# User-Controllable Arbitrary Style Transfer via Entropy Regularization

**Jiaxin Cheng**[1,*], **Yue Wu**[2], **Ayush Jaiswal**[2], **Xu Zhang**[2], **Pradeep Natarajan**[2], **Prem Natarajan**[2]

[1] USC Information Sciences Institute
[2] Amazon Alexa Natural Understanding
chengjia@isi.edu; {wuayue,ayujaisw,xzhnamz,natarap,premknat}@amazon.com

## Abstract

Ensuring the overall end-user experience is a challenging task in arbitrary style transfer (AST) due to the subjective nature of style transfer quality. A good practice is to provide users many instead of one AST result. However, existing approaches require to run multiple AST models or inference a diversified AST (DAST) solution multiple times, and thus they are either slow in speed or limited in diversity. In this paper, we propose a novel solution ensuring both efficiency and diversity for generating multiple user-controllable AST results by systematically modulating AST behavior at run-time. We begin with reformulating three prominent AST methods into a unified assign-and-mix problem and discover that the entropies of their assignment matrices exhibit a large variance. We then solve the unified problem in an optimal transport framework using the *Sinkhorn-Knopp* algorithm with a user input $\varepsilon$ to control the said entropy and thus modulate stylization. Empirical results demonstrate the superiority of the proposed solution, with speed and stylization quality comparable to or better than existing AST and significantly more diverse than previous DAST works. Code is available at https://github.com/cplusx/eps-Assign-and-Mix.

## Introduction

Neural style transfer (NST) refers to the process of rendering a pastiche image $P$ from a content image $C$ and a style image $S$ through a deep neural network (DNN), such that the resulting $P$ displays the content of $C$ in the style of $S$. Early NST methods like (Gatys, Ecker, and Bethge 2016) were developed for a single pair of $S$ and $C$, requiring the model to be re-trained for every such pair. These were followed by a series of methods that support a single $S$ but arbitrary $C$ (Johnson, Alahi, and Fei-Fei 2016; Ulyanov et al. 2016; Ulyanov, Vedaldi, and Lempitsky 2017; Li and Wand 2016), and those developed for a static set of $S$ but arbitrary $C$ (Ghiasi et al. 2017; Chen et al. 2017; Dumoulin, Shlens, and Kudlur 2016; Li et al. 2017a; Zhang and Dana 2018; Kotovenko et al. 2019a; Sanakoyeu et al. 2018). In recent years, arbitrary style transfer (AST) methods have become the most popular approaches for NST as they can be utilized for arbitrary, including unseen, $S$ and $C$. Contemporary research in AST includes deformable style transfer (Kim et al. 2020; Liu, Yang, and Hall 2021), brush-based style transfer (Kotovenko et al. 2021), better style transfer loss functions (Risser, Wilmot, and Barnes 2017; Sanakoyeu et al. 2018; Cheng et al. 2021; Wu et al. 2022), transformer-based AST (Deng et al. 2021; Wu et al. 2021; Cheng et al. 2019; Deng et al. 2022), text-driven AST (Kwon and Ye 2022) *etc*.

Despite the existence of a large number of AST approaches, with each generating pleasing stylized images, no method produces results that are single-handedly considered the best among users. This is because style transfer is an art form and human perception of art is subjective. Indeed, a review of user preference studies in previous works (Li et al. 2019; Park and Lee 2019; Liu et al. 2021; Lin et al. 2021; Yao et al. 2019) provides the following insights: (1) the most preferred AST method typically achieves less than two-thirds of the user votes, (2) the difference between the top two most preferred methods commonly ranges from 10 to 30 percentage points, and (3) even the method with the least votes usually receives at least 5% approval.

To increase stylization diversity and improve user experience, one solution is to run multiple AST models simultaneously, referred to as *Multi-AST-*$K$, where $K$ is the number of AST methods. However, despite achieving good diversity and style transfer quality, this approach demands significant computational resources and training efforts. Alternatively, ArtIns (Xie et al. 2022) generates diverse AST results by decomposing style into many components and linearly combining them during sampling, but it does not preserve style faithfully (See appendix for comparison.). In contrast, diversified style transfer techniques (Li et al. 2017a; Ulyanov, Vedaldi, and Lempitsky 2017; Wang et al. 2020; Li et al. 2020; Wang et al. 2021) generate multiple stylized images for the same input through random noise in the AST process (Li et al. 2017a; Wang et al. 2020; Li et al. 2020). However, the output diversity of such methods is limited as the noise must be small enough to avoid low-quality results and is challenging to control in practice. For example, PWCT (Wang et al. 2020) introduces random noise perturbation in the WCT (Li et al. 2017b) AST process, and SP (Li et al. 2020) designs random style feature permutation for an AST process like AdaIN (Huang and Belongie 2017).

In this paper, we propose a novel $\varepsilon$-Assign-and-Mix ($\varepsilon$-AM ) formulation to address the preference diversity issue in AST by allowing a user input $\varepsilon$ at run-time to systematically

[*]This work was done during Jiaxin Cheng's internship at Amazon.

| Framework | Multi-AST-$K$ | DAST | Ours |
|---|---|---|---|
| **Param.** | AST-Index | Random noise | $\varepsilon$ param. |
| **Controllable?** | Yes | No | Yes |
| **#Models** | $K$ | 1 | 1 |
| **#Outputs** | $K$ | $\infty$ | $\infty$ |
| **Diversity** | High | Low-Med | High |
| **Quality** | High | Low-Med | High |
| **Speed** | Slow | Fast | Fast |

Table 1: Comparisons of style transfer frameworks for increasing stylization diversity.

change the behavior of the assignment matrix that defines the correspondence between content and style features, and consequently, the nature of the generated stylized images.

**Our work makes three major contributions**: (1) we show that popular AST methods, *e.g*. AdaIN (Huang and Belongie 2017), DFR (Gu et al. 2018) and SANet (Park and Lee 2019), can all be unified into our generalized assign-and-mix formulation, (2) we propose a novel $\varepsilon$-AM framework to allow users to systematically modulate the assignment matrix and subsequent stylization at run-time, and (3) we propose framing the problem as optimal transport, which can be solved efficiently using the *Sinkhorn-Knopp* algorithm (Cuturi 2013) via entropy regularization. Results of extensive experiments show that our framework can explicitly and actively control the AST behavior, and produce high-quality and diverse outputs with high inference speed to please a wide range of users. Fig. 1 shows the overview of our solution and **??** highlights its similarities and differences with the existing ones.

## $\varepsilon$-Assign-and-Mix

In this section, we first review the core transfer function used in AST. We then revisit three well-known AST approaches and show that a large family of existing AST approaches can be viewed as solving a generalized two-step Assign-and-Mix (AM ) problem. We later present our analysis that shows that a major difference among different AST methods is the entropy of the assignment matrix that defines style-content feature correspondence. Based on these findings, we propose to solve the proposed AST problem using the general AM formulation. By introducing a new input parameter $\varepsilon$ to control the entropy of the assignment matrix explicitly, our proposed $\varepsilon$-AM approach can output diverse AST results at run-time. We explain the assignment step and the mix step and show that the problem can be efficiently solved by the *Sinkhorn-Knopp* algorithm (Cuturi 2013).

### AST Transfer Function

An essential task in AST is to construct a transfer function $t(\cdot)$ that maps content feature $f_C$ and style features $f_S$ to a pastiche feature $f_P$ carrying content and style information.

$$f_P = t(f_C, f_S) \tag{1}$$

Thus, we broadly classify AST approaches into three families, namely: (1) content feature transform, (2) style feature assignment, and (3) others. The family of content feature transform methods typically obtains $f_P$ via a heuristic or learnable transform of the content feature $f_C$ with the style feature $f_S$ as the reference for controlling the implicit transfer process (Li et al. 2019; Guo et al. 2018), or through explicit statistical matching (Huang and Belongie 2017; Li et al. 2017b,c; Huo et al. 2021) between $f_P$ and $f_S$, *e.g*. mean and variance (Huang and Belongie 2017), covariance (Li et al. 2017b), and maximum mean discrepancy (Li et al. 2017c). In contrast, the family of style feature assignment methods reconstructs $f_P$ from the style feature vectors in $f_S$, while the content feature $f_C$ is used as the reference to either implicitly guide the style assignment (Huang et al. 2020) or explicitly ensure the optimal global and/or local assignment (Chen and Schmidt 2016; Gu et al. 2018). The third family spans AST approaches (Park and Lee 2019; Shen, Yan, and Zeng 2018; Zhang, Zhu, and Zhu 2019; Kotovenko et al. 2019b; Yao et al. 2019; Cheng et al. 2019; Liu et al. 2021; Deng et al. 2021; Wu et al. 2021) that do not show a strong preference between $f_S$ and $f_C$, and typically use a sub-network to implicitly represent the function.

### Generalized Assign-and-Mix Problem

To simplify discussion, we view all deep features $f_C$ and $f_S$ as matrices, *i.e*., for a convolutional feature tensor $f_*$ of size $h \times w \times d$, we view it as an $n_* \times d$ matrix in the paper, where $n_* = h \cdot w$ and $d$ is the feature dimension. In addition, we denote $f_{*,i}$ as the $i$-th row in $f_*$. The row vector is named the content/style vector.

As aforementioned, the core task in AST is to design the transfer function Eq. (1). Existing approaches have different design criteria, but we discover that many (Huang and Belongie 2017; Park and Lee 2019; Gu et al. 2018; Huang et al. 2020; Deng et al. 2020) follow the same AM transfer function of the abstract form below,

$$f_P = t^{\text{AM}}(f_C, f_S) = \underbrace{A \times f_S}_{\texttt{Assign}} \underbrace{+ M \odot \phi(f_C)}_{\texttt{Mix}} \tag{2}$$

where content $f_C$ and style of $f_S$ are of size $n_C \times d$ and size $n_S \times d$, respectively, $f_P$ is the output feature with content from $f_C$ and style from $f_S$, $A$ is the assignment matrix of size $n_C \times n_S$ denoting style-content feature correspondence, $M$ is the mixing matrix of size $n_C \times d$ that fuses content and style features, $\phi(\cdot)$ is an element-wise projection function, and $\times$ and $\odot$ represents matrix multiplication and element-wise multiplication, respectively. Further, with $A[i, j]$ denoting the element in $A$ at position $(i, j)$, we have $A[i, j] \geq 0$ and $\sum_j A[i, j] = 1$.

We now show how the transfer functions of three well-known and representative AST methods, namely, Adaptive Instance Normalization (AdaIN) (Huang and Belongie 2017) from the content feature transform family, Deep Feature Reshuffle (DFR) (Gu et al. 2018) from the style feature assignment family, and Style Attentional Network (SANet) (Park and Lee 2019) from the implicit network-based family, can be rewritten in the AM formulation.

**Adaptive Instance Normalization** AdaIN (Huang and Belongie 2017) is a representative content feature transform AST method. It adopts the following linear transfer function with style mean and standard deviation used as the bias

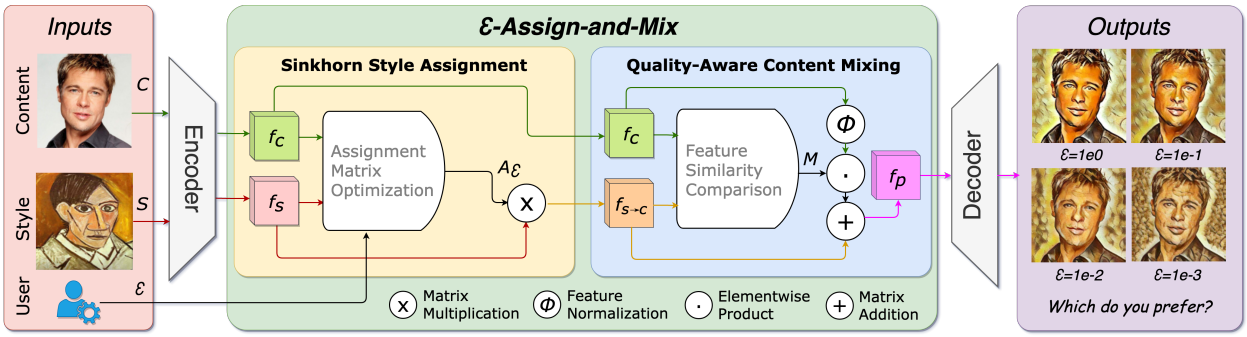


Figure 1: Overview of our $\varepsilon$-Assign-and-Mix ($\varepsilon$-AM ) arbitrary style transfer framework. Besides content and style inputs $C$ and $S$, we introduce a new user-controllable parameter $\varepsilon$ to modulate our two-step style transfer process. For given content and style features $f_C$ and $f_S$, Sinkhorn Style Assignment finds the $\varepsilon$-dependent optimal assignment matrix $A_\varepsilon$ to reconstruct content features using style features $f_S$. Quality-aware Content Mixing compares feature similarity between the reconstructed content $f_{S\to C}$ and the original content $f_C$ to dynamically generate a mixing matrix $M$ for extra content blending. In conjunction, they produce diversified and high quality outputs by varying $\varepsilon$ using the same network.

and slope, respectively,

$$f_P^{\text{AdaIN}} = \mu_{f_S} + \sigma_{f_S} \odot f_C^{\mathcal{N}} \tag{3}$$

where $f_*^{\mathcal{N}}$ indicates a normalized feature,

$$f_*^{\mathcal{N}} = \text{normalize}(f_*) = (f_* - \mu_{f_*}) \oslash \sigma_{f_*}; \tag{4}$$

$\odot$ and $\oslash$ denote element-wise product and division, respectively, and $\mu_{f_*}$ and $\sigma^2_{f_*}$ are the mean and variance of $f_*$. Thus, AdaIN (Huang and Belongie 2017) aligns the first two statistical moments between $f_P$ and $f_S$. Following the AM formulation, the AdaIN transfer function can be rewritten as Eq. (5), where $\mathbb{1}_{h\times w}$ is an all-one matrix of size $h\times w$.

$$f_P^{\text{AdaIN}} = \underbrace{\frac{\mathbb{1}_{n_C\times n_S}}{n_S}}_{=A} \times f_S + \underbrace{\sigma_{f_S}}_{=M} \odot \underbrace{\text{normalize}(f_C)}_{=\phi(\cdot)} \tag{5}$$

**Deep Feature Reshuffle**: DFR (Gu et al. 2018) is the first work in the style feature assignment family of AST methods. It has the following transfer function:

$$f_P^{\text{DFR}} = B \times f_S + \beta \cdot f_C \tag{6}$$

where $B$ is a binary matrix whose 1s indicate the optimal correspondences between one content vector and one style vector, $\cdot$ and $\times$ refer to scalar product and matrix multiplication, respectively, and $\beta$ is a scalar. Following the AM formulation, Eq. (6) can be rewritten as Eq. (7), where $\mathcal{I}(\cdot)$ is the identity function.

$$f_P^{\text{DFR}} = \underbrace{B}_{=A} \times f_S + \underbrace{\beta \cdot \mathbb{1}_{n_C\times d}}_{=M} \odot \underbrace{\mathcal{I}(f_C)}_{=\phi(\cdot)} \tag{7}$$

**Style Attentional Networks**: SANet (Park and Lee 2019) uses an attention module to dynamically compute similarities between content and style vectors. The similarity matrix is then used to assign style features and mix content features. We omit learnable feature projections in SANet to simplify discussion as they don't change the content or style semantics of a feature, and arrive the transfer function below,

$$f_P^{\text{SANet}} = \text{softmax}\left(f_C^{\mathcal{N}} \times (f_P^{\mathcal{N}})^T\right) \times f_P + f_C \tag{8}$$

where $(\cdot)^T$ is the matrix transpose and softmax$(\cdot)$ indicates the softmax function along rows. Following the AM formulation, we rewrite Eq. (8) as Eq. (9) below.

$$f_P^{\text{SANet}} = \underbrace{\text{softmax}\left(f_C^{\mathcal{N}} \times (f_S^{\mathcal{N}})^T\right)}_{=A} \times f_S + \underbrace{\mathbb{1}_{n_C\times d}}_{=M} \odot \underbrace{\mathcal{I}(f_C)}_{=\phi(\cdot)} \tag{9}$$

## AST Modulation via $\varepsilon$-Assign-and-Mix

One important but unanswered question in the last section is *why the three studied AST solutions produce very different results*. We investigate this by studying their similarities and differences in **??** . As shown, the methods have different dependencies on style and content. We also observe that, since the mixing matrix $M$ is always heuristic or constant matrix, the assignment matrix $A$ generally plays a more important role in determining the AST output. Hence, it is crucial to analyze the assignment matrix $A$.

We recharacterize $A$ by normalizing it as $\tilde{A} = A/n_C$. Since $\sum_i \sum_j \tilde{A}[i,j] = 1$ and $\tilde{A}[i,j] \geq 0$, $\tilde{A}$ can be viewed as a probability matrix with entropy as:

$$h(\tilde{A}) = -\sum_{i=1}^{n_c}\sum_{j=1}^{n_S} \tilde{A}[i,j] \cdot \log(\tilde{A}[i,j]) \tag{10}$$

The differences in the three studied AST methods can be largely attributed to their distinct nature of $h(\tilde{A})$. Specifically, AdaIN (Huang and Belongie 2017) requires a fully-flat assignment matrix to represent the style mean, corresponding to maximum $h(\tilde{A}^{\text{AdaIN}})$; DFR (Gu et al. 2018) uses a binary assignment matrix to reshuffle style feature vectors *w.r.t.* content, resulting in a small $h(\tilde{A}^{\text{DFR}})$; and SANet (Park and Lee 2019) adopts an attention module to predict an dynamic assignment matrix, with $h(\tilde{A}^{\text{SANet}})$ lying between the previous two extremes.

We reformulate the AST transfer function (Eq. (2)) as our $\varepsilon$-AM transfer function in Eq. (11) with a new and external user-adjustable parameter $\varepsilon$ that allows users to directly

modulate the AST output at runtime by controlling the entropy of the assignment matrix.

$$f_P = t_\varepsilon^{\text{AM}}(f_C, f_S) = \underbrace{A_\varepsilon \times f_S}_{\varepsilon-\texttt{Assign}} \underbrace{+M \odot \phi(f_C)}_{\texttt{Mix}} \quad (11)$$

The resulting $\varepsilon$-AM formulation is illustrated in Fig. 1 and the *Sinkhorn Style Assignment* and the *Quality-aware Content Mixing* processes implement the *Assignment* and *Mixing* steps in Eq. (11), respectively. In the assignment step, the external parameter $\varepsilon$ is used to constrain the entropy of the assignment probability matrix $\tilde{A}_\varepsilon$ during optimization for reconstructing content from style features as below

$$f_{S\to C} = A_\varepsilon \times f_S. \quad (12)$$

In the mixing step, we quantify the content discrepancy between $f_{S\to C}$ and $f_C$, and mix content features accordingly. Details discussion can be found later.

| | **AdaIN** | **DFR** | **SANet** | **Ours** |
|---|---|---|---|---|
| $M$ dep. on style? | Yes | No | No | Yes |
| $M$ dep. on content? | No | No | No | Yes |
| $M$'s type | Heuri. | Const. | Const. | Attention |
| $A$ dep. on style? | Yes | Yes | Yes | Yes |
| $A$ dep. on content? | No | Yes | Yes | Yes |
| $A$'s type | Heuris. | Optim. | Attention | Optim. |
| Entropy of $\tilde{A}$ | High | Low | Med | Adjustable |
| Multi-output? | No | No | No | Yes |

Table 2: AST assignment and mixing matrices comparisons. Heuris. means Heuristic, Const. means Constant, and Optim. means Optimization.

## Sinkhorn Style Assignment

The design of the assignment matrix $A_\varepsilon$ needs to fulfill two requirements: (1) its entropy should be easily controlled by $\varepsilon$, and (2) $A_\varepsilon$ can be efficiently computed. We adopt the optimal transport framework for computing the style-content correspondence in $A_\varepsilon$ and use *Sinkhorn distances* (Cuturi 2013) for efficiency.

In particular, we define the assignment cost matrix $\mathcal{C}$ of size $n_C \times n_S$ as:

$$\mathcal{C}[i,j] = \rho(f_{C,i}, f_{S,j}), \quad (13)$$

where $\rho(\cdot,\cdot)$ measures the cosine distance, following (Kolkin et al. 2020), and $\mathcal{C}[i,j]$ represents the distance between the $i$-th content vector $f_{C,i}$ and the $j$-th style vector $f_{S,j}$.

To control the entropy of $A_\varepsilon$ with an external parameter $\varepsilon \geq 0$, we formulate the optimal $A_\varepsilon$ computation as solving the optimal transport problem with an entropic constraint as:

$$A_\varepsilon = \underset{A}{\operatorname{argmin}} \left\langle A/n_C, \mathcal{C} \right\rangle - \varepsilon \cdot h(A/n_C) \quad (14)$$

where $\tilde{A} = A/n_C$ is a probability matrix, $\langle\cdot,\cdot\rangle$ stands for the Frobenius dot-product and $h(\cdot)$ is the entropy function. According to (Cuturi 2013), this is a strictly convex problem that can be solved at "lightspeed" with the *Sinkhorn-Knopp matrix scaling* algorithm (Cuturi 2013) (see details in the supplemental material).

$A_\varepsilon$ exhibits two interesting properties: (1) the entropy $h(A_\varepsilon/n_C)$ decreases monotonically as $\varepsilon$ decreases, and (2) $h(A_\varepsilon/n_C)$ is differentiable *w.r.t.* $f_C$ and $f_S$. Property (1) allows control over the assignment matrix via $\varepsilon$, *i.e.* as $\varepsilon \to 0$, $A_\varepsilon$ approaches to a binary matrix, which is similar to the solution in DFR, and as $\varepsilon \to \infty$, $A_\varepsilon$ approaches a uniformly distributed matrix, which is similar to the solution in AdaIN. Property (2) allows us to train an encoder using gradients from the assignment matrix. Finally, we obtain the reconstructed content feature $f_{S\to C}$ via Eq. (12).

Empirically, we notice that when $\varepsilon = 1$, the resulting $A_\varepsilon$ tends to follow a uniform distribution and does not change significantly for $\varepsilon > 1$. Further, for $\varepsilon < 1e\text{-}4$, the numerical solution could be unstable. Hence, we set the adjustment range to $\varepsilon \in [1e\text{-}4, 1]$ in this work. The resulting algorithm typically converges in three to ten iterations.

## Quality-Aware Content Mixing

Instead of using constant or heuristic-based content mixing like previous works (see **??** ), we develop a novel quality-aware content mixing method for fusing content and style features. Intuitively, if a reconstructed content vector $f_{S\to C,i}$ is close to the corresponding original content vector $f_{C,i}$, no mixing is needed since both content and style information are well-preserved. However, if $f_{S\to C,i}$ is far from $f_{C,i}$, it is necessary to mix it with $f_{C,i}$ to reduce content distortion. We define the quality-aware mixing matrix $M$ as,

$$M[i,j] = \text{sigmoid}\big(w \cdot \rho(f_{C,i}, f_{S\to C,i}) + b\big), \forall j \in [1,d] \quad (15)$$

where sigmoid$(\cdot)$ is the activation, $w$ and $b$ are learnable scalars, and $\rho(\cdot,\cdot)$ is the cosine distance function as following the work of (Kolkin et al. 2020). Hence, $\rho(f_{C,i}, f_{S\to C,i})$ quantifies the feature similarity, and $M[i,j]$ indicates the dynamic coefficient to mix extra content. As a result, we could obtain output feature $f_P$ that carries both content and style information via Eq. (11).

# Experimental Results

## Training Settings

Although the proposed two-step $\varepsilon$-AM transfer function module can be plugged into many AST networks (see Fig. 1), *e.g.*, skip-connection networks (Huang et al. 2020), U-Net-like models (Liu et al. 2021), neural flows (An et al. 2021) and others (Lin et al. 2021), we adopt the classic AdaIN network (Huang and Belongie 2017). The model contains a VGG-19 (Simonyan and Zisserman 2014) encoder (up to the layer `relu4_1`) and its mirrored decoder, where `MaxPool2D` layers are replaced with `UpSample2D` layers. Once the VGG-19 encoder outputs the content and style features $f_C$ and $f_S$, we pass them through the two-step transfer function, and obtain the pastiche feature $f_P$, which is then used by the decoder to produce the stylized image $P$.

In particular, the encoder is initialized with a VGG-19 (Simonyan and Zisserman 2014) pre-trained on ImageNet and the decoder is initialized with random weights. We use the suggested VGG layer set from (Huang and Belongie 2017)

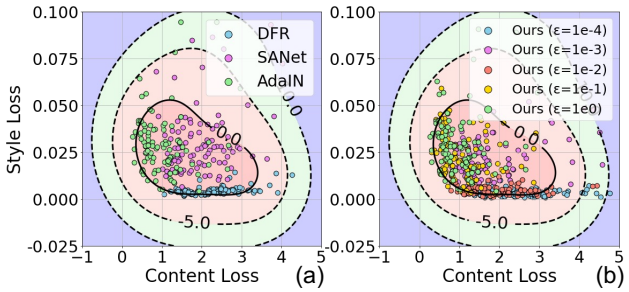


Figure 2: Influence of $\varepsilon$ on stylization diversity: (a) shows the combined loss distribution of AdaIN, DFR, and SANet methods, and (b) shows that of ours with different $\varepsilon$ values, respectively. Each $\circ$ indicates a sample, and colors denote AST methods. The contours are fitted via `OneClassSVM` using data points from (a), for providing reference across (a) and (b). Solid contours indicate the decision boundaries while numbers on contours indicate margins. The diversity generated by $\varepsilon$-AM is comparable to that spanned by all three methods in (a).

to compute content and style losses and the style-aware normalized losses in (Cheng et al. 2021) for AST optimization. The MS-COCO (Lin et al. 2014) and the Painter-By-Numbers (Nichol 2016) (PBN) datasets are used for content and style images, respectively. We randomly sample $\varepsilon$ from $[1e\text{-}4, 1]$, and (content, style) pairs. We resize images to 256×256, and train models for 160,000 iterations. We tune the entire network end-to-end with the `Adam` optimizer using $1e$-4 as the learning rate and $1e$-5 as the weight decay.

## Study of $\varepsilon$ Impact on Stylization and Diversity

We begin with studying the relationship between $\varepsilon$ and the overall AST output diversity in terms of content and style loss distributions. More precisely, we randomly select 100 pairs of content and style images, and apply the above-mentioned three AST methods, namely, AdaIN (Huang and Belongie 2017), DFR (Gu et al. 2018), and SANet (Park and Lee 2019). We also perform AST on these pairs using our method with $\varepsilon \in \{1e0, 1e\text{-}1, 1e\text{-}2, 1e\text{-}3, 1e\text{-}4\}$.

We then compute content and style losses for all samples and plot them in Fig. 2. The plot shows that (1) as $\varepsilon$ varies, $\varepsilon$-AM changes the AST behavior in terms of output content and style losses, and (2) the diversity of $\varepsilon$-AM is comparable to that spanned in conjunction by AdaIN, DRF, and SANet.

We further study the relationship between $\varepsilon$ and the entropy of the resulting probability matrix using the same data. Tab. 3 summarizes our findings, showing that the entropy of the assignment matrix $h(\tilde{A}_\varepsilon)$ varies as $\varepsilon$ changes. Specifically, the resulting assignment matrix has similar entropy to that of AdaIN (flat matrix) for large $\varepsilon = 1e0$ and DFR (sparse matrix) for small $\varepsilon = 1e\text{-}4$. More importantly, we can modulate the resulting entropy from that of AdaIN to that of DFR by tuning $\varepsilon$, which is the exact behavior that we expect to achieve.

| **Entropy** | **Adjustable Parameter $\varepsilon$** | | | | | **AdaIN** | **DFR** | **SANet** |
|---|---|---|---|---|---|---|---|---|
| $h(\tilde{A}{=}A/n_C)$ | $1e0$ | $1e$-1 | $1e$-2 | $1e$-3 | $1e$-4 | | | |
| **Mean** | 16.5 | 14.9 | 12.4 | 10.5 | 8.9 | 16.6 | 8.3 | 9.6 |
| **Std** | 0.01 | 0.14 | 0.59 | 0.48 | 0.39 | - | - | 0.34 |

Table 3: The entropy of the probability assignment matrices trained by different AST methods.

## Study of Impact of Assignment and Mixing Steps

We study the effectiveness of the assignment and mixing steps in our method by comparing the full model with three ablation versions, namely, (1) *Assignment-No-BP*, where we stop the gradient from the assignment matrix to update the encoder, (2) *No-Mix*, where we skip the content mixing step, and (3) *Constant-Mix*, where we mix constant content features like DFR(Gu et al. 2018) (see Eq. (6)). We control for the impact of $\varepsilon$ by fixing it to $1e$-2 in this study.

We find that both the proposed modules are effective for generating high quality images. Tab. 4 validates the proposed assignment and mixing choices in terms of the model's content and style losses. It is clear that the losses increase significantly when the gradient from the assignment matrix is not used (*Assignment-No-BP*), which makes it difficult to tune the encoder. Removing quality-aware content feature mixing (*No-Mix*) or using the constant content mixing (*Constant-Mix*) also increases both the losses compared to the full model. However, this has less impact compared to the assignment ablation. Overall, both the proposed differentiable Sinkhorn style assignment method and the quality-aware content mixing improve AST. Qualitatively, we notice that *Assignment-No-BP* achieves the worst image quality, and fails to preserve local content or style. while the output image from the full model has the best quality. We provide corresponding visualization in the supplementary material.

| **Settings** | **Content Loss↓** | **Style Loss↓** | **Total Loss↓** |
|---|---|---|---|
| Assign-No-BP | 2.26 | 3.02 | 5.28 |
| No-Mix | 2.09 | 2.82 | 4.91 |
| Const.-Mix | 2.01 | 2.59 | 4.60 |
| Full | **1.97** | **2.28** | **4.25** |

Table 4: Evaluation losses for models trained with different assignment methods and mixing methods.

## Comparison with the State-of-the-Art

In this section, we conduct experiments to compare our $\varepsilon$-AM solution with the state-of-the-art (SoTA) style transfer works, including AST methods AdaIN (Huang and Belongie 2017),WCT (Li et al. 2017b),SANet (Park and Lee 2019),MANet (Deng et al. 2020),ArtFlow (An et al. 2021),DFR (Gu et al. 2018) and DAST methods PWCT (Wang et al. 2020), and SP(Li et al. 2020). Unless otherwise stated, we use the default settings for previous works. Since not all studied metrics are applicable to both AST and DAST approaches, we do the following adjustment to ensure fair comparisons – for an AST metric defined on

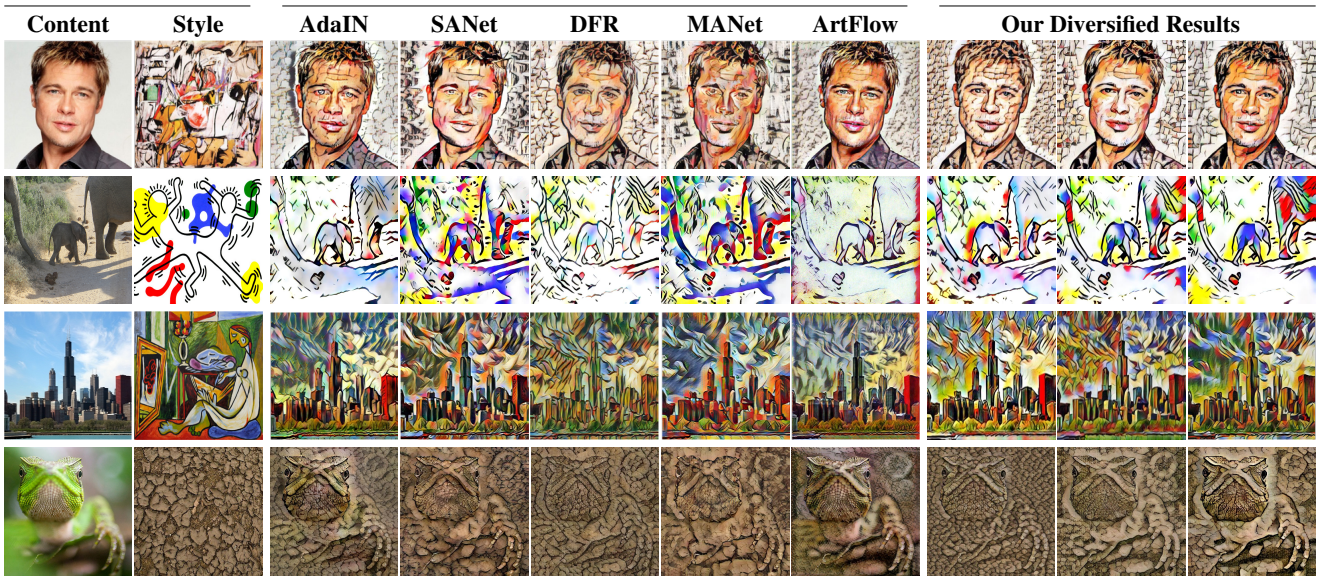


Figure 3: Qualitative comparisons of the proposed $\varepsilon$-AM ($\varepsilon \in (1e\text{-}4, 1)$) to SoTA AST approaches. $\varepsilon$-AM achieves comparable to or better style transfer quality than SoTA AST solutions. Best viewed digitally and zoomed in. See more results in the supplemental materials.

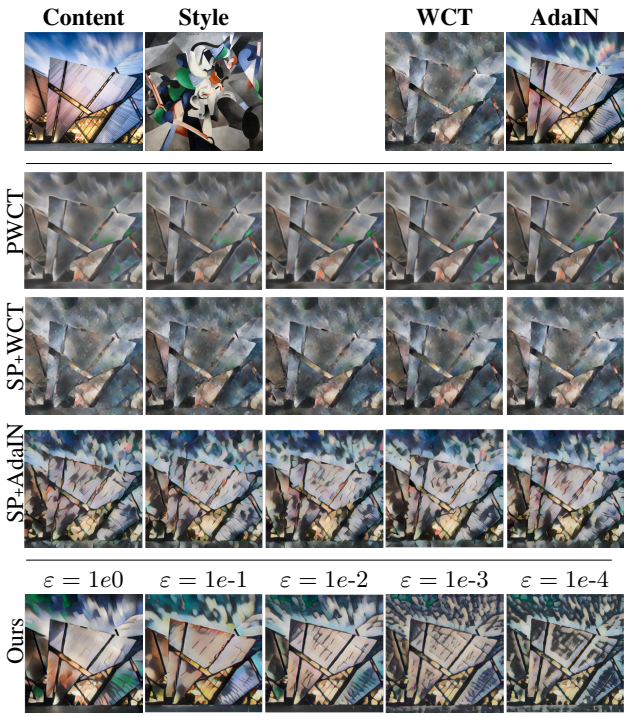


Figure 4: Qualitative comparisons of the proposed $\varepsilon$-AM ($\varepsilon \in (1e\text{-}4, 1)$) to SoTA DAST approaches. $\varepsilon$-AM attains obviously higher diversity than SoTA DAST solutions.

a deterministic output, we fix the random seed in a DAST solution and use a fixed $\varepsilon$ for the proposed $\varepsilon$-AM solution; for a DAST metric defined on a set of outputs, we group AST methods as Multi-AST solutions (*i.e.* a set of AST solutions is treated as single DAST solution to generate *diversified* outputs) and use random $\varepsilon$ values for our method. In particular, we name the set of all eight studied AST methods as *Multi-AST-8*. In the following sections, we show that the $\varepsilon$-AM solution outperforms SoTA methods for diversity, deception rate, inference time and user-like rate for most cases and leave the discussion of failure cases in appendix.

**Diversity** metrics used in DAST include pixel distance (PD) (Wang et al. 2020) and learned perceptual image patch similarity (LPIPS) (Zhang et al. 2018). The former computes the average pixel difference in RGB space, while the latter measures that in the feature space of `conv1_5` of the ImageNet pretrained AlexNet.

| Method | PD↑ | LPIPS↑ |
|---|---|---|
| PWCT | 0.059 | 0.242 |
| SP + AdaIN | 0.050 | 0.229 |
| SP + WCT | 0.088 | 0.248 |
| Ours | **0.117** | **0.375** |

Table 5: Diversity evaluation for SoTA DAST approaches.

We compute diversity metrics by randomly sampling 1,000 image pairs with content images from MS-COCO (Lin et al. 2014) and style images from PBN (Nichol 2016) for evaluation. For each solution, we generate three diversified samples for each image pair and compute their pairwise distances, which results in $C_3^2 = 6$ scores for each style-content image pair. These results are presented in Tab. 5. Multi-AST-8 represents a setting that uses eight very different AST models. This setting achieves the highest diversity

performance, as expected, and we treat its scores as empirical upper bounds. In contrast, the SoTA DAST solutions using a single model attain much lower PD and LPIPS. Although $\varepsilon$-AM also has only one network, it largely promotes the diversity scores of DAST solutions – leading the best DAST approach by 33% in PD and 51% in LPIPS relatively, and comes close to Multi-AST-8. Qualitative results can be found in Figs. 3 and 4.

**Deception Rate** (DR) is defined as the success rate of an AST method's output stylized images at deceiving an expert artist classification model. This score is highly correlated to human expert scores (Sanakoyeu et al. 2018). Consequently, a higher deception rate indicates better stylization quality.

We follow the protocol in (Cheng et al. 2021) to compute DR and first generate 5,000 stylized images by randomly matching 1,798 PBN style images from 34 artists (Cheng et al. 2021) who have at least 30 paintings in the testing set that have not been seen during training, and 5,000 MS-COCO content images. We construct a nearest neighbor classifier based on the winning solution of the PBN challenge[1]. It computes 2,048-$d$ features for all style and stylized images. Finally, for a stylized image, a successful deception means that its artist matches that of its nearest style image.

Results are summarized in Tab. 6. It is clear that our approach achieve much higher DR comparing to SoTA AST and DAST approaches, and the best DR is attained when we use $\varepsilon = 1e\text{-}2$. Qualitative results in Fig. 4 show that our method generates high-quality diversified stylized images.

| | Method | DR (%) ↑ | Inf. (ms) ↓ | Multi. of AdaIN ↓ |
|---|---|---|---|---|
| AST | AdaIN | 28.80 | **22** | **1.00**× |
| | WCT | 18.45 | 172 | 7.82× |
| | SANet | **40.65** | 37 | 1.68× |
| | UST | 22.00 | 49 | 2.23× |
| | MANet | 33.05 | 39 | 1.77× |
| | Art Flow | 30.70 | 183 | 8.32× |
| | DFR | 38.45 | 4,501 | 204.59× |
| | MultiModal | 21.10 | 105 | 4.77× |
| DAST | PWCT | 17.45 | 203 | 9.23× |
| | SP+WCT | 17.55 | 180 | 8.18× |
| | SP+AdaIN | **33.10** | **23** | **1.05**× |
| Ours | $\varepsilon = 1e\text{-}0$ | 34.70 | **23** | **1.05**× |
| | $\varepsilon = 1e\text{-}1$ | 39.05 | 26 | 1.18× |
| | $\varepsilon = 1e\text{-}2$ | **47.35** | 48 | 2.18× |
| | $\varepsilon = 1e\text{-}3$ | 43.45 | 33 | 1.50× |
| | $\varepsilon = 1e\text{-}4$ | 38.60 | 26 | 1.18× |

Table 6: Comparison of deception rate (DR) and inference time.

**Inference Time** is critical for real-world applications. We report inference time based on the average of 100 inference runs of 256×256 images on an *NVIDIA Titan X* GPU. We run this evaluation under single sample AST mode, which treats all DAST solutions as AST and uses fixed $\varepsilon$ for $\varepsilon$-AM. As shown in Tab. 6, our $\varepsilon$-AM runs faster than most SoTA AST and DAST methods. Despite requiring assignment optimization, $\varepsilon$-AM is much faster than other solutions that require optimization, namely, multi-iterations of constraint nearest neighbor search in DFR (Gu et al. 2018) and *Graph-Cut* in MultiModal (Zhang et al. 2019). We also find that there is a trade-off between speed and quality for $\varepsilon$-AM – among all $\varepsilon$ values, $\varepsilon = 1e\text{-}0$ leads to the shortest inference time and also the lowest DR, while $\varepsilon = 1e\text{-}2$ takes the longest time but attains the highest DR (see Tab. 6).

[1]https://github.com/inejc/painters

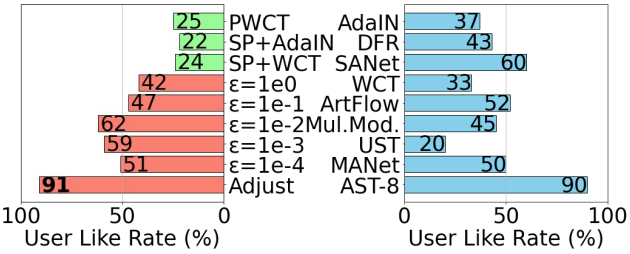


Figure 5: Like-rate user study. AST, DAST and ours. The "Multi-AST-8" and "Adjust" are the aggregated like rate for all AST baselines and five $\varepsilon$ values, respectively.

**User-Like Rate** is a subjective study we conducted on 10 individuals. We randomly select 100 content-style pairs for each user, and generate corresponding stylized images for all AST and DAST methods as well as our $\varepsilon$-AM. Users are presented with tuples of (content, style, output), and they vote either *like* or *dislike*. No verbal guidelines or method information is provided to the users. Fig. 5 shows the user-like study for our $\varepsilon$-AM and SoTA works. The proposed $\varepsilon$-AM with $\varepsilon = 1e\text{-}2$ attains the best user-like rate, and leads the second best SANet by 2%. We further study performance in the context of user-control by considering the Multi-AST-8 setting that aggregates likes of eight AST methods and our Adjustable setting that aggregates likes for our method across five $\varepsilon$ values. Votes for Multi-AST-8 are obtained by combining the votes of all eight AST methods – it is considered *dislike* if none of the votes is *like*, otherwise it is taken as *like*. We use the same setting to aggregate votes across five $\varepsilon$ for the "Adjustable" version of our method. Results show that our "Adjustable" setting performs the best across the board while Multi-AST-8, which uses eight different AST models, comes close with 1% lower like rate.

## Conclusion

In this paper, we introduce $\varepsilon$ as an external parameter to augment AST. Our approach includes a generalized AST transfer function with content-style feature assignment optimization and quality-aware content feature mixing. We demonstrate that popular AST works, such as AdaIN, DFR, and SANet, are special cases of our unified framework. By modulating the entropy of the assignment matrix via $\varepsilon$, users can control the output AST images. Our approach outperforms SoTA AST and DAST methods in both quantitative and qualitative evaluations. We generate high-quality and diverse outputs at fast inference speeds. User-study results validate the effectiveness of our method, mitigating the user preference diversity issue in AST.

## References


An, J.; Huang, S.; Song, Y.; Dou, D.; Liu, W.; and Luo, J. 2021. ArtFlow: Unbiased Image Style Transfer via Reversible Neural Flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 862–871.

Chen, D.; Yuan, L.; Liao, J.; Yu, N.; and Hua, G. 2017. Stylebank: An explicit representation for neural image style transfer. In *IEEE conference on Computer Vision and Pattern Recognition*, 1897–1906.

Chen, T. Q.; and Schmidt, M. 2016. Fast patch-based style transfer of arbitrary style. In *NIPS Workshop on Constructive Machine Learning*.

Cheng, J.; Jaiswal, A.; Wu, Y.; Natarajan, P.; and Natarajan, P. 2021. Style-Aware Normalized Loss for Improving Arbitrary Style Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 134–143.

Cheng, M.-M.; Liu, X.-C.; Wang, J.; Lu, S.-P.; Lai, Y.-K.; and Rosin, P. L. 2019. Structure-preserving neural style transfer. *IEEE Transactions on Image Processing*, 29: 909–920.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26: 2292–2300.

Deng, Y.; Tang, F.; Dong, W.; Ma, C.; Pan, X.; Wang, L.; and Xu, C. 2022. StyTr2: Image Style Transfer with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11326–11336.

Deng, Y.; Tang, F.; Dong, W.; Sun, W.; Huang, F.; and Xu, C. 2020. Arbitrary style transfer via multi-adaptation network. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2719–2727.

Deng, Y.; Tang, F.; Pan, X.; Dong, W.; Xu, C.; et al. 2021. StyTrˆ 2: Unbiased Image Style Transfer with Transformers. *arXiv preprint arXiv:2105.14576*.

Dumoulin, V.; Shlens, J.; and Kudlur, M. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *IEEE conference on Computer Vision and Pattern Recognition*, 2414–2423.

Ghiasi, G.; Lee, H.; Kudlur, M.; Dumoulin, V.; and Shlens, J. 2017. Exploring the structure of a real-time, arbitrary neural artistic stylization network. In *British Machine Vision Conference*.

Gu, S.; Chen, C.; Liao, J.; and Yuan, L. 2018. Arbitrary style transfer with deep feature reshuffle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8222–8231.

Guo, M.; Haque, A.; Huang, D.-A.; Yeung, S.; and Fei-Fei, L. 2018. Dynamic task prioritization for multitask learning. In *European Conference on Computer Vision*, 270–287.

Huang, S.; Xiong, H.; Wang, T.; Wang, Q.; Chen, Z.; Huan, J.; and Dou, D. 2020. Parameter-Free Style Projection for Arbitrary Style Transfer. *arXiv preprint arXiv:2003.07694*.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, 1501–1510.

Huo, J.; Jin, S.; Li, W.; Wu, J.; Lai, Y.-K.; Shi, Y.; and Gao, Y. 2021. Manifold Alignment for Semantically Aligned Style Transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14861–14869.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 694–711. Springer.

Kim, S. S.; Kolkin, N.; Salavon, J.; and Shakhnarovich, G. 2020. Deformable style transfer. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, 246–261. Springer.

Kolkin, N. I.; Shechtman, E.; Paris, S.; and Shakhnarovich, G. 2020. Less is More, Faithful Style Transfer without Content Loss. *https://home.ttic.edu/ñickkolkin/Paper/NNST_Preprint.pdf*.

Kotovenko, D.; Sanakoyeu, A.; Lang, S.; and Ommer, B. 2019a. Content and style disentanglement for artistic style transfer. In *IEEE International Conference on Computer Vision*, 4422–4431.

Kotovenko, D.; Sanakoyeu, A.; Ma, P.; Lang, S.; and Ommer, B. 2019b. A content transformation block for image style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10032–10041.

Kotovenko, D.; Wright, M.; Heimbrecht, A.; and Ommer, B. 2021. Rethinking Style Transfer: From Pixels to Parameterized Brushstrokes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12196–12205.

Kwon, G.; and Ye, J. C. 2022. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18062–18071.

Li, C.; and Wand, M. 2016. Precomputed real-time texture synthesis with Markovian generative adversarial networks. In *European Conference on Computer Vision*, 702–716. Springer.

Li, P.; Zhang, D.; Zhao, L.; Xu, D.; and Lu, D. 2020. Style Permutation for Diversified Arbitrary Style Transfer. *IEEE Access*, 8: 199147–199158.

Li, X.; Liu, S.; Kautz, J.; and Yang, M.-H. 2019. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3809–3817.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017a. Diversified texture synthesis with feed-forward networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3920–3928.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017b. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, 386–396.

Li, Y.; Wang, N.; Liu, J.; and Hou, X. 2017c. Demystifying neural style transfer. In *International Joint Conference on Artificial Intelligence*, 2230–2236.

Lin, T.; Ma, Z.; Li, F.; He, D.; Li, X.; Ding, E.; Wang, N.; Li, J.; and Gao, X. 2021. 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*, 5141–5150.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 740–755. Springer.

Liu, S.; Lin, T.; He, D.; Li, F.; Wang, M.; Li, X.; Sun, Z.; Li, Q.; and Ding, E. 2021. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6649–6658.

Liu, X.-C.; Yang, Y.-L.; and Hall, P. 2021. Learning To Warp for Style Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3702–3711.

Nichol, K. 2016. *Painter by numbers*.

Park, D. Y.; and Lee, K. H. 2019. Arbitrary style transfer with style-attentional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5880–5888.

Risser, E.; Wilmot, P.; and Barnes, C. 2017. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*.

Sanakoyeu, A.; Kotovenko, D.; Lang, S.; and Ommer, B. 2018. A style-aware content loss for real-time hd style transfer. In *European Conference on Computer Vision*, 698–714.

Shen, F.; Yan, S.; and Zeng, G. 2018. Neural style transfer via meta networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 8061–8069.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Ulyanov, D.; Lebedev, V.; Vedaldi, A.; and Lempitsky, V. S. 2016. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. In *International Conference on Machine Learning*, volume 1, 4.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2017. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 6924–6932.

Wang, Z.; Zhao, L.; Chen, H.; Qiu, L.; Mo, Q.; Lin, S.; Xing, W.; and Lu, D. 2020. Diversified arbitrary style transfer via deep feature perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7789–7798.

Wang, Z.; Zhao, L.; Chen, H.; Zuo, Z.; Li, A.; Xing, W.; and Lu, D. 2021. Diversified Patch-based Style Transfer with Shifted Style Normalization. *arXiv preprint arXiv:2101.06381*.

Wu, X.; Hu, Z.; Sheng, L.; and Xu, D. 2021. StyleFormer: Real-Time Arbitrary Style Transfer via Parametric Style Composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14618–14627.

Wu, Z.; Zhu, Z.; Du, J.; and Bai, X. 2022. CCPL: Contrastive Coherence Preserving Loss for Versatile Style Transfer. In *European Conference on Computer Vision*.

Xie, X.; Li, Y.; Huang, H.; Fu, H.; Wang, W.; and Guo, Y. 2022. Artistic Style Discovery With Independent Components. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19870–19879.

Yao, Y.; Ren, J.; Xie, X.; Liu, W.; Liu, Y.-J.; and Wang, J. 2019. Attention-aware multi-stroke style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1467–1475.

Zhang, C.; Zhu, Y.; and Zhu, S.-C. 2019. Metastyle: Three-way trade-off among speed, flexibility, and quality in neural style transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1254–1261.

Zhang, H.; and Dana, K. 2018. Multi-style generative network for real-time transfer. In *European Conference on Computer Vision*, 0–0.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhang, Y.; Fang, C.; Wang, Y.; Wang, Z.; Lin, Z.; Fu, Y.; and Yang, J. 2019. Multimodal style transfer via graph cuts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5943–5951.