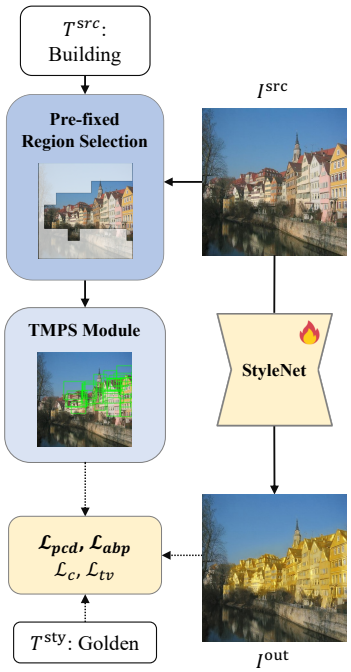
Style-Editor: Text-driven object-centric style editing Supplementary Material

A. Detailed training process of the Style-Editor



- The model takes three inputs: a source image (I^{src}), which is the subject of the style editing; a source text (T^{src}), which identifies the specific object in the source image to be modified; and a style text (T^{sty}), describing the desired style to be applied to the object.
- In the early iterations, the Pre-fixed Region Selection (PRS) is used to divide I^{src} into coarse foreground and background regions.
- Within the coarsely segmented foreground region, patches are generated. The Text-Matched Patch Selection (TMPS) module then comes into play, selecting those patches that correspond to the object mentioned in T^{src}.
- The training of the model incorporates a combination of loss functions: Patch-wise Co-Directional loss ($\mathcal{L}_{\text{pcd}}$), Adaptive Background Preservation loss ($\mathcal{L}_{\text{abp}}$), Content loss ($\mathcal{L}_{\text{c}}$), and Total Variation loss ($\mathcal{L}_{\text{tv}}$).

Figure 7. Simplified overview of the training process.

A.1. Pre-fixed Region Selection (PRS)

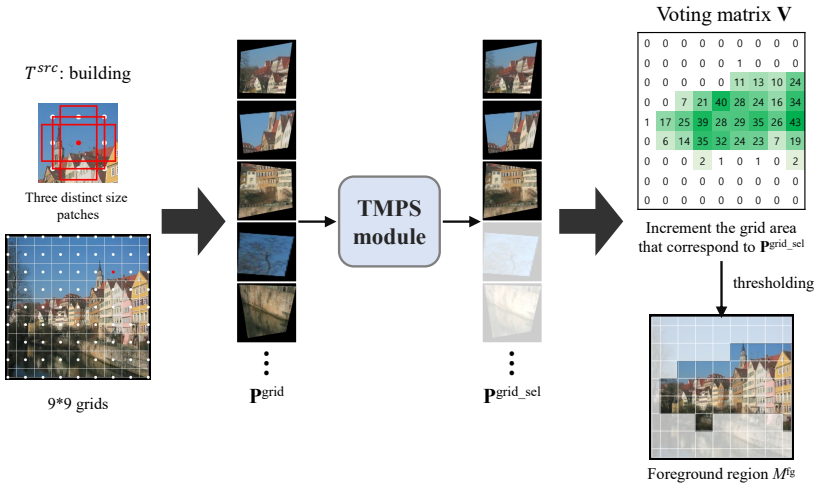


Figure 8. Overview of Pre-fixed Region Selection (PRS).

In our approach, we process the source image (I^{src}) by dividing it into a 9×9 grid. From each grid point, we generate three patches of varying sizes, centered on these points, resulting in a total of 243 patches. These patches are then passed

through the Text-Matched Patch Selection (TMPS) module, which identifies and selects the patches that are most relevant to the source text (T^{src}). We use a specific threshold τ (we set as 2) to determine the foreground region (M^{fg}) of the image: grids that exceed this threshold in terms of patch relevance are classified as part of the foreground. After the initial iterations, further patch generation is concentrated within this coarse foreground region. The number of patches generated in subsequent iterations is adjusted proportionally to the count of grids identified as part of the foreground region. This method ensures a focused and relevant application of style editing where it is most pertinent according to the text input.

B. Comprehensive Analysis of our Style-Editor

In this section, we provide a detailed analysis of our Style-Editor model by evaluating its components and performance through various experiments. Specifically, we focus on the impact of the patch distribution consistency loss, visualize the style editing process, assess the computational overhead of the TMPS and PRS modules, explore the effect of patch size, and compare our patch-wise approach with segmentation-based methods. This detailed examination and presentation of results not only highlight the efficacy of our Style-Editor model but also provide valuable insights into the model's operational dynamics throughout its training phase.

B.1. The patch distribution consistency loss

To demonstrate the impact of the patch distribution consistency loss, denoted as $\mathcal{L}_{\text{con}}$, we conduct an ablation study. The findings from this study are displayed in Fig. 9. A key observation from our experiments is that the inclusion of $\mathcal{L}_{\text{con}}$ significantly contributes to the preservation of vital features in the images, even after the style editing process. Notable examples include the retention of text on a T-shirt, the intricate pattern of a tropical fish, and the clarity of numbers on a bowling ball. These results highlight that the implementation of the $\mathcal{L}_{\text{con}}$ loss is crucial in maintaining a focused distribution of patches on the object. This focused approach is what enables the preservation of essential details and textures during the style editing, ensuring that the core visual elements of the object remain intact and recognizable post-transformation. This study thus underscores the effectiveness of the $\mathcal{L}_{\text{con}}$ loss in enhancing the quality and fidelity of object-centric style editings in our Style-Editor model.

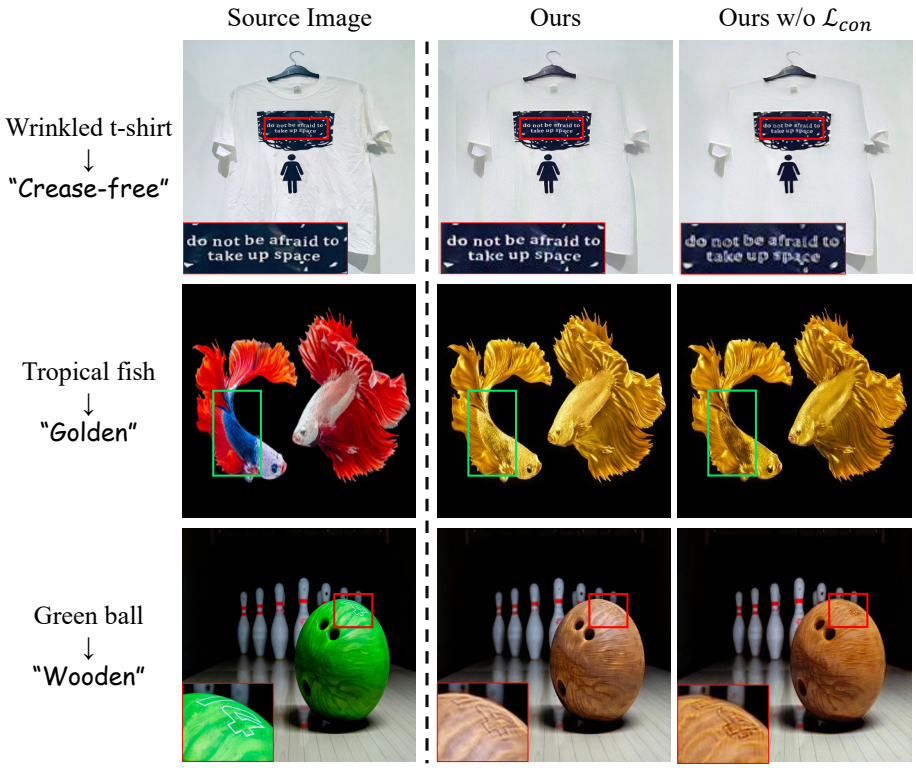


Figure 9. Qualitative comparison demonstrating the effect of the $\mathcal{L}_{\text{pcd}}$ loss by contrasting results with and without the $\mathcal{L}_{\text{con}}$ loss.

B.2. Visualization of style editing process

Fig. 10 shows the iterative process of style editing as conducted by our model. Initially, the model prioritizes the preservation of the background, ensuring it remains as close to the original image as possible. This early focus on background fidelity is a crucial step in maintaining the overall integrity and context of the source image. As the training progresses through successive iterations, the model begins to refine its approach. It gradually learns to enhance and accentuate the details within both the object and the background. This progression illustrates the model's sophisticated capability to strike a balance between maintaining background fidelity and enhancing the object of interest. This iterative learning and adaptation process is a key aspect of our model's functionality. It demonstrates how the model evolves to effectively manage the complexities of style editing, ensuring that both the object's details and the background's essence are harmoniously preserved and enhanced.

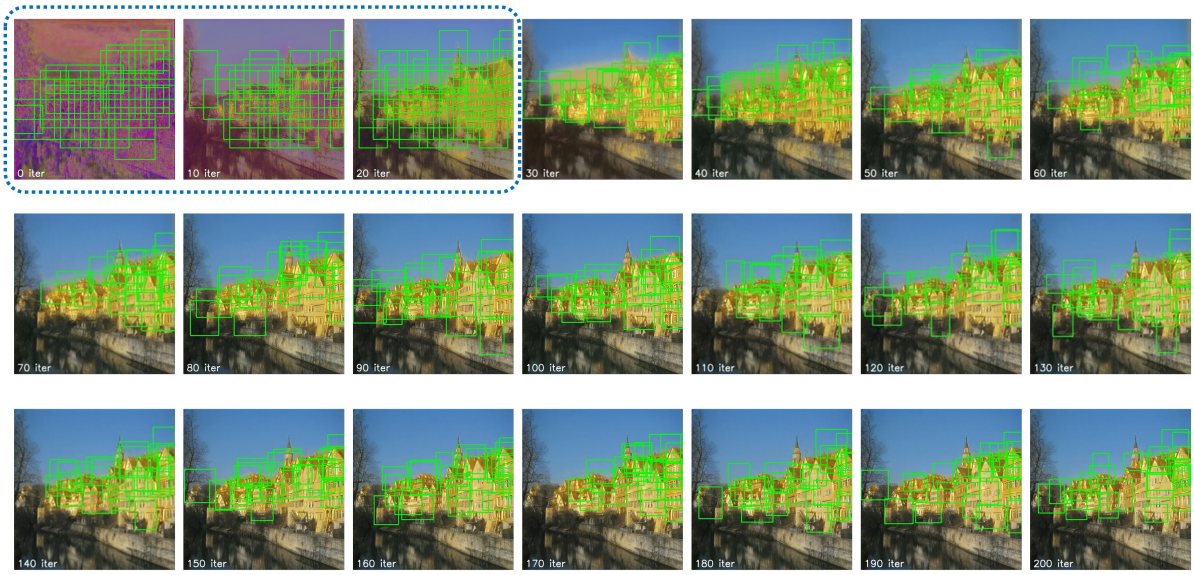


Figure 10. Visualization of the style editing process at intervals of every 10 iterations. This sequence illustrates the progressive transformation and refinement of the image style over time. The parts marked with blue dashed lines denote the iterations where the Pre-fixed Region Selection (PRS) module operates, and the green patches in each iteration denote the patches selected through the Text-Matched Patch Selection (TMPS). Areas not included in the patch are the regions where the Adaptive Background Preservation loss ($\mathcal{L}_{\text{abp}}$) is applied.

B.3. Computational overhead of TMPS and PRS

We conducted additional experiments to assess the computational overhead of the TMPS and PRS modules. In the course of training, we processed the model with 5 distinct styles of text and 3 varieties of source images 10 times to calculate the average time spent. These time estimates encompass both the loading of the CLIP model and the image processing depicted in Tab. 3-(a), as well as the actual training period shown in Tab. 3-(b). This data reveals that the proposed modules add merely an extra 10 seconds to the process (see Ours), and that the PRS can indeed reduce training time by approximately 8-9 seconds. This supports our assertion. During the inference phase, which merely involves loading a pre-saved training checkpoint and introducing an image, the presence or absence of any modules does not alter the runtime as shown in Tab. 3-(c), ensuring consistency across different scenarios. Thus, our experiments illustrate that while the inclusion of the TMPS and PRS modules extends training and inference times, the increase is not significantly detrimental.

Method (seconds)	(a) Total training time (+model & data load)	(b) Training time	(c) Inference time
w/o PRS, w/o TMPS	40.7s	35.2s	0.28s
w/o PRS, w TMPS	59.9s (+19.2s)	53.3s (+18.1s)	0.28s (+0.0s)
w PRS, w TMPS (Ours)	51.9s (+11.2s)	44.3s (+9.1s)	0.28s (+0.0s)

Table 3. Comparison of different methods on training and inference times

B.4. Impact of Patch Size

We conducted additional experiments to investigate the impact of patch size in the TMPS module. The experimental setup and results are presented in Tab. 4 and Tab. 5, highlighting how variations in patch sizes affect key performance metrics. Our findings demonstrate that the TMPS module remains robustness across different patch sizes. Notably, our adaptive selection of patch sizes, ranging from 64 to 128 for images of 512×512 size, yields well-rounded results throughout our experiments. This adaptive approach reflects a careful trade-off: smaller patches enhance image quality metrics, such as PSNR, Con_F, and Con_B, while larger patches help preserve stylization consistency, especially around object boundaries. Nevertheless, it is worth noting that such a reduction in patch size might also decrease the CLIP similarity for the foreground (Sim_F), potentially compromising the content's integrity.

Methods	Foreground quality metrics		Background quality metrics					
	$\text{Sim}_F \uparrow$	$\text{Con}_F \downarrow$	$\text{L1}_B \downarrow$	$\text{Con}_B \downarrow$	$\text{Sty}_B \downarrow$	$\text{SSIM}_B \uparrow$	$\text{DISTS}_B \downarrow$	$\text{PSNR}_B \uparrow$
patch 64	0.3	**3.46**	**0.09**	1.2	0.11	**0.90**	0.08	**27.92**
patch 96	0.32	3.94	**0.09**	1.25	0.12	0.89	0.08	27.56
patch 128	**0.33**	4.24	0.10	1.29	0.13	0.89	0.08	27.17
Ours (patch 64-128)	**0.33**	3.75	0.10	**1.15**	**0.10**	**0.90**	**0.07**	27.65

Table 4. Quantitative evaluation of fixed patch sizes. Metrics include similarity (Sim_F, L1_B), content loss (Con_F, Con_B), style loss (Sty_B), structural similarity index (SSIM), perceptual distance (DISTS), and peak signal-to-noise ratio (PSNR). ↑ indicates higher values are better, while ↓ indicates lower values are better.

Methods	Foreground quality metrics		Background quality metrics					
	$\text{Sim}_F \uparrow$	$\text{Con}_F \downarrow$	$\text{L1}_B \downarrow$	$\text{Con}_B \downarrow$	$\text{Sty}_B \downarrow$	$\text{SSIM}_B \uparrow$	$\text{DISTS}_B \downarrow$	$\text{PSNR}_B \uparrow$
patch 32-64	0.29	**2.90**	**0.09**	**1.12**	0.11	**0.91**	0.08	**28.41**
Ours (patch 64-128)	0.33	3.75	0.10	1.15	**0.10**	0.90	**0.07**	27.65
patch 128-256	**0.34**	4.43	0.11	1.37	0.15	0.87	0.09	26.05

Table 5. Quantitative evaluation of adaptive patch sizes. Metrics include similarity (Sim_F, L1_B), content loss (Con_F, Con_B), style loss (Sty_B), structural similarity index (SSIM), perceptual distance (DISTS), and peak signal-to-noise ratio (PSNR). ↑ indicates higher values are better, while ↓ indicates lower values are better.

B.5. Comparison with Segmentation method

To evaluate the effectiveness of our approach compared to segmentation-based methods, we conducted experiments by applying a mask to our results using the open-vocabulary segmentation model ODISE [61]. The comparative results are illustrated in Fig. 11. When applying a 'Sunlight' style to a towel image, our model generates results where light naturally diffuses around the object. In contrast, segmentation-based methods rely on per-pixel binary classification (0 or 1) to determine the presence of an object, resulting in abrupt and unnatural transitions at the boundaries where the style is applied. This disrupts the overall continuity and realism of the stylized image. Furthermore, segmentation-based methods typically require additional training on datasets, such as the MSCOCO [38] dataset. These findings highlight the advantage of our model in maintaining seamless and natural style editing, avoiding the artifacts commonly observed at object boundaries in segmentation-based methods. Moreover, by leveraging the CLIP encoder for both style transfer and object patch identification, our approach eliminates the need for pre-trained segmentation networks, achieving both simplicity and effectiveness.

C. Visualization of evaluation metric

To evaluate the performance of object-centric style editing, we conducted evaluations using the annotations from the MS COCO 2017 dataset. Fig. 12 presents the visualized results used as reference examples for the evaluation metrics employed in our experiments. In particular, $M^{\text{fg\_gt}}$ denotes the ground truth (GT) mask corresponding to the target class. Using this mask, we calculated the foreground quality metrics by masking regions outside the class object areas and cropping the images according to the mask area, as shown in $I^{\text{src}} \odot M^{\text{fg\_gt}}$ and $I^{\text{out}} \odot M^{\text{fg\_gt}}$ in the Fig. 12. Conversely, the background quality

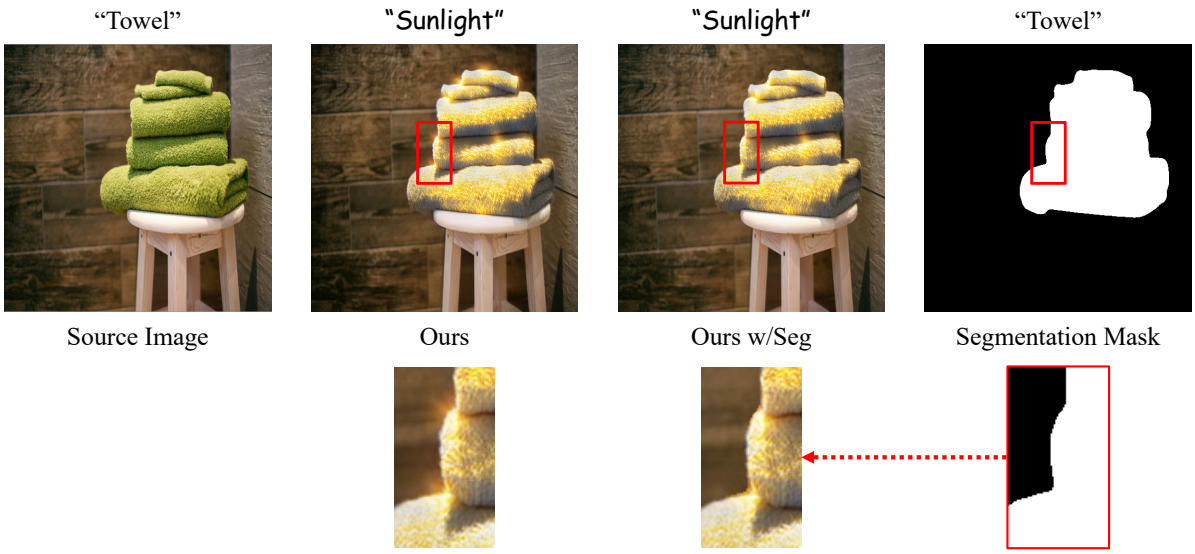


Figure 11. Qualitative comparison between our results and our results with segmentation masks applied.

metrics were measured using images where the object regions were excluded, as represented by $I^{\text{src}} \odot M^{\text{bg\_gt}}$ and $I^{\text{out}} \odot M^{\text{bg\_gt}}$ in the Fig. 12.

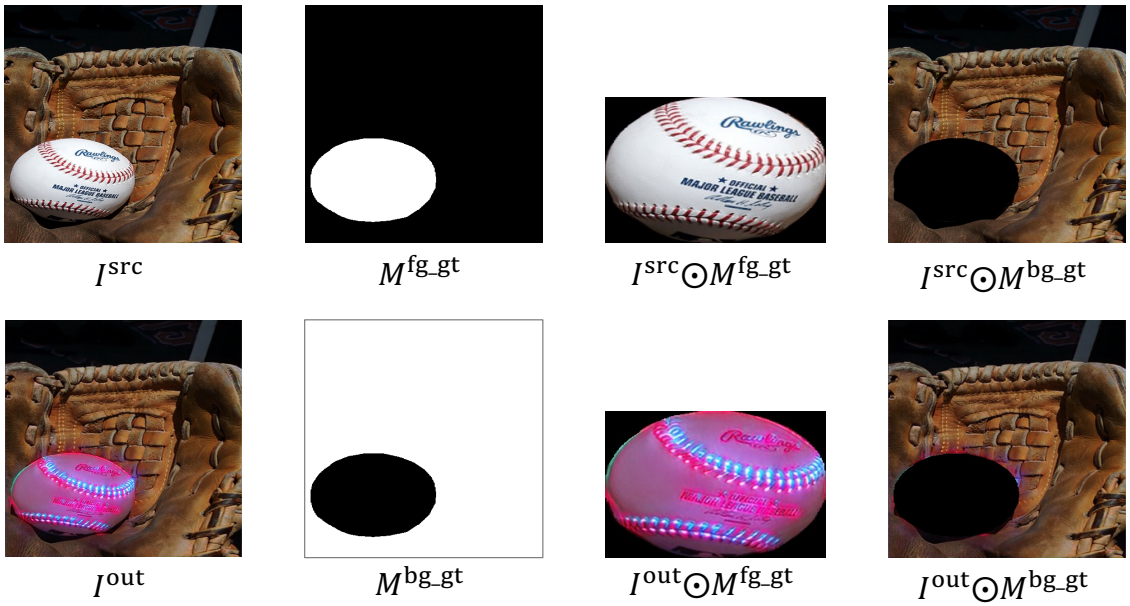


Figure 12. Example images for evaluating foreground quality metrics and background quality metrics.

D. Complex input texts

We have conducted further experiments using a diverse set of examples featuring intricate 'source' and 'style' texts resulting in Fig. 13. For example, in the 'cake to emerald' scenario, although the stylized image with simple text retains aspects of the original cake's style, the detailed source text enables the creation of a cake styled purely in emerald. Similarly, in the 'barn to snowy' scenario, the model adeptly preserves the background's style while effectively applying style editing to the foreground object. Furthermore, our experiments incorporating complex 'style' texts illustrate the complexity of the text does not impede our method's ability to achieve the intended editing effects. These results clearly demonstrate that our method is adept not only at managing simple texts but is equally effective with complex textual inputs. This expansion of our testing framework underscores our method's robustness and versatility in accommodating a broad spectrum of textual complexities.

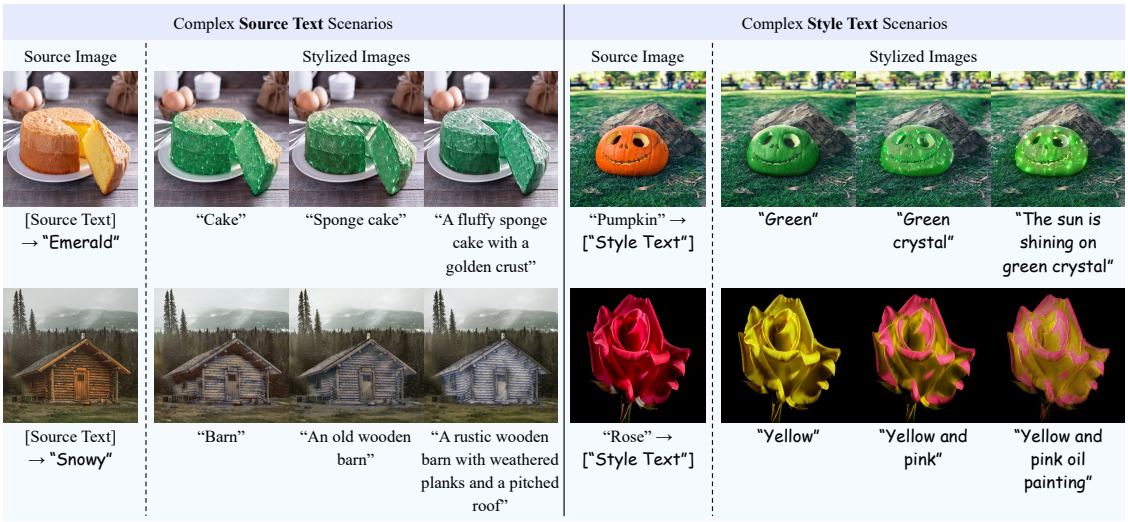


Figure 13. Our style editing results with complex image-text scenarios.

E. Limitations of Style-Editor

A noted limitation of Style-Editor is its reliance on the CLIP model's feature space and classification capabilities. Consequently, its performance may diminish for styles or objects less represented or absent in the CLIP training dataset, such as recent art or gadgets, as illustrated in Fig. 14. This dependency highlights a potential area for future development, aiming to enhance Style-Editor's adaptability to emerging styles and objects.

Figure 14. Failure cases of Style-Editor. The numbers following the style text denote the year of the art work was created.

F. User study

We conducted two rounds of user studies to enhance our evaluation process, the results of which are presented in Tab. 6. This study involved 50 participants, whose ages ranged from their 20s to 50s. We designed the study to assess three key aspects: stylization quality, background preservation, and content preservation, providing nine examples within each category for evaluation. The evaluation included the mask-based editing models Glide [41], and Blended Diffusion [1], along with the top-4 models from Tab. 1 — FlexIT [7], LEDITS ++ [3], Null text inversion [40], and Text2LIVE [2]. In each round, we compared the stylized images produced by three competing models with those generated by our model. Our analysis indicates that our model excels in achieving a balanced object-centric style editing, effectively maintaining the semantic integrity of the object and the background without compromise. In particular, compared with mask-based methods, our model delivers more natural and stable performance. The user study question examples are illustrated in Fig. 15.

Method	StyQual ↑	BackPre ↑	ContPre ↑
Blended Diffusion [1]	4.4 %	7.3 %	2.4 %
Glide [41]	1.1 %	12.0 %	3.6 %
Text2LIVE [2]	29.3 %	14.9 %	11.8 %
FlexIT [7]	14 %	5.1 %	2.9 %
LEDITS++ [3]	13.3 %	6 %	6.2 %
Null-text inversion [40]	5.8 %	25.8 %	16 %
Ours (Avg.)	**66** %	**64.4** %	**78.6** %

Table 6. User study detailing preference percentages.

G. Additional quantitative results

To assess the robustness of our model, we conducted additional quantitative experiments on the top-5 models from Tab. 1, with training time evaluated following the settings in Sec. B.3. The test set was expanded by randomly selecting 50 images from the MSCOCO 2017 dataset and pairing each with 10 different style text descriptions, yielding a total of 500 stylized images for evaluation. As presented in Tab. 7, our model consistently outperforms others on the extended test set while ranking second in speed. Notably, even compared to LEDITS++ [3], the fastest model, our approach demonstrates superior performance in both foreground and background metrics. These results validate its strong generalization ability and highlight its well-balanced stylized output at a relatively fast pace.

Methods	Foreground metrics		Background metrics						
	Sim_F ↑	Con_F ↓	L1_B ↓	Con_B ↓	Sty_B ↓	SSIM_B ↑	DISTS_B ↓	PSNR_B ↑	Time (s) ↓
FlexIT [7]	0.24	7.03	0.20	4.09	0.36	0.66	0.14	21.17	59.8
LEDITS++ [3]	0.21	5.98	0.19	2.70	0.42	0.75	0.13	21.43	**10.4**
NTI [40]	0.20	4.64	0.16	3.02	0.32	0.74	0.12	23.46	102.3
LPM [46]	0.20	8.62	0.25	4.81	0.73	0.67	0.19	19.53	119.9
Text2LIVE [2]	<u>0.30</u>	<u>3.65</u>	<u>0.13</u>	<u>1.25</u>	<u>0.17</u>	<u>0.87</u>	**0.08**	<u>25.47</u>	412.6
Ours	**0.31**	**3.09**	**0.10**	**0.95**	**0.08**	**0.89**	**0.08**	**26.71**	<u>44.3</u>

Table 7. Additional quantitative comparison with 500 stylized images. ↑ indicates that higher values are better, while ↓ indicates that lower values are better.

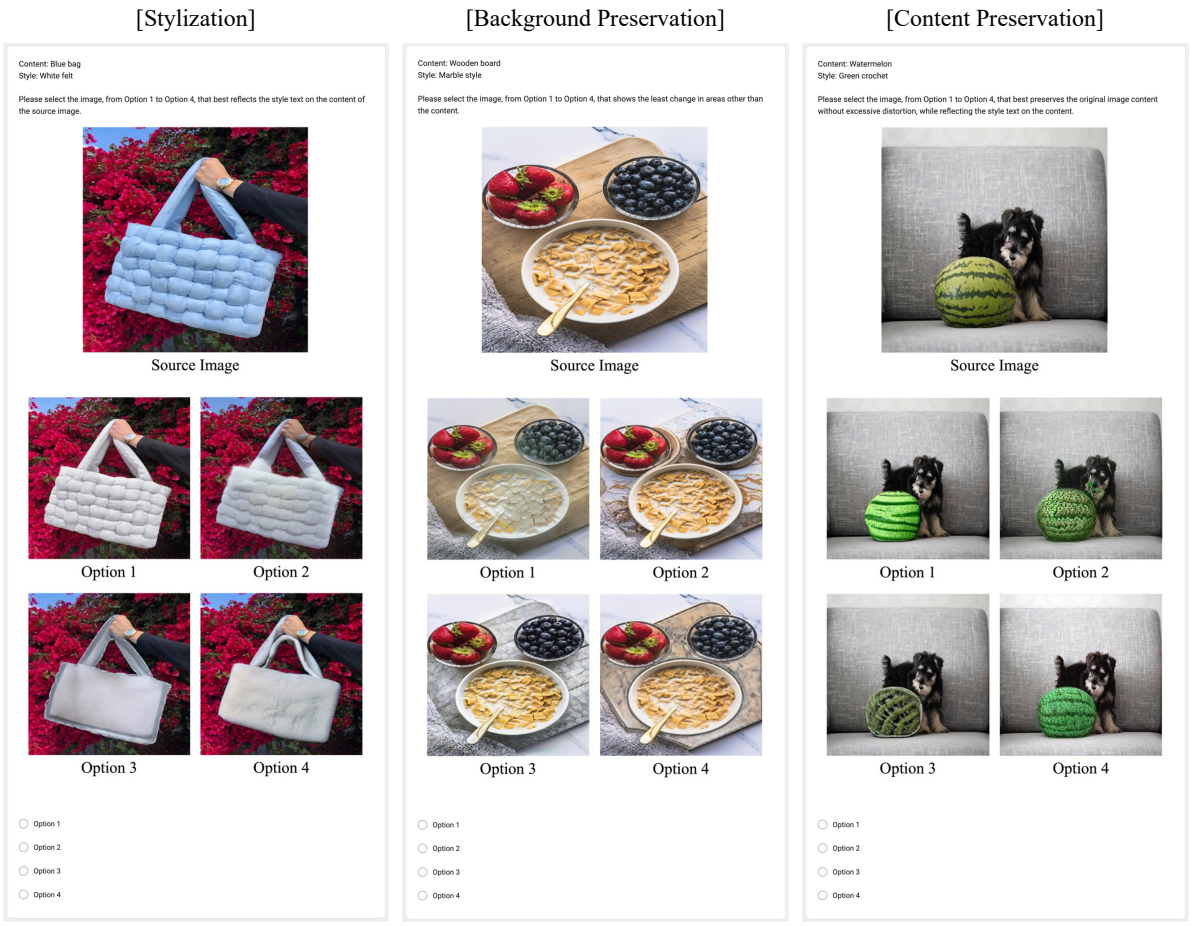


Figure 15. User study question examples. The order of options was shuffled for each question.

H. Additional object-centric style editing results using our Style-Editor

We show additional results of our Style-Editor method across diverse scenes, as seen in Fig. 16, Fig. 17, Fig. 18, and Fig. 19. These figures illustrate the effectiveness of our approach in various style editing scenarios, emphasizing the versatility of Style-Editor. TMPS plays a crucial role in initiating the style editing process. It specifically targets image areas that correspond with the input text, ensuring that the chosen style is applied seamlessly and appropriately to the relevant objects. This targeted approach results in a harmonious blend of the new style with the original image, particularly in areas corresponding to the source text. Furthermore, our method incorporates an innovative ABP loss. This component of Style-Editor is vital in maintaining the integrity of the background areas during the style editing process. It ensures that these areas remain unaffected by the changes applied to the object of interest. This loss function is key to achieving a balanced and natural-looking result where the style changes are confined to the targeted object, while the rest of the image retains its original appearance. The results in Fig. 16, Fig. 17, Fig. 18, and Fig. 19 collectively showcase the robust customizability and adaptability of the Style-Editor method. They demonstrate its capability to handle a diverse range of styles and scenarios, effectively adapting the chosen style to the specific objects in the image as dictated by the source text, all while preserving the overall aesthetic and integrity of the background.

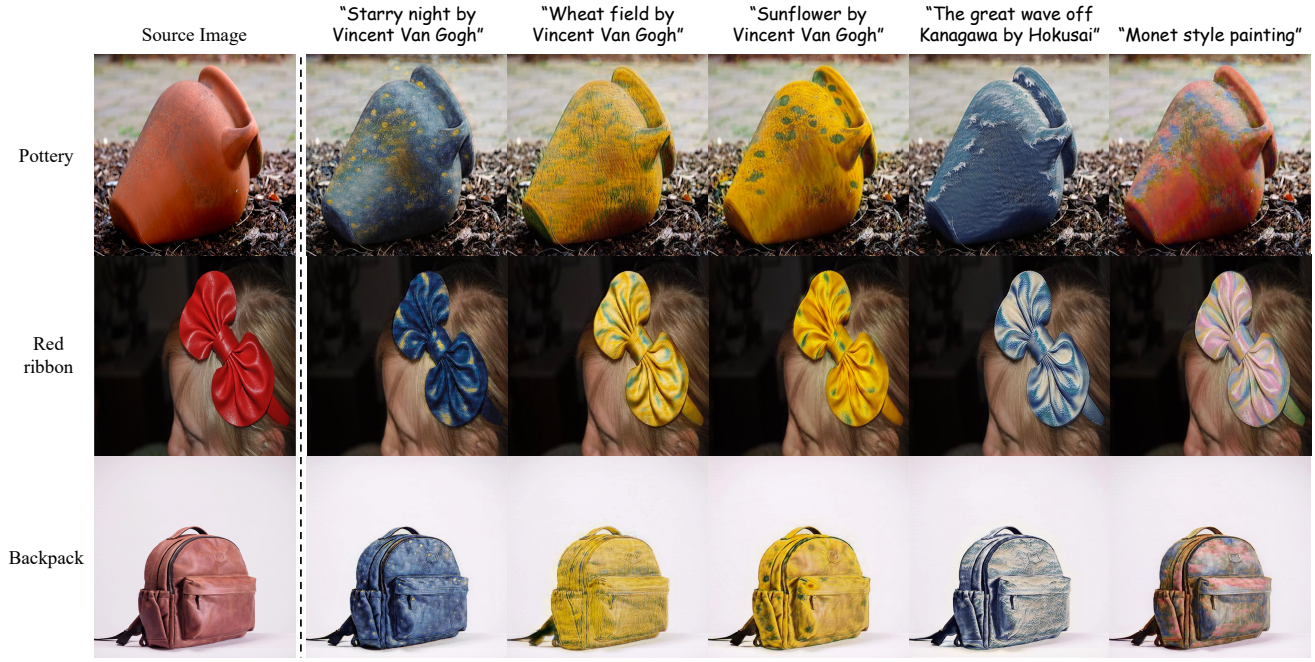


Figure 16. Stylization results demonstrating various **"artistic"** styles guided by text using our Style-Editor model.

Figure 17. Stylization results demonstrating various **"color"** styles guided by text using our Style-Editor model.

Figure 18. Stylization results demonstrating various **"texture"** styles guided by text using our Style-Editor model.

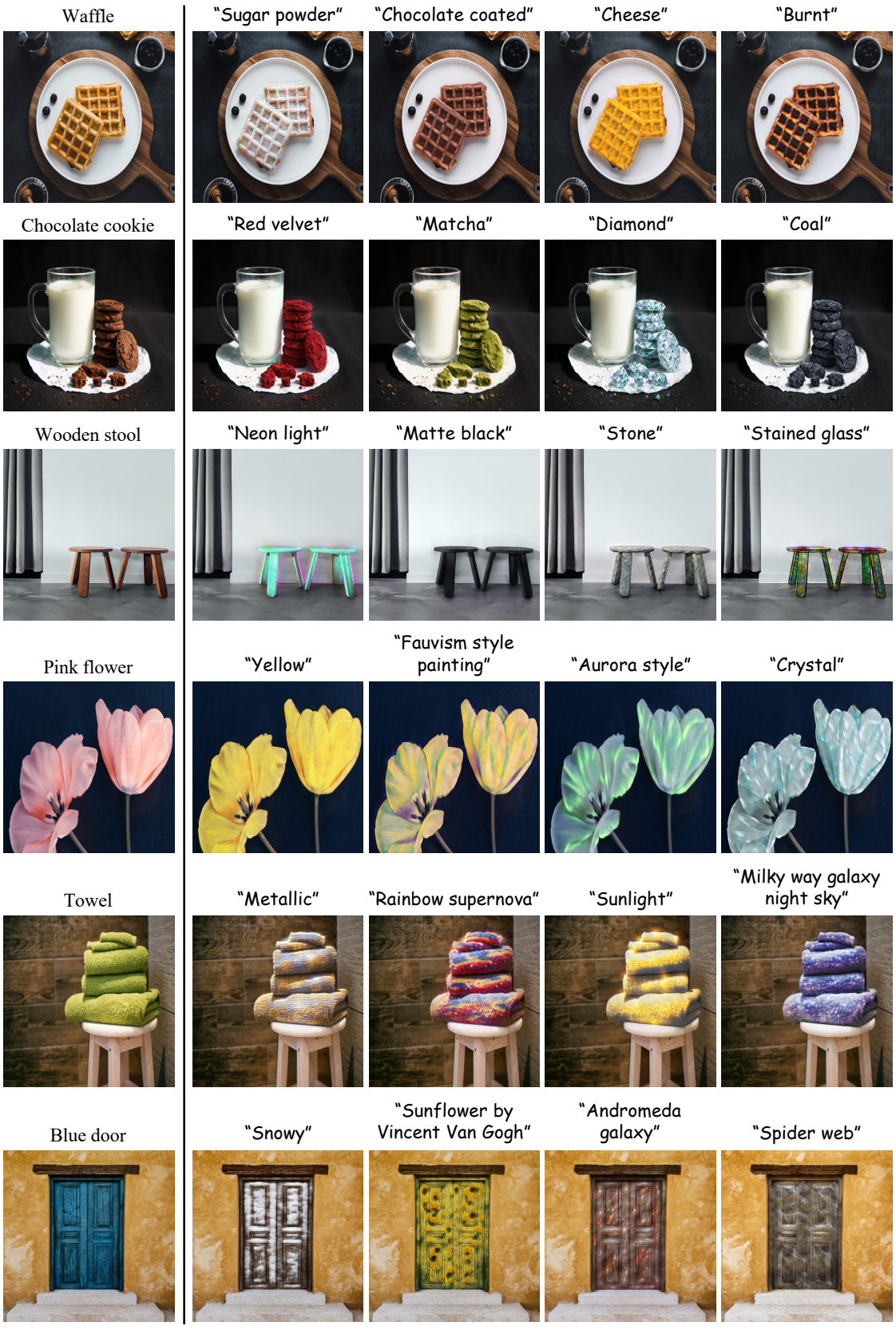


Figure 19. Our additional stylization results with various image-text scenarios.