Stable Score Distillation –Supplementary Material–

Anonymous ICCV submission

Paper ID 2976

This supplementary material contains the following parts:

- Sec. 1 provides the implementation details of our method.
- Sec. 2 explains the cross-trajectory term and null-text branch.
- Sec. 3 shows the quantitative evaluation in ablation study.
- Sec. 4 explains the connection with IP2P.
- Sec. 5 provides the comparison with PDS.
- Sec. 6 shows the more comparison in 2D image editing.
- more 3D scene editing results in video.

1. Implementation Details

For 3DGS [5] as a 3D representation, we implemented our method using the official codes of GS-Editor [1]. During editing, we utilized a maximum of 96 views for different scenes and trained the model for 1500 iterations (about **650 seconds** on one RTX3090 GPU for editing 512×512 image). By default, we set the classifier scale to 7.5 for the weight of **cross-prompt**, 2.0 for **cross-trajectory**, and 7.5 for **prompt-enhancement**. The diffusion model used is SD-2.1, along with InstructionP2P, which samples noise from $t \in [0.02, 0.98]$ and employs the DDIM [9] scheduler with 1000 steps.

For NeRF [7] as a 3D representation, we applied our method to the official codes of IN2N [2], the scale weights and diffusion model settings are the same as described above and set the number of iterations to 3K. For PDS [6], we used the default settings (30K iterations) from the official codes.

For image editing, we followed the same settings as DDS [3], using the DDIM scheduler with 1000 steps, with the same weights as used for 3D editing. The diffusion model is set to SD-1.5 by default, for both our method and the competitors.

2. More explanation about cross-trajectory

2.1. Theoretical explanation

In the main manuscript, we discussed that the cross-trajectory term represents the change in the current latent state, which is why we refer to it as cross-trajectory. Below, we provide a more detailed theoretical explanation:

If the source prompt is set as null-text, L_{ssd} corresponds to the generation process from the source latent:

$$L_{\text{ssd}} = s\left(\epsilon_\phi(z_t, y) - \epsilon_\phi(z_t, \varnothing)\right) + \left(\epsilon_\phi(z_t, \varnothing) - \epsilon_\phi(\hat{z}_t, \varnothing)\right), \tag{1}$$

the second term $\epsilon_\phi(z_t, \varnothing) - \epsilon_\phi(\hat{z}_t, \varnothing)$ represents the difference between predictions for the new trajectory and the old one under the null-text prompt. Hence, it can be regarded as the direction that transitions an image from the old trajectory to the new one. Notable, DDS yields the same result as Eq. 1 when the source prompt is set to null-text.

$$L_{\text{dds}} = s\left(\epsilon_\phi(z_t, y) - \epsilon_\phi(\hat{z}_t, \hat{y})\right) + (1-s)\left(\epsilon_\phi(z_t, \varnothing) - \epsilon_\phi(\hat{z}_t, \varnothing)\right), \tag{2}$$

However, in editing tasks, given a source prompt, the second term $(1-s)\left(\epsilon_\phi(z_t, \varnothing) - \epsilon_\phi(\hat{z}_t, \varnothing)\right)$ in Eq. 2, when regarded as the cross-trajectory term, has a negative weight $(1-s)$, which continually subtracts from the trajectory and has a detrimental effect on the optimization process.

In our design, term $\epsilon_\phi(z_t, \hat{y}) - \epsilon_\phi(\hat{z}_t, \varnothing)$ offers an interpretable explanation on the cross-trajectory term. Here, $\epsilon_\phi(z_t, \hat{y})$ is the distance between the current latent z_t and the source prompt $\hat{y}$, while $\epsilon_\phi(\hat{z}_t, \varnothing)$ is the distance between the source latent $\hat{z}_t$ and the null-text. The latter is a constant distance derived from the diffusion model, and we subtract it from the former to determine the trajectory direction.

2.2. Why introduce the null-text branch and the "clear" gradient of the cross-trajectory term

In the main manuscript, we introduce the null-text branch to guide the optimization process. Here, we further explain the null-text and the concept of "clear" gradient. First, we visualize how ϵ affects the image. As discussed in NFSD [4], subtracting the initial noise $(-\epsilon)$ yields a clean image. The predicted $\epsilon_\phi(\hat{z}_t, \hat{y}_t)$ and the residual $\epsilon_\phi(\hat{z}_t, \hat{y}_t) - \epsilon$ are shown in the first two rows of Fig. 1. The residual $\epsilon_\phi(\hat{z}_t, \hat{y}_t) - \epsilon$ generates much better images while preserving the source image structure. We want to emphasize that DDS eliminates the initial noise $(-\epsilon)$ through the "Delta" operation by subtracting the source branch from the target branch.

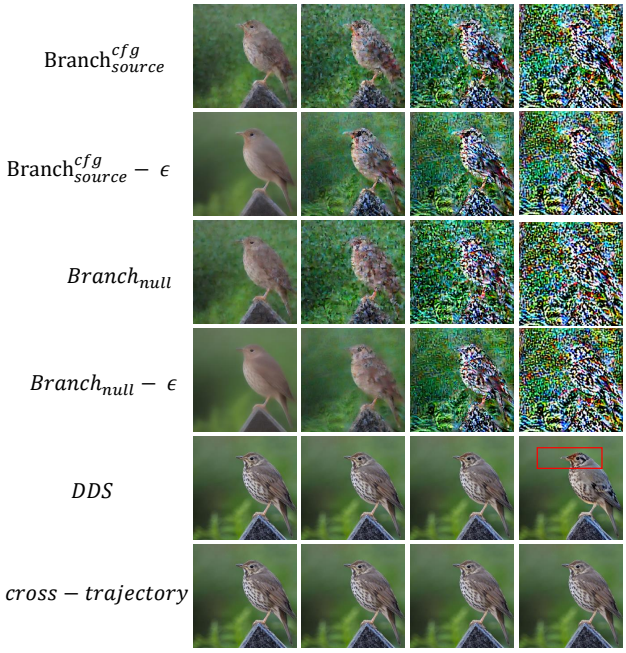

Figure 1. Illustration of the gradient of DDS and Our design. The first two rows show the source branch predicted $\epsilon_\phi(\hat{z}_t, \hat{y}_t)$ and the residual $\epsilon_\phi(\hat{z}_t, \hat{y}_t) - \epsilon$. The middle two rows show the predicted null-text. The last two rows illustrate the gradients of DDS and our **cross-trajectory** term. From left to right, the figures represent the optimization iteration process.

Secondly, regarding the null-text branch, we found that it can also estimate model bias, as shown in Fig. 1. The null-text branch produces the same effect as the source branch. Ideally, the null-text branch should not include the edit direction, which not provide any information about the source prompt. Thus, we can consider the null-text branch as a "clear" gradient.

Furthermore, we conducted an experiment to demonstrate the "clear" gradient. In this experiment, the target prompt was set to be the same as the source prompt, and our L_{ssd} only included the **cross-trajectory** term. Ideally, the gradient in this setup should not modify the source image. As shown in Fig. 1, during the optimization process, the gradient of DDS modifies the head (red box) and wing of bird, whereas the gradient of our **cross-trajectory** term does not modify the image at all, indicating that the gradient of **cross-trajectory** term is "clear".

3. Quantitative Evaluation in Ablation Study

Due to space limitations in the main manuscript, we provide the quantitative evaluation in Tab. 1. Additionally, we include the metric for "CLIP Image", which calculates the similarity between the edited image and the source image, evaluating the performance in preserving the source image's structure. Notably, the results in Tab. 1 are calculated using

Table 1. Quantitative evaluation in ablation study.

Method	CLIP Sim ↑	Sim Dire ↑	CLIP Image ↑
L_ssd	**0.1977**	**0.1169**	91.47
$L\_ssd + L\_ID$	0.1937	0.0970	**92.42**
$L\_ssd + L\_align$	0.1954	0.1111	91.75
Full	0.1938	0.1040	92.10

scene cases processed by the SD model, which differs from Table 1 in the main manuscript that uses both the SD model and InstructionP2P. As analyzed in the main manuscript, InstructionP2P does not include the L_{align} and L_{ID} components.

In the first two rows of Tab. 1, we observe that using L_{ID} improves the "CLIP Image" score while correspondingly decreasing the "CLIP Sim" and "Sim Dire" metrics. The L_{ID} term strictly preserves the source image, and in some cases, it constrains the editing strength. In our design, the L_{ID} term is intended to prevent gradient explosions and ultimately leads to more visually appealing results.

The L_{align} component effectively enhances style editing for 2D images. For 3D scene editing as shown in the third row of Tab. 1, L_{align} improves the "CLIP Image" score while reducing "CLIP Sim" and "Sim Dire." This is achieved by setting $w_t = 2.0$ in L_{ssd} to maintain **cross-trajectory**. Simply adding L_{align} to the loss function, however, may lead to a decrease in performance.

Overall, the full model achieves balanced performance across all three metrics, which is desirable for editing tasks.

4. Connection with IP2P

We talk about the two terms perspectively in the following.

(i) The term $\epsilon_\phi(z_t, \hat{y}) - \epsilon_\phi(\hat{z}_t, \varnothing)$ in Eq.6 corresponds to the effect of the initial point (constant term $\epsilon_\phi(\hat{z}_t, \varnothing)$) on the current latent z_t, where the source latent $\hat{z}_t$ and source prompt $\hat{y}$ influence the optimization. In IP2P, the source image is embedded into the model architecture, whereas our method provides the source latent and prompt implicitly.

(ii) The cross-prompt term $\epsilon_\phi(z_t, y) - \epsilon_\phi(z_t, \hat{y})$ in Eq.6 corresponds to the influence of the transition from the source prompt to the target prompt y on the current latent z_t. In IP2P, $\epsilon_\theta(z_t; c_I; c_T) - \epsilon_\theta(z_t; c_I; \varnothing)$ in Eq.10 describes how the instruction c_T affects the latent representation. In a word, SSD provides a constant term in the optimization process, while IP2P embeds the source image into the model. Two terms in SSD are equivalent to the two terms in IP2P, respectively. We have presented the experimental results of reweighting the two terms in IP2P, which produce the same effect as shown in Fig.4.

5. Comparison with PDS

We provide additional visual results comparing our method with PDS [6], a scene editing approach based on posterior

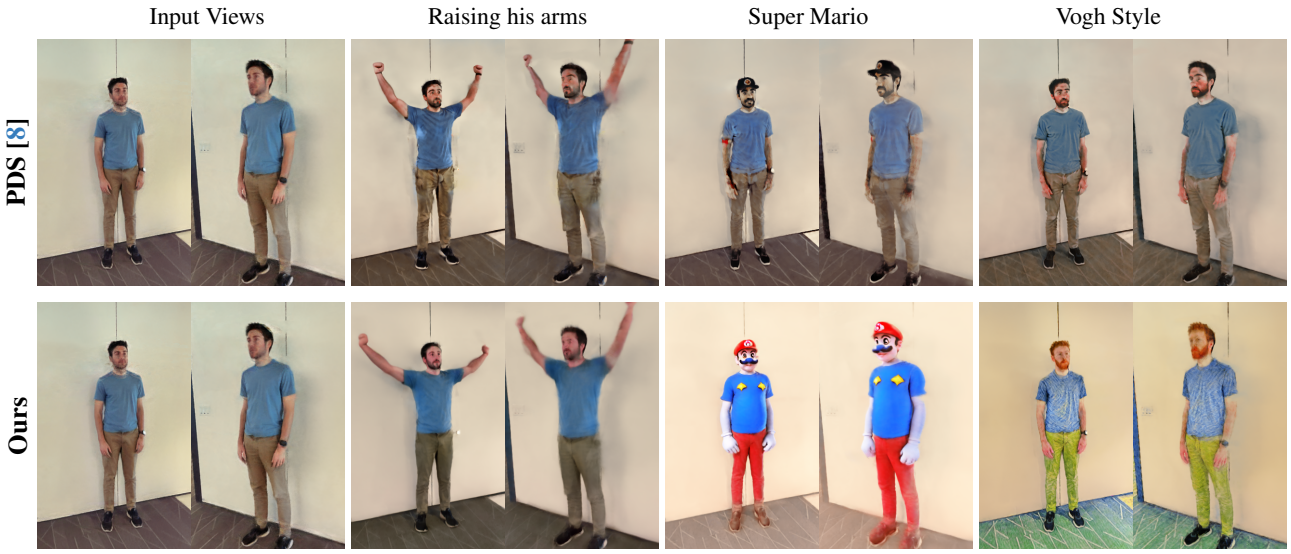

Figure 2. Qualitative comparison with PDS, our approach (SDS) demonstrates superior performance in pose change, object change, and style editing tasks, providing more realistic and visually appealing results.

Table 2. Quantitative comparison with PDS.

Method	CLIP Sim ↑	Sim Dire ↑
PDS	0.1432	0.018
Ours	**0.1503**	**0.026**

sampling. PDS attempts to match the stochastic latent between the source and target prompts, however, it suffers from slow convergence (30K iterations, about **10 hours**) and inferior editing quality.

In Fig. 2, we present a comparison between PDS and our method (3K iterations). It is evident that our method produces more realistic and visually appealing results in pose change, object change, and style editing tasks. The performance of PDS is constrained by the stochastic latent, showing strengths in pose changes and object additions, but falling short in content editing. In contrast, our method offers more stable and precise editing results, making it better suited for editing tasks.

In Tab. 2, we provide a quantitative comparison with PDS. Our method outperforms PDS in both "CLIP Sim" and "Sim Dire" metrics, demonstrating the effectiveness of our design in 3D scene editing tasks.

6. More Comparison in 2D Image Editing

We provide additional comparisons of 2D image editing results. In Fig. 3, we present comparison results for content editing tasks, including object change and pose change. Our method produces more realistic and visually appealing results compared to DDS and CDS, while inversion-based editing methods often degrade the source images. In Fig. 4, we show the comparison results for style editing tasks. Our method achieves superior performance in style editing, demonstrating its effectiveness across various editing scenarios.

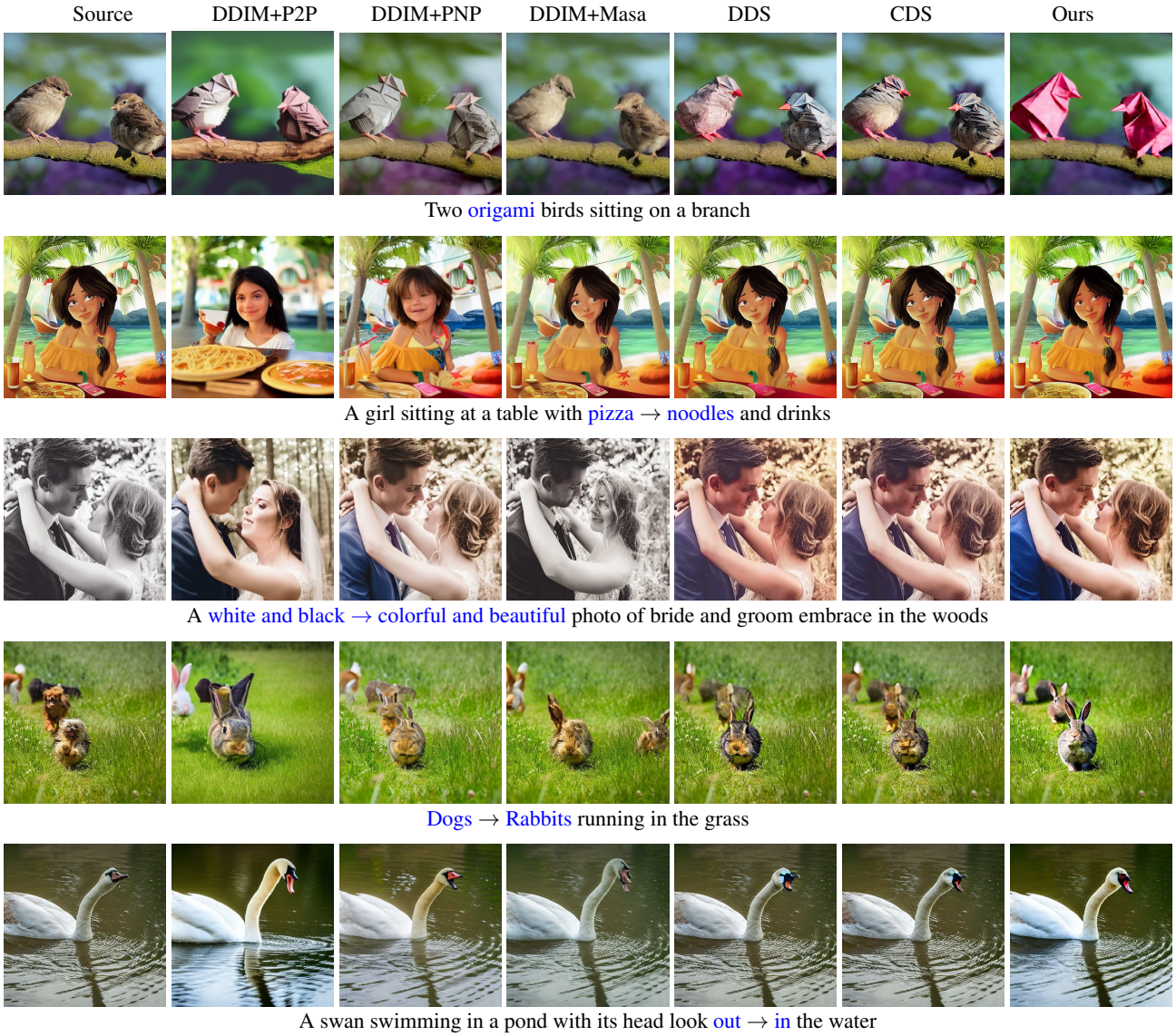

Figure 3. Comparison of different editing methods in content editing.

References

[1] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *CVPR*, pages 21476–21485, 2024. 1

[2] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *ICCV*, 2023. 1

[3] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *ICCV*, pages 2328–2337, 2023. 1

[4] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. In *ICLR*, 2024. 1

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 42(4):1–14, 2023. 1

[6] Juil Koo, Chanho Park, and Minhyuk Sung. Posterior distillation sampling. In *CVPR*, pages 13352–13361, 2024. 1, 2

[7] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421, 2020. 1

[8] JangHo Park, Gihyun Kwon, and Jong Chul Ye. ED-NeRF: Efficient text-guided editing of 3d scene with latent space nerf. In *ICLR*, 2024. 3

[9] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1

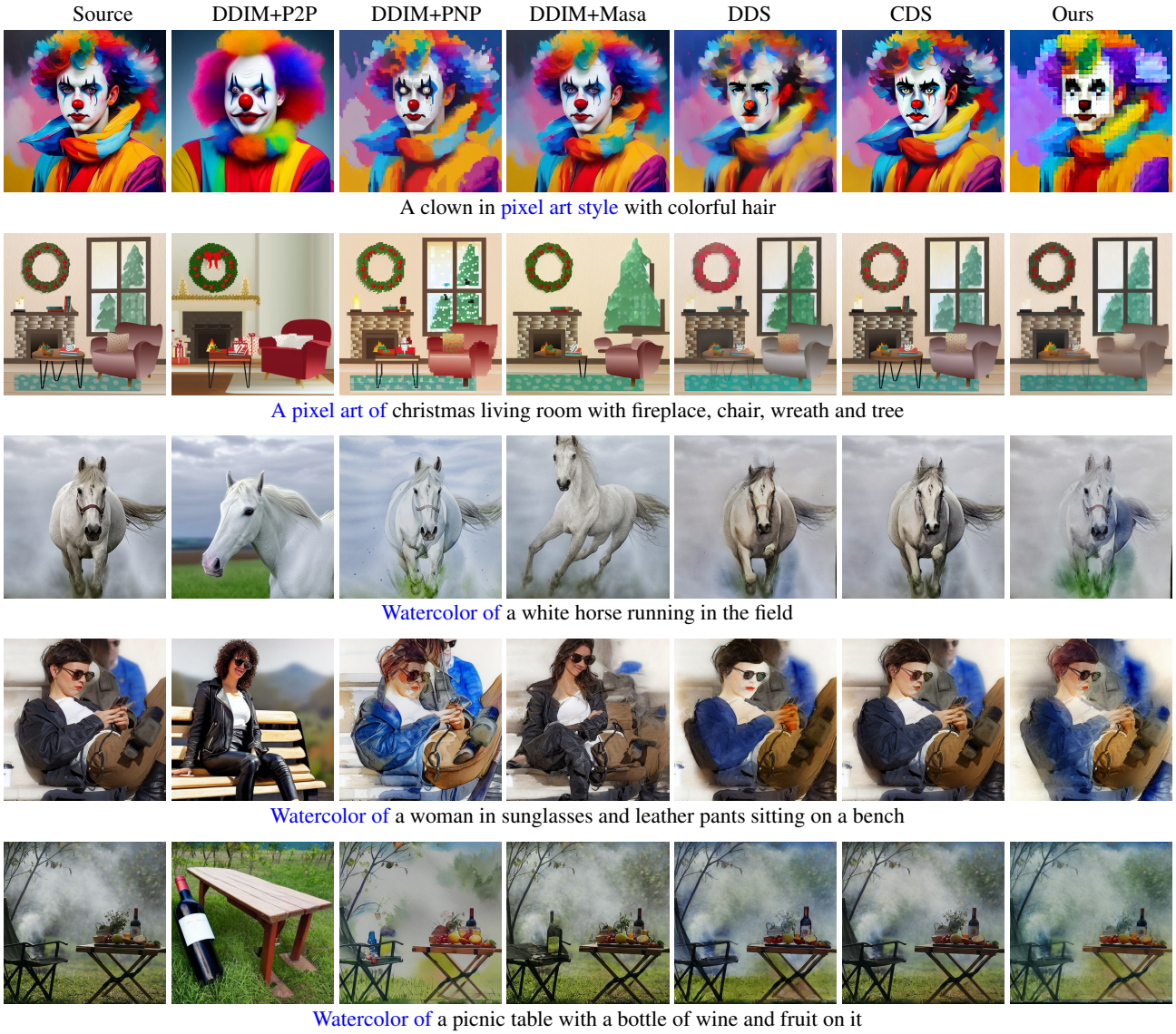

Figure 4. Comparison of different editing methods in style editing.