2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

APRstyler: Adaptive Patch and Reversible Text-Guided Single Image Style Transfer

Zhichao Liu[1], Fang Yang[1][0000-0001-5948-3965] and Xiufang Miao[1]

[1] School of Cyber Security and Computer, Hebei University, Baoding 071002, China
yangfang@hbu.edu.cn

Abstract. Text-guided image style transfer enables stylization based on natural language prompts and has garnered increasing attention. CLIPstyler pioneered this paradigm by leveraging CLIP's image-text alignment but suffers from content distortion and uneven style application due to random patch sampling and weak structural constraints. To address these issues, we propose APRstyler, a CLIP-based framework integrating two key modules: an adaptive patch weighting strategy and a reverse generative network. The former evaluates the semantic relevance of each patch to the content image, suppressing noisy regions and enhancing local style fidelity. The latter reconstructs the content image from the stylized output, enforcing structural preservation through bidirectional supervision. Extensive experiments demonstrate that APRstyler yields stylizations with improved semantic consistency and content retention. For reproducibility, all experiments are conducted with fixed random seeds and averaged over multiple runs. Code will be released upon acceptance.

Keywords: Image processing, Text-guided Style Transfer, CLIP

1 Introduction

In recent years, text-guided artistic style transformation has emerged as a new type of style transformation technology. By using natural language to describe styles, it eliminates the need for reference style images.

CLIP (Contrastive Language-Image Pre-training) [1], as a cutting edge cross-modal pre-training model, has its core in the construction of a unified embedding space to jointly process image and text data. In recent years, the CLIP model has created a sensation in fields such as image generation [2] and image editing [3]. Against this backdrop, CLIPstyler [4], the first model to achieve text-guided image stylization, came into being. Its achievements have drawn widespread attention in the field of image stylization and demonstrated remarkable stylization capabilities.

CLIPstyler [4] transforms the original image content by integrating a lightweight convolutional neural network (CNN). At the same time, the method leverages the strong representation abilities of the CLIP [1] model to assess the similarity between the generated stylized image and the textual description within the feature space, enabling the production of images that reflect the intended textual style. This approach has opened

up a new era of text-driven style transformation. However, it also faces some challenges: CLIPstyler [4] adopts a patch-wise method to calculate the loss to improve the consistency of the style. However, data augmentation through random cropping may introduce low-quality image patches. These low-quality patches lack key semantics and cannot accurately reflect the target style. Frequent learning of such patches can mislead the model, resulting in uneven style transfer, loss of details, and poor results. Furthermore, when performing style transfer, CLIPstyler [4] often overfits the target text style and damages the content structure of the original image.

To overcome these challenges, we introduce APRstyler, a novel style transfer approach guided by text and built upon the CLIP [1] framework. To avoid model training being misled by poor quality patches, we propose an end-to-end adaptive patch adjustment scheme. Specifically, we calculate the feature distance between the randomly cropped patch and the original image and transform this distance into a weight to constrain the importance of the current image patch. The larger the feature distance of the current image block, the smaller the language information of the original image contained in the image block, so the weight of the current patch is smaller. This scheme can adaptively adjust the weights of different patches without the need for manual presetting. It makes the patches that are beneficial to style transfer have larger weights and those that are not conducive to style transfer have smaller weights.

In addition, to maximize the preservation of the content structure during the style transfer process and achieve the best balance between the details of the content and the degree of stylization, inspired by CycleGAN [5], we propose a reverse generative network. Through this, the image can be restored to the original image as much as possible after style transfer. This indicates that the image after style transfer retains the structural information of the original image to the greatest extent possible, enabling the model to self-regulate how to achieve a better balance between style transfer and maintaining structural integrity.

Experimental results indicate that our model generates artistic images which effectively retain fine-grained content details while achieving a higher degree of stylistic consistency with the intended style. Moreover, the resulting stylized outputs align more closely with human aesthetic preferences.

The contributions of this paper are as follows.

- We propose an end-to-end adaptive patch adjustment scheme, which enhances the model's ability to align with the target style and the stability of stylization.
- We propose a reverse generative network. Through the reverse loss, style transformation is achieved while better preserving the details of the content images.
- Extensive quantitative and qualitative experiments demonstrate that the model outperforms current advanced methods. It can stylize images in the deep feature space while better retaining the detailed information of the content images.

2 Related works

2.1 Traditional style transfer

Style transfer focuses on merging the visual content of a source image with the characteristics of a target style. Gatys et al. [6] pioneered the use of convolutional neural networks (CNNs) for style transfer by minimizing content and style losses through Gram matrix representations; however, this approach demands substantial computational resources.

To enhance efficiency, Johnson et al. [7] introduced a feedforward neural network for rapid style transfer, achieving real-time performance at the expense of sometimes compromising structural fidelity.

CycleGAN [5] introduces cycle consistency loss to enable unsupervised image translation, helping retain content during stylization. However, it often suffers from unstable training and overfitting to high-frequency details, leading to excessive stylization and content loss.

To support arbitrary style transformations, Huang and Belongie presented Adaptive Instance Normalization (AdaIN) [8], aligning the mean and variance of content and style features. While efficient and flexible, it lacks spatial awareness and often misses local textures.

2.2 Using text to guide style transfer

Unlike traditional methods [9-11], text-guided style transfer enables more flexible control using natural language. The introduction of CLIP [1] by OpenAI enabled multimodal contrastive learning, allowing style transfer guided solely by textual descriptions.

CLIPstyler [4] proposed by Kwon et al. in 2022 is a pioneering work in style transfer based on CLIP [1]. It utilizes the image-text matching feature provided by CLIP [1] to guide style transfer based on text prompts. This approach applies multimodal contrastive learning of vision and language to image stylization, making style transfer more flexible and no longer limited to fixed style images. CLIPstyler [4] performs outstandingly in generating diverse styles. However, due to excessive reliance on the contrast of global features, these methods are prone to damaging the original content during the transfer process. In particular, they lack fine control over the details of local structures, and the quality of the generated images needs improvement. Moreover, CLIPstyler [4] aligns the source and target text-image pairs in the CLIP space based on the CLIP loss of patches. A large portion of the image patches obtained by random cropping are of poor quality and not conducive to model training, which seriously affects the output quality of the model. Our adaptive patch selection and reversible generative network address these issues, enhancing image quality.

In recent years, Diffusion Models [12] have also made remarkable progress in the field of style transfer. Their applications in style transfer and image generation fields demonstrate their powerful generative capabilities and flexibility. For example, models such as stable diffusion [13] and DEADiff [14] have achieved efficient and controllable

image style transfer through technical means such as decoupled representation and semantic guidance. These models not only improve the flexibility and controllability of style transfer but also further promote the development of computer vision technology.

3 Method

3.1 Overall Architecture

This section presents the overall framework of the proposed APRstyler, which is composed of two synergistic components: an adaptive patch weighting strategy and a reverse generative network.

As shown in Fig. 1, inspired by CLIPstyler [4], we adopt its CNN encoder-decoder model G. The model is capable of extracting multi-level visual representations from the content image and performing stylization within the deep feature domain to achieve a more lifelike texture appearance. The process begins with computing semantic patch importance using CLIP-based similarity, followed by stylization using a generator network. To regularize the stylization and reduce structural distortion, a reverse generative network F reconstructs the original image from the stylized output. Both modules are trained jointly in an end-to-end fashion, with a comprehensive loss function combining style, content, reconstruction, and smoothness constraints.

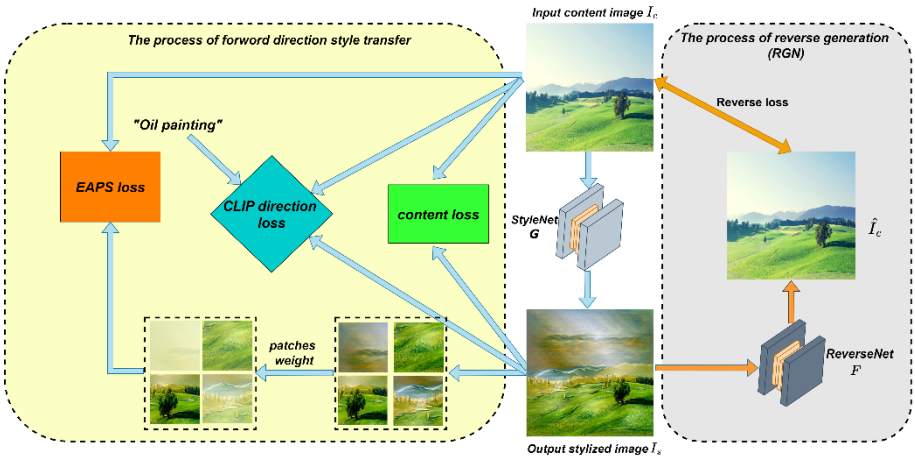

Fig. 1. Overall architecture of APRstyler. The left side represents the forward style transfer process, while the right side depicts the reverse generation process.

3.2 End-to-end Adaptive Patch Weighting Strategy(EAPS)

The end-to-end adaptive patch adjustment scheme is a key innovation of APRstyler. CLIPstyler [4] randomly divides the style image into many blocks and sends them to CLIP [1] for calculating the loss function to ensure that the generated image can reflect the style described in the target text in the local area. Random cropping of images is a

commonly used data augmentation technique [15], aiming to improve the model's robustness to different viewpoints and partial occlusions.

However, this augmentation method may result in the cropped patches missing important target objects and sampling low-quality patches. These patches may not effectively display the semantic information of the original image and are not suitable for the target text style (such as solid-color areas, blurred parts, or irrelevant areas at the edges of the image). During the training process, if the model frequently learns from these low-quality patches, it may mislead the optimization direction of the model. As a result, the model fails to correctly capture the required style features during training, leading to problems such as uneven style transfer, inability to align with the target style, and poor style transfer effects.

To address this issue, we introduce an End-to-end Adaptive Patch Selection (EAPS) mechanism that dynamically adjusts the contribution of each patch based on its semantic similarity to the original content image in CLIP [1] feature space.

Specifically, let I_c be the content image, and I_s be the stylized image at a given iteration. We randomly crop a set of N patches $\{P_i\}$ from I_s. For each patch P_i, we compute its CLIP [1] feature f_i and calculate its cosine distance to the CLIP [1] feature of the original image f_c. The similarity weight w_i for patch P_i is defined as:

$$w_i = \frac{1}{1+\cos(f_i, f_c)} \tag{1}$$

We normalize the weights using softmax:

$$w_i = \frac{\exp(w_i)}{\Sigma_j \exp(w_j)} \tag{2}$$

These weights are used to modulate the patch-wise CLIP loss:

$$L_{EAPS} = \Sigma_{i=1}^{N} \bar{w}_i \cdot L_{CLIP}(P_i, t) \tag{3}$$

Here, L_{CLIP} represents the loss obtained by comparing patches and text styles in CLIP [1].

This weight mechanism ensures that patches conducive to style transfer have a greater influence, while the impact of unfavorable patches is minimized. By reducing the influence of low-quality patches on the overall loss, this adaptive weight mechanism enables the model to focus on more informative patches during the learning process, thereby reducing the interference of irrelevant or incorrect samples on contrastive learning. Theoretically, this weighted loss scheme can reduce the gradient contribution of poor samples during the optimization process, stabilize the model training process, avoid overfitting to meaningless regions, and better align with the target style.

During the experiment, we found that in the first round of training, the generated images lack effective CLIP [1] guidance, resulting in poor style transfer and large feature distances. This can reduce patch weights to near zero, limiting gradient updates and causing training bottlenecks.

To address this, we adopt a progressive weight adjustment:

(1) Fixed weights initially: For the first 20 rounds, all patches are assigned a high fixed weight, allowing the model to learn global style features.

(2) Gradual weight introduction: As generated images approach the target style, feature-based weights are progressively applied, ensuring stable training while refining patch importance.

3.3 Reverse Generative Network (RGN)

CLIPstyler [4] enables text-guided style transfer without requiring reference style images, but traditional methods often overlook structural preservation, leading to content loss or over-stylization.

To mitigate this, Inspired by CycleGAN[5], we introduce a Reverse Generative Network (RGN) to reinforce structural retention. By incorporating reverse loss into training, we mitigate content degradation and excessive stylization. Unlike CycleGAN [5], which addresses unpaired training, APRstyler focuses on balancing style adaptation and content preservation, preventing overfitting to target text styles while maintaining structural integrity.

Let G denote the stylization generator and F denote the reverse generator. The forward path generates the stylized image $I_s = G(I_c, t)$, while the reverse path attempts to reconstruct $\hat{I}_c = F(I_s)$.

To ensure accurate reconstruction, we apply a content consistency loss using both pixel-wise L1 loss and perceptual VGG loss:

$$L_{rev-pixel} = \left\| I_c - \hat{I}_c \right\|_1 \tag{4}$$

$$L_{rev-vgg} = \left\| V(I_c) - V\left(\hat{I}_c\right) \right\|_2 \tag{5}$$

where $V(\cdot)$ denotes features extracted from a pretrained VGG network.

This reverse pathway provides a strong regularization signal, promoting better content preservation in the forward stylization process. It also enables bidirectional consistency, which is critical for applications requiring high structural fidelity.

3.4 Loss Funcitions

To optimize the stylization while maintaining content and structural integrity, we employ the following losses:

Adaptive Patch Style Loss (EAPS). This is the core patch-based CLIP style loss introduced in Section 3.2. Each patch's contribution is adaptively weighted based on its semantic similarity to the original content image.

CLIP Direction Loss (L_{dir}). StyleGAN-NADA [18] introduced a directional CLIP loss, which aligns the CLIP-space direction between the source and the output text-image pairs. This method has been proven effective, so we also adopt the directional CLIP loss to achieve better results.

$$\Delta T = \mathrm{E_T}(\mathrm{t_{sty}}) - \mathrm{E_T}(\mathrm{t_{src}}) \tag{6}$$

$$\Delta I = \mathrm{E_I}(f(I_c)) - \mathrm{E_I}(I_c) \tag{7}$$

$$L_{dir} = 1 - \frac{\Delta T \cdot \Delta I}{|\Delta T| \times |\Delta I|} \tag{8}$$

In the equation, E_T and E_I are the text and image encoders of CLIP; t_{sty} and t_{src} represent the semantic texts for the style target and input content. When using a specific image as the content image, t_{src} is simply set to "Photo."

Content Loss (Forward). To preserve high-level structure during stylization, we add a VGG-based perceptual loss between the I_s and I_c .

$$L_{content} = \|V(I_s) - V(I_c)\|_2 \tag{9}$$

This ensures that the forward stylization path does not overly. distort the original content

Reconstruction Losses (Reverse). Only the content loss during the forward style transfer process is insufficient to improve the problem of content distortion. As described in Section 3.3, the reverse generator reconstructs the original image. We include:

$L_{rev-pixel}$: L1 loss between $\hat{I}_c$ and I_c

$L_{rev-vgg}$: VGG perceptual loss between $\hat{I}_c$ and I_c

Total Variation Loss. To enhance spatial smoothness and reduce artifacts, we apply total variation (TV) loss on the stylized image.

$$L_{tv} = \sum_{n=1}^{\infty} \left(\left(I_s^{i+1,j} - I_s^{i,j} \right) \right)^2 + \left(\left(I_s^{i,j+1} - I_s^{i,j} \right) \right)^2 \tag{10}$$

Final Loss. The overall objective combines all components.

$$L_{total} = \lambda_s L_{EAPS} + \lambda_d L_{direction} + \lambda_c L_{content} + \lambda_r \left(L_{rev-pixel} + L_{rev-vgg} \right) + \lambda_{tv} L_{TV} \tag{11}$$

4 Experiments

4.1 Implementation details

Our model is trained on a single content image and supports arbitrary resolutions, but we use 512×512 images for consistency and resource efficiency.

For the forward network, the loss functions include content loss $L_{content}$, directional CLIP loss L_{dir}, Adaptive Patch Style Loss L_{EAPS} and total variation loss L_{tv}. The corresponding weights are set to 1×10^2, 5×10^2, 9×10^3, and 2×10^{-3}, respectively,

balancing content retention, stylization, and smoothness. For the reverse generative network, we apply $L_{rev-pixel}$ and $L_{rev-vgg}$, with weight 1×10^2. Both content losses are computed using VGG [16] conv4_2 and conv5_2 layers.

For image cropping, we use 128×128 patches, extracting 64 patches per image, as this configuration yielded optimal visual quality.

The StepLR scheduler is employed to reduce the learning rate by half every 100 steps, aiding the model's stability in the later training phases. We utilize the Adam optimizer with a specified learning rate. The training process spans 200 iterations, with the learning rate halved at iteration 100. On a single RTX 3080 Ti GPU, each text input takes about 50 seconds to process.

Our dataset is sourced from Night2Day, Summer2Winter, MS-COCO, WikiArt, and various online resources.

4.2 Qualitative assessment

Fig. 2 shows some representative style transfer results of our method. In the result images we present, the images generated by our method not only match the target text style in terms of global color, but also undergo meticulous transformations in local textures. Whether it is complex textures or abstract artistic effects, high-quality style transfer can be achieved while maintaining the structural information of the original image. This qualitative evaluation demonstrates the excellent performance of our model in handling delicate style transformations. For example, under the style text of "Oil painting", our method successfully simulates the rough brushstrokes and thick paint texture of an oil painting. Even under complex style descriptions (such as "Post Impressionist painting"), the model can still accurately capture the changes in color and light and shadow while preserving the detailed structure of the original image.

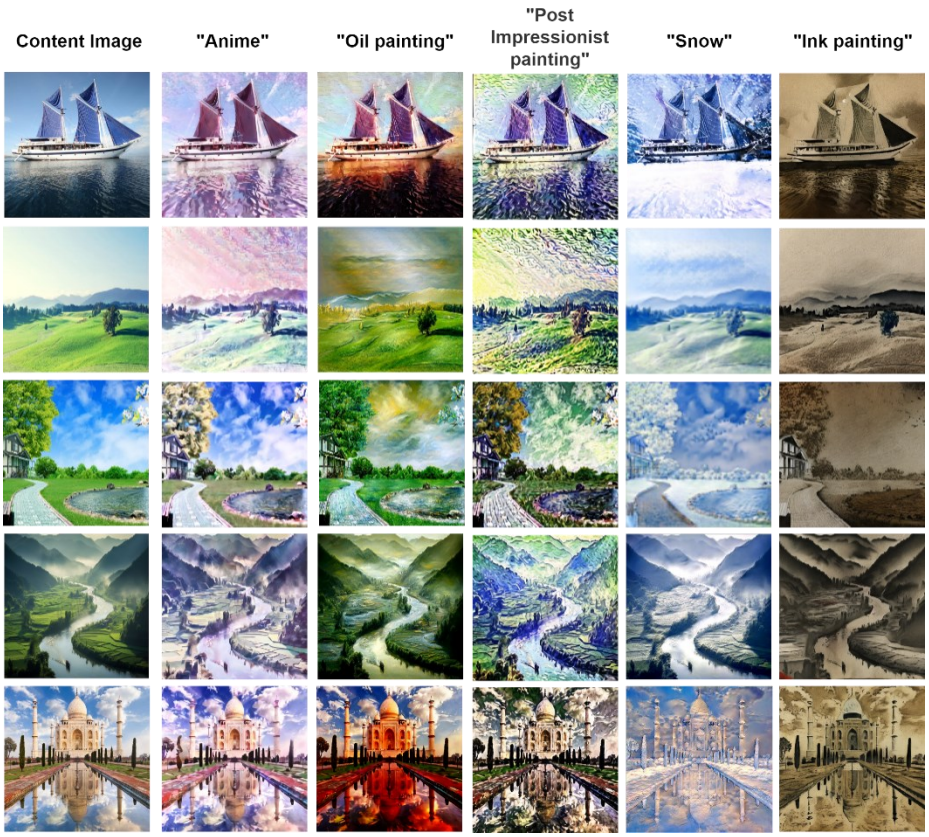

Fig. 2. The results of style transfer by APRstyler with various styles as text conditions

4.3 Comparison with Other Text-Guided Manipulation Models

To evaluate the effectiveness of our method, we qualitatively compared our method with CLIPstyler [4], DiffStyler [13], LDAST [19], and Text2LIVE [20] using five style prompts: "Snow," "Sketch with black pencil," "Mosaic," "Night," and "An ink wash painting"

Our model, benefiting from the end-to-end patch selection scheme, generates high-quality, detail-rich images that closely align with the target style. For instance, in the "Snow" style, our method accurately renders winter hues and snow textures, preserving fine details like textured grass and water reflections. In contrast, CLIPstyler [4] loses structural details, becoming blurry; DiffStyler [13] struggles with water surfaces and sky details; LDAST [19] maintains structure but produces grayscale images, failing to reflect the target style; and Text2LIVE [20] merely shifts the color tone without adding snow details, resulting in a less aesthetic output.

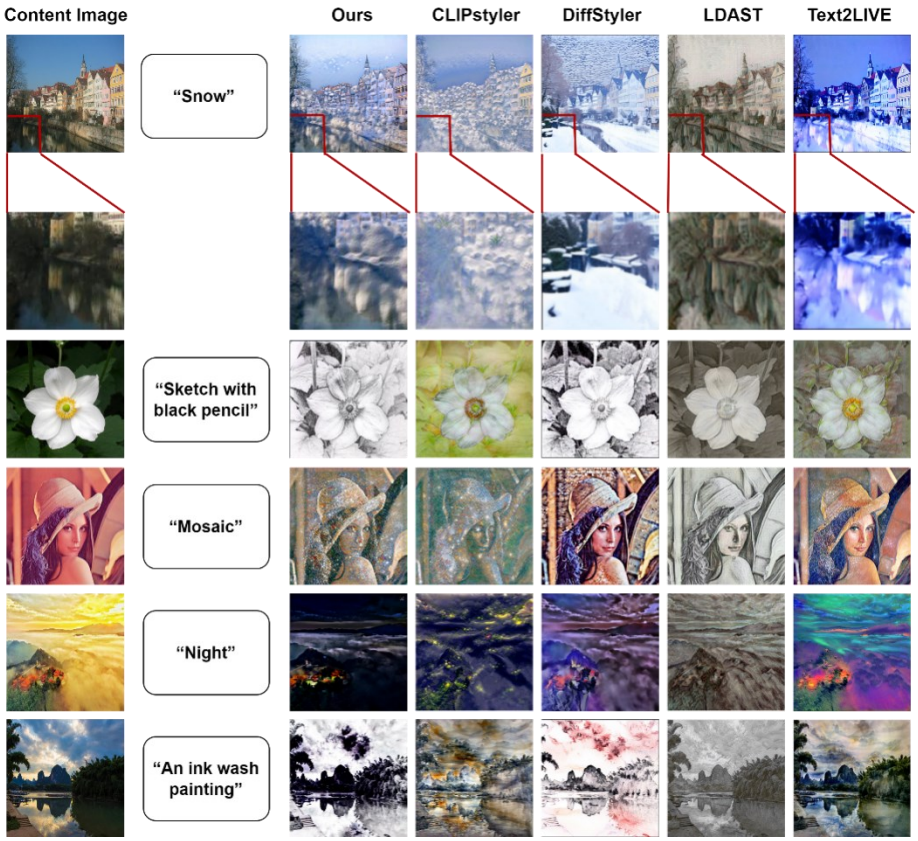

Fig. 3. Comparison with Other Text-Guided Manipulation Models

4.4 Quantitative evaluation

In order to better prove the progressiveness of our proposed model, we use NIMA [21], DISTS [22], LPIPS [23], and CLIPscore to evaluate the APRstyler framework. We randomly selected 10 content images and defined text prompts containing 20 style descriptions. Each method generated 200 stylized images.

NIMA [21]. NIMA [21] is a no-reference image quality assessment method that evaluates both the physical and aesthetic quality of images, considering factors like composition, color, and contrast to simulate human aesthetic judgment.

DISTS [22]. DISTS [22] is a full-reference metric that assesses structural and textural similarities, commonly used in image generation tasks like style transfer and super-resolution.

LPIPS [23]. LPIPS [23] is a widely used metric for evaluating the perceptual similarity between generated and real images. It better reflects human visual perception compared to traditional metrics, where a lower LPIPS value indicates higher similarity.

CLIPscore. CLIPScore quantitatively evaluates the alignment between images and text by computing the cosine similarity between text embeddings and image embeddings from the CLIP encoder, measuring how well the stylized image matches the text prompt.

Table 1 shows the results of the quantitative evaluation. In terms of the DISTS [22] metric, LDAST [19] ranks second only to the APRstyler scheme. However, it has the lowest NIMA [21] index, indicating that although LDAST [19] can restore the structural information of the content image relatively well, its visual perception effect in style transfer is poor. Our approach outperforms other advanced models in terms of NIMA [21] and CLIPscore metrics, demonstrating that the APRstyler approach can better align styles and provide high-quality output. Although APRstyler achieved suboptimal results in LPIPS [23], overall, our model is still the best.

All results are averaged over 5 independent runs with fixed random seeds. The reported standard deviation reflects the robustness of each method under different style prompts and content images. APRstyler consistently outperforms other methods across most metrics, especially in CLIPScore and NIMA, indicating better alignment with the target style and visual aesthetics.

Table 1. Quantitative evaluation results of different methods. The best results are presented in bold, and the second-best results are underlined.

	DISTS ↑	NIMA ↑	LPIPS ↓	CLIPscore ↑
CLIPstyler	0.370 ± 0.013	<u>6.35 ± 0.21</u>	0.590 ± 0.018	0.280 ± 0.015
DiffStyler	0.332 ± 0.017	5.28 ± 0.19	**0.512 ± 0.022**	<u>0.291 ± 0.014</u>
LDAST	<u>0.410 ± 0.012</u>	4.56 ± 0.18	0.760 ± 0.020	0.214 ± 0.013
Text2LIVE	0.353 ± 0.015	5.96 ± 0.16	0.682 ± 0.019	0.251 ± 0.012
Ours	**0.432 ± 0.010**	**6.72 ± 0.14**	<u>0.519 ± 0.016</u>	**0.321 ± 0.011**

4.5 Ablation Experiment.

To assess the effectiveness of our proposed method, we conducted ablation experiments to analyze how each component affects the model's output, as illustrated in Fig. 4. As can be seen from Fig. 4(b), when we remove the end-to-end patch selection scheme (EAPS), although the structure of the original image can be retained (for example, the mountain at the back of the second row is still clearly visible), from the perspective of visual aesthetics, the effect is far inferior to that of our method (for example, an unexpected green tone appears when using "Sketch with black pencil" as the text guidance in the first row).

After removing the reverse generative network scheme (RGN), as can be observed in Fig. 4(c), it is very difficult to preserve the structural details of the original image well when dealing with images with complex structural textures (for example, the details of the leaves in the upper left corner of the first row are completely lost, and the outline of the distant mountain and the shape of the clouds in the sky in the second row have become very blurred).

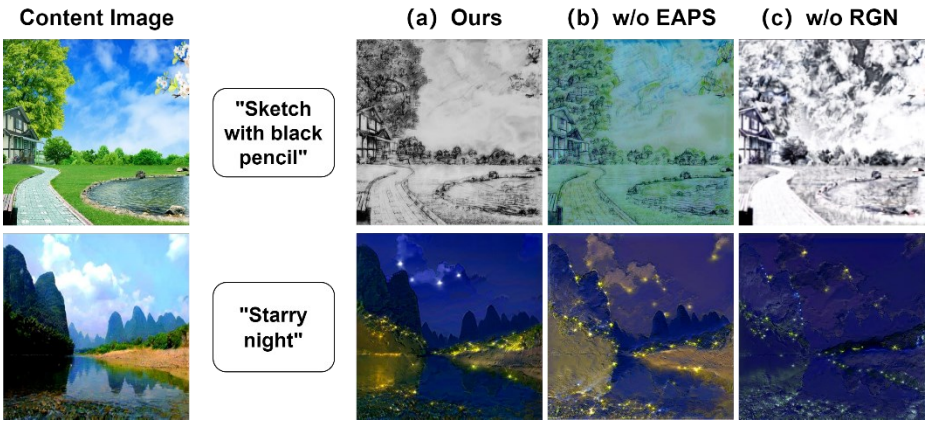

Fig. 4. Visualization results of ablation study

Table 2 presents the quantitative evaluation results of the ablation experiment, further validating the effectiveness of our proposed method. The ablation study demonstrates that both the reverse generative network (RGN) and the adaptive patch selection (EAPS) contribute to performance improvements. Notably, RGN improves content preservation (DISTS, LPIPS), while EAPS enhances stylization alignment (NIMA, CLIPScore). Their combination achieves the best overall performance.

Table 2. Quantitative evaluation results of different methods. The best results are presented in bold.

	DISTS ↑	NIMA ↑	LPIPS ↓	CLIPscore ↑
Baseline	0.370 ± 0.013	6.35 ± 0.21	0.590 ± 0.018	0.280 ± 0.015
RGN	0.420 ± 0.011	5.63 ± 0.18	0.648 ± 0.019	0.290 ± 0.013
EAPS	0.360 ± 0.014	6.55 ± 0.17	0.550 ± 0.021	0.270 ± 0.011
RGN+EAPS(ours)	**0.432 ± 0.010**	**6.72 ± 0.14**	**0.519 ± 0.016**	**0.321 ± 0.011**

From the results of the ablation experiments, it can be seen that different components of APRstyler each play a role, and the reverse generative network scheme and the end-to-end patch selection scheme have different advantages. The reverse generative network scheme plays a significant role in retaining the content structure information, while the end-to-end patch selection scheme focuses more on enhancing the aesthetic feeling of the image, making the style alignment ability more excellent.

When the two are used in combination, it can ensure both the retention of the content structure and the improvement of the aesthetic quality. Ultimately, the model performs best in all indicators. This further validates the effectiveness of the method we proposed, indicating that the combination of these two schemes can complement each other in the image style transfer task and significantly improve the quality of the generated images.

5 Conclusion

In this paper, we introduced APRstyler, a novel CLIP-based text-guided style transfer framework that addresses two major limitations of existing methods: the impact of low-quality patch sampling and the loss of content structure during stylization. By incorporating an adaptive patch weighting mechanism and a reverse generative network, our approach improves both the visual quality and structural consistency of the stylized output.

Experimental results demonstrate that APRstyler produces aesthetically pleasing and semantically aligned stylizations while effectively mitigating common issues such as over-stylization, content distortion, and patch-level inconsistency.

In future work, we aim to explore more advanced patch weighting strategies and integrate diffusion-based modules to further enhance stylization flexibility and detail fidelity.

We believe that APRstyler offers a promising direction for controllable, text-driven visual stylization, with potential applications in digital art, personalized image editing, and multimodal content generation.

References

1. Radford, A., Kim, J.W., Hallacy, C., et al.: Learning Transferable Visual Models from Natural Language Supervision. In: Proc. of the 38th International Conference on Machine Learning. PMLR, pp. 8748–8763 (2021)
2. Crowson, K., Biderman, S., Kornis, D., et al.: VQGAN-CLIP: Open domain image generation and editing with natural language guidance. In: European Conference on Computer Vision, pp. 88–105. Springer, Cham (2022)
3. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18208–18218 (2022)
4. Kwon, G., Ye, J.C.: CLIPstyler: Image style transfer with a single text condition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18062–18071 (2022)
5. Zhu, J.Y., Park, T., Isola, P., et al.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)
6. Crowson, K., Biderman, S., Kornis, D., et al.: VQGAN-CLIP: Open domain image generation and editing with natural language guidance. In: European Conference on Computer Vision, LNCS, vol. XXXX, pp. 88–105. Springer, Cham (2022)
7. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Computer Vision – ECCV 2016, LNCS, vol. XXXX, pp. 694–711. Springer, Cham (2016)
8. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1501–1510 (2017)

9. Li, X., Liu, S., Kautz, J., et al.: Learning linear transformations for fast image and video style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3809–3817 (2019)
10. Yoo, J., Uh, Y., Chun, S., et al.: Photorealistic style transfer via wavelet transforms. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9036–9045 (2019)
11. Xu, W., Long, C., Wang, R., et al.: DRB-GAN: A dynamic resblock generative adversarial network for artistic style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6383–6392 (2021)
12. Yang, L., Zhang, Z., Song, Y., et al.: Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys* 56(4), 1–39 (2023)
13. Huang, N., Zhang, Y., Tang, F., et al.: DiffStyler: Controllable dual diffusion for text-driven image stylization. *IEEE Transactions on Neural Networks and Learning Systems* (2024)
14. Qi, T., Fang, S., Wu, Y., et al.: DEADiff: An Efficient Stylization Diffusion Model with Disentangled Representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8693–8702 (2024)
15. Shorten, C., Khoshgoftaar, T.M., Furht, B.: Text data augmentation for deep learning. *Journal of Big Data* 8(1), 101 (2021)
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
17. Wang, L., Zhang, Y., Feng, J.: On the Euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(8), 1334–1339 (2005)
18. Gal, R., Patashnik, O., Maron, H., et al.: StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)* 41(4), 1–13 (2022)
19. Fu, T.J., Wang, X.E., Wang, W.Y.: Language-driven artistic style transfer. In: European Conference on Computer Vision, LNCS, vol. XXXX, pp. 717–734. Springer, Cham (2022)
20. Bar-Tal, O., Ofri-Amar, D., Fridman, R., et al.: Text2Live: Text-driven layered image and video editing. In: European Conference on Computer Vision, LNCS, vol. XXXX, pp. 707–723. Springer, Cham (2022)
21. Talebi, H., Milanfar, P.: NIMA: Neural image assessment. *IEEE Transactions on Image Processing* 27(8), 3998–4011 (2018)
22. Ding, K., Ma, K., Wang, S., et al.: Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44(5), 2567–2581 (2020)
23. Zhang, R., Isola, P., Efros, A.A., et al.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)