RLMiniStyler: Light-weight RL Style Agent for Arbitrary Sequential Neural Style Generation

Jing Hu[1], **Chengming Feng**[1], **Shu Hu**[2], **Ming-Ching Chang**[3], **Xin Li**[3], **Xi Wu**[1] and **Xin Wang**[3*]
[1]Chengdu University of Information Technology, China
[2]Purdue University, USA
[3]University at Albany, SUNY, USA
jing_hu09@163.com, fengxiaoming520@gmail.com, hu968@purdue.edu, mchang2@albany.edu, xli48@albany.edu, xi.wu@cuit.edu.cn, xwang56@albany.edu

Abstract

Arbitrary style transfer aims to apply the style of any given artistic image to another content image. Still, existing deep learning-based methods often require significant computational costs to generate diverse stylized results. Motivated by this, we propose a novel reinforcement learning-based framework for arbitrary style transfer *RLMiniStyler*. This framework leverages a unified reinforcement learning policy to iteratively guide the style transfer process by exploring and exploiting stylization feedback, generating smooth sequences of stylized results while achieving model lightweight. Furthermore, we introduce an uncertainty-aware multi-task learning strategy that automatically adjusts loss weights to adapt to the content and style balance requirements at different training stages, thereby accelerating model convergence. Through a series of experiments across image various resolutions, we have validated the advantages of *RLMiniStyler* over other state-of-the-art methods in generating high-quality, diverse artistic image sequences at a lower cost. Codes are available at https://github.com/fengxiaoming520/RLMiniStyler.

1 Introduction

The goal of style transfer is to alter the style of an image while preserving its content. Arbitrary style transfer (AST), a key task in this domain, involves the challenge of using a single model to apply any desired artistic style to any given content. Since the pioneering work of Gatys et al.[Gatys *et al.*, 2015] in neural style transfer, subsequent research[An *et al.*, 2021; Wu *et al.*, 2022; Deng *et al.*, 2022; Kwon *et al.*, 2023; Lin *et al.*, 2021; Liu *et al.*, 2021; Park and Lee, 2019; Wang *et al.*, 2023] has made significant strides in enhancing model generalization capabilities, optimizing result quality, and accelerating inference speeds. Due to the varying preferences for the degree of stylization among individuals, precisely controlling the level of stylization to meet diverse needs is a challenging task. Mainstream approaches typically rely on manual tuning of hyperparameters to balance content and style, achieving results with varying degrees of stylization [Gatys *et al.*, 2015; Huang and Belongie, 2017; Liu *et al.*, 2021; Park and Lee, 2019], including the mixing ratio of content features to style features, as well as the individual weightings of content loss and style loss. However, repetitive process of trial and adjustment for achieving suitable weighting parameters, along with the complexity of networks with over 7 million parameters, limits their applicability. To simplify network models, MicroAST [Wang *et al.*, 2023] employs a streamlined model without pre-trained networks for faster inference, and AesFA [Kwon *et al.*, 2023] decomposes images into frequency components for efficient stylization. Even though these lightweight AST methods ensure computational efficiency and can perform style transfer on any style, they also require manual adjustment of hyperparameters and retraining to achieve varying degrees of stylization for specific styles. Importantly, achieving a good balance between content and style through manual hyperparameter tuning is challenging and often results in under-stylization or over-stylization. Hence, it is necessary to develop a new arbitrary style transfer technique that not only facilitates the transfer of any style but also offers a rich array of style degree options for each particular style, relies less on manual hyperparameter tuning, and remains computationally efficient. Recently, RL-NST [Feng *et al.*, 2023] pioneered the application of reinforcement learning to the single style transfer task, achieving precise control over the degree of stylization for one specific style. But it struggles to distinguish between diverse styles, requiring retraining when faced with new styles.

This paper proposed a novel framework named RLMiniStyler that leverages reinforcement learning for controlling the process of the arbitrary style transfer using a unified policy and uncertainty-aware automatic multi-task learning. Leveraging the autonomous exploration inherent in reinforcement learning, our proposed method refines style expression, resulting in a diverse range of stylized results. By integrating a unified policy capable of effectively encoding both content and style images within a single neural network without feature confusion, RLMiniStyler can use one encoder to extract content and style features, thereby reducing model complexity and ensuring a consistent approach to learning and adaptation. Compared to using two encoders, this design is

*Corresponding Author

Figure 1: Illustration of our arbitrary style sequence generation process. **Top Left:** Content and Style Images (5 style examples). **Right:** The sequence number of the results. Content images are progressively stylized with increasing strength along prediction sequences (see the index). Our method allows for easy control over stylization degree, preserving content details in early sequences and synthesizing more style patterns in later sequences, resulting in a user-friendly approach.

more conducive to stable training in the reinforcement learning process. Additionally, the uncertainty-aware automatic multi-task learning allows for dynamic adjustment of learning priorities based on the current performance state. Capable of rapidly generating a diverse array of results with varying degrees of stylization under limited resources, our method offers a richer visual experience beyond a singular result, as shown in Fig. 1.

RLMiniStyler empowers the agent to autonomously learn and explore various style transformation strategies without being constrained by pre-defined rules, resulting in more diverse and innovative stylized images. In summary, we summarize the main contributions of this work as follows:

- We present the first method of arbitrary style transfer based on reinforcement learning. RLMiniStyler provides a stable and flexible control of stylization. It allows flexible control over the degree of stylization by progressively incorporating style patterns into the results over time.
- We propose a unified policy within RLMiniStyler to ensure it remains sufficiently lightweight to operate efficiently in resource-constrained environments, while still maintaining high performance.
- We propose an uncertainty-aware, multi-task learning optimization strategy within our RLMiniStyler to automatically balance style learning and content preservation.
- Through comprehensive experiments on diverse image resolutions, we show the effectiveness of RLMiniStyler in creating high-quality and varied artistic sequences, showcasing its lightweight model advantage and superior or comparable performance across various evaluation metrics relative to both existing lightweight and state-of-the-art style transfer methods.

2 Related Work

Arbitrary Style Transfer (AST). AST aims to enable style transfer using a single trained model, achieving a balance between content and style across various style images without requiring additional training. While recent advancements [Deng *et al.*, 2022; Gu *et al.*, 2018; Hu *et al.*, 2023; Huang and Belongie, 2017; Liu *et al.*, 2021; Park and Lee, 2019; Wang *et al.*, 2020] have been made in this area, many methods have complex models and offer limited diversity in stylization results. Although recently proposed lightweight methods [Wang *et al.*, 2023; Kwon *et al.*, 2023] have employed lightweight models, they necessitate retraining to realize results with varying degrees of stylization for a particular style. Using pruning techniques [Wu *et al.*, 2024] can also achieve lightweight style transfer models, but this approach inevitably leads to a decline in style transfer performance, such as insufficient stylization.

Deep Reinforcement Learning for Neural Style Transfer. The agent in reinforcement learning (RL) focuses on developing optimal strategies through continual exploration and exploitation to maximize cumulative rewards. Handling high-dimensional continuous state and action spaces is particularly challenging for RL agents. Maximum Entropy Reinforcement Learning (MERL) methods [Haarnoja *et al.*, 2017; Haarnoja *et al.*, 2018; Hu *et al.*, 2023; Zhao *et al.*, 2019] demonstrate robust performance in high-dimensional continuous RL tasks by encouraging exploration. However, they may face limitations when applied to generative tasks such as Image-to-Image Translation (I2IT), as they are not inherently designed for generative models. SAEC [Luo *et al.*, 2021], a framework that extends the traditional MERL approach, introduces a generative component to effectively handle I2IT tasks, but the 1D action space limits its effectiveness when attempting to process images with resolution higher than 128×128. Recently, RL-NST [Feng *et al.*, 2023] has successfully extended SAEC to the style transfer task by expanding the action space to 2D and 3D. However, as a single-style transfer method, it requires retraining for each new style, making it unsuitable for the AST task.

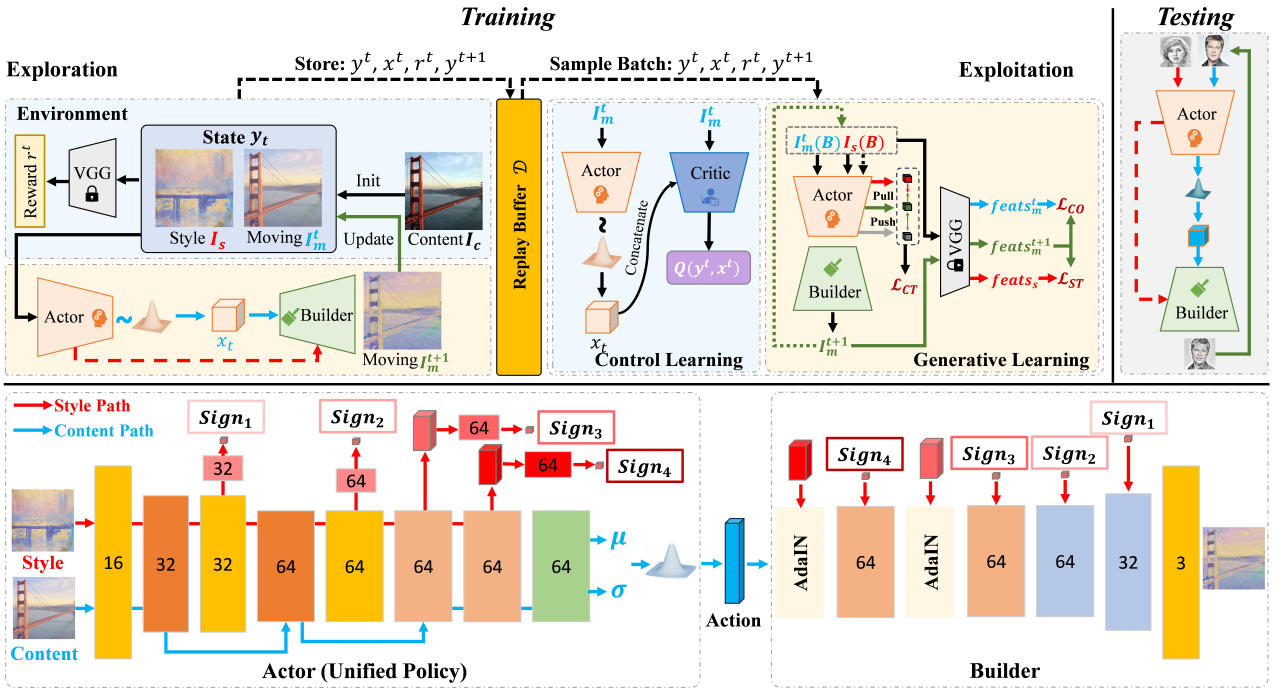


Figure 2: Overview of the RLMiniStyler model. **Top:** The state y_t is initialized with the content image I_c and the style image I_s. Latent-action $\mathbf{x}_t$ is sampled from a high-dimensional Gaussian distribution and is concatenated with the critic's output. It is estimated by the policy P_κ: $\mathbf{x_t} \sim P_\kappa(\mathbf{x}_t|\mathbf{y}_t)$. The predicted moving image I_m^{t+1} is generated by builder B_τ. 'Pull' and 'Push' refer to minimize and maximize the distance between two feature maps, respectively. *Note that the pre-trained VGG network is used only to extract features for calculating rewards and losses during training.* **Bottom:** The structure of the actor and the builder. And $Sign_{1,2,3,4}$ refer to the style signals derived from the calculation of style features. Different colors in the network represent different network architectures, and details of the network structure can be found in supplementary materials.

3 Method

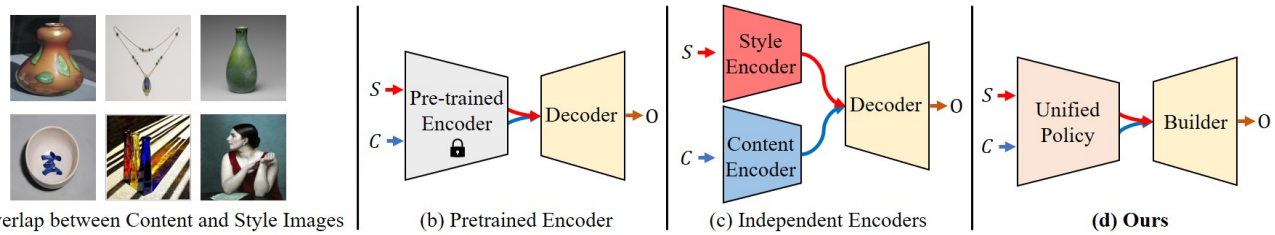
Existing AST methods usually use complex neural networks for one-step inference, limiting stylization diversity and restricting user preferences. Based on MERL framework [Haarnoja *et al.*, 2018], we propose a novel, lightweight RL method for AST to enhance the richness of artistic stylization. In our method, style transfer is regarded as a sequential decision-making problem. In the guidance of a well-defined reward function, our RL agent selects optimal actions at each time step, and generates intermediate stylized results with varying style degrees accordingly. The overview of our method is shown in Fig. 2. Our approach includes three key components: the actor P_κ characterized by parameters κ, the builder B_τ characterized by parameters τ, and the critic Q_δ characterized by parameters δ. The actor serves as the unified policy network responsible for making decisions and style feature extraction based on the current state composed of both the moving image and the style image, the builder acts as the generation network responsible for executing the actor's stylized decisions, and the critic acts as the scoring network responsible for evaluating the actor's decisions. The actor and the critic constitute the RL learning path for style control, while the actor and the builder constitute the generative learning part for generating stylized image. We next describe our method in details.

3.1 Deep Reinforcement NST Framework

In our RL environment Υ, C_D and S_D represent the content dataset and the style dataset, respectively. The **state** $\mathbf{y}^t \in \Upsilon$ is composed of two parts: the moving image I_m^t and the style image $I_s \in S_D$. The moving image I_m^t is initialized using the content image $I_c \in C_D$. The **action** $\mathbf{x}^t$ is determined by the agent based on its observation of the current state. To extract high-level abstract actions from the actor, we establish a model of stochastic latent actions conditioned on the current state (that is, the action $\mathbf{x}^t$ follows the conditional probability $\mathbf{x}^t = P_\kappa(\mathbf{x}^t|\mathbf{y}^t)$). In practice, we employ the reparameterization technique [Kingma and Welling, 2013] to obtain these actions. The moving image I_m^{t+1} in state $\mathbf{y}^{t+1}$ is created by the builder based on $\mathbf{x}^t$ and current state $\mathbf{y}^t$. The **reward** $\mathbf{r}^t$ is derived from the measurement of style discrepancy between the moving image I_m^t and the style image I_s. The reward is inversely proportional to this discrepancy, such that a smaller style difference results in a larger reward.

3.2 Unified Policy for Efficient Style and Content Representation

Existing AST models [Deng *et al.*, 2022; Huang and Belongie, 2017; Johnson *et al.*, 2016; Kwon *et al.*, 2023; Liu *et al.*, 2021; Park and Lee, 2019; Wang *et al.*, 2023] widely employ an encoder-decoder network as backbone architecture. Most of them [Gu *et al.*, 2023; Huang and Belongie, 2017;



(a) Overlap between Content and Style Images (b) Pretrained Encoder (c) Independent Encoders **(d) Ours**

Figure 3: The overlap between style and content images, and the illustration from Pre-trained Encoder to Non-pretrained Encoder. In the figure, 'S', 'C', and 'O' represent the style image, content image, and stylized output, respectively. The images shown in (a) are drawn from both content and style datasets, but the boundaries between them are so blurred that it's challenging to clearly distinguish their original sources. Most existinig style transfer methods typically employ two encoding approaches: one directly utilizes single complex pre-trained encoder (b), while the other trains separate encoders for content and style (c). In contrast, our method adopts a novel approach, using single mini-unified policy for both content and style (d). We detail the specifics of this unified policy as shown in Fig. 2.

Li *et al.*, 2017; Liu *et al.*, 2021; Park and Lee, 2019] use pre-trained VGG [Simonyan and Zisserman, 2014] as the encoder due to its strong capability to capture a wide range of features useful for representing both content and style in images, as shown in Fig. 3(b). But the the complexity of the pre-trained VGG can lead to substantial computational expenses and may introduce unwanted style patterns, like "eyes". An alternative approach [Deng *et al.*, 2022; Wang *et al.*, 2023; Kwon *et al.*, 2023] involves training two encoders to independently process content and style images, treating them as separate distributions, as shown in Fig. 3(c). In this way, more appropriate encoding of content and style images is achieved to avoid incorrect style patterns. However, as shown in Fig. 3(a), there are no clear boundaries between content images and style images in practice. Hence, due to the inherent overlap between these two types of images being overlooked, using two separate encoders complicates the AST model and slows down the training process.

In light of this, RLMiniStyler leverages a unified policy for modeling content and encoding style using a single encoder, as shown in Fig. 3(d). To enable the encoder to have the ability to simultaneously process both content and style images, we draw inspiration from the StyleBank [Chen *et al.*, 2017], which decouples content and style images through explicit style representation. Specifically, we integrated two additional style space dedicated to style encoding at different positions within the encoder's architecture, while the other parts of the encoder were used for general feature extraction, as shown in Fig. 2. This design not only maintains the efficiency of a single encoder but also enhances our control over the subtleties between content and style images by processing style features at different levels. Compared to the design of two separate encoders, the actor, capable of perceiving content and style simultaneously, can make more precise decisions based on the current state. In other words, we can more accurately manipulate the outcome of style transfer to achieve a richer variety of stylistic fusion effects.

3.3 Joint Learning

Our framework employs a joint learning strategy, integrating two mutually coordinated optimization processes: control learning and generative learning. In the control learning, our model learns control policies, while in the generative learning, it learns stylized image generation. Training alternates between these two parts. The generative learning consists of the Actor P_κ and the Builder B_τ, and the design of the loss function helps in effectively propagating gradient information between the Actor P_κ and the Builder B_τ. The control learning consists of the Actor P_κ and the Critic Q_δ, and the training of the Actor can be conducted jointly through the control learning and the generative learning to ensure rapid and stable convergence. Algorithm in appendix describes the RLMiniStyler algorithm. All parameters are optimized based on the samples from replay pool $\mathcal{D}$.

Control Learning

In the control learning, adhering to the MERL framework [Haarnoja *et al.*, 2018], we iteratively refine a stochastic policy P_κ utilizing reward signals $\mathbf{r}^t$ and soft Q-values $Q_\delta(\mathbf{y}^t, \mathbf{x}^t)$. Here, the action $\mathbf{x}^t$ is generated by the actor in response to the current state $\mathbf{y}^t$, following the policy $P_\kappa(\mathbf{x}^t|\mathbf{y}^t)$. The soft Q-function $Q_\delta(\mathbf{y}^t, \mathbf{x}^t)$, computed by the critic network, provides an estimation of the expected cumulative reward for the state-action pair $(\mathbf{y}^t, \mathbf{x}^t)$ under the current policy. During the evaluation phase, we guide the improvement of the stochastic policy through the minimization of the soft Bellman residual, defined as:

$$J_Q(\delta) = \mathbb{E}_{(\mathbf{y}^t, \mathbf{x}^t, r^t, \mathbf{y}^{t+1}) \sim \mathcal{D}} \Big[\frac{1}{2} \Big(Q_\delta(\mathbf{y}^t, \mathbf{x}^t) - \big(r^t + \gamma \mathbb{E}_{\mathbf{y}^{t+1}} \big[V_{\bar{\delta}}(\mathbf{y}^{t+1}) \big] \big) \Big)^2 \Big], \tag{1}$$

where $\mathcal{D}$ is the replay pool and $V_{\bar{\delta}}(\mathbf{y}^t) = \mathbb{E}_{\mathbf{x}^t \sim P_\kappa}[Q_{\bar{\delta}}(\mathbf{y}^t, \mathbf{x}^t) - \alpha \log P_\kappa(\mathbf{x}^t|\mathbf{y}^t)]$. We set the reward signal $\mathbf{r}^t$ as the negative of the style loss to ensure that the agent learns to control the stylization process. Given our objective is style-related, such a choice of reward function is reasonable. Specifically, the style loss serves as a simple and effective means to assess the similarity between the stylized output and the target style image, hence we set the reward function as the negative of the style loss $\mathbf{r}^t = -\mathcal{L}_{ST}$, where the detailed definition of $\mathcal{L}_{ST}$ is shown in Eq. (5).

The target critic network, denoted as $Q_{\bar{\delta}}$, plays a crucial role in stabilizing the training process. The parameters of this network, represented as $\bar{\delta}$, are determined by calculating the exponential moving average of the parameters from the critic

network [Lillicrap *et al.*, 2015]: $\bar{\delta} \rightarrow \omega\delta + (1-\omega)\bar{\delta}$, with hyperparameter $\omega \in [0,1]$. To optimize $J_Q(\delta)$, we use the gradient descent with respect to parameters δ as:

$$\begin{aligned}\delta \leftarrow & \delta - \rho_Q \nabla_\delta Q_\delta(\mathbf{y}^t, \mathbf{x}^t)\Big(Q_\delta(\mathbf{y}^t, \mathbf{x}^t) - r^t \\ & - \gamma\left[Q_{\bar{\delta}}(\mathbf{y}^{t+1}, \mathbf{x}^{t+1}) - \alpha \log P_\kappa(\mathbf{x}^{t+1}|\mathbf{y}^{t+1})\right]\Big),\end{aligned} \tag{2}$$

where ρ_Q is the learning rate. In the RL framework, the critic evaluates the actions taken by the actor, which in turn influences the policy decisions of the actor. Consequently, the following objective can be applied to minimize the Kullback-Leibler (KL) divergence between the policy induced by the actor and a Boltzmann distribution, as determined by the Q-function:

$$\begin{aligned}J_P(\kappa) =& \mathbb{E}_{\mathbf{y}^t \sim \mathcal{D}}\left[\mathbb{E}_{\mathbf{x}^t \sim P_\kappa}\left[\alpha \log(P_\kappa(\mathbf{x}^t|\mathbf{y}^t)) - Q_\delta(\mathbf{y}^t, \mathbf{x}^t)\right]\right] \\ =& \mathbb{E}_{\mathbf{y}^t \sim \mathcal{D}, \mathbf{n}^t \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}\left[\alpha \log(P_\kappa(f_P(\mathbf{n}^t, \mathbf{y}^t)|\mathbf{y}^t)) \right. \\ & \left. - Q_\delta(\mathbf{y}^t, f_P(\mathbf{n}^t, \mathbf{y}^t))\right].\end{aligned} \tag{3}$$

The last equation holds because $\mathbf{x}^t$ can be evaluated by $f_P(\mathbf{n}^t, \mathbf{y}^t)$, where $\mathbf{n}^t$ is a noise vector sampled from a 3D Gaussian distribution with mean $\boldsymbol{\mu} = 0$ and standard deviation $\boldsymbol{\Sigma} = 1$. Note that hyperparameter α can be automatically adjusted by using the method proposed in [Haarnoja *et al.*, 2018]. Similarly, we apply the gradient descent method with the learning rate ρ_κ to optimize parameters as:

$$\begin{aligned}\kappa \leftarrow & \kappa - \rho_\kappa\Big(\nabla_\kappa \alpha \log(P_\kappa(\mathbf{x}^t|\mathbf{y}^t)) + \big(\nabla_{\mathbf{x}^t} \alpha \log(P_\kappa(\mathbf{x}^t|\mathbf{y}^t)) \\ & - \nabla_{\mathbf{x}^t} Q_\delta(\mathbf{y}^t, \mathbf{x}^t)\big)\nabla_\kappa f_\kappa(\mathbf{n}^t, \mathbf{y}^t)\Big).\end{aligned}$$

Generative Learning

Generative learning, through specific training strategies, enhances our model's generation capability in style transfer, ensuring the production of high-quality stylized image results. We assess the similarity between the stylized image I_m^{t+1} and the input images by comparing their high-level features. Specifically, we employ a pre-trained VGG [Simonyan and Zisserman, 2014] as a feature extraction backbone ϕ to extract features independently from the moving image I_m^t, the style image I_s, and the stylized image I_m^{t+1}. We calculate the content loss $\mathcal{L}_{CO}$ by comparing the semantic similarity between I_m^{t+1} and I_m^t and the style loss $\mathcal{L}_{ST}$ by comparing the style similarity between I_m^{t+1} and I_s.

Content Loss. We evaluate how closely the stylized image resembles the content image, by maximizing perceptual similarity using the widely adopted perceptual loss [Johnson *et al.*, 2016]. Let $\phi^{(j)}$ denote the activation of the j-th layer, producing a feature map with dimensions $C^j \times H^j \times W^j$, where C^j, H^j, and W^j represent the number of channels, height, and width of the feature map, respectively. The content loss $\mathcal{L}_{CO}$ is calculated by:

$$\mathcal{L}_{CO}(I_m^{t+1}, I_m^t) = \frac{1}{C^j H^j W^j} \parallel \phi^{(j)}(I_m^{t+1}) - \phi^{(j)}(I_m^t) \parallel_2^2 . \tag{4}$$

Style Loss. The style loss $\mathcal{L}_{ST}$ estimates the style deviations between the stylized image I_m^{t+1} and style image I_s. Let J represent the layer number of the network ϕ. It calculates statistical measures of μ and standard deviation σ to penalize I_m^{t+1}, inspired by [Huang and Belongie, 2017]:

$$\begin{aligned}\mathcal{L}_{ST}(I_m^{t+1}, I_s) = & \sum_{j=1}^{J} \|\mu(\phi^{(j)}(I_m^{t+1})) - \mu(\phi^{(j)}(I_s))\|_2^2 \\ & + \sum_{j=1}^{J} \|\sigma(\phi^{(j)}(I_m^t)) - \sigma(\phi^{(j)}(I_s))\|_2^2 .\end{aligned} \tag{5}$$

Hierarchical Style Representation Contrastive Loss (HSRCL). Recent studies [Chen *et al.*, 2021; Wang *et al.*, 2023] have demonstrated that the lightweight network struggles to fully capture and express the style features of style images in a single inference process, and incorporating a contrastive learning loss can mitigate this issue. For instance, MicroAST [Wang *et al.*, 2023] employs a style signal contrastive learning loss to deal with this, but it predominantly relys on deep-layer features for contrastive learning, overlooking the contributions of shallow-layer features to the overall style representation. To this end, we introduce a novel hierarchical style representation contrastive loss, which integrates contrastive learning between deep and shallow feature representations, so as to enhance the style representation. More specifically, when sampling a batch of data from the replay buffer $\mathcal{D}$, we construct both positive and negative sets for each sample's deep and shallow features. And the feature contrastive loss respectively computed from the deep features and shallow features are combined to create a hierarchical style contrastive loss function $\mathcal{L}_{CT}$, which is defined as:

$$\mathcal{L}_{CT} = \sum_{i=1}^{N}\sum_{k=1}^{K} \frac{\parallel P_\kappa(I_m)^{(i,k)} - P_\kappa(I_s)^{(i,k)} \parallel_2^2}{\sum_{j\neq i}^{N} \|P_\kappa(I_m)^{(i,k)} - P_\kappa(I_s)^{(j,k)} \|_2^2}, \tag{6}$$

where K represents the number of feature layers in the unified policy network, N represents the batch size. The batch comprises N states $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$. Each state $\mathbf{y}_i \in \mathbf{Y}$ consists of a moving image I_m and a style image I_s. For each I_m, we consider the style image I_s from $\mathbf{y}_i$ as a positive sample and the style images I_s from other $\mathbf{y}_j$ as negative samples, where $j \neq i$.

Uncertainty-aware Automatic Multi-task Learning. A common way to enhance the quality of style transfer involves quantifying the semantic similarity to the content image and the style similarity to the style image through content and style loss functions, along with auxiliary loss functions, such as adversarial loss. But the weights for these loss functions are usually heuristically selected before training and remain unchanged throughout the training process, which is not sufficient enough to handle images with different style and content.

To this end, as inspired by [Kendall *et al.*, 2018], we propose to use a multi-task learning framework that treats content learning, style learning, and contrastive learning as distinct but interconnected tasks. Using *homoscedastic uncertainty*, we dynamically adjust the loss weights of each task derived from a principled probabilistic model, achieving a balanced optimization objective that adapts throughout training. Unlike traditional methods requiring manual tuning of

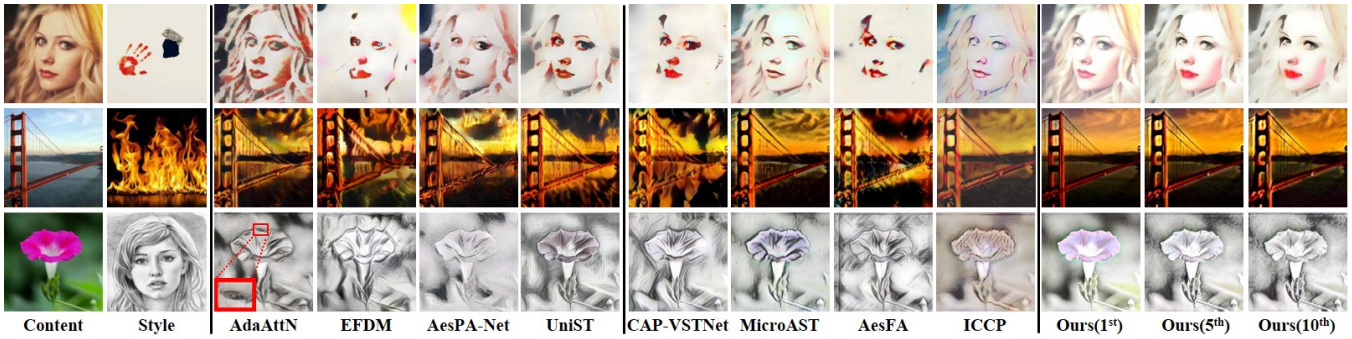


Figure 4: Qualitative Comparison with several AST algorithms in 256 pixel resolution. The 1st and 2nd columns present the content and style images, respectively. The subsequent four columns display the results from the current SOTA AST methods. The three columns immediately following showcase the results of the lightweight methods. Lastly, we present the sequential stylization results generated by our method, including sequences 1st, 5th, and 10th.

	Model Complexity		256×256 Pixel Resolution					512×512 Pixel Resolution				
Method	Params (1e6) ↓	Storage (MB) ↓	Content Loss ↓	SSIM ↑	Style Loss ↓	Time(s) ↓	Pref.(%) ↑	Content Loss ↓	SSIM ↑	Style Loss ↓	Time(s) ↓	Pref.(%) ↑
AdaAttN(2021)	13.6299	128.4020	3.0668	0.4987	0.6027	0.0117	12.00	2.4280	0.5341	0.5516	0.1032	10.67
EFDM(2022)	7.0110	26.7000	3.6671	0.3165	0.4233	0.0073	4.67	2.9439	0.3788	0.3268	0.0079	6.67
CAP-VSTNet(2023)	4.0899	15.6719	3.5984	0.4501	**0.3151**	0.0423	7.33	2.7459	0.4864	**0.2234**	0.1209	7.33
AesPA-Net(2023)	23.6737	92.3340	2.6822	0.4504	0.8266	0.3110	5.67	2.0412	0.5195	0.8756	0.4628	10.00
UniST(2023)	65.2545	302.9424	2.8888	0.4305	0.4137	0.0295	9.67	2.4080	0.4567	0.2952	0.0347	7.33
MicroAST(2023)	0.4720	1.8570	2.6382	0.4753	0.6247	**0.0066**	9.00	2.0349	0.5034	0.4960	**0.0069**	7.67
AesFA(2024)	3.2208	12.3100	3.3734	0.4115	0.3945	0.0167	8.33	2.7624	0.4466	0.3024	0.0187	7.00
ICCP(2024)	**0.0790**	**0.3447**	2.7964	0.5152	1.3025	0.0087	3.33	2.1236	0.5559	1.1415	0.0098	4.00
Ours(1^{st})	0.3712	1.4750	**1.1684**	**0.6444**	1.0487	0.0094	**15.00**	**0.9292**	**0.6517**	0.8927	0.0150	**17.33**
Ours(5^{th})	0.3712	1.4750	**2.1508**	**0.5509**	0.6974	0.0336	**20.33**	**1.6491**	**0.5711**	0.5528	0.0852	**13.67**
Ours(10^{th})	0.3712	1.4750	2.7518	0.4898	0.6209	0.0631	4.67	2.0871	0.5191	0.4892	0.1733	8.33

Table 1: Quantitative Comparison of Model Complexity and Performance with Various AST Algorithms at Standard Resolutions. 'Pref.' represents user preferences from our user study.

loss weights, our approach learns the relative importance of each task's loss function directly from the data. This not only simplifies the training process but also enables the dynamic modulation of the content loss and the style loss ratios to find the optimal solution.

Let λ_c, λ_s, λ_{ct} denote the loss weights for content loss, style loss, and contrastive loss, respectively. These weights can adapt based on homoscedastic uncertainty σ_1^2, σ_2^2, and σ_3^2, reflecting the noise level or task confidence. And they are inversely proportional to the noise parameters. The final loss is:

$$\mathcal{L}_{final}(\kappa, \tau, \lambda_c, \lambda_s, \lambda_{ct}) = \lambda_c \mathcal{L}_{CO} + \lambda_s \mathcal{L}_{ST} + \lambda_{ct} \mathcal{L}_{CT} + \epsilon,$$
$$\lambda_c = \frac{1}{\sigma_1^2}, \lambda_s = \frac{1}{\sigma_2^2}, \lambda_{ct} = \frac{1}{\sigma_3^2}, \epsilon = \log(\sigma_1 \sigma_2 \sigma_3), \tag{7}$$

where $\log(\sigma_1 \sigma_2 \sigma_3)$ acts as a regularizer to prevent excessive increase in noise. Lastly, we employ a gradient descent method with the learning rate η to update the Actor and Builder parameters (κ and τ) as well as $\sigma_i (i = 1, 2, 3)$:

$$\kappa \leftarrow \kappa - \eta_\kappa \nabla_\kappa \mathcal{L}_{final}, \quad \tau \leftarrow \tau - \eta_\tau \nabla_\tau \mathcal{L}_{final}, \tag{8}$$

$$\sigma_i \leftarrow \sigma_i - \eta_{\sigma_i} \nabla_{\sigma_i} \mathcal{L}_{final}. \tag{9}$$

4 Experiments

4.1 Experimental Setup

Datasets and evaluation metric: Like most AST methods [Deng *et al.*, 2022; Huang and Belongie, 2017; Liu *et al.*, 2021; Park and Lee, 2019; Wang *et al.*, 2023], we utilize the MS-COCO dataset [Lin *et al.*, 2014] for content and the WikiArt dataset [Phillips and Mackintosh, 2011] for style. During training, images are first scaled to 512×512 pixels, then randomly cropped to 256×256, while testing can handle any input size. Following MicroAST [Wang *et al.*, 2023], we assess all algorithms across seven aspects: *visual effect, inference time, parameter count, content loss, style loss, SSIM [Wang* et al.*, 2004], and storage space*.

Implementation details: We use the Adam optimizer [Kingma and Ba, 2014] with a learning rate 2e-4, the batch size in the environment set to 1, and the batch size sampled from the replay buffer set to 8. All experiments are conducted on a single NVIDIA Tesla P100 (16GB) GPU.

4.2 Comparisons with Prior Arts

Baselines: We compare our method with four light-weight AST methods: CAP-VSTNet [Wen *et al.*, 2023], MicroAST [Wang *et al.*, 2023], AesFA [Kwon *et al.*, 2023], and ICCP [Wu *et al.*, 2024], as well as four state-of-the-art AST methods: AdaAttN [Liu *et al.*, 2021], EFDM [Zhang *et al.*, 2022], AesPA-Net [Hong *et al.*, 2023] and UniST [Gu *et al.*, 2023]. All codes used in the experiment are sourced from their respective public repositories, and we use the default settings provided.

Qualitative comparison: We visually compare our method with all baseline methods in Fig. 4. AdaAttN shows a repetitive style pattern resembling the eyes (the third row), while EFDM and CAP-VSTNet lose a significant semantic

and structural content (first and second rows). AesPA-Net produces inconsistent results, especially in the eye area (first row). UniST, MicroAST and ICCP show insufficient stylization (third row), and AesFA has severe boundary artifacts (third row). In contrast, our approach generates a sequence of results with increasing stylization levels while maintaining coherent content structure. Our method has also been compared with lightweight baselines at higher resolutions (512, 4K). Due to space constraints, the detailed comparison results are included in the supplementary materials.

Quantitative comparison: Table 1 provides a comprehensive comparison between our approach and baseline models. Our method consistently achieves competitive scores in content loss, SSIM, style loss, and inference time, demonstrating its efficiency and effectiveness in producing outputs that balance style expression with content preservation. As the sequence progresses, our method enhances style richness while maintaining content fidelity. In terms of model complexity, our model outperforms the minimally pruned model in performance and features a lower parameter count and reduced complexity compared to the smallest non-pruned model. Similarly, more comparative results with lightweight methods at high resolutions (1K, 2K, 4K) are included in the supplementary materials. Additionally, it is the first AST method capable of automatically controlling the degree of stylization on images ranging from 256 to 4K resolution.

4.3 Ablation Study

With and without RL: We discussed the effectiveness of RL in style control. In Fig. 5, without RL, the Actor-Builder (AB) in (a) initially preserves semantic information. But as shown in (e), at sequence 10, notable content information is lost. In contrast, our method in (d) produces smoother and clearer stylized images from the start, and stably maintains high-quality results throughout the sequence in (h) at sequence 10. This consistent performance highlights the significant enhancement of RL provides to DL-based AST models.

Automatic multi-task learning (AML) vs. manual settings: We manually tuned the loss weights in our method, based on the settings of MicroAST and empirical adjustments. Specifically, we set the content loss weight $\lambda_c = 1$, the style loss weight $\lambda_s = 3$, and the HSRCL loss weight $\lambda_{ct} = 3$, while keeping all other settings unchanged. As shown in Fig. 5 , compared to the fixed loss weight method in (b,f), our approach using AML demonstrates superior content preservation in both sequence 1 in (d) and sequence 10 in (h). Our study indicates that AML significantly enhances model performance and accelerates network convergence.

Hierarchical style representation contrastive loss (HSRCL) vs. style signal contrastive loss: We investigated the effectiveness of HSRCL by comparing with the deep-feature based contrastive loss proposed in MicroAST [Wang *et al.*, 2023]. As shown in Fig. 5, for sequence 1, using only deep features for contrastive learning (c) exhibits less of style diversity as compared with the result in (d). Comparing with sequence 10 in (h), there is a noticeable decline in (g) in terms of content affinity due to incoherent style expression. This experiment demonstrates that HSRCL significantly enhances the model's capacity in style expression.

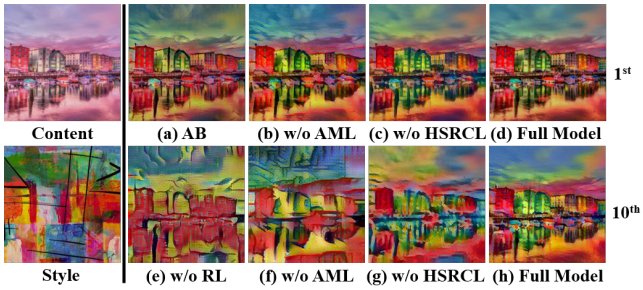


Figure 5: Ablation Study Results Comparing the Impact of RL, Automatic Multi-task Learning (AML), and Hierarchical Style Representation Contrastive Loss (HSRCL) vs. Style Signal Contrastive Loss on Style Transfer Performance. The visual comparison underscores the contributions of RL, AML, and HSRCL to the fidelity and stability of stylized results across sequences. More results are presented in the supplementary materials.

4.4 User Study

We conducted user study on nine different methods. We recruited 30 participants representing a diverse range of ages, genders, and professional backgrounds. Each participant was randomly presented with 20 ballots: 10 at a 256×256 resolution and 10 at a 512×512 resolution. Each ballot included the content image, the style image, and 11 randomly shuffled stylized results. Note that since our method produces sequential results, we present the outcomes at the first, fifth, and tenth sequences. We collected 300 valid ballots for each resolution, and the detailed results are shown in Table 1. It is evident that the majority of users prefer the stylized results generated by our method. In other words, although the assessment of stylized results is inherently subjective, our lightweight style transfer agent is designed to generate a diverse array of sequential outputs tailored to meet the varying preferences and requirements of different users.

5 Conclusion

In this paper, we introduce a lightweight Arbitrary Style Transfer method using reinforcement learning. Our approach employs a unified policy to simultaneously learn from content and style images through a coherent encoding and decoding process, thereby more effectively capturing the distinguishing information between content and style. Our novel hierarchical style representation contrastive loss differentiates between shallow and deep style representations, enriching the expressiveness of the style transfer. Furthermore, Automatic Multi-task Learning facilitates training across various stages, accelerating the convergence of the model. Extensive experiments have demonstrated that our method not only generates visually harmonious and aesthetically pleasing artistic images across different resolutions but also produces a diverse range of stylized outcomes. The simplicity and effectiveness of our approach are expected to accelerate the miniaturization of style transfer networks. Although this work has successfully achieved miniaturization and diversification in arbitrary style transfer for images, the challenge remains in applying it to video, which involves temporal processing. Our future goal is to extend our approach to video arbitrary style transfer.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 42375148, Sichuan province Key Technology Research and Development project under Grants (Nos.2024ZHCG0190, No. 2024ZHCG0176), CUIT Science and Technology Innovation Capacity Enhancement Program project under Grant KYQN202305.

Contribution Statement

The contributions of Jing Hu and Chengming Feng to this paper were equal.

References

[An *et al.*, 2021] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 862–871, 2021.

[Chen *et al.*, 2017] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1897–1906, 2017.

[Chen *et al.*, 2021] Haibo Chen, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, Dongming Lu, et al. Artistic style transfer with internal-external learning and contrastive learning. *Advances in Neural Information Processing Systems*, 34:26561–26573, 2021.

[Deng *et al.*, 2022] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11326–11336, 2022.

[Feng *et al.*, 2023] Chengming Feng, Jing Hu, Xin Wang, Shu Hu, Bin Zhu, Xi Wu, Hongtu Zhu, and Siwei Lyu. Controlling neural style transfer with deep reinforcement learning. *arXiv preprint arXiv:2310.00405*, 2023.

[Gatys *et al.*, 2015] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[Gu *et al.*, 2018] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. Arbitrary style transfer with deep feature reshuffle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8222–8231, 2018.

[Gu *et al.*, 2023] Bohai Gu, Heng Fan, and Libo Zhang. Two birds, one stone: A unified framework for joint learning of image and video style transfers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23545–23554, 2023.

[Haarnoja *et al.*, 2017] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *ICML*, pages 1352–1361. PMLR, 2017.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[Hong *et al.*, 2023] Kibeom Hong, Seogkyu Jeon, Junsoo Lee, Namhyuk Ahn, Kunhee Kim, Pilhyeon Lee, Daesik Kim, Youngjung Uh, and Hyeran Byun. Aespa-net: Aesthetic pattern-aware style transfer networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22758–22767, 2023.

[Hu *et al.*, 2023] Jing Hu, Zhikun Shuai, Xin Wang, Shu Hu, Shanhui Sun, Siwei Lyu, and Xi Wu. Attention guided policy optimization for 3d medical image registration. *IEEE Access*, 2023.

[Huang and Belongie, 2017] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li 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*, pages 694–711. Springer, 2016.

[Kendall *et al.*, 2018] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Kwon *et al.*, 2023] Joonwoo Kwon, Sooyoung Kim, Yuewei Lin, Shinjae Yoo, and Jiook Cha. Aesfa: An aesthetic feature-aware arbitrary neural style transfer. *arXiv preprint arXiv:2312.05928*, 2023.

[Li *et al.*, 2017] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30, 2017.

[Lillicrap *et al.*, 2015] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014:*

13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014.

[Lin *et al.*, 2021] Tianwei Lin, Zhuoqi Ma, Fu Li, Dongliang He, Xin Li, Errui Ding, Nannan Wang, Jie Li, and Xinbo Gao. Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5141–5150, 2021.

[Liu *et al.*, 2021] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6649–6658, 2021.

[Luo *et al.*, 2021] Ziwei Luo, Jing Hu, Xin Wang, Siwei Lyu, Bin Kong, Youbing Yin, Qi Song, and Xi Wu. Stochastic actor-executor-critic for image-to-image translation. *arXiv preprint arXiv:2112.07403*, 2021.

[Park and Lee, 2019] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5880–5888, 2019.

[Phillips and Mackintosh, 2011] Fred Phillips and Brandy Mackintosh. Wiki art gallery, inc.: A case for critical thinking. *Issues in Accounting Education*, 26(3):593–608, 2011.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Wang *et al.*, 2004] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[Wang *et al.*, 2020] Zhizhong Wang, Lei Zhao, Haibo Chen, Lihong Qiu, Qihang Mo, Sihuan Lin, Wei Xing, and Dongming Lu. Diversified arbitrary style transfer via deep feature perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7789–7798, 2020.

[Wang *et al.*, 2023] Zhizhong Wang, Lei Zhao, Zhiwen Zuo, Ailin Li, Haibo Chen, Wei Xing, and Dongming Lu. Microast: Towards super-fast ultra-resolution arbitrary style transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2742–2750, 2023.

[Wen *et al.*, 2023] Linfeng Wen, Chengying Gao, and Changqing Zou. Cap-vstnet: Content affinity preserved versatile style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18300–18309, 2023.

[Wu *et al.*, 2022] Zhijie Wu, Chunjin Song, Guanxiong Chen, Sheng Guo, and Weilin Huang. Completeness and coherence learning for fast arbitrary style transfer. *Transactions on Machine Learning Research*, 2022.

[Wu *et al.*, 2024] Kexin Wu, Fan Tang, Ning Liu, Oliver Deussen, Weiming Dong, Tong-Yee Lee, et al. Lighting image/video style transfer methods by iterative channel pruning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3800–3804. IEEE, 2024.

[Zhang *et al.*, 2022] Yabin Zhang, Minghan Li, Ruihuang Li, Kui Jia, and Lei Zhang. Exact feature distribution matching for arbitrary style transfer and domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8035–8045, 2022.

[Zhao *et al.*, 2019] Xujiang Zhao, Shu Hu, Jin-Hee Cho, and Feng Chen. Uncertainty-based decision making using deep reinforcement learning. In *2019 22th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2019.