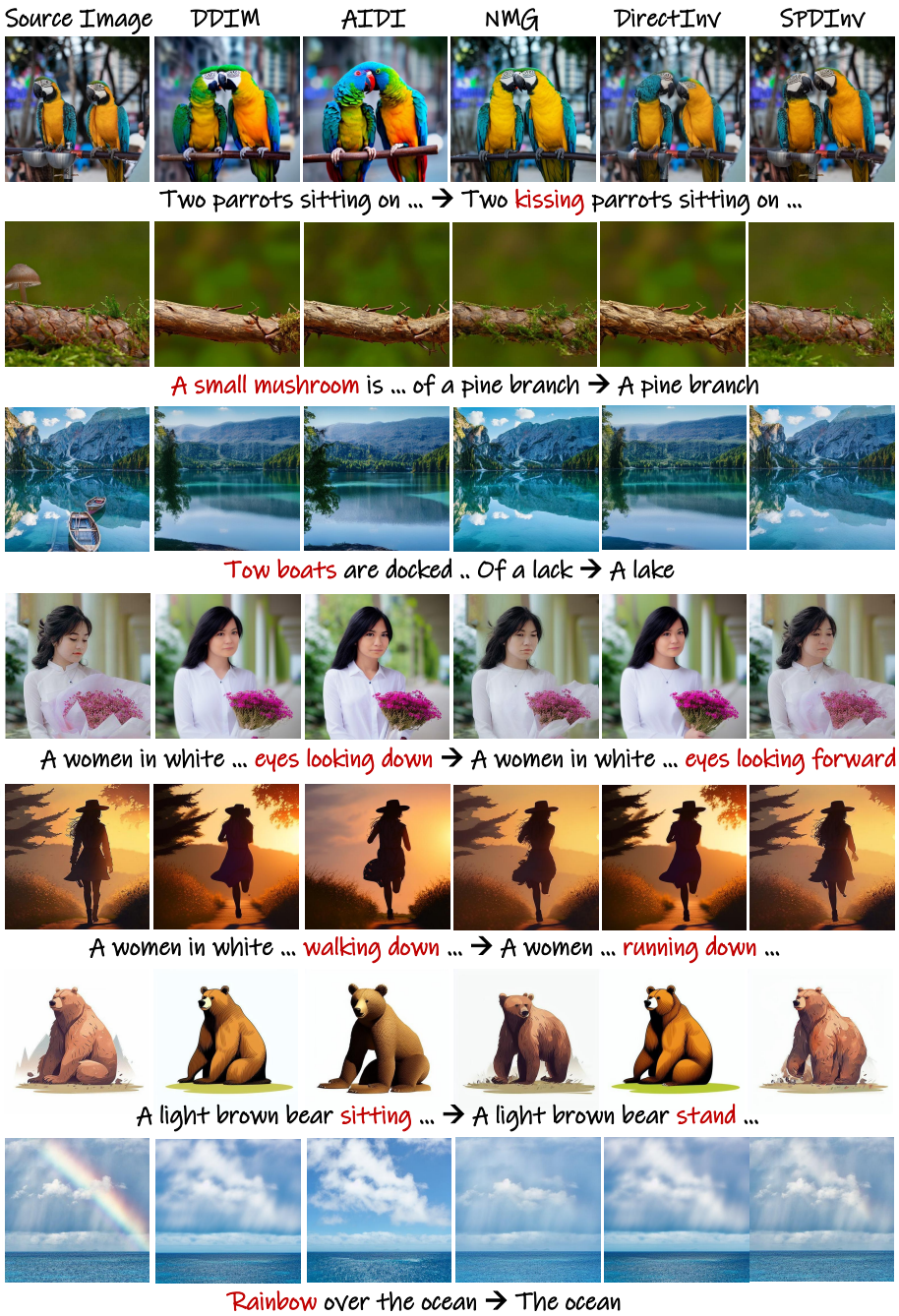

Fig. 5: Visual comparisons of different editing methods with MasaCtrl.

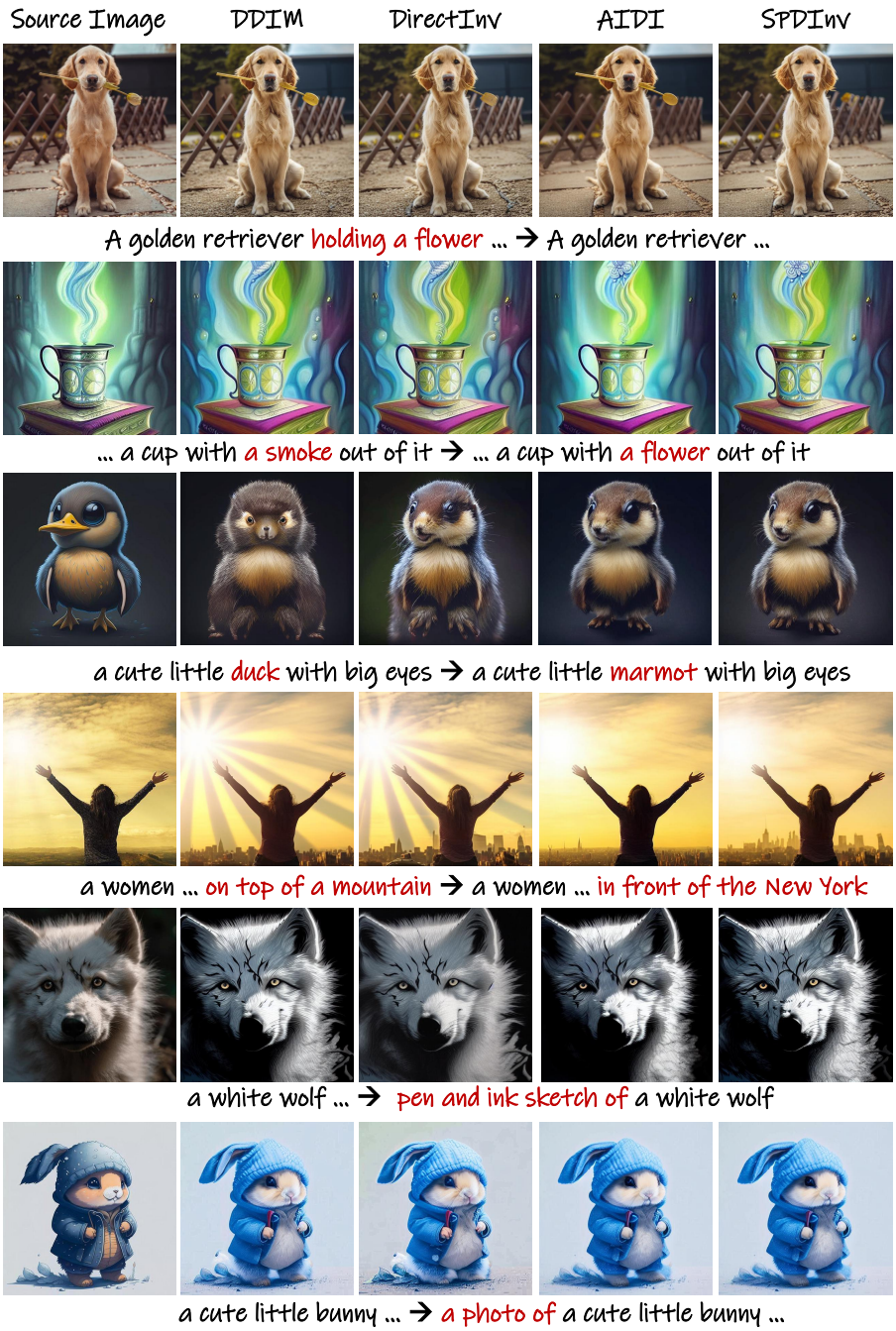

Fig. 6: Visual comparisons of different editing methods with PNP.

Fig. 7: Visual results of localized image editing with custom-diff.

Fig. 8: Failure cases of SPDInv.

References

1. Cao, M., Wang, X., Qi, Z., Shan, Y., Qie, X., Zheng, Y.: Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. arXiv preprint arXiv:2304.08465 (2023)
2. Choi, J., Choi, Y., Kim, Y., Kim, J., Yoon, S.: Custom-edit: Text-guided image editing with customized diffusion models. arXiv preprint arXiv:2305.15779 (2023)
3. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
4. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
5. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2020)
6. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-play diffusion features for text-driven image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1921–1930 (2023)
7. Wei, Y., Zhang, Y., Ji, Z., Bai, J., Zhang, L., Zuo, W.: Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. arXiv preprint arXiv:2302.13848 (2023)