Multimedia Systems (2024) 30:112
https://doi.org/10.1007/s00530-024-01316-w

REGULAR PAPER

Intelligent-paint: a Chinese painting process generation method based on vision transformer

Zunfu Wang[1] · Fang Liu[1] · Zhixiong Liu[1] · Changjuan Ran[1] · Mohan Zhang[1]

Received: 28 October 2023 / Accepted: 6 March 2024 / Published online: 3 April 2024

Abstract
The generation of painting steps can help people understand how artistic works are created and assist beginners in learning through copying. However, this task faces significant challenges: achieving the generation of clear and plausible intermediate painting steps while maintaining consistency with the real painting process. Existing related research mainly focuses on generating painting steps for oil painting using brush stroke rendering methods. However, such approaches often result in significant discrepancies between the generated process and the real painting process, making it challenging to reflect the principles and techniques of painting accurately, and they are not applicable to Chinese painting. To better address the issue of generating painting steps for Chinese painting, we propose "Intelligent-paint". First, considering the unique painting principles of Chinese painting, we interpret the painting process as a mapping from the final artwork to a series of intermediate painting stages. To ensure the quality of the generated intermediate stages, we use a generator based on Vision Transformer (Vit) for this mapping process. We enhance the image generation quality by adversarial learning with a real/fake discriminator. In addition, to capture the characteristics of Chinese painting, such as void and brush strokes, we employ void loss constraint and brush stroke loss constraint to ensure consistency with the features of Chinese painting. To ensure the coherence between the generated painting sequence and the real painting process, we employ a sequence discriminator to constrain the generated painting sequence. Expert evaluations and quantitative assessments indicate that our method outperforms existing approaches. Through ablation experiments and applicability evaluations, our method demonstrates strong rationality and applicability, providing significant assistance to beginners in learning Chinese painting.

Keywords Chinese painting · Vision transformer · Painting steps · Image generation

1 Introduction

Painting is an important artistic form of human expression that has evolved over thousands of years, gradually developing various techniques for visual representation from its earliest cave drawings. Human painters often require extensive learning and practice to master this skill. Is it possible for machines or artificial intelligence agents to learn and master this human skill? Research on intelligent painting agents can enhance people's understanding of painting, helping them comprehend how exquisite art pieces are created and reducing uncertainties for beginner painters during their self-learning process.

Intelligent painting agents refer that, given a specified image input, generate realistic and plausible painting processes. This area of research is fraught with challenges. First, painting is a highly complex and specialized skill, where different painting styles and types possess different methods

Communicated by B. Bao.

✉ Fang Liu
fangl@hnu.edu.cn

Zunfu Wang
wangzunfu@hnu.edu.cn

Zhixiong Liu
zhixiongliu@hnu.edu.cn

Changjuan Ran
s220800533@hnu.edu.cn

Mohan Zhang
designzmh@hnu.edu.cn

[1] School of Design, Hunan University, Changsha 410082, Hunan, China

and principles. Particularly, in the context of Chinese painting, artists seek a delicate balance between realism and artistic expression to achieve a distinctive artistic conception, using specific perspectives and techniques for visual expression. Second, unlike video generation or prediction tasks in traditional computer vision research, painting involves a diverse array of subjects. For instance, when creating artworks depicting still life or human figures, artists may adopt various drawing strategies. In addition, inferring the possible painting process solely from the final artwork lacks necessary temporal and motion information. Moreover, human painters often employ editing and retouching techniques during the creative process, introducing extra uncertainty for machine learning. Finally, real painting processes are not linear; artists may pause and resume their work, leading to personalized approaches that defy capturing common underlying patterns.

Existing research on painting process generation can be divided into two categories: stroke-based rendering methods [1] and pixel-based video prediction methods [2]. Stroke-based rendering methods are the most commonly used approaches for generating painting processes. These methods imitate the real painting process by decomposing the input image into a series of coarse-to-fine strokes. The challenge lies in generating stroke decompositions of varying granularity and producing authentic brushstrokes. While existing studies have made considerable progress, these methods assume a priori that painting processes involve a transition from coarse-to-fine strokes and do not consider the sequential order of brushstrokes in real painting processes. As a result, although these methods can generate logically consistent drawing processes for oil paintings, they exhibit significant differences when applied to painting forms such as watercolor or traditional Chinese painting, where brushstroke variations are minimal. Video prediction methods for generating painting processes are a relatively new research area. Zhao et al. [3] utilized CVAE [4] to learn pixel variations in oil painting processes and synthesized painting process sequences that align with real painting processes using a temporal sequence oil painter. This method interprets the painting process as pixel variations from a blank canvas to the final painted image. Although it can generate seemingly reasonable drawing processes, it suffers from high computational complexity, limited output image size (e.g., 64×64), and the inability to capture brushstroke variations, resulting in a tremendous gap from real painting processes.

We propose a novel painting step generation approach based on the Vit generator of a generative adversarial network (GAN) [5] to generate a sequence of painting steps that both maximizes the preservation of realistic painting brushstrokes and adheres to authentic painting processes. We interpret the Chinese painting process as a mapping from the final artwork to a series of intermediate stages, where we specify the drawing stage labels and generate corresponding stage images from the final artwork. Our generator accepts two conditional inputs: the stage labels and the final artwork. We employ a discriminator similar to PatchGANs [6–8], capable of producing two different outputs based on the settings of the output layer, one for judging the authenticity of the input image and the other for predicting its corresponding stage label. The adversarial interplay between the discriminator and generator encourages the generator to produce more realistic and coherent intermediate images that align with the painting stages.

Unlike traditional GANs, our ViT-based Unet [9] generator achieves better image generation quality. To address the unique characteristics of Chinese painting, such as void and brushstroke techniques, we use specific constraints to generate more authentic images. For void, we utilize adversarial loss combined with L1 loss to generate images that resemble real effects and leave appropriate areas blank. For brushstroke constraints, similar to the method used by He et al. [10], we employ a pre-trained holistically nested edge detector [11] to impose loss constraints on the edge effects of generated and real images, ensuring the generated images closely match the brushstroke edges in real Chinese paintings. We compare our method with brush rendering-based methods, video prediction-based methods, and other generator-based GANs. The quantitative evaluations, qualitative assessments, and expert evaluations consistently demonstrate the superiority of our proposed method across multiple metrics. Our main contributions can be summarized as follows:

(1) We propose a new approach for generating painting steps, tailored to the characteristics of Chinese painting, including blank space and brushstroke techniques. This approach enables the generation of realistic and reasonable image painting processes while preserving authentic brushstrokes. (2) We employ a ViT-based generator method for image generation, extending the research from single-image domains to multi-target domains in image generation. (3) Through experiments, we demonstrate that our method can generate authentic painting steps and, to some extent, assist beginners in the process of learning painting.

The remainder of this paper is organized as follow. Section 2 summarizes related works. We analyzed the technical characteristics of Chinese painting in Sect. 3. Then we present the problem overview and the details of our methods in Sect. 4. The evaluation method of our solution and results are presented in Sects. 5 and 6 separately. Finally, Sect. 7 concludes our work.

2 Related work

2.1 Research on reasoning the painting process

2.1.1 Stroke-based rendering methods

Existing researches on painting process generation primarily focus on stroke-based rendering methods. Early works in this field concentrated on stylizing real images through abstract strokes [1, 12–15]. Typically, these methods process real images using an ordered set of strokes, controlling the color, shape, size, and direction of individual strokes. Subsequently, researchers began simulating painting processes by controlling the appearance order and timing of strokes within the image. Fu et al. [16] generated realistic painting processes for line sketches by controlling the visibility order of strokes based on painting cognition mapping. Similarly, Fan et al. [17] decomposed the strokes in traditional Chinese ink paintings and used a natural evolution strategy to determine the order of brushstroke appearance, producing realistic painting processes. These stroke order-based painting process generation methods require the construction of specialized libraries of fundamental brushstrokes and rules, which involves considerable time and effort for decomposition and encoding of the original artwork. With the development of machine learning techniques, stroke order can be automatically optimized in a self-supervised or semi-supervised manner. Kevin et al. [18] employed self-supervised deep neural networks to learn the mapping between paintings and brushstrokes, providing a foundation for image-to-brushstroke decomposition. Huang et al. [19] used a reinforcement learning-based method to determine the position and color of each stroke and made long-term plans to decompose texture-rich images into strokes. Jaskirat et al. [20] introduced a semantic segmentation module based on reinforcement learning to recognize foreground and background elements in the image, generating more realistic painting processes. However, reinforcement learning has the characteristics of unstable training and low efficiency, limiting its universality and practicality [21].

Zou et al. [22] reformulated stroke prediction as a "parameter search" process, distinct from previous stroke-based rendering methods that employed stepwise greedy search [12, 23], recurrent neural networks [24], or reinforcement learning [19, 25–27]. This method maximizes the similarity between the input and rendered output in a self-supervised manner. Zheng et al. [28] employed self-supervised learning to map brushstrokes to the final image, generating painting processes for images such as doodles and handwritten characters. This method is more efficient compared to reinforcement learning-based methods. Liu et al. [21] proposed a novel Transformer-based framework for predicting stroke set parameters.

While stroke-based rendering methods transfer high-level cognitive painting knowledge to intelligent agents and generate planning processes consistent with human cognition to some extent, their assumption that decomposed brushstrokes align with the actual brushstroke variations in real painting processes is challenging to meet. Therefore, existing methods simulate painting processes by transitioning from coarse-grained to fine-grained stroke variations. Consequently, the generated painting processes differ significantly from real ones and are applicable only to certain painting types with significant brushstroke variations, such as oil paintings and tapestries [22].

2.1.2 Pixel-wise prediction methods

If stroke-based rendering methods operate in a higher level cognitive dimension of painting process knowledge, pixel-wise prediction for painting process generation works at a more foundational level of knowledge reasoning. Compared to stroke-based rendering methods, pixel-based prediction methods retain more original painting knowledge information and align more closely with real painting steps but introduce challenges such as increased computational complexity and less discernible cognitive features. Zhao et al. [29] employed CVAE-based learning for painting process modeling and optimized the generated painting process sequences through sequence optimization. This method achieves expected results for oil paintings and watercolor paintings. However, it has limitations: due to computational constraints, the input and output image sizes are only 50×50, and the generated process sequences exhibit significant variations only in the early frames, making it difficult to observe distinct differences in later frames.

In our method, we preprocess real painting data based on expert knowledge, aiming to retain as much information about authentic brushstrokes as possible while achieving brushstroke stacking orders consistent with real painting processes.

2.2 Image generation research

Painting process generation can be understood as an image generation task that generates time-series images by inputting final paintings. Therefore, we also need to review the related research on image generation. With the continuous development of deep learning technology, significant progress has been made in image generation tasks, achieving the ability to generate visually indistinguishable fake images. Early deep neural network

image generation methods focused on autoregressive models, and autoregressive models such as PixelRNN [30] and PixelCNN [31] achieved good experimental results. However, autoregressive models have drawbacks such as high computational cost and unsuitability for generating large-sized images. Variational autoencoder (VAE) models, which adopt an encoder–decoder structure and generate images by sampling from a latent vector following a Gaussian distribution, have become popular image generation algorithms. VAE models are easy to converge and computationally efficient compared to autoregressive models, but they suffer from image blurriness. Subsequently, the introduction of generative adversarial networks (GANs) greatly promoted the development of image generation research. GAN models optimize the quality of generated images through the continuous adversarial optimization between the generator and discriminator. GAN models can generate realistic large-sized images and achieve diverse image generation tasks by employing different generator and discriminator settings, obtaining impressive generation results. These tasks include text-to-image generation, image-to-image translation, and cross-domain image transformation. Although GANs have limitations in terms of limited generation diversity and training instability, GAN-based image generation methods remain the mainstream solution in image generation tasks and have evolved into many domain-specific image generation and processing studies. Recently, the remarkable progress of attention mechanisms and Transformer-style architectures in natural language processing has drawn attention to their potential in the field of image generation, leading to extensive research on Vision Transformers. Compared to traditional CNN-based generation models, Transformers can more effectively capture non-local patterns, which are common in nature [9]. The applications of Transformers in computer vision were introduced in [32], while recent work has shown that a CNN–Transformer hybrid can achieve better performance [33, 34]. Diffusion models [35], a class of likelihood-based models with a stationary training objective, can obtain better sample quality than state-of-the-art GANs. Although diffusion models represent a promising direction in generative modeling, they are slower than GANs during sampling due to multiple denoising steps (thus forward propagation). Furthermore, in scenarios with relatively small-scale data, GANs still have advantages in terms of training and inference time while ensuring image generation quality.

3 Technical characteristics of Chinese painting

Before undertaking the intelligent generation of Chinese painting processes, a brief analysis and summary of Chinese painting are necessary. Unlike other types of artworks, Chinese painting possesses unique drawing techniques and expressive methods, as shown in Fig. 1. One prominent feature is the extensive use of void in Chinese paintings, leaving significant areas of the composition unpainted. In addition, the brushstrokes in Chinese painting differ significantly from those in oil painting and other styles. Moreover, Chinese painting often employs a flat perspective rather than the realistic perspective sought in Western art.

There are numerous books and studies that analyze and introduce Chinese painting techniques throughout history, with Xie He's "Guohuapinlu," proposed during the Southern Dynasty of China, being a classic example. He advocated a systematic approach to painting, emphasizing drawing according to different parts and shapes of the subject, followed by color filling based on the subject's various colors. Similarly, other literature [36, 37] also highlights that the drawing order in Chinese painting is influenced

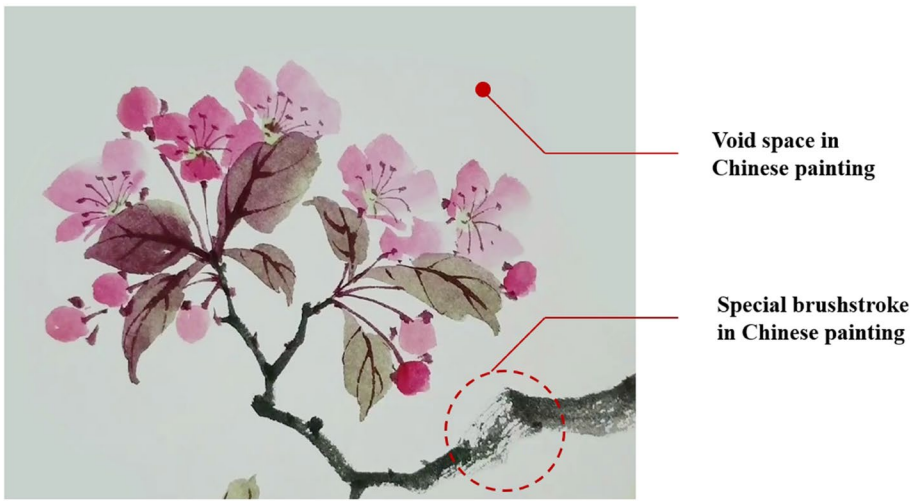

Fig. 1 Void space and unique brushstrokes in Chinese paintings

by factors such as shape, shadow, color, and semantics. After summarizing relevant literature and consulting with Chinese painting experts, we can outline the main principles of meticulous brushwork flower painting in Chinese painting as follows:

Shape: Chinese paintings are drawn based on the different shapes of the objects. For instance, leaves with similar shapes are drawn together, and petals with similar shapes are drawn together. Color: objects of the same color are usually drawn together to reduce the need for frequent ink changes. For example, green leaves and branches are typically drawn together, followed by the drawing of petals in different colors. Darker colors are usually drawn first, followed by lighter ones. Position: objects with close proximity are usually drawn together in the drawing order. Moreover, there is a general tendency to draw from the center to the periphery, and from major elements to minor details. Importance: the main theme elements are generally drawn first, with details or decorative parts drawn subsequently. The above principles may sometimes conflict with each other, and there is no strict hierarchy of priority. As long as the drawing order adheres to one or more of these principles, we consider the process to be reasonable and compliant with the specifications. Figure 2 shows the actual drawing process of Chinese painting.

In traditional research on painting process generation, two main approaches are stroke-based rendering methods and pixel-wise prediction methods. Stroke-based rendering methods are suitable for painting types with distinct stroke segmentation, such as oil paintings, but they are not well-suited for realistic paintings like watercolor paintings, and the stroke segmentation process differs significantly from real painting processes, as shown in Fig. 2. Pixel-wise prediction methods can generate more diverse variations in the image, with richer granularity. However, their generated painting processes are non-linear (with significant changes in the early stages and less noticeable changes in the later stages), which does not effectively reflect painting techniques and principles. To generate realistic and reasonable image painting processes while preserving authentic brushstrokes, we propose a new pixel-wise prediction method for painting step generation. We decompose the real painting process into a series of intermediate stages based on expert knowledge. Each drawing stage corresponds to a unique label. During model inference, given the final artwork and stage labels as input, the model generates the final painting process sequence. This knowledge-based learning approach allows the painting agent to generate realistic painting process sequences while ensuring that the painting sequences adhere to painting principles.

4 Method

4.1 Problem overview

Painting process varies greatly depending on the materials used, the subject of the painting, and the artist's painting habits. For example, some artists prefer to complete their work quickly, while others prefer to spend a long time refining their pieces. It is challenging to enable machines or intelligent agents to master this complex process. Our goal is to enable intelligent agents to generate drawing steps that closely resemble real painting processes. As mentioned earlier, painting processes are diverse, and our objective is not to generate all possible painting processes but rather to generate a reasonable and consistent painting process that adheres to painting principles and habits. Therefore, our research question can be summarized as follows: given or inputting an artwork to an intelligent agent, how can we generate an image sequence that approximates the real painting process? In other words, how can we transform a single image into a sequence of target images, ensuring the coherence between the target image sequences?

Assuming we have a completed artwork represented as the mathematical formula: x_T, our goal is to infer the previous creative sequence as a sequence of formulas: $x_1, x_2, \ldots, x_{T-1}$. If we have real painting video sequences obtained based on expert knowledge represented as the set of formulas: $\left\{X^{(i)} = x_1^i, x_2^i \ldots, x_{T^{(i)}}^i\right\}$, we first construct the mapping model from the x_T to the previous $x_1, x_2, \ldots, x_{T-1}$ frame by frame: $G{:}\{x_T, c_1\} \rightarrow x_1, G{:}\{x_T, c_2\} \rightarrow x_2, \ldots, G{:}\{x_T, c_{(T-1)}\} \rightarrow x_{T-1}$,

Fig. 2 Comparison of the painting process between Chinese painting and Western painting. Chinese paintings usually draw different parts of the object one after another, while Western paintings usually use the background–foreground layered drawing strategy

where different input labels c_i are used to distinguish each mapping. During the training process, the model parameters are adjusted by learning from all video data. During testing, the trained model combined with different labels is used to achieve the image translation from the final artwork to each frame before it.

4.2 System framework

Our model architecture is illustrated in Fig. 3, consisting primarily of three components: the generator, the discriminator, and the brush stroke constraint. The generator takes the final artwork and specified painting stage labels as input to output corresponding stage images of the painting process. It generates a sequence of continuous stage images based on a series of consecutive stage labels. Unlike traditional generators using CNN, our generator adopts the ViT-Unet architecture, incorporating attention mechanisms, enabling our model to produce better image results. Detailed design aspects of our model will be discussed in the following section.

The discriminator adopts the PatchGAN structure, configured to simultaneously discriminate the realness of input images and classify their corresponding painting stage labels. We refer to these classifiers as D_{src} and D_{cls}, respectively. It should be noted that D_{cls} inputs a real drawing stage image and corresponding stage label each time. By sequentially learning the input of a complete drawing process, D_{cls} has the ability to distinguish the entire drawing process. The dual discriminator structure ensures that the generator not only produces realistic images but also maintains the correlation and distinction between the generated painting stage image sequences, aiming to closely resemble authentic painting processes. To address the unique characteristics of Chinese painting, such as blank space and distinctive brushstrokes, we supervise the generator using both Void constraint and brush stroke constraint to generate more authentic images.

For the blank space constraint, we apply L1 loss to constrain the generated images to their corresponding real images, as L1 loss produces clearer details compared to L2, ensuring that the images generate white blank areas in the correct positions. For the brush stroke constraint, we use a pre-trained holistically nested edge detector to extract edge features from both real and generated images. By imposing the edge feature loss constraint, we encourage the generator to better preserve the authentic brushstroke effects during image generation.

Regarding the generation of painting steps, it is essential not only to produce realistic intermediate images but also to ensure that the generated image sequences are both interconnected and independent. Specifically, for a certain intermediate stage image, the content already drawn in the previous stage's drawing image should be completed in this stage, and this stage's image should continue based on the previous stage's work. In other words, the content in the intermediate stage image should be more than the previous stage and less than the following stage. We achieve this objective through an adversarial interplay between the generator and the discriminator's stage label classification. The generator needs to adversarially confront D_{src}, making it difficult to discern whether the generated images are real or fake to improve image generation quality. In addition, the generator needs to continuously engage with D_{cls} to ensure that the discriminator's output stage labels align with the original input stage labels, ensuring that the generator can produce stage images that comply with authentic painting processes. Through this design, during testing, we only need to input the final painting and the stage label of each stage into the generator G, and then the serialized drawing process can be output, and the continuity and consistency of the sequenced drawing process can be ensured.

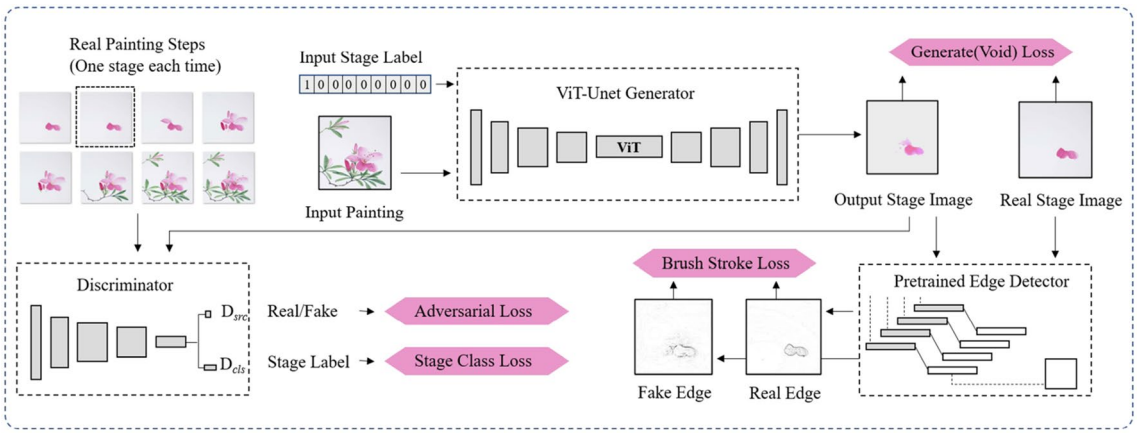

Fig. 3 Schematic diagram of intelligent-paint

4.2.1 Generator model architecture

Our generator model adopts a UNet-Vit structure similar to UVCGAN [9]. The generator consists of a UNet [38] with a pixel-wise Vision Transformer (ViT) [32] at the bottleneck. UNet was originally designed for medical image segmentation, using an encoder–decoder structure with a U-shape. Compared to earlier models like Fully Convolutional Networks, UNet employs skip connections for feature map fusion, enabling the capture of more image features.

In the traditional UNet model, the encoder path extracts features from the input through four convolutional layers and downsampling. The features extracted at each layer are passed through skip connections to the corresponding layers in the decoder path, with the bottom-level features also being passed to ViT. We assume that skip connections are effective in transferring high-frequency features to the decoder, while ViT provides an effective method for learning paired relationships of low-frequency features.

On the encoder path of UNet, the preprocessing layer transforms the image into a tensor with dimensions (w_0, h_0, f_0). The tensor undergoes halving of width and height at each downsampling block, while the feature size doubles in the last three downsampling blocks. The output from the encoder path, with dimensions $(w, h, f) = (w_0/16, h_0/16, 8f_0)$, serves as the input to the pixel-wise ViT bottleneck.

Our generator model details are shown in Fig. 4. The pixel-wise ViT mainly consists of a stack of Transformer encoder blocks [39]. To build the input for the stack, ViT first flattens the encoded image along the spatial dimensions to form a token sequence. The length of the token sequence is $w \times h$, and each token in the sequence is a vector of length f. It then concatenates each token with a two-dimensional Fourier positional embedding of dimension f_p [40] and linearly maps the result to a dimension of f_v. To enhance the convergence of Transformers, we adopt the rezero regularization scheme [41] and introduce a trainable scaling parameter α, which adjusts the size of the non-trivial branch of the residual block. The output of the Transformer stack is projected back to size f and is not attenuated to have width w and height h. In this study, we use 12 Transformer encoder blocks and set $f, f_p, f_v = 384$, and $f_h = 4f_v$ for the feed-forward networks in each block.

4.2.2 Discriminator model architecture

Our discriminator model adopts the PatchGAN [dd1] structure. PatchGAN maps the input to an $N \times N$ patch (matrix) X, where X_{ij} represents the probability of each patch being a real sample. Taking the average of X_{ij} values yields the final output of the discriminator. This encourages the GAN discriminator to model high-frequency structures while relying on an L1 term to enforce low-frequency correctness. To model high-frequency structures, it is sufficient to focus our attention on the local image patches.

We modified the PatchGAN structure by introducing two different branches at the output layer to separately output the image's real/fake classification and the painting stage label classification. For D_{src}, its final output is a $4 \times 4 \times 1$ tensor representing the real/fake classification labels. As for D_{cls}, the number of convolution operations determines its output size, generating a label sequence of size T. In our experiments, we set T to 9. For detailed model specifications, please refer to Fig. 5.

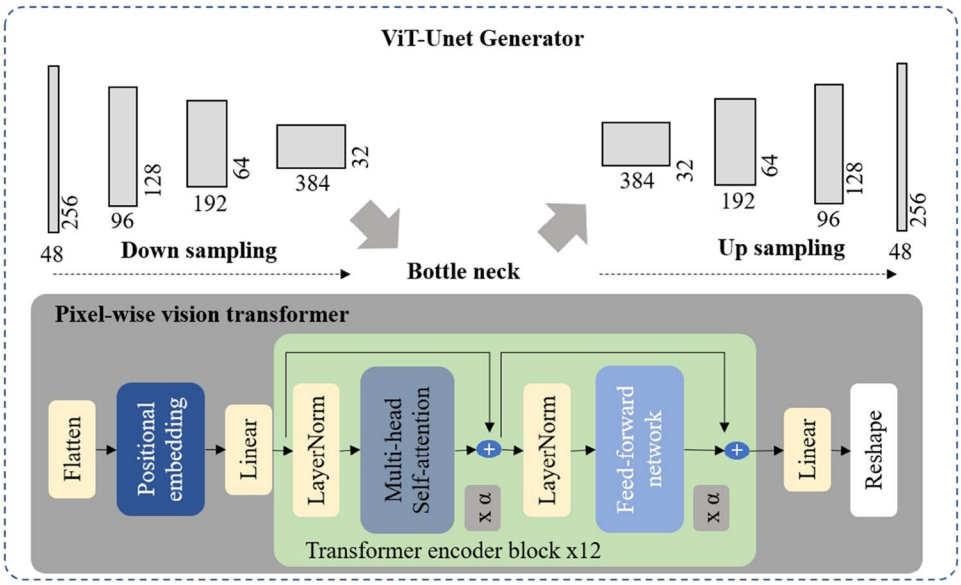

Fig. 4 Vit-Unet generator framework

4.3 Loss functions

4.3.1 Adversarial loss

We employ the GAN architecture to achieve the generation of target images. Therefore, we aim to make the generator G generate realistic images that the discriminator D cannot distinguish from real images. Thus, we define the adversarial loss as follows:

$$L_{\text{adv}} = \mathbb{E}x[\log D_{\text{src}}(x)] + \mathbb{E}x, c[\log(1 - D_{\text{src}}(G(x, c)))] \quad (1)$$

Here, G generates the target image $G(x, c)$ based on the input painting image x_t and the label information c, while D attempts to discriminate between real and generated images. $D_{\text{src}}(x)$ represents the probability distribution of D's output. The generator G tries to minimize this objective, while the discriminator D tries to maximize it.

4.3.2 Label classification loss

Given the input painting image x_T and the target domain label c_i, our goal is to generate the target domain image $\bar{x}_i$ that can be correctly classified by the discriminator and assigned the label c_i. Since the discriminator needs to correctly classify both real and generated images, we need to compute the classification loss separately for real and generated images.

For the classification loss of real images, it is defined as:

$$L_{\text{cls}}^{T} = \mathbb{E}(x, c')[-\log D_{\text{cls}}(c'|x)] \quad (2)$$

Here, $D_{\text{cls}}(c'|x)$ represents a probability distribution over domain labels generated by D. To minimize this objective, D tries to classify a real image x into its corresponding original domain c'. The input image and label pair (x, c') is given by the training data.

As for the fake image, the loss function for domain classification is defined as:

$$L_{\text{cls}}^{f} = \mathbb{E}(x, c)[-\log D_{\text{cls}}(c|G(x, c))] \quad (3)$$

In this case, G tries to minimize this objective to generate images that can be classified as the target domain c.

4.3.3 Void constraint and brush stroke constraint

Void constraint involves leaving blank spaces at appropriate locations on the canvas [10]. Since we employ the pix2pix method in our task, the adversarial loss and generative loss help create proper voids in the output image while preserving essential information from the source image.

To ensure that the generated results contain the same painted areas as real painting processes, we adopt the brush stroke constraint method proposed in Ref. [10]. This method utilizes a pre-trained VGG-19 feature extractor E to obtain the edge map of both the real image and the generated image. The brush stroke loss is then employed to ensure that the result $E(G(x_i))$ and the real image $E(x_i)$ share similar areas. The brush stroke loss is defined as follows:

$$\mathcal{L}bs = \mathbb{E}x \sim p_{data}(x)\left[-\frac{1}{N}\sum_{i=1}^{N}(\mu E(x)i \log E(G(x))i + (1-\mu)(1 - E(x)i)\log(1 - E(G(x))i))\right] \quad (4)$$

Here, N represents the number of elements in the edge map, and μ is a hyperparameter used to balance the contributions of positive and negative examples in the loss calculation.

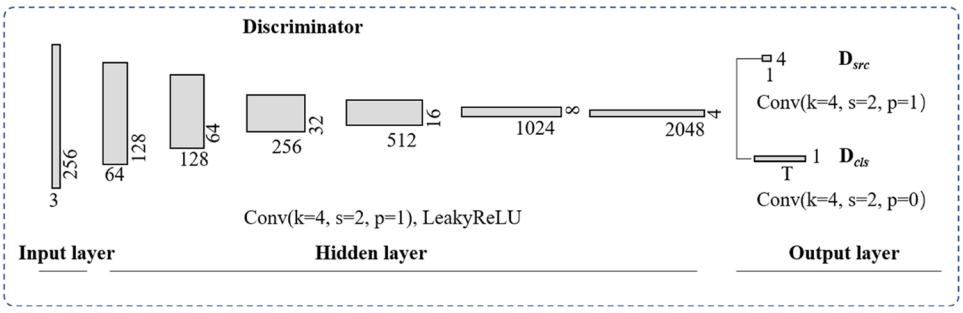

Fig. 5 Discriminator model networks

This brush stroke constraint ensures that the generated image captures similar brush stroke areas as seen in the real image.

4.3.4 Generation loss

For the generator G, its task is not only to generate images that are indistinguishable from real ones by the discriminator D but also to generate results that are as close to real images as possible. Isola et al. [8] proposed using the L1 distance to generate less blurry results compared to the L2 distance. Therefore, we use the L1 loss and define it as:

$$L_{L1} = \mathbb{E}_{(x_T, x_i, c_i)}[||(x_i - G(x_T, c_i))||_1] \tag{5}$$

4.3.5 Overall loss

Finally, the objective functions to optimize G and D are written, respectively, as

$$\mathcal{L}_D = -\mathcal{L}_{\text{adv}} + \lambda_{\text{cls}}\mathcal{L}^T_{\text{cls}} \tag{6}$$

$$\mathcal{L}_G = \mathcal{L}_{\text{adv}} + \lambda_{\text{cls}}\mathcal{L}^f_{\text{cls}} + \lambda_{L1}\mathcal{L}_{L1} + \lambda_{bs}\mathcal{L}_{bs} \tag{7}$$

λ_{cls} and λ_{L1} are the coefficients to control the relevant loss weights, we use both in train and test $\lambda_{\text{cls}} = 1$, $\lambda_{L1} = 10$, $\lambda_{bs} = 10$.

5 Experimental setup

5.1 Datasets

Due to the focus on stroke-based rendering methods in previous research on painting process generation, there are not many publicly available datasets of painting process videos. Only zhao [29] et al. have publicly released datasets of watercolor and digital painting videos. Therefore, we created our own painting process dataset—the Chinese Flower Painting Dataset (traditional Chinese painting can be divided into freehand painting and realistic painting, and our dataset mainly focuses on realistic style paintings using the technique of "meiguhua").

We collected 296 Chinese flower painting process videos from the internet. These videos were shot from a fixed perspective, and the painting paper remained stationary during the process. The average length of the videos is 3.5 min, and they have a resolution of 720P or 1080P. We decomposed all the videos into individual frames and selected clean frames that only contain the painting content without any irrelevant objects such as the artist's hands, painting tools, or subtitles. We then invited painting experts to further select frames that accurately reflect the painting process and maintain a linear progression, meaning that the degree of change between frames is relatively consistent. During the training of our model, we split the entire dataset into training, validation, and testing sets in a ratio of 70:15:15.

Since our model requires fixed-length frame sequences of length T for training and testing, we selected T frames at regular intervals from the processed datasets for model training. The training was conducted using image resolutions of 256×256.

5.2 Training procedure

5.2.1 Pre-training

In the training process of GAN networks, mode collapse can occur, where the generator fails to faithfully reproduce the target distribution of images. It has been shown that transfer learning in GANs is an effective approach to improve performance on small training datasets [42–46]. Therefore, before training the entire model, we performed pre-training on the generator. To create non-matching images, we tiled the 256×256 images into non-overlapping 32×32 pixel blocks and randomly masked 40% of the blocks by setting their pixel values to zero. We used the Adam optimizer, cosine annealing learning rate scheduler, and standard data augmentation techniques such as small random rotations, random cropping, random flipping, and color jittering. During pre-training, we did not differentiate the image sequence labels, which means that the main focus was on training the generator to generate target images with specified labels in subsequent image sequence generation training.

5.2.2 Training for painting process image sequence generation

We trained the model for 50 epochs on the collected dataset. We used the Adam optimizer with a constant learning rate of 0.0001 during the first half of the training, which was then linearly annealed to zero during the second half. We applied three data augmentations: resizing, random cropping, and random horizontal flipping. Before randomly cropping the images to 256×256, we enlarged them from 256×256 to 286×286.

5.3 Baselines

To comprehensively compare the performance of our method in painting process generation, we compared it with stroke-based rendering methods and pixel prediction methods. In addition, to test the effectiveness of our Vit-Unet generator compared to other generators, we also compared it with the performance of the pix2pix generator. Finally,

we used linear interpolation from a blank canvas to the final artwork as a baseline test.

LearningToPaint [19] is a typical method for generating painting process using stroke-based rendering. LearningToPaint determines the position and color of each stroke through reinforcement learning and can decompose images with rich textures into strokes. LearningToPaint can be trained without the experience of human painters or stroke tracking data. Therefore, we used the author's pre-trained model for comparison. In the comparative testing, we selected T frames of the painting process at regular intervals and compared them with other methods.

PaintTransformer [21] is a method that uses stroke-based rendering for painting process generation and has achieved good performance. PaintTransformer addresses the issues of training instability and limited applicability in traditional reinforcement learning-based stroke rendering methods. It proposes a Transformer-based framework that defines the painting process as a stroke set prediction problem and predicts the parameters of stroke sets using a feed-forward network. It also designs a self-training workflow that can be trained without any existing datasets. PaintTransformer has good generalization capabilities and can achieve good painting performance. We did not retrain PaintTransformer but used the best parameters provided by the authors for comparison. Since PaintTransformer can generate painting process sequences of arbitrary lengths, in the quantitative evaluation tests, we resized all the test images to 50×50 and compared them by selecting T frames at regular intervals.

PaintingManyPasts [29] is a pixel prediction-based painting process generation method. Zhao et al. use a VAE model to generate intermediate images of the painting process and employ a sequence optimizer based on convolutional neural networks to make the generated image sequences more consistent with the real painting process. PaintingManyPasts has achieved good performance in the watercolor and digital painting datasets. For the Chinese flower painting dataset, we conducted retraining according to the author's initial settings. It should be noted that PaintingManyPasts assumes a fixed output length of 40 frames and image size of 50×50. Therefore, in the quantitative evaluation tests, we resized all the test images to 50×50 and compared them by selecting T frames at regular intervals.

Since our method employs the Vit-Unet generator for image generation, to test the effectiveness of the attention mechanism and compare it with traditional image-to-image translation tasks, we also conducted tests by replacing ourVit-Unet generator with the generator from the pix2pix method [8]. During the model training, we did not pre-train the generator separately but trained the entire model for 200 epochs and selected the best-performing epoch for comparison.

Finally, to demonstrate the effectiveness of each method, we designed a linear interpolation method from a blank image to the final artwork as a baseline. We calculated the L1 pixel distance between uniformly distributed intermediate images as the measure of the painting process. We refer to this baseline method as the "inter method" and consider it as a quantitative lower bound for comparison.

6 Experimental results

6.1 Qualitative evaluation

The comparison of our method with various baseline methods in generating results is shown in Figs. 6 and 7. During the testing process, we used T=9 as the sequence length for painting process inference. For PaintingManyPasts, LearningToPaint, and PaintTransformer, as their generated sequence lengths are inconsistent and longer than 9, we selected 9 frames at regular intervals for comparison.

It can be observed that, our method generates painting process sequences that closely resemble the real process of watercolor painting and Chinese flower painting, as shown in Figs. 6 and 7. The Pix2Pix method generates painting process sequences with unclear region boundaries and color distortions. The PaintingManyPasts method fails to accurately reflect the painting order of different regions in the process, resulting in color blurriness and unrealistic painting processes. Although the LearningToPaint and PaintTransformer methods can generate seemingly reasonable painting processes, the variations in strokes deviate significantly from real painting processes, and they fail to reflect the order of painting different regions. Therefore, from a visual perspective, our method demonstrates superior performance in the tested paintings.

6.2 Expert evaluation

As painting is a professional skill that requires expertise and knowledge, we invited 20 painting experts to evaluate the generated images and the rationality of the painting process using a five-point Likert scale for each criterion. All the painting experts have received education and practical experience in Chinese painting. For the 6 different model methods, we randomly selected 10 images from the test set for evaluation by the painting experts, and each expert evaluating 60 test images. From the expert evaluation results in Table 1, we can see that our method outperforms existing methods both in terms of image generation quality and rationality of the rendering process.

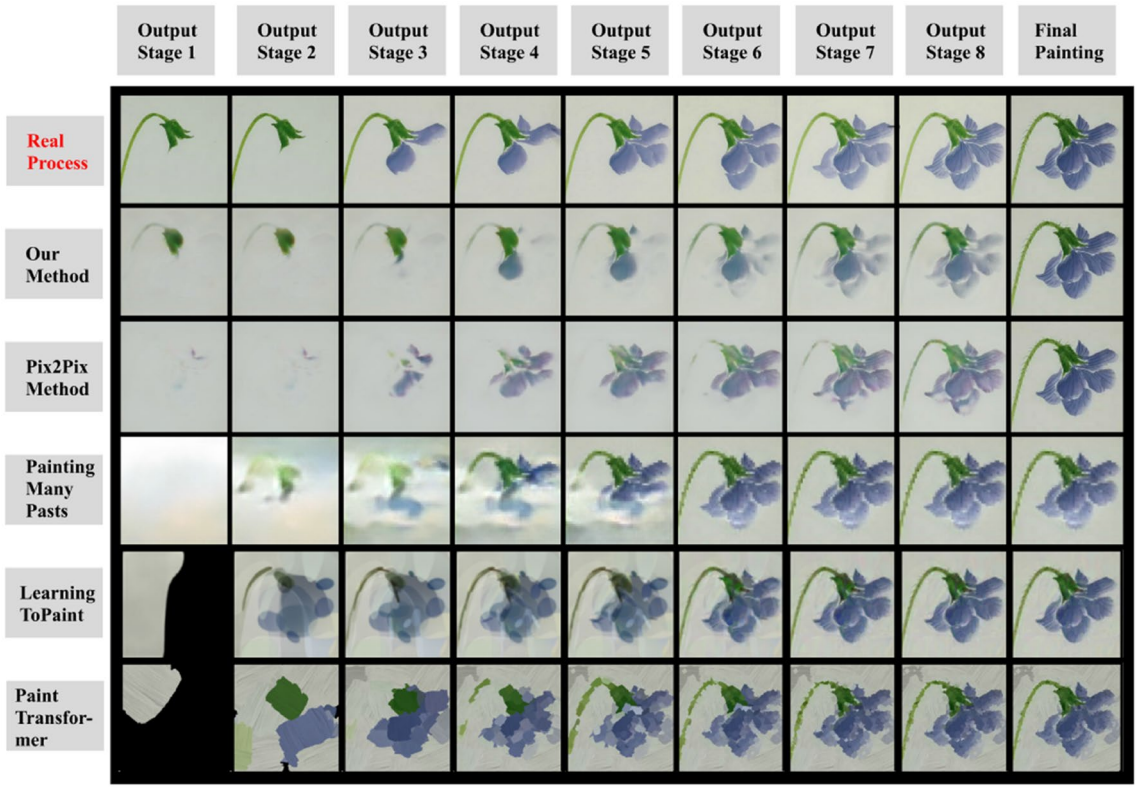

Fig. 6 Comparison of the quality of our method and other methods in the Chinese painting process

6.3 Quantitative evaluation

6.3.1 Evaluation metrics

Fréchet inception distance (FID) [47], structural similarity index (SSIM) [48], and peak signal-to-noise ratio (PSNR) [49] are commonly used metrics for evaluating the image generation quality in various image generation tasks. FID measures the similarity between two sets of images based on the statistical similarity of their computer vision features, which are calculated using the Inception v3 image classification model. A lower FID score indicates greater similarity between the image sets, with a score of 0.0 representing identical images. PSNR is another metric used to measure image quality. It is based on mean squared error (MSE) and is commonly used to represent the distortion level of images. Higher PSNR values indicate better image quality, typically categorized as follows: PSNR $>$ 40 dB: excellent image quality (very close to the original image); 30–40 dB: good image quality (perceptible distortion but acceptable); 20–30 dB: poor image quality; below 20 dB: unacceptable image quality. SSIM measures the similarity between two images by calculating their luminance, contrast, and structure, and it produces values between 0 and 1, with higher values indicating greater structural similarity between the images. We use the above three metrics to evaluate the image quality of our model.

From the metrics in Table 2, it can be seen that our method outperforms other methods in terms of image quality across multiple dimensions. Although these metrics can evaluate the quality of generated images, some researchers have pointed out that for stochastic tasks, comparing synthesized results to a "ground truth" is ill-defined [8, 50]. Therefore, we adopted the evaluation strategy used by zhao et al. to evaluate PaintingManyPasts. The first method is called Best Result Distance (lower is better): for all the test results, the most similar results were selected to calculate the L1 distance to the corresponding real videos. A lower distance indicates more realistic results. Another method is called Best Painting Change Shape Similarity (higher is better): for the test results, the real change shape was compared to the most similarly shaped change synthesized by each method, measured by intersection-over-union (IOU). This metric captures whether a method paints in similar semantic regions to the artist.

From the evaluation results in Table 3, it can be observed that our method achieves superior image generation quality and closer resemblance to the real painting process compared to other methods.

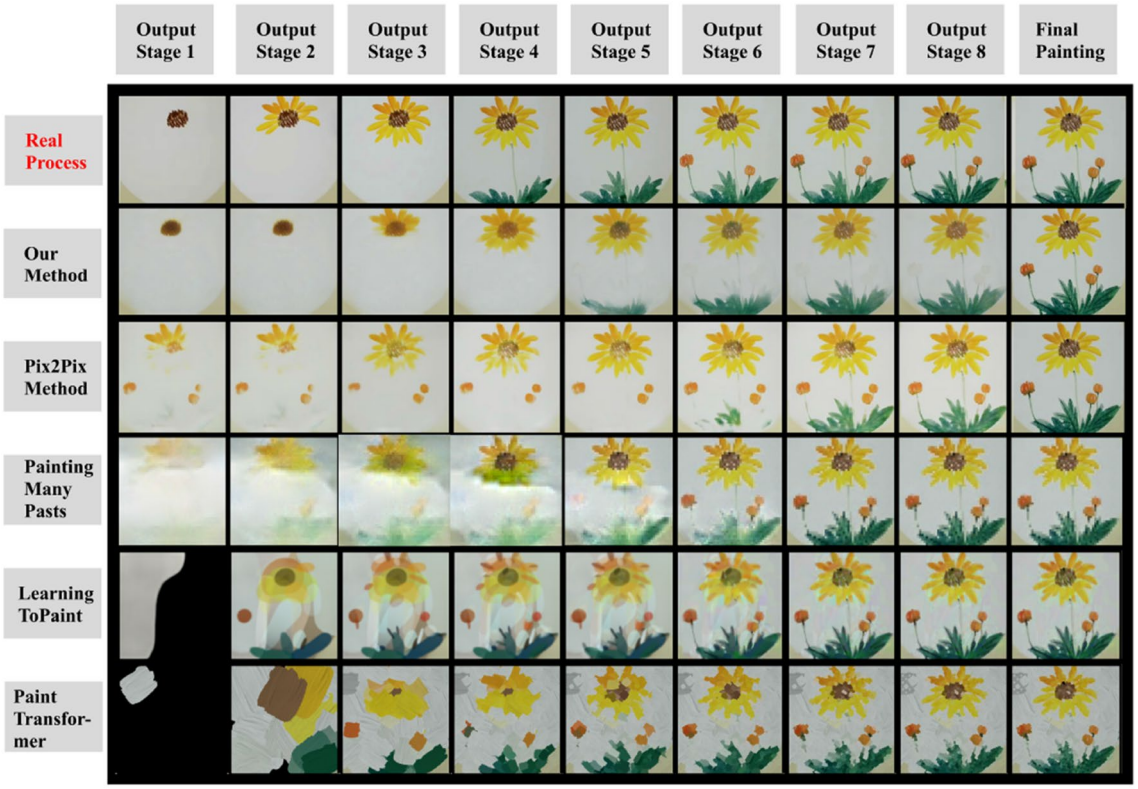

Fig. 7 Comparison of the quality of our method and other methods in the Chinese painting process

Table 1 Comparison results with expert evaluation of existing painting step generation methods

Methods	Image quality ↑	Drawing process ↑
LearningToPaint	3.2	2.6
PaintTransformer	3.4	2.8
PaintingManyPasts	1.6	1.8
Pix2Pix method	2.2	3.2
Inter method	0.5	0.1
Our method	3.5↑	4.2↑

Table 2 Quantitative image quality comparison with existing painting step generation methods

Methods	FID ↓	SSIM ↑	PSNR ↑
LearningToPaint	320.42	6.75	0.11
PaintTransformer	140.83	14.32	0.27
PaintingManyPasts	150.45	10.91	0.21
Pix2Pix method	164.78	15.14	0.31
Inter method	132.14	14.53	0.39
Our method	130.27	15.48	0.42

6.4 Interpretation of attention

Since we use the Vit-Unet structure with attention mechanism in the bottleneck of the generator, visualizing the attention mechanism can help understand its role in the process. We visualized the 12 Transformer encoder blocks in the pixel Vit. When multi-head attention is used, each head produces an attention matrix. For simplicity, we averaged the attention weights over all heads and target tokens for each block in the Transformer encoder stack. We generated a heatmap as follows: reshaped a feature vector to a square of size 16×16, upscaled it 16 times to match the dimension of the input image, and then applied a Gaussian filter with $\sigma = 16$. The results are shown in Fig. 8. It can be seen that

Table 3 Drawing step quantitative evaluation results

Methods	L1 ↓	Chang IOU ↑
LearningToPaint	0.85	0.15
PaintTransformer	0.82	0.14
PaintingManyPasts	0.94	0.14
Pix2Pix method	0.68	0.32
Inter method	0.82	0.1
Our method	0.16	0.54

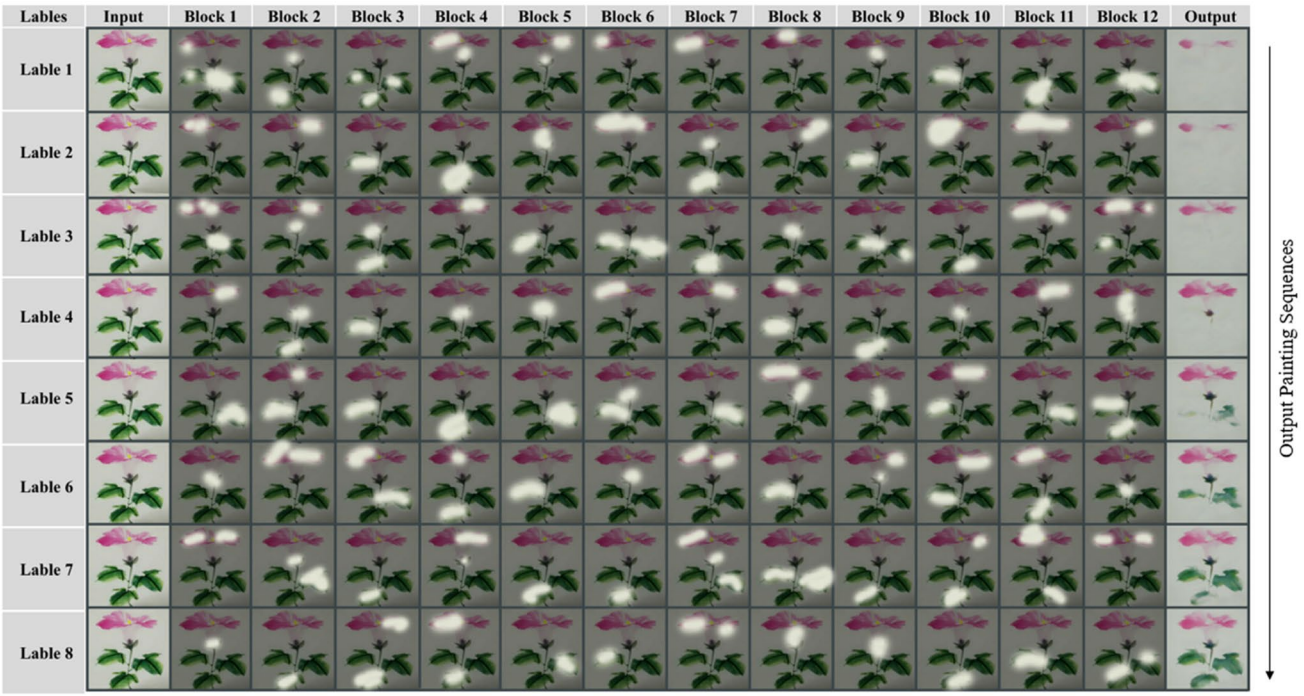

Fig. 8 Visualization results of the attention mechanism

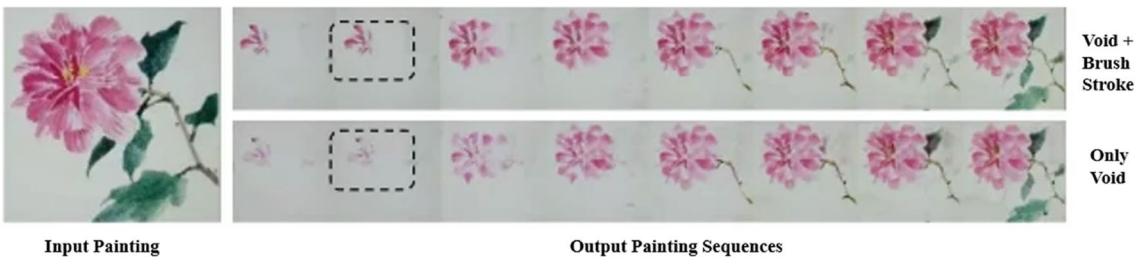

Fig. 9 Visual quality comparison of different variants of our method

for Chinese flower painting, the attention is mainly focused on the composition elements of the flowers' petals, leaves, and stems. This is consistent with the principle of painting different parts based on their locations.

6.5 Ablation study

We proposed two essential constraints to deal with voids and brush strokes. Since the void constraint is the combination of generative adversarial loss, which cannot be ablation from the complete network, we focus on evaluating the importance of brush stroke constraint. We, therefore, train two variant networks, one for void constraint only, the other for void and brush stroke constraints.

We show the results of ablation study experiments in Fig. 9. For the parts with complex strokes, the method with the brush stroke constraint can better present the delicate strokes (e.g., the black boxes in the figure).

Table 4 Image quality evaluation results under different loss constraints

Methods	FID ↓	SSIM ↑	PSNR ↑	Experts score ↑
Void + brush stroke	167.31	15.33	0.29	4.1
Only void	132.25	15.58	0.46	3.6

To more accurately detect the differences in image generation performance between two different loss settings, we utilized FID, SSIM, and PSNR metrics to measure the results of different methods on the test set. In addition, we invited 20 painting experts to subjectively rate the generated images from different methods using a 5-point Likert scale

Table 5 Model training time comparison

T value	Pix2Pix method	PaintingManyPasts	Our method
6	~ 9 h	~ 20 h	~ 14 h
9	~ 16 h	~ 26 h	~ 22 h
12	~ 38 h	~ 80 h	~ 36 h
24	~ 80 h	~ 125 h	~ 40 h

Table 6 Quality evaluation results under different T values

T value	L1↓	Chang IOU↑
6	0.08	0.64
9	0.16	0.54
12	0.14	0.48
24	0.18	0.38

to determine their closeness to real paintings (higher scores indicating better results). The evaluation results are shown in Table 4. From the results in the table, it can be observed that the image quality of the generated pictures decreased when the brush stroke constraint was removed, as reflected by various evaluation metrics and methods. In other words, both blank space and brush stroke constraints effectively enhance the quality of Chinese painting image generation.

6.6 Model utility evaluation

6.6.1 Painting sequence length test

Our model can generate painting processes with (T-1) frames. In theory, as long as there is a sufficiently long training dataset, T can be any value. To investigate the generalization performance of the model, we tested the training time and output quality of the model with different T values. Since the methods of LearningToPaint and PaintTransformer are based on stroke rendering and can be trained without the need for a dataset, comparing them is not meaningful, so we did not include them in the comparison. PaintingManyPasts can theoretically generate painting sequences of unlimited length, so we conducted the comparison tests with the same sequence length and input–output size. We trained the model using two Tesla V100 GPUs on the Chinese flower painting dataset, with input and output sizes both set to 256×256.

From Table 5, it can be seen that as the value of T increases, both the training time for the Pix2Pix and PaintingManyPasts methods significantly increases. By the use of pre-training, the increase in training time for our method is not significant as T increases.

To examine the changes in image quality and resemblance to the real painting process with different sequence lengths, we tested the Best Result Distance and Change IOU for different lengths in the Chinese flower dataset, and the results are summarized in Table 6.

From the table, it can be observed that as the length of T increases, both the image generation quality and the resemblance to the real painting process decrease. This may be due to the difficulty of capturing more painting variation patterns as T increases. In addition, it could be related to the quality of the dataset. As the sequence length increases, it becomes challenging to create consistent real image painting sequences with coherent variation patterns. It is important to note that despite the decrease in quality with increasing T, our method still outperforms other methods.

6.6.2 Test on oil painting

While different painting techniques indeed involve distinct artistic skills, there still exists some commonality among them. For instance, whether it is oil painting, watercolor, or Chinese painting, artists typically paint the same color parts simultaneously, and they often start with darker areas before moving on to lighter ones. To evaluate the scalability of our model, we conducted testing in the domain of oil painting, and the results are depicted in Fig. 10.

It can be observed that in the case of real oil painting processes, artists usually start by outlining the rough contours of the objects using a single color and then proceed with applying paint of different colors as references, with colors that are closer in appearance often being painted together. Our method can accurately reflect the painting order of different color regions in a realistic manner. While Sun's method may have slight differences in the specific order of painting in certain areas compared to the real process, it still follows the general principles of painting. Similarly, although the Pix2Pix method can depict the painting order, it exhibits incomplete strokes in the segmentation of painting regions. The PaintingManyPasts method focuses more on reflecting the background-to-foreground order in the painting process, but it suffers from severe stroke distortions and blurry images. Both the LearningToPaint and PaintTransformer methods use stroke rendering to represent the painting process, but LearningToPaint uses color blocks to represent strokes, while PaintTransformer closely resembles the strokes in real oil painting. However, except for the first frame, both methods treat the entire image as a whole in subsequent frames, which deviates from the general practice of painting different regions in a different order. Therefore, overall, our method can generate painting process sequences that adhere to real painting principles while retaining realistic strokes, and it exhibits better visual performance compared to other methods.

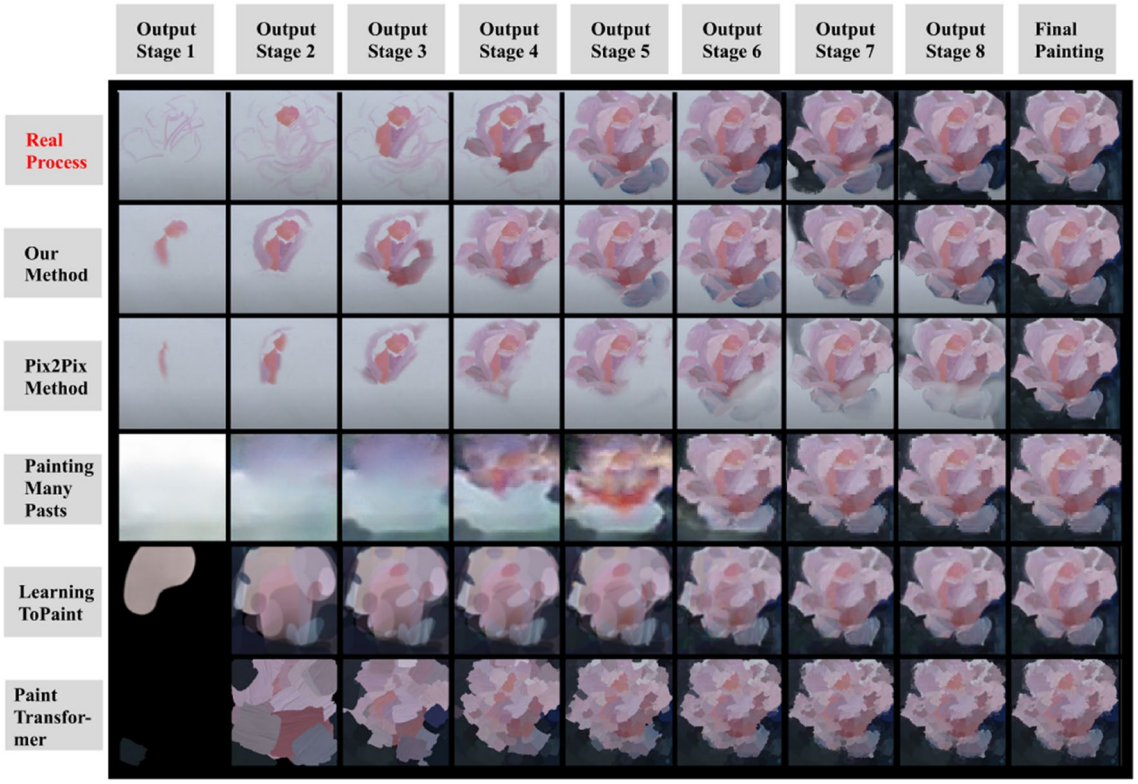

Fig. 10 Comparison of the quality of our method and other methods in the Chinese painting process

6.6.3 User study

To test the practicality of our approach, specifically its ability to alleviate the burden on beginners during the self-learning process, we conducted a user test. We invited five novice Chinese painting learners and provided them with our generated painting process to assist in their copying practice. After the exercise, we conducted interviews with the participants.

From the interview results, we found that our generated painting steps could indeed help beginners to a certain extent. For novice learners, they often struggle with how to start and proceed during the copying practice. Our decomposed steps simplified this process and reduced the cognitive burden on beginners, making it easier for them to follow the step-by-step guidance.

7 Conclusion

In this paper, we proposed a novel method for generating painting processes that differs from traditional stroke-based rendering methods. Our method can preserve realistic strokes while being more consistent with real painting processes. We understand the painting process as an image translation task from the final artwork to a series of intermediate key stages, and we use a single generator with stage labels to generate image sequences. To improve the quality and efficiency of image generation, we employed a Vit-Unet structure with attention mechanism for the generator and validated its effectiveness through attention visualization. We conducted experimental comparisons with stroke-based rendering methods and pixel prediction-based methods, and validated the superiority of our method through qualitative evaluation, expert evaluation, and quantitative evaluation. We conducted experiments on different painting type and different sequence lengths to demonstrate the applicability of our method. The evaluation results showed that our method surpasses existing methods in terms of generated image quality and has strong adaptability.

However, due to the limitation of dataset size, our current method is only suitable for relatively simple content in artworks. When the composition becomes too complex, the generated results may not be as accurate. Nonetheless, these limitations were expected during the development process. Our vision is to assist beginners in alleviating their burden during the self-learning process through an intelligent painting agent. For beginners, they often start with relatively simple content in their artworks, and our method can still provide helpful guidance for them to practice copying in these cases.

Author contributions Z.W. and C.R. wrote the main manuscript text. M.Z. prepared all the figures, F. L. and Z. L. designed and executed the experimental section. All authors reviewed the manuscript.

Funding This work is supported by the National Social Science Fund of China (Grant No. 23BG115) and the Fundamental Research Funds for the Central Universities.

Data availability The data are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Hertzmann, A.: A Survey of Stroke-Based Rendering. Institute of Electrical and Electronics Engineers (2003)
2. Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J.A., Orts-Escolano, S., Garcia-Rodriguez, J., Argyros, A.: A review on deep learning techniques for video prediction. IEEE Trans. Pattern Anal. Mach. Intell. **44**(6), 2806–2826 (2020)
3. Zhao, A., Balakrishnan, G., Lewis, K.M., Durand, F., Guttag, J.V., Dalca, A.V.: Painting many pasts: Synthesizing time lapse videos of paintings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8435–8445 (2020)
4. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. In: Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14, pp. 776–791. Springer (2016)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Adv. Neural Inf. Process. Syst. **7**, 2672–2680 (2014)
6. Li, C., Wand, M.: Precomputed real-time texture synthesis with Markovian generative adversarial networks. In: Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III 14, pp. 702–716. Springer (2016)
7. Zhu, J.-Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)
8. Chuang, Y., Goldman, D., Zheng, K., Curless, B., Salesin, D., Szeliski, R.: Animating pictures with stochastic motion textures. ACM Trans. Graph. **24**(3), 853–860 (2005)
9. Torbunov, D., Huang, Y., Yu, H., Huang, J., Yoo, S., Lin, M., Viren, B., Ren, Y.: UVCGAN: UNet vision transformer cycle-consistent gan for unpaired image-to-image translation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 702–712 (2023)
10. He, B., Gao, F., Ma, D., Shi, B., Duan, L.-Y.: ChipGAN: a generative adversarial network for chinese ink wash painting style transfer. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 1172–1180 (2018)
11. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1395–1403 (2015)
12. Haeberli, P.: Paint by numbers: abstract image representations. In: Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques, pp. 207–214 (1990)
13. Lewis, J.-P.: Texture synthesis for digital painting. In: Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, pp. 245–252 (1984)
14. Reeves, W.T., Blau, R.: Approximate and probabilistic algorithms for shading and rendering structured particle systems. ACM SIGGRAPH Comput. Graph. **19**(3), 313–322 (1985)
15. Hertzmann, A.: Painterly rendering with curved brush strokes of multiple sizes. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 453–460 (1998)
16. Fu, H., Zhou, S., Liu, L., Mitra, N.J.: Animated construction of line drawings. In: Proceedings of the 2011 SIGGRAPH Asia Conference, pp. 1–10 (2011)
17. Tang, F., Dong, W., Meng, Y., Mei, X., Huang, F., Zhang, X., Deussen, O.: Animated construction of Chinese brush paintings. IEEE Trans. Vis. Comput. Graph. **24**(12), 3019–3031 (2017)
18. Frans, K., Cheng, C.-Y.: Unsupervised image to sequence translation with canvas-drawer networks. arXiv preprint arXiv:1809.08340 (2018)
19. Huang, Z., Heng, W., Zhou, S.: Learning to paint with model-based deep reinforcement learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8709–8718 (2019)
20. Singh, J., Zheng, L.: Combining semantic guidance and deep reinforcement learning for generating human level paintings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16387–16396 (2021)
21. Liu, S., Lin, T., He, D., Li, F., Deng, R., Li, X., Ding, E., Wang, H.: Paint transformer: feed forward neural painting with stroke prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6598–6607 (2021)
22. Zou, Z., Shi, T., Qiu, S., Yuan, Y., Shi, Z.: Stylized neural painting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15689–15698 (2021)
23. Litwinowicz, P.: Processing images and video for an impressionist effect. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 407–414 (1997)
24. Ha, D., Eck, D.: A neural representation of sketch drawings. arXiv preprint arXiv:1704.03477 (2017)
25. Ganin, Y., Kulkarni, T., Babuschkin, I., Eslami, S.A., Vinyals, O.: Synthesizing programs for images using reinforced adversarial learning. In: International Conference on Machine Learning, pp. 1666–1675. PMLR (2018)
26. Xie, N., Hachiya, H., Sugiyama, M.: Artist Agent: a reinforcement learning approach to automatic stroke generation in oriental ink painting. IEICE Trans. Inf. Syst. **96**(5), 1134–1144 (2013)
27. Zhou, T., Fang, C., Wang, Z., Yang, J., Kim, B., Chen, Z., Brandt, J., Terzopoulos, D.: Learning to sketch with deep q networks and demonstrated strokes. arXiv preprint arXiv:1810.05977 (2018)
28. Zheng, N., Jiang, Y., Huang, D.: StrokeNet: a neural painting environment. In: International Conference on Learning Representations (2018)
29. Zhao, A., Balakrishnan, G., Lewis, K.M., Durand, F., Guttag, J.V., Dalca, A.V.: Painting many pasts: Synthesizing time lapse videos of paintings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8435–8445 (2020)
30. Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International Conference on Machine Learning, pp. 1747–1756. PMLR (2016)
31. Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. Adv. Neural Inf. Process. Syst. **29** (2016)

32. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16×16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
33. Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollár, P., Girshick, R.: Early convolutions help transformers see better. Adv. Neural Inf. Process. Syst. **34**, 30392–30400 (2021)
34. Guo, J., Han, K., Wu, H., Tang, Y., Chen, X., Wang, Y., Xu, C.: CMT: convolutional neural networks meet vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12175–12185 (2022)
35. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Adv. Neural Inf. Process. Syst. **34**, 8780–8794 (2021)
36. Ming, R.: Chinese brush painting: an academic approach for painting flowers and fish [paperback]
37. Dwight, J.: The Chinese brush painting bible: over 200 motifs with step-by-step illustrated instructions (2011)
38. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, pp. 234–241. Springer (2015)
39. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
40. Anokhin, I., Demochkin, K., Khakhulin, T., Sterkin, G., Lempitsky, V., Korzhenkov, D.: Image generators with conditionally-independent pixel synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14278–14287 (2021)
41. Bachlechner, T., Majumder, B.P., Mao, H., Cottrell, G., McAuley, J.: Rezero is all you need: Fast convergence at large depth. In: Uncertainty in Artificial Intelligence, pp. 1352–1361. PMLR (2021)
42. Wang, Y., Wu, C., Herranz, L., Weijer, J., Gonzalez-Garcia, A., Raducanu, B.: Transferring gans: generating images from limited data. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 218–234 (2018)
43. Noguchi, A., Harada, T.: Image generation from small datasets via batch statistics adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2750–2758 (2019)
44. Zhao, M., Cong, Y., Carin, L.: On leveraging pretrained gans for limited-data generation. In: Proc. ICML, pp. 11340–11351 (2020)
45. Wang, Y., Gonzalez-Garcia, A., Berga, D., Herranz, L., Khan, F.S., Weijer, J.v.d.: Minegan: effective knowledge transfer from gans to target domains with few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9332–9341 (2020)
46. Grigoryev, T., Voynov, A., Babenko, A.: When, why, and which pretrained gans are useful? arXiv preprint arXiv:2202.08937 (2022)
47. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local Nash equilibrium. Adv. Neural Inf. Process. Syst. **30** (2017)
48. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
49. Zhai, G., Min, X.: Perceptual image quality assessment: a survey. Sci. China Inf. Sci. **63**, 1–52 (2020)
50. Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., Chopra, S.: Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604 (2014)