Look, Compare and Draw: Differential Query Transformer for Automatic Oil Painting

Lingyu Liu, Yaxiong Wang†, Li Zhu, Lizi Liao, Zhedong Zheng†

***Abstract*—This work introduces a new approach to automatic oil painting that emphasizes the creation of dynamic and expressive brushstrokes. A pivotal challenge lies in mitigating the duplicate and common-place strokes, which often lead to less aesthetic outcomes. Inspired by the human painting process, *i.e.*, observing, comparing, and drawing, we incorporate differential image analysis into a neural oil painting model, allowing the model to effectively concentrate on the incremental impact of successive brushstrokes. To operationalize this concept, we propose the Differential Query Transformer (DQ-Transformer), a new architecture that leverages differentially derived image representations enriched with positional encoding to guide the stroke prediction process. This integration enables the model to maintain heightened sensitivity to local details, resulting in more refined and nuanced stroke generation. Furthermore, we incorporate adversarial training into our framework, enhancing the accuracy of stroke prediction and thereby improving the overall realism and fidelity of the synthesized paintings. Extensive qualitative evaluations, complemented by a controlled user study, validate that our DQ-Transformer surpasses existing methods in both visual realism and artistic authenticity, typically achieving these results with fewer strokes. The stroke-by-stroke painting animations are available on our project website [1].**

***Index Terms*—Automatic Oil Painting, Stroke-based Rendering, Style Transfer, Sequence Prediction**

I. Introduction

PAINTING is a common form of human artistic expression, but it requires a certain level of technical skill. Computer-aided art [1]–[8] enables people without professional drawing skills to create their own artistic works. Neural oil painting [9]–[13] has emerged as a promising paradigm for artistic image transformation by simulating the brushstrokes of oil paintings through hierarchical stroke rendering. It aims to guide machines in progressively generating images by emulating authentic oil painting brushstrokes, from coarse to fine, on a digital canvas, thereby imparting to the images the characteristic texture of oil paintings.

Traditional stroke-based rendering methods typically rely on step-wise greedy search and heuristic optimization, which often lead to low efficiency [14]–[17]. As noted by Hu *et al.* [18], deep learning-based methods have gained traction, employing a variety of strategies such as reinforcement learning [12], [18], [19], neural networks [20], and optimization-based approaches [21], [22]. While these methods have validated promising painting results, challenges in achieving higher efficiency and effectiveness in practical applications persist. For example, Hu *et al.* [18] develop a reinforcement learning-based agent trained on real images to dynamically determine the painting sequence, but it struggles with generalization, and becomes unstable when faced with unseen images. Similarly, Zou *et al.* [21] introduce a stroke optimization method that achieves high-quality results but requires extremely long inference times. On the other hand, Liu *et al.* [20] directly construct a neural network to efficiently predict a set of strokes. However, this method often produces coarse strokes and particularly fails to capture fine details at the canvas boundaries.

Despite varying learning strategies within specific models, the prevailing works all adhere to the iterative learning paradigm, that is, predicting the subsequent brushstroke based on the current one. In line with this paradigm, existing methodologies employ a rather direct approach by generating the forthcoming brushstroke directly using the existing stroke as input. We contend that this predictive approach suffers from the absence of an intermediate guidance from the current stroke to the next, which becomes particularly challenging when there is a significant divergence between the paintings in the early steps of prediction. Conversely, in the human painting process, artists frequently observe and compare differences between their current work and the target painting before deciding on the subsequent brushwork. Motivated by this procedure, we propose the incorporation of image discrepancy as a form of intermediate guidance to address the neural oil painting problem, aiming to bridge the gap between the current iteration and the ultimate artistic vision, thereby enhancing the fidelity and effectiveness of the neural painting process.

In light of the aforementioned considerations, we adopt PaintTransformer [20] as our baseline and propose a new differential image-guided painter framework: the Differential Query Transformer (DQ-Transformer). The DQ-Transformer learns differential image features between the current canvas and the target image, focusing on the discrepancies between the images, thereby enabling more accurate stroke predictions. In particular, we employ local encoders comprised of convolutional neural networks to learn three position-aware image features separately: the current canvas, the target image, and the differential image between these two. The differential image features are then transformed into query tokens, which are used

Lingyu Liu and Li Zhu are with the School of Software, Xi'an Jiaotong University, Xi'an, 710049, China. (e-mail: liulingyu@stu.xjtu.edu.cn; zhuli@mail.xjtu.edu.cn).

Yaxiong Wang is with the School of Computer and Information Science, Hefei University of Technology, Hefei, 230000, China. (e-mail: wangyx15@stu.xjtu.edu.cn).

Lizi Liao is with the Singapore Management University, 188065, Singapore (e-mail: lzliao@smu.edu.sg).

Zhedong Zheng is with Faculty of Science and Technology, and Institute of Collaborative Innovation, University of Macau, Macau, 999078, China. (e-mail: zhedongzheng@um.edu.mo).

† Corresponding author.

[1]https://differential-query-painter.github.io/DQ-painter/

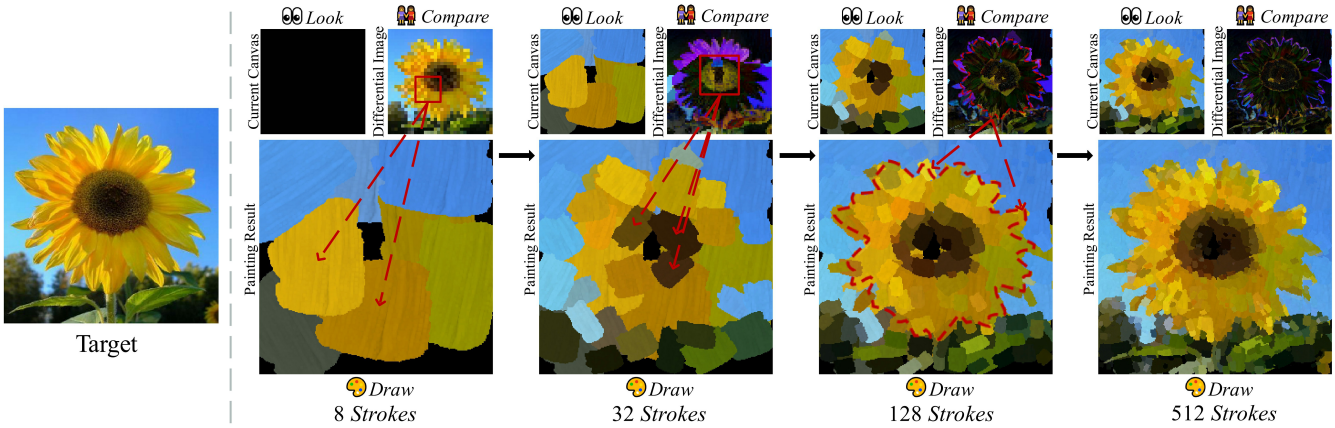


Fig. 1: Differential image-guided inference process. We present four intermediate stages of oil painting according to a real target image (left). Each stage is illustrated with a diagram, where the top-left corner shows the current canvas, the top-right corner displays the corresponding differential image for that stage, and the bottom part presents the painting result inferred by our model. We observe that since we explicitly compare the content in the differential images during training, our model tends to add strokes in areas where discrepancies are more pronounced, thereby progressively reducing the discrepancy content within the differential images.

as dynamic queries to the DQ-Transformer to decode the stroke parameters. The final painting result is obtained by rendering these decoded strokes onto the canvas. We first minimize the L_1 distance between the target image and the rendered image, as well as the L_1 distance between the predicted strokes and the ground-truth strokes. Furthermore, we train the DQ-Transformer with a WGAN-based discriminator [18], [23]. The discriminator is utilized during training to enhance the precision of predicted strokes, by treating the rendered images as fake samples and striving to penalize the generation of erroneous strokes. Compared with the baseline framework [20], our DQ-Transformer retains its efficient inference advantage while innovatively introducing differential-guided dynamic queries. By explicitly focusing on image discrepancies through differential features, our method effectively eliminates duplicated stroke predictions and simultaneously captures subtle texture details.

The "look, compare and draw" painting process of our model is illustrated in Figure 1, where we present four intermediate stages of completing a real image with several strokes. It can be observed that our model evaluates the content of the differential image and introduces strokes precisely in areas exhibiting more significant disparities. This dynamic querying mechanism allows our model to prioritize areas that require refinement, progressively reducing visual differences and guiding the painting toward a highly detailed and structurally accurate final output. Unlike existing stroke-based oil painting methods that often rely on static representations or fixed attention patterns, our approach is fundamentally **observation-first**: it continuously re-evaluates the evolving canvas in relation to the target, making each stroke placement both context-aware and purpose-driven. This design is conceptually simple and remarkably effective. To prove that the oil paintings produced by our method are of high quality, we compare them with other state-of-the-art stroke-based oil painting methods. Qualitative comparisons indicate that our method can generate images with more authentic oil painting textures while maintaining the fidelity of the original images. We have conducted a Mean Opinion Score (MOS) test and invited volunteers to evaluate the quality of oil paintings created by the above methods. The paintings of our method attained the highest preference ratings from the users. The primary contributions of our work are:

- **Differential Image Analysis Integration:** We introduce a new painting pipeline that embeds differential image analysis within the neural oil painter framework. By focusing on the incremental changes wrought by successive brushstrokes, this simple and effective enhancement sharpens the attention to localized details, yielding a more intuitive and nuanced rendering process.
- **Differential Query Transformer Architecture:** Inspired by the spirit of human artists, *i.e.*, observing, comparing and drawing, we further introduce a Differential Query Transformer (DQ-Transformer) that explicitly leverages position-aware differential features as dynamic queries to guide stroke prediction.
- **Superior Performance:** Both quantitative and qualitative experiments on three public datasets, *i.e.*, Landscapes, FFHQ, and Wiki Art, affirm that the proposed method achieves better pixel-level and perception-level reconstruction, as well as higher user preference across various painting themes. Furthermore, the proposed method is stroke-efficient, *i.e.*, it achieves competitive painting quality with fewer strokes.

II. Related Work

Stroke-based painting and pixel-wise painting represent two distinct paradigms in digital art creation. We first review related work on pixel-wise generation [24]–[28]. To enhance robustness, DreamAnime [29] disentangles anime style and identity into separate latent codes for independent text control. 3DArtmator [30] and MVCGAN [31] incorporate 3D awareness through an interpretable stylization subspace and multi-view

consistency, respectively. Huang *et al.* [32] propose a cross-art attention mechanism for style transfer, while DG-Net [33] disentangles style and content representations. For improved generation quality, Zhang *et al.* [34] introduce DPTN-TA, which uses dual-task correlation and a texture affinity loss for pose-guided person image synthesis and view synthesis. TextIR [35] leverages CLIP to align textual and visual features, achieving effective performance across multiple image restoration tasks. Despite their success, these pixel-based methods manipulate images holistically and do not reflect the stepwise, stroke-driven logic of human painting.

Unlike pixel-based generative models, automatic oil painting deploys brushstrokes as the fundamental unit of creation. Traditional stroke-based methods [14], [15], [36], [37] rely on handcrafted rules to generate strokes. For example, Hertzmann *et al.* [38] apply multi-sized curved brush strokes to transform photographs into painterly renderings. Im2Oil [17] combines adaptive sampling based on probability density maps to produce high-quality results. However, these rule-based approaches suffer from low search efficiency in large stroke spaces, leading to long runtimes. Recently, deep learning based methods have gained increasing popularity, and various learning strategies have been explored to address stroke-based rendering. As noted by Hu *et al.* [18], existing automatic oil painting methods based on deep neural networks can primarily be classified into three categories as follows:

Optimization-based methods. Optimization-based methods aim to determine the optimal stroke order to improve drawing efficiency. Fan *et al.* [39] deconstruct brushstrokes in traditional Chinese ink paintings and introduce a natural evolution strategy to infer their best application sequence. To support stroke decomposition, Ashcroft *et al.* [40] propose a generative model for complex vector drawings and demonstrate its effectiveness on intricate anime line art. Stylized Neural Painting [21] treats stroke prediction as a parametric search process, mimicking a vector graphics renderer to adapt painting techniques to real images. Parameterized Brushstrokes [22] searches over parameterized stroke styles to complete a painting. Liu *et al.* [41] learn stroke style distributions and use semantic-aware placement to enhance artistic quality. Hertzmann *et al.* [4] leverage segmentation and dynamic attention maps to efficiently adjust stroke parameters. These methods can be optimized jointly with neural style transfer but suffer from long optimization times for each image.

Neural network-based methods. Neural network based methods directly use basic architectures to predict painting strokes. Early work employs Recurrent Neural Networks (RNNs) [42] to decompose images into sequences, but relies on detailed manual annotations, limiting scalability. To overcome this, Frans *et al.* [43] apply self-supervised deep networks to learn the mapping from completed paintings to their brushstrokes. Paint Transformer [20] reformulates stroke prediction as a feed forward set generation task using a Transformer, enabling parallel stroke parameter prediction and efficient self supervised training without manual labels. Based on this work, Dong *et al.* [44] further study the efficient test-time adaptation. Similarly, Song *et al.* [45] propose HairstyleNet, which combines parametric controllable strokes with neural rendering for high quality interactive hairstyle editing. Although these methods are annotation free and computationally efficient, their predicted strokes are often coarse and lack fine details near canvas boundaries.

Reinforcement learning-based methods. Reinforcement learning-based methods [9], [12], [46]–[48] aim to learn the textures and styles of real-world images to improve the painting quality. As a seminal effort, Huang *et al.* [19] employ a more complicated reinforcement learning model to paint complex real-world images with a watercolor brush. Moreover, Compositional Neural Painter [18] incorporates object detection learning into the reinforcement learning model, dynamically segmenting and predicting stroke regions. Training a stable reinforcement learning agent is challenging due to the dynamic interactions among its components, as this process typically leads to instability.

Although the aforementioned methods achieve satisfactory results in rendering paintings, they suffer from issues such as boundary inconsistencies and struggle with more intricate images. We address these limitations by introducing a DQ-Transformer architecture that leverages differentially derived image representations, augmented with positional information, to guide informed stroke prediction. Our model is both sensitive to position and capable of producing higher-quality renderings.

III. Methodology

Overview. Neural painting simplifies the painting task into predicting a sequence of brush strokes. In this section, we offer a comprehensive description of the training process for our painter framework, along with the inference process utilized for generating artworks. A brief overview of our painter framework is illustrated in Figure 2. We utilize a self-supervised pipeline, originally introduced by [20], in which the current canvas and target images are constructed using randomly synthesized strokes, thereby eliminating the need for real images during the training process. Our objective is to guide the model to concentrate on the regions of discrepancy between the canvas and the target image, thereby predicting more accurate strokes to minimize these differences, without the necessity of considering the semantic information of the images. Furthermore, we construct a differential image between the target image and the current canvas, which subsequently serves as the query tokens for our DQ-Transformer. The differential operation approximates how the human visual system processes image information, emphasizing the incremental effects resulting from consecutive brushstrokes.

A. Preliminaries

Stroke Renderer. We adhere to the settings commonly employed in stroke-based painting methods [17], [18], [20], [21] for stroke rendering, adjusting the properties of real static brushstrokes,*i.e.*, oil brushstrokes, to generate various stroke variants based on the specified parameters. The stroke parameters are defined as $s = \{x, y, h, w, \theta, r, g, b\}$, where (x, y) denotes the coordinates of the center point, h represents the height, w represents the width, θ denotes the rotation angle, and (r, g, b) indicates the RGB color values of the stroke. At

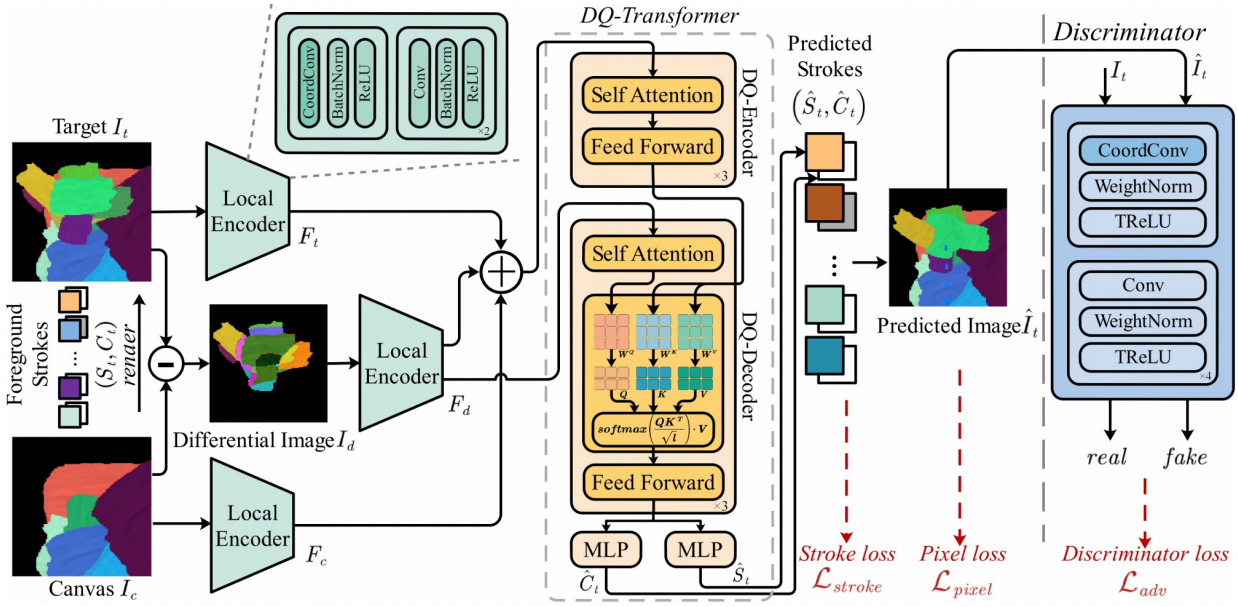


Fig. 2: A brief overview of our painter framework. Given the canvas image I_c and the target image I_t generated by the renderer, we first obtain their differential image I_d by simply subtracting one input from the other. Three local encoders comprised of convolutional neural networks are employed to extract image features F_c, F_t, and F_d with positional information. DQ-Transformer has two components, *i.e.*, the DQ-encoder and the DQ-decoder. These visual features F_c, F_t and F_d, are concatenated and then fed to the DQ-encoder to obtain the fused feature F_{kv}. Next, we transform the differential image features F_d into query tokens to query the key and value pairs generated by the fused feature F_{kv}. Finally, the DQ-Transformer outputs a set of predicted strokes $\hat{S}_t$, each accompanied by its respective confidence $\hat{C}_t$. The predicted image $\hat{I}_t$ is generated by rendering these strokes onto the canvas. The discriminator operates by treating the target images I_t as real samples and the predicted images $\hat{I}_t$ as fake samples.

each step n, the stroke renderer is employed to render the stroke parameters into a stroke image R_n and a binary mask M_n, where M_n is a single-channel alpha map of R_n. These stroke images are then sequentially added to the current canvas, potentially covering any previous strokes if they exist. The iterative rendering process can be formulated as:

$$I_n = R_n \odot c_n M_n + I_{n-1} \odot (1 - c_n M_n), \tag{1}$$

where c_n is the confidence of the stroke, indicating whether the stroke is valid. $\odot$ is the element-wise multiplication, while I_{n-1} is the previous painting result. The entire rendering process is based on differentiable linear transformations and does not contain any trainable parameters.

Canvas Construction. In each training iteration, we first randomly sample two strokes sets: a background strokes set S_b to generate the canvas I_c, and a foreground strokes set S_t to create the target image I_t based on I_c. Background strokes are rendered onto an empty canvas to establish the current canvas I_c. Subsequently, the foreground strokes are superimposed onto the current canvas to produce the target image I_t. Notably, the background strokes are coarser in granularity than the foreground strokes. This construction methodology mirrors the human artistic process, which evolves from broad outlines to detailed refinements.

B. Painter Framework

The painter framework aims to reconstruct the target image I_t using a sequence of predicted strokes. Given the current canvas $I_c \in \mathbb{R}^{3\times P\times P}$ and the target image $I_t \in \mathbb{R}^{3\times P\times P}$, where P is the pre-defined patch size that acts as the basic unit for subsequent painting. Then the differential image is obtained by performing a pixel-wise subtraction: $I_d = I_t - I_c$. Our painter framework takes I_c, I_t, and I_d as input and predicts a stroke set $\hat{S}_t$. The predicted image is generated by rendering these strokes onto the canvas.

Local Encoder. As shown in Figure 2, the painter framework first employs separate local encoders, comprised of convolutional neural networks, to individually extract their feature maps, denoted as $F_c, F_t, F_d \in \mathbb{R}^{3\times \frac{P}{4}\times \frac{P}{4}}$. It is worth noting that traditional convolutional layers lack explicit positional encoding, and stacking them directly can lead to the loss of coordinate information. To address this issue, we substitute traditional convolutional layers with Coordinate Convolution (CoordConv) [49], implementing it in the first layer of the convolutional network. CoordConv introduces additional channels to the input feature map, representing the X-Y coordinates of each feature pixel, thereby enabling the convolutional learning process to have a degree of awareness about the spatial positions. Then, F_c, F_t, and F_d, endowed with positional encoding, are concatenated and flattened as the input of DQ-Transformer.

DQ-Transformer. DQ-Transformer consists of two main parts: a DQ-Encoder and a DQ-Decoder. The DQ-Encoder block consists of a self-attention layer and a feed-forward layer, and it learns a mapping from the concatenated features

$\{F_c,\ F_t,\ F_d\}$ to produce the fused features F_{kv}. The DQ-Decoder block comprises a self-attention layer, a cross-attention layer, and a feed-forward layer. In the DQ-Decoder, the differential image features F_d are transformed into query tokens. This transformation helps the model focus on local changes introduced by incremental strokes. The DQ-Decoder then considers the correspondences between the differential query tokens F_d and the fused features F_{kv} output by the DQ-encoder. The self-attention layer learns the relative attention and interactions among the various elements of differential query tokens. The cross-attention layer implements $CrossAttention\left(Q;K;V\right)=softmax\left(\frac{QK^T}{\sqrt{l}}\right)\cdot V$, and l is the output dimension of key and query features, while

$$Q=W^Q F_d, K=W^K F_{kv}, V=W^V F_{kv}, \tag{2}$$

where W^Q, W^K, and W^V are learnable weights that project F_d to query, and map F_{kv} to key and value, respectively. Finally, the differential query tokens are fed through two MLPs to predict stroke parameters $\hat{S}_t=\{\hat{s}_i\}_{i=1}^N$ and their corresponding confidences $\hat{C}_t=\{\hat{c}_i\}_{i=1}^N$ respectively. During the inference phase, we determine whether the predicted stroke is valid based on the sign of confidence $\hat{c}_i$. If $\hat{c}_i \geqslant 0$, we draw this stroke, otherwise, we skip it. We draw all predicted valid strokes onto the canvas, yielding the final painting $\hat{I}_t$.

C. Training Objective

Pixel Loss. The most direct goal of neural painting is to reconstruct the target image. Therefore, similar to [9], [20], we minimize the L_1 distance between the predicted image $\hat{I}_t$ and the target image I_t as:

$$\mathcal{L}_{pixel}=\lambda_p\left\|I_t-\hat{I}_t\right\|_1, \tag{3}$$

where λ_p is a weight term.

Stroke Loss. Given that the target image is rendered from the canvas image using the set of foreground strokes, we can constrain the difference between the ground-truth and the prediction at the stroke level. We follow the stroke loss [20] on the re-matched strokes as:

$$\mathcal{D}_{match}=\frac{1}{|S_t|}\sum_{u=1}^{|S_t|}\left(c_u\left(\mathcal{D}_{L_1}^u+\lambda_W\mathcal{D}_W^u\right)+\mathcal{D}_{bce}^u\right), \tag{4}$$

where u and $\hat{u}$ represent the target strokes and predicted strokes respectively. $\mathcal{D}_{L_1}^u, \mathcal{D}_W^u$, and $\mathcal{D}_{bce}^u$ represent the pixel loss, rotation loss, and classification loss of the stroke set, respectively, as proposed by [20]. λ_W is a weight term, and $|S_t|$ is the number of strokes.

Further, to encourage the model to reconstruct the target using the minimum number of valid strokes, we impose an additional regularization on the confidence $\hat{C}_t$ of the predicted strokes. We derive the stroke loss:

$$\mathcal{L}_{stroke}=\mathcal{D}_{match}+\lambda_c\frac{1}{|S_t|}\sum_{u=1}^{|S_t|}\left\|\hat{c}_u\right\|_1, \tag{5}$$

where λ_c is a weight term for the confidence regularization.

Adversarial Loss. Treating our painting network as a generator, we design a simple discriminator, which regards the generated images as fake samples, encouraging the model to predict strokes that make the painting closer to the target image. As shown in Figure 2, the discriminator consists of five blocks. In the first block, we replace the Conv layer with a CoordConv layer. The training process employs a WGAN-GP loss [18] as:

$$\mathcal{L}_{adv}=Dis\left(\hat{I}_t\right)-Dis\left(I_t\right)+\lambda_{dis}\left(\left\|\nabla_{\tilde{I}_t}Dis\left(\tilde{I}_t\right)\right\|_2-1\right)^2, \tag{6}$$

where $Dis\left(\cdot\right)$ represents the discriminator score for a given sample. $\tilde{I}_t$ is a linear interpolation between real samples I_t and fake samples $\hat{I}_t$. $\left\|\nabla_{\tilde{I}_t}Dis\left(\tilde{I}_t\right)\right\|_2$ is the $L2$ norm of the gradient of the discriminator on the interpolation point. λ_{dis} is the hyperparameter for the gradient penalty.

Overall loss. Finally, our network is optimized by the pixel loss, the stroke loss, and the adversarial loss as:

$$\mathcal{L}_{total}=\mathcal{L}_{pixel}+\mathcal{L}_{stroke}+\gamma\mathcal{L}_{adv}, \tag{7}$$

where $\gamma=\frac{\|\mathcal{L}_{pixel}\|}{\|\mathcal{L}_{adv}\|}$ is an adaptive balancing factor [18].

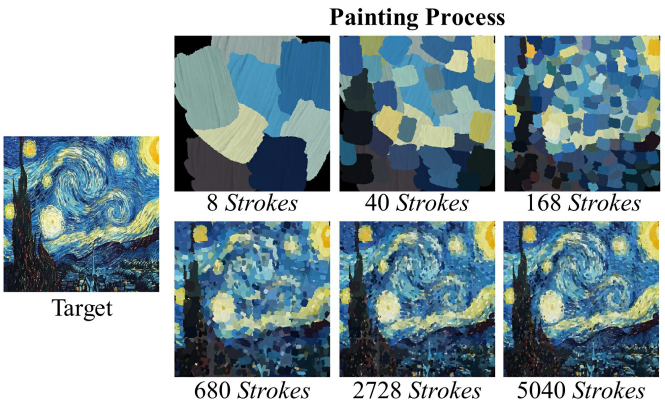


Fig. 3: Our painting progress following a coarse-to-fine manner.

D. Painting Inference

Following the painting strategies of [20], [21], our model generates paintings in a progressive manner, starting from a coarse sketch and gradually refining details across multiple scales. It is worthy noting that the stroke number of our method is not fixed. Because our network also predicts "skip" when the current painting area is already satisfactory. Our coarse-to-fine painting process is illustrated in Figure 3. Moreover, our method produces seamless results without visible patch seams. This is enabled by two design choices: (1) the use of spatial positional embeddings that preserve location awareness even near patch edges; (2) our differential-query mechanism, which conditions stroke prediction on the residual error map across the full canvas context. As a result, strokes near boundaries are not suppressed, and the final composition remains visually coherent.

IV. Experiment

A. Implementation Details

Datasets. Our model is trained exclusively using synthesized stroke images, without relying on any real-world datasets. We

TABLE I: Quantitative comparison with competitive methods under pixel-level and perception-level reconstruction on unseen real-world datasets at different levels of stroke counts. Lower values indicate better reconstruction. Bold indicates best. Benefiting from the observation-first mechanism, our model adapts to varying stroke budgets while preserving fine-grained details and global structure, consistently achieving strong performance across a wide range of stroke counts. This shows its robustness and efficiency in high-fidelity neural painting under resource constraints.

Stroke	Method	Landscape		FFHQ		Wiki Art		Average	
		$\mathcal{L}_{pixel}\downarrow$	$\mathcal{L}_{pcpt}\downarrow$	$\mathcal{L}_{pixel}\downarrow$	$\mathcal{L}_{pcpt}\downarrow$	$\mathcal{L}_{pixel}\downarrow$	$\mathcal{L}_{pcpt}\downarrow$	$\mathcal{L}_{pixel}\downarrow$	$\mathcal{L}_{pcpt}\downarrow$
500	Stylized Neural Painting	0.068	0.941	0.057	1.047	0.064	0.998	0.063	0.995
	Paint Transformer	0.080	0.851	0.067	1.052	0.072	0.934	0.073	0.946
	Im2Oil	0.096	0.992	0.077	1.071	0.089	1.036	0.087	1.033
	Learning To Paint	0.065	0.793	**0.050**	**0.850**	0.062	0.833	0.059	0.825
	Compositional Neural Painter	0.069	0.886	0.053	0.996	0.062	0.907	0.062	0.930
	Ours	**0.063**	**0.751**	0.051	0.881	**0.058**	**0.812**	**0.057**	**0.815**
1000	Stylized Neural Painting	0.072	0.921	0.060	1.012	0.067	0.974	0.066	0.969
	Paint Transformer	0.079	0.843	0.064	1.045	0.069	0.913	0.071	0.934
	Im2Oil	0.094	0.983	0.071	1.040	0.087	1.022	0.084	1.015
	Learning To Paint	0.063	0.805	**0.046**	0.833	0.057	0.829	0.055	0.822
	Compositional Neural Painter	0.063	0.848	0.048	0.946	0.056	0.864	0.056	0.886
	Ours	**0.062**	**0.751**	0.047	**0.830**	**0.056**	**0.789**	**0.055**	**0.790**
5000	Stylized Neural Painting	0.068	0.939	0.057	1.044	0.064	0.996	0.063	0.993
	Paint Transformer	0.070	0.807	0.056	0.934	0.061	0.841	0.062	0.861
	Im2Oil	0.064	0.720	0.042	0.742	0.052	0.718	0.053	0.727
	Learning To Paint	0.055	0.718	**0.032**	0.697	0.047	0.705	**0.045**	0.707
	Compositional Neural Painter	0.056	0.732	0.037	0.772	0.046	0.715	0.046	0.740
	Ours	**0.054**	**0.579**	0.039	**0.631**	**0.045**	**0.593**	0.046	**0.601**

conduct evaluation on three distinct datasets: Landscapes [50], FFHQ [51], and Wiki Art [52]. The Landscapes dataset comprises the natural landscape images sourced from the Flickr website. FFHQ is a high-quality face image dataset that covers a variety of ages, genders, races, and expressions. WikiArt is a compilation comprising a large number of artistic pieces with diverse styles, each piece created through genuine human painting. For each dataset, we randomly select 100 images as test samples.

Settings. We set patch size P as 32 and the maximum number of brushstrokes $|S_t|$ in one patch as 8. During training, parameters for target strokes are randomly generated from a uniform distribution. We sequentially render these strokes, and if a stroke covers more than 75% of the area of the preceding stroke, its confidence is set to 0 to ensure that the rendered strokes do not overly overlap. We follow existing works [20] to set hyper-parameters $\lambda_p = 8$, and $\lambda_W = 10$. For the adversarial loss weight, we follow [18] and set $\lambda_{dis} = 10$. We have conducted experiments to determine the appropriate weight in Eq. 5 and ultimately set $\lambda_c = 0.1$ as default. We use the AdamW optimizer [53] with an initial learning rate of 1×10^{-4} and set weight decay to 1×10^{-2}. The model is trained for 100,000 iterations using a batch size of 64. The first 50,000 iterations are dedicated to pre-training the painting network without the adversarial loss. This strategy helps to avoid mode collapse, ensuring that the generator can faithfully reconstruct the target images.

B. Comparison with State-of-the-Art Methods

Quantitative Comparison. We conduct a quantitative comparison between our method and four state-of-the-art oil painting methods: Stylized Neural Painting [21] (an optimization-based model), Paint Transformer [20] (a neural network-based model), Im2Oil [17] (a traditional search-based model), Learning to Paint [19] (a reinforcement learning-based model) and Compositional Neural Painter [18] (a reinforcement learning-based model). Since the main objective of neural painting is to recreate original images, we directly use the pixel loss $\mathcal{L}_{pixel}$ and the perceptual loss $\mathcal{L}_{pcpt}$ [54] as evaluation metrics. $\mathcal{L}_{pixel}$ calculates the mean L_1 distance between the rendered images and the target images at the pixel level. $\mathcal{L}_{pcpt}$ is a perceptual metric based on neural network features, which measures the similarity between a target image and a generated image by comparing their differences in high-level feature maps. Lower values of $\mathcal{L}_{pixel}$ and $\mathcal{L}_{pcpt}$ both indicate a better image reconstruction quality. All painting results are produced at a resolution of 512×512 pixels. Among the five methods we compare, Stylized Neural Painting, Learning to Paint, and Compositional Neural Painter can set the exact number of strokes. Paint Transformer and Im2Oil can only roughly control the number of strokes by adjusting the setting parameters. For a fair comparison, we conduct experiments at 500, 1,000, and 5,000 strokes respectively.

Table I shows our results on various datasets at different levels of stroke counts. It is intriguing to observe that all methods exhibit loss fluctuations across different datasets, indicating a substantial influence of image content complexity on the painting results. For example, our paintings achieve a lower pixel loss and a higher perceptual loss on the FFHQ dataset compared to the Landscapes and Wiki Art datasets. This difference can be attributed to the nature of the images in each dataset. Although plein-air paintings from the Landscapes dataset exhibit complex compositions, they possess less high-level semantic information compared to the high-definition facial images in the FFHQ dataset. Consequently, the plein-air paintings experience higher pixel loss but lower perceptual

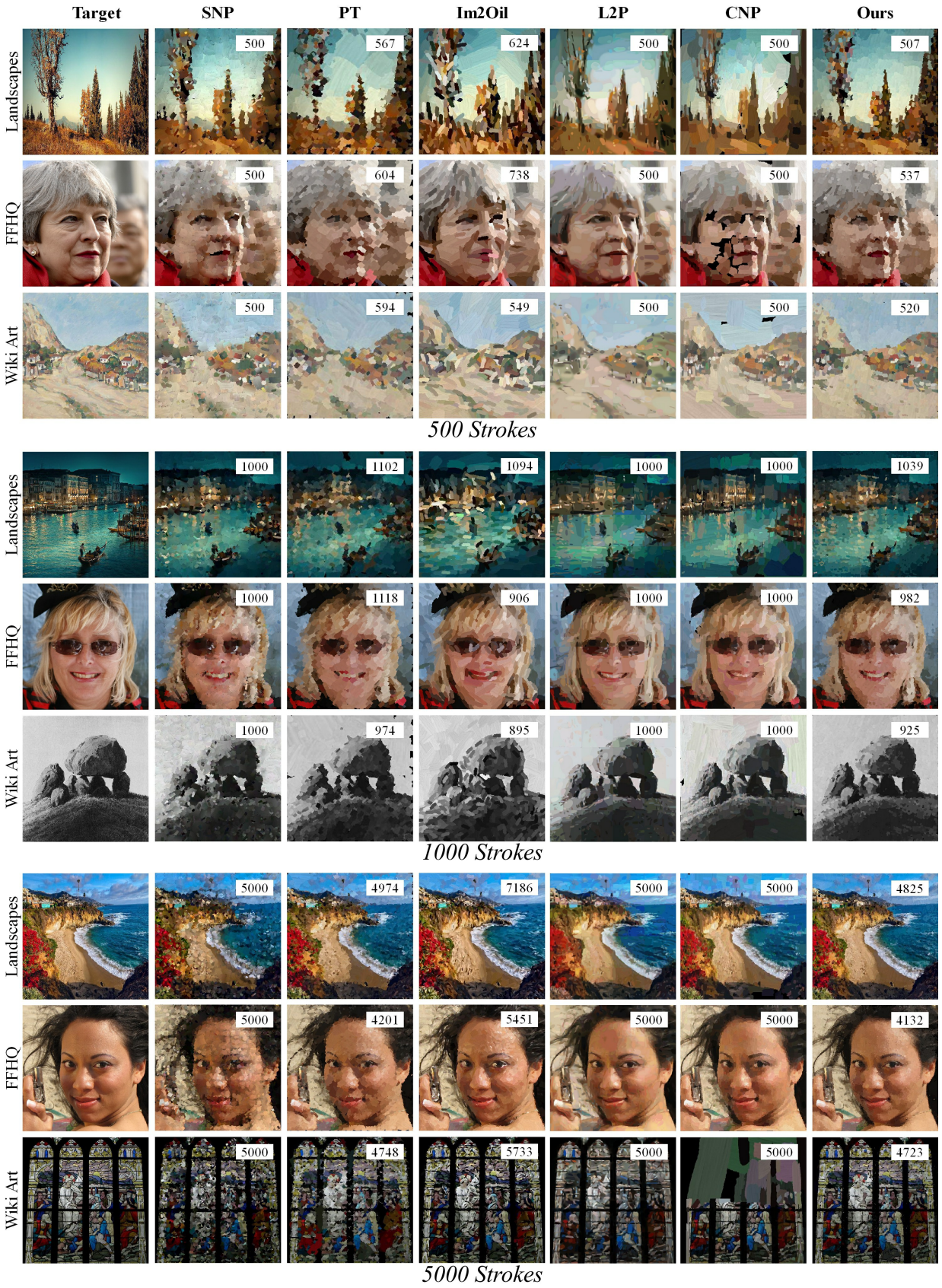


Fig. 4: Qualitative comparison between our model and state-of-the-art neural painting methods on unseen real-world datasets at different levels of stroke counts. The actual number of strokes used in the painting is annotated in the top right corner of the image. Our method leverages the difference image as a dynamic query for each painting step. This observation-first approach enables our model to achieve superior visual quality with relatively fewer strokes, effectively reproducing complex details with high fidelity. Please zoom in to obtain a more detailed view.

loss. This also illustrates the necessity of incorporating both pixel and perceptual loss as evaluation metrics, as they capture different aspects of the painting quality.

As shown in Table I, our approach achieves the best overall balance: at 500 strokes, we obtain the lowest average perceptual loss (0.815) and competitive pixel loss (0.057); at 5000 strokes, we further reduce perceptual loss to 0.601, significantly outperforming all baselines. Notably, Learning to Paint achieves the lowest pixel loss on FFHQ, which can be attributed to its training on additional human face datasets (*e.g.*, CelebA [55]) as reported in the original work. In contrast, our method is trained solely on random strokes without any domain-specific images, yet still achieves competitive pixel accuracy (0.039 at 5000 strokes) while significantly outperforming Learning to Paint in perceptual quality (0.631 vs. 0.697 on FFHQ). While methods such as Stylized Neural Painting, Paint Transformer, Im2Oil, and Compositional Neural Painter achieve competitive results in certain settings, their overall performance remains inferior to ours. The quantitative results highlight the robustness and effectiveness of our approach in reconstructing high-quality images under increasingly complex stroke configurations.

Qualitative Comparison. Figure 4 presents a comprehensive qualitative comparison across three diverse image categories and three stroke budgets (500, 1000, 5000). Stylized Neural Painting produces blocky results with visible grid artifacts, especially at high stroke counts, and yields blurred facial details on FFHQ. Paint Transformer generates coarse strokes that miss fine structures, leading to poor edge definition across all datasets. Im2Oil over-samples strokes in textured regions, such as sand or hair, causing cluttered and disordered outputs due to its density-based sampling strategy. Learning to Paint achieves low pixel loss on faces by leveraging extra face-specific training data, but its renderings appear over-smoothed and airbrushed, lacking authentic brushstroke expressiveness. Compositional Neural Painter, relying on object priors, often leaves blank regions or misaligns strokes on novel or complex scenes like WikiArt, indicating limited generalization. In contrast, our method accurately reconstructs image content while preserving vivid and coherent brushwork, requires no domain-specific image data, and consistently delivers superior visual quality across diverse image types and stroke budgets.

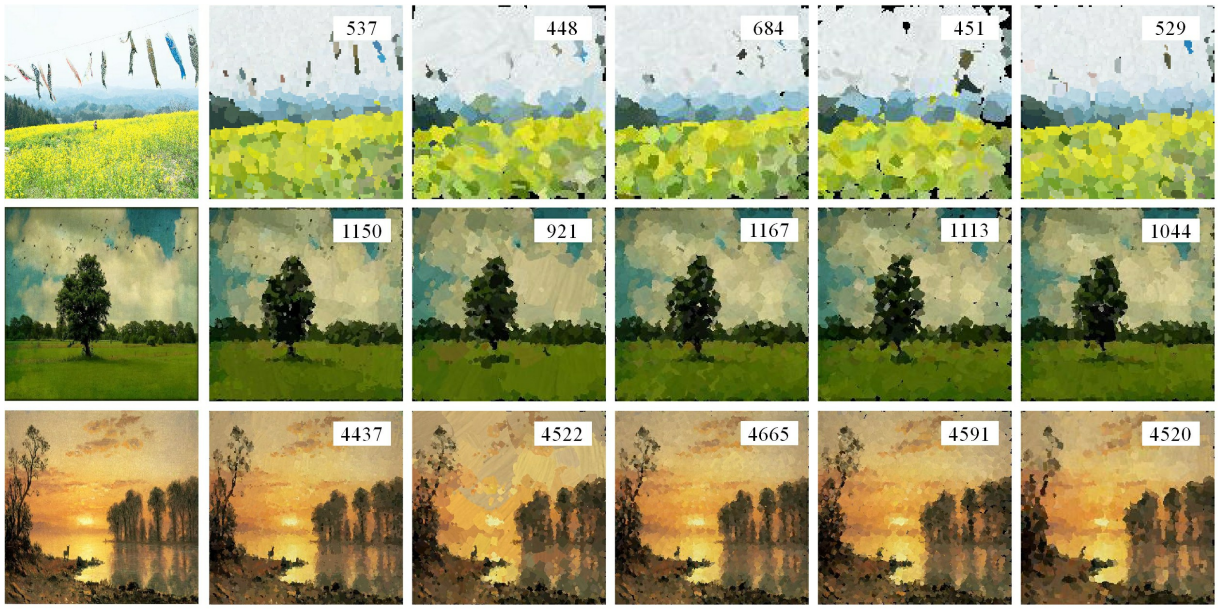
User Study. To further evaluate the painting quality of our model, we conducted a Mean Opinion Score (MOS) study [17] to assess user preferences for automatic oil painting methods. We recruit a total of 30 graduate students from diverse disciplines across our university to participate in the MOS test. We launch a questionnaire website through Gradio [56]. Each questionnaire involves the random selection of 30 image sets, wherein each set comprises one target image alongside five corresponding oil paintings. The identities of these five oil paintings are anonymized within each set, and their presentation sequence is randomized to mitigate order effects. Participants are instructed to evaluate each set of oil paintings and identify the two works they deem to exhibit superior quality. By limiting participants to selecting their top-2 choices, we aim to focus on the most outstanding results while avoiding the potential ambiguity and difficulty of ranking lower-quality paintings. The average user voting rate of each method is shown on the vertical axis of Figure 6. Collectively, the data indicate a pronounced user preference for our proposed oil painting method relative to alternative approaches. Although Compositional Neural Painter and Im2Oil show commendable painting quality, their performance is inconsistent across different images, leading to slightly lower votes. Stylized Neural Painting and Paint Transformer have limitations in detail rendering, which negatively impacts their overall voting.

Efficiency Analysis. The training and inference times of all methods, measured on a single NVIDIA RTX 3090 Ti GPU using their official implementations and default settings, are summarized in Table III. Our method achieves a training time of only 10 hours, substantially outpacing reinforcement learning–based approaches such as Learning to Paint (50 hours) and Compositional Neural Painter (90 hours), which rely on costly policy optimization and extensive environment exploration. It further surpasses Stylized Neural Painting in efficiency and matches the training speed of Paint Transformer. At inference time, our approach renders each image in just 0.72 seconds, on par with Paint Transformer and orders of magnitude faster than Stylized Neural Painting, Im2Oil, and Compositional Neural Painter. Notably, this high computational efficiency is attained without compromising visual fidelity.

C. Ablation Studies

Quantitative Effect of Primary Components. To validate the effectiveness and robustness of each component in our framework, we conduct an extensive ablation study across three representative stroke budgets: 500, 1000, and 5000 strokes. We train four ablated models: one variant without the differential image; one variant without the confidence regularization in Eq. 5; one variant without CoordConv layers; and one variant without the WGAN-based discriminator. As shown in Table II, removing the differential image leads to the most significant degradation, especially under low stroke budgets (*e.g.*, +0.082 in L_{pixel} at 500 strokes), highlighting that error-driven dynamic queries are essential for guiding efficient stroke placement. This confirms that our formulation enables the model to focus directly on reconstruction residuals, improving sample efficiency. The benefit of confidence regularization becomes increasingly evident as the stroke budget grows: its absence leads to higher perceptual loss at 5,000 strokes on both FFHQ and WikiArt. Similarly, discarding the adversarial loss degrades performance on complex WikiArt scenes, where fine textural details are critical. Crucially, the full model maintains a consistent advantage over all ablated versions across all stroke budgets, underscoring the complementary roles of each component in achieving both high fidelity and rendering efficiency.

Qualitative Effect of Primary Components. The qualitative results are shown in Figure 5. Without the differential image, the model suffers from redundant strokes and poor refinement, as seen in the over-painted grass regions. Removing confidence regularization leads to noisy and unstable stroke generation, particularly evident in fine details like tree edges. The absence of CoordConv degrades spatial coherence, resulting in blurred boundaries and distorted structures. Finally, eliminating the



(a) Target (b) Our (Full) (c) *w/o* I_d (d) *w/o* Reg (e)*w/o* CoordConv (f) *w/o* Discriminator

Fig. 5: Ablation study on the primary components of our framework at different stroke counts. The actual number of brushstrokes used in the painting is annotated in the top right corner of the image. Please zoom in to obtain a more detailed view.

TABLE II: Quantitative Effect of Primary Components at different levels of stroke counts. *w/o* I_d denotes that we do not use the differential image, while *w/o* Reg ($\lambda_c = 0$) means the model without confidence regularization in Eq. 5, *w/o* CoordConv represents we solely employ conventional convolutional layers to extract image features, *w/o* Discriminator denotes that we train the model without the discriminator.

Stroke	Method	Landscape		FFHQ		Wiki Art		Average	
		$\mathcal{L}_{pixel} \downarrow$	$\mathcal{L}_{pcpt} \downarrow$	$\mathcal{L}_{pixel} \downarrow$	$\mathcal{L}_{pcpt} \downarrow$	$\mathcal{L}_{pixel} \downarrow$	$\mathcal{L}_{pcpt} \downarrow$	$\mathcal{L}_{pixel} \downarrow$	$\mathcal{L}_{pcpt} \downarrow$
500	*w/o* I_d	0.095	0.883	0.088	1.079	0.090	0.935	0.091	0.966
	w/o Reg ($\lambda_c = 0$)	0.085	0.849	0.077	1.048	0.081	0.907	0.081	0.935
	w/o CoordConv	0.103	0.935	0.104	1.145	0.101	0.994	0.103	1.025
	w/o Discriminator	0.070	0.920	0.062	1.037	0.065	0.975	0.066	0.977
	Ours (Full)	**0.063**	**0.751**	**0.051**	**0.881**	**0.058**	**0.812**	**0.057**	**0.815**
1000	*w/o* I_d	0.084	0.858	0.071	1.027	0.075	0.899	0.077	0.928
	w/o Reg ($\lambda_c = 0$)	0.072	0.794	0.058	0.949	0.065	0.837	0.065	0.860
	w/o CoordConv	0.086	0.911	0.076	1.081	0.082	0.959	0.081	0.984
	w/o Discriminator	0.068	0.821	0.055	0.916	0.061	0.867	0.061	0.868
	Ours (Full)	**0.062**	**0.751**	**0.047**	**0.830**	**0.056**	**0.789**	**0.055**	**0.790**
5000	*w/o* I_d	0.078	0.833	0.064	0.975	0.066	0.868	0.069	0.892
	w/o Reg ($\lambda_c = 0$)	0.064	**0.476**	0.048	0.791	0.055	0.736	0.056	0.668
	w/o CoordConv	0.075	0.854	0.059	0.976	0.067	0.899	0.067	0.910
	w/o Discriminator	0.059	0.735	0.047	0.713	0.051	0.770	0.052	0.739
	Ours (Full)	**0.054**	0.579	**0.039**	**0.631**	**0.045**	**0.593**	**0.046**	**0.601**

TABLE III: Comparison of training and inference time across different painting methods.

Method	SNP	PT	Im2Oil	L2P	CNP	Ours
Training (*hours*)	11	4	-	50	90	10
Inference (*seconds*)	89	0.70	125	3	12	0.72

TABLE IV: Ablation study on the weight λ_c. We set $\lambda_c = 0.1$ as the default value.

λ_c	0.05	0.1	0.2	0.5	1	5	10
$\mathcal{L}_{pixel} \downarrow$	0.048	**0.046**	**0.046**	0.050	0.050	0.055	0.058
$\mathcal{L}_{pcpt} \downarrow$	0.668	**0.607**	0.614	0.686	0.685	0.786	0.791

discriminator causes a loss of artistic style and perceptual realism, producing less expressive paintings. Upon zooming into the detailed sections of the image, the painting produced by the full model appears smoother.

Effect of the Weight λ_c. Furthermore, we investigate the influence of varying weights (λ_c) for the confidence regularization loss on model performance. Table IV shows the pixel loss and perceptual loss of the model on the test set under

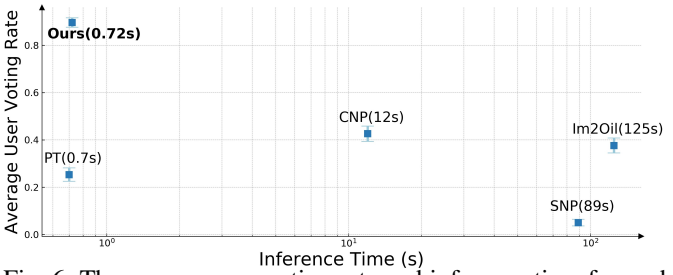


Fig. 6: The average user voting rate and inference time for each method. Methods positioned closer to the upper left corner are characterized by higher user votes and faster inference speeds. Our approach surpasses the comparison methods in preference score by a clear margin and also offers faster inference speed.

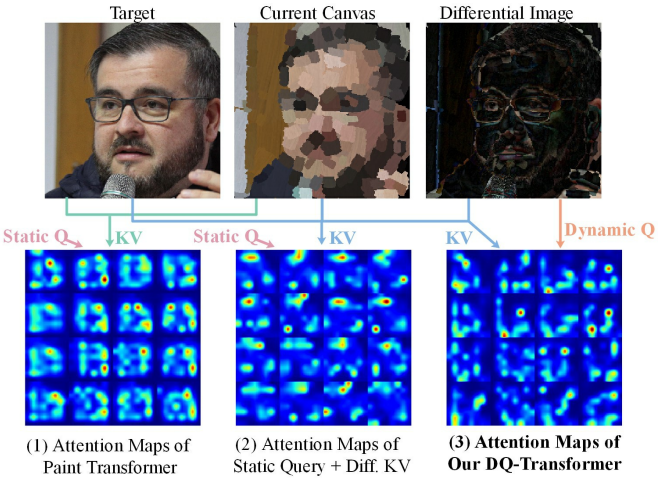


Fig. 7: Attention visualization comparing three configurations. (1) Paint Transformer uses fixed learnable queries and key-value from the target and current canvas. (2) A variant with static queries and key-value from target, canvas, and differential image. (3) Our DQ-Transformer uses the differential image as dynamic query and combines all three inputs for key-value. Our model generates attention maps that sharply focus on regions with significant reconstruction errors.

different weights. We observe that when $\lambda_c > 1$, both the pixel loss and perceptual loss of the model are relatively high, indicating poor image quality. When $\lambda_c < 0.5$, the model exhibits relatively lower pixel loss, and when $\lambda_c = 0.1$, the model achieves the minimum perceptual loss. Consequently, based on the experimental results, we set $\lambda_c = 0.1$ as the default value.

D. Further Discussion

Attention Analysis. To better understand how our differential query mechanism influences stroke placement, we visualize cross-attention maps from three configurations: (1) the original Paint Transformer with fixed learnable queries; (2) a variant where the differential image is used as key-value but queries remain static; (3) our DQ-Transformer, where the differential image serves as dynamic queries. As shown in Figure 7, we visualize the cross-attention maps from the first decoder layer of

Fig. 8: Failure case under an extremely limited stroke budget. With only two strokes, the model produces a highly abstract output that cannot capture the structure of the target image, illustrating a fundamental limitation of stroke-based neural painting methods.

transformer. It's important to note that the full-image attention map is stitched from 16 local maps, as the model processes the image in a 4×4 patch grid during inference. The Paint Transformer exhibits scattered attention patterns with no clear spatial correlation to reconstruction errors. When the differential image is used only as key-value, attention becomes slightly more focused but still fails to consistently highlight under-reconstructed regions. In contrast, our method produces sharp, localized attention peaks that align closely with high-error areas in the differential image. By formulating the differential image as a dynamic query, our model embodies the "look, compare, and draw" painting paradigm: it first looks at the current canvas, compares it with the target to compute residual errors, and then draws strokes guided by those discrepancies. This feedback-driven loop enables the model to allocate brushstrokes adaptively, focusing on regions that need refinement rather than applying uniform coverage.

Comparison with General Image Stylization Methods. Recent advances in diffusion models and large vision-language systems, such as StyleAligned [57] and B-LoRA [58], have achieved impressive results in global style transfer and semantic image manipulation. However, these methods operate in pixel or latent space and generate images holistically, without explicitly modeling the painting process. In contrast, our work falls within the neural painting paradigm, where the core objective is to learn stroke-level prediction through a coarse-to-fine autoregressive sequence. This formulation naturally yields a complete painting trajectory that can be rendered as a temporally coherent animation. Moreover, the generated stroke sequences and their intermediate renderings constitute high-quality, scalable training data for models that aim to learn from procedural creation dynamics. For instance, ProcessPainter [59] leverages neural painting pipelines to synthesize video datasets capturing stroke-by-stroke artistic generation. Crucially, our method requires no real-world oil-painting images for training, relying solely on synthetic strokes, and thus avoids dependence on scarce artistic datasets. Beyond data synthesis, our explicit stroke programs are directly executable by robotic painting systems. Each predicted stroke includes geometric and appearance parameters that can be translated into motor commands for physical brushes. While producing a finished artwork via printing or direct pixel rendering is technically straightforward, the ability to generate a painting step by step

in real time, adapting brushstrokes based on ongoing canvas feedback, mirrors how humans create art and enables truly interactive and observable artistic behavior. This makes our approach particularly attractive for applications such as robot art education and human-robot co-creation. Our method is not intended to replace general-purpose stylization tools, but rather to complement them by offering a process-driven approach to art generation.

Limitations. We acknowledge that our model shares a common limitation with other neural painting approaches. When restricted to an extremely small number of strokes, such as two, it cannot faithfully reconstruct the input image. As shown in Figure 8, the output under this setting is highly abstract and lacks structural fidelity. This behavior stems from the nature of stroke based generation. Each brushstroke is a local and spatially constrained operation. With only a few strokes available, the model has insufficient capacity to represent complex shapes or fine details. It reflects a fundamental constraint of the neural painting paradigm, which relies on iterative refinement over many steps. As the stroke budget increases, for example to 200 strokes, the reconstruction quality improves significantly. Therefore, very low stroke counts should be understood as early sketching stages rather than final outputs. Future work will explore hybrid strategies that combine semantic priors, *e.g.*, keypoints [60], with stroke based rendering to enhance early stage expressiveness.

V. Conclusion

In this work, we introduce a new automatic oil painting method guided by differential images, which generates brushstrokes akin to those created by human artists. We design a Differential Query Transformer and incorporate the differential image features as queries for decoding the brushstrokes. This "Look, Compare and Draw" approach enables the model to precisely focus on the visual effects produced by the incremental addition of strokes. Coupled with adversarial training, this mechanism significantly improves stroke prediction accuracy and, subsequently, enhances the fidelity of the output images. We have conducted comparisons against state-of-the-art stroke-based painting methods on unseen real-world datasets and validated the superiority of our method through a combination of qualitative and quantitative evaluations, as well as a user study, assessing both pixel-level and perception-level reconstruction accuracy.

References

[1] A. Hertzmann, "Generative models for the psychology of art and aesthetics," *Empirical Studies of the Arts*, vol. 43, no. 1, pp. 23–43, 2025.

[2] ——, "Toward a theory of perspective perception in pictures," *Journal of Vision*, vol. 24, no. 4, pp. 23–23, 2024.

[3] B. D. Campbell, N. Hedley, and A. Hertzmann, "Art and artificial intelligence," *IEEE Computer Graphics and Applications*, vol. 44, no. 2, pp. 10–11, 2024.

[4] M. L. de Guevara, M. Fisher, and A. Hertzmann, "Segmentation-based parametric painting," in *2024 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. IEEE, 2024, pp. 1–6.

[5] T. Isenberg, P. Neumann, S. Carpendale, M. C. Sousa, and J. A. Jorge, "Non-photorealistic rendering in context: an observational study," in *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, 2006, pp. 115–126.

[6] J. E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg, "State of the" art": A taxonomy of artistic stylization techniques for images and video," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 5, pp. 866–885, 2012.

[7] P. Rosin and J. Collomosse, *Image and video-based artistic stylisation*. Springer Science & Business Media, 2012, vol. 42.

[8] P. L. Rosin, Y.-K. Lai, D. Mould, R. Yi, I. Berger, L. Doyle, S. Lee, C. Li, Y.-J. Liu, A. Semmo *et al.*, "Nprportrait 1.0: A three-level benchmark for non-photorealistic rendering of portraits," *Computational Visual Media*, vol. 8, no. 3, pp. 445–465, 2022.

[9] J. Singh, C. Smith, J. Echevarria, and L. Zheng, "Intelli-paint: Towards developing human-like painting agents," *arXiv*, 2021.

[10] Y. Liang, J. Tenenbaum, T. A. Le *et al.*, "Drawing out of distribution with neuro-symbolic generative models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 244–15 254, 2022.

[11] Q. Wang, H. Deng, Y. Qi, D. Li, and Y.-Z. Song, "Sketchknitter: Vectorized sketch generation with diffusion models," in *The Eleventh International Conference on Learning Representations*, 2023.

[12] Z. Wang, F. Liu, Z. Liu, C. Ran, and M. Zhang, "Intelligent-paint: a chinese painting process generation method based on vision transformer," *Multimedia Systems*, vol. 30, no. 2, p. 112, 2024.

[13] K. Frans, L. Soros, and O. Witkowski, "Clipdraw: Exploring text-to-drawing synthesis through language-image encoders," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5207–5218, 2022.

[14] P. Haeberli, "Paint by numbers: Abstract image representations," in *SIGGRAPH*, 1990, pp. 207–214.

[15] P. Litwinowicz, "Processing images and video for an impressionist effect," in *SIGGRAPH*, 1997, pp. 407–414.

[16] A. Hertzmann, "A survey of stroke-based rendering." Institute of Electrical and Electronics Engineers, 2003.

[17] Z. Tong, X. Wang, S. Yuan, X. Chen, J. Wang, and X. Fang, "Im2oil: stroke-based oil painting rendering with linearly controllable fineness via adaptive sampling," in *ACMMM*, 2022, pp. 1035–1046.

[18] T. Hu, R. Yi, H. Zhu, L. Liu, J. Peng, Y. Wang, C. Wang, and L. Ma, "Stroke-based neural painting and stylization with dynamically predicted painting region," in *ACMMM*, 2023, pp. 7470–7480.

[19] Z. Huang, W. Heng, and S. Zhou, "Learning to paint with model-based deep reinforcement learning," in *ICCV*, 2019, pp. 8709–8718.

[20] S. Liu, T. Lin, D. He, F. Li, R. Deng, X. Li, E. Ding, and H. Wang, "Paint transformer: Feed forward neural painting with stroke prediction," in *ICCV*, 2021, pp. 6598–6607.

[21] Z. Zou, T. Shi, S. Qiu, Y. Yuan, and Z. Shi, "Stylized neural painting," in *CVPR*, 2021, pp. 15 689–15 698.

[22] D. Kotovenko, M. Wright, A. Heimbrecht, and B. Ommer, "Rethinking style transfer: From pixels to parameterized brushstrokes," in *CVPR*, 2021, pp. 12 196–12 205.

[23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[24] X. Li, C.-C. Lin, Y. Chen, Z. Liu, J. Wang, R. Singh, and B. Raj, "Paintseg: Painting pixels for training-free segmentation," in *Advances in Neural Information Processing Systems*, vol. 36. Curran Associates, Inc., 2023, pp. 35–56.

[25] C. Chen, F. Lv, Y. Guan, P. Wang, S. Yu, Y. Zhang, and Z. Tang, "Human-guided image generation for expanding small-scale training image datasets," *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[26] Z. Zheng, J. Zhu, W. Ji, Y. Yang, and T.-S. Chua, "3d magic mirror: Clothing reconstruction from a single image via a causal perspective," *npj Artificial Intelligence*, 2026.

[27] H. Yi, Z. Zheng, X. Xu, and T.-S. Chua, "Progressive text-to-3d generation for automatic 3d prototyping," *ACM Trans. Multimedia Comput. Commun. Appl.*, 2026.

[28] Y. Suo, Z. Zheng, X. Wang, B. Zhang, and Y. Yang, "Jointly harnessing prior structures and temporal consistency for sign language video generation," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 6, pp. 1–18, 2024.

[29] C. Xu, Y. Xu, H. Zhang, X. Xu, and S. He, "Dreamanime: Learning style-identity textual disentanglement for anime and beyond," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[30] C. Zheng, B. Liu, X. Xu, H. Zhang, and S. He, "Learning an interpretable stylized subspace for 3d-aware animatable artforms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 2, pp. 1465–1477, 2024.

[31] X. Zhang, Z. Zheng, D. Gao, B. Zhang, Y. Yang, and T.-S. Chua, "Multi-view consistent generative adversarial networks for compositional 3d-aware image synthesis," *IJCV*, vol. 131, no. 8, pp. 2219–2242, 2023.

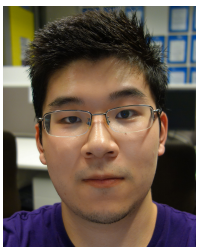
[32] N. Huang, W. Dong, Y. Zhang, F. Tang, R. Li, C. Ma, X. Li, T.-Y. Lee, and C. Xu, "Creativesynth: Cross-art-attention for artistic image synthesis with multimodal diffusion," *IEEE Transactions on Visualization and Computer Graphics*, 2025.
[33] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," in *CVPR*, 2019, pp. 2138–2147.
[34] P. Zhang, L. Yang, X. Xie, and J. Lai, "Pose guided person image generation via dual-task correlation and affinity learning," *IEEE Transactions on Visualization and Computer Graphics*, 2023.
[35] Y. Bai, C. Wang, S. Xie, C. Dong, C. Yuan, and Z. Wang, "Textir: A simple framework for text-based editable image restoration," *IEEE Transactions on Visualization and Computer Graphics*, 2025.
[36] J. Collomosse and P. Hall, "Painterly rendering using image salience," in *Proceedings 20th Eurographics UK Conference*. IEEE, 2002, pp. 122–128.
[37] A. Hertzmann, "Paint by relaxation," in *Proceedings. Computer Graphics International 2001*. IEEE, 2001, pp. 47–54.
[38] ——, "Painterly rendering with curved brush strokes of multiple sizes," in *SIGGRAPH*, 1998, pp. 453–460.
[39] F. Tang, W. Dong, Y. Meng, X. Mei, F. Huang, X. Zhang, and O. Deussen, "Animated construction of chinese brush paintings," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 12, pp. 3019–3031, 2017.
[40] A. Ashcroft, A. Das, Y. Gryaditskaya, Z. Qu, and Y.-Z. Song, "Modelling complex vector drawings with stroke-clouds," in *The Twelfth International Conference on Learning Representations*, 2024.
[41] X.-C. Liu, Y.-C. Wu, and P. Hall, "Painterly style transfer with learned brush strokes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 9, pp. 6309–6320, 2023.
[42] A. Graves, "Generating sequences with recurrent neural networks," *arXiv*, 2013.
[43] K. Frans and C.-Y. Cheng, "Unsupervised image to sequence translation with canvas-drawer networks," *arXiv*, 2018.
[44] Q. Dong, L. Liu, Y. Wang, J. J. Liu, and Z. Zheng, "Domain-agnostic neural oil painting via normalization affine test-time adaptation," in *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025, pp. 12 390–12 398.
[45] X. Song, C. Liu, Y. Zheng, Z. Feng, L. Li, K. Zhou, and X. Yu, "Hairstyle editing via parametric controllable strokes," *IEEE Transactions on Visualization & Computer Graphics*, vol. 30, no. 07, pp. 3857–3870, 2024.
[46] Y. Ganin, T. Kulkarni, I. Babuschkin, S. A. Eslami, and O. Vinyals, "Synthesizing programs for images using reinforced adversarial learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1666–1675.
[47] T. Zhou, C. Fang, Z. Wang, J. Yang, B. Kim, Z. Chen, J. Brandt, and D. Terzopoulos, "Learning to sketch with deep q networks and demonstrated strokes," *arXiv*, 2018.
[48] J. Singh and L. Zheng, "Combining semantic guidance and deep reinforcement learning for generating human level paintings," in *CVPR*, 2021, pp. 16 387–16 396.
[49] R. Liu, J. Lehman, P. Molino, F. Petroski Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution," *Advances in neural information processing systems*, vol. 31, 2018.
[50] Y. Chen, Y.-K. Lai, and Y.-J. Liu, "Cartoongan: Generative adversarial networks for photo cartoonization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9465–9474.
[51] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.
[52] F. Phillips and B. Mackintosh, "Wiki art gallery, inc.: A case for critical thinking," *Issues in Accounting Education*, vol. 26, no. 3, pp. 593–608, 2011.
[53] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7
[54] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 694–711.
[55] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018.
[56] "Gradio: Build machine learning web apps — in python," Gradio, 2023. [Online]. Available: https://gradio.app/
[57] A. Hertz, A. Voynov, S. Fruchter, and D. Cohen-Or, "Style aligned image generation via shared attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4775–4785.
[58] Y. Frenkel, Y. Vinker, A. Shamir, and D. Cohen-Or, "Implicit style-content separation using b-lora," in *European Conference on Computer Vision*. Springer, 2024, pp. 181–198.
[59] Y. Song, S. Huang, C. Yao, X. Ye, H. Ci, J. Liu, Y. Zhang, and M. Z. Shou, "Processpainter: Learn painting process from sequence data," *arXiv:2406.06062*, 2024.
[60] J. Lin, Z. Zheng, Z. Zhong, Z. Luo, S. Li, Y. Yang, and N. Sebe, "Joint representation learning and keypoint detection for cross-view geo-localization," *IEEE Transactions on Image Processing*, vol. 31, pp. 3780–3792, 2022.

Lingyu Liu received her B.E. degree in information management and information system from Northwest A&F University, China, in 2017. She is currently a Ph.D. candidate in the School of Software Engineering at Xi'an Jiaotong University. Her research interests include image caption and image generation.

Yaxiong Wang received the B.S. degree from Lanzhou University, Lanzhou, China, in 2015, and Ph.D. degree at School of software Engineering, Xi'an Jiaotong University, Xi'an, China, in 2021. He is now a associate professor in Hefei University of Technology. His research interests include cross-modal retrieval, image generation, semantic segmentation, and ReID.

Li Zhu received the B.S. degree from Northwestern Polytechnical University, Xi'an, China, in 1989, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, in 1995 and 2000, respectively.,He is currently a Professor with the School of Software, Xi'an Jiaotong University. His main research interests include multimedia processing and communication, parallel computing, and networking.

Lizi Liao is an assistant professor at the Singapore Management University. She obtained her Ph.D. from National University of Singapore in 2019. Dr. Liao's research interests center on task-oriented dialogues, proactive conversational agents, and multimodal conversational search and recommendation as the application target. She serves as senior PC member or area chair of these prestigious conferences and organizing committee members of SIGIR'24, WWW'24, WSDM'23 and ACM MM'19 etc.



Zhedong Zheng is an assistant professor with the University of Macau. He was a research fellow at School of Computing, National University of Singapore. He received the Ph.D. degree from the University of Technology Sydney, Australia, in 2021 and the B.S. degree from Fudan University, China, in 2016. He received the IEEE Circuits and Systems Society Outstanding Young Author Award of 2021. He serves as an area chair at ACM MM and ICASSP, and a publication chair at ACM MM'25.