Language-Driven Image Style Transfer

Tsu-Jui Fu, William Yang Wang
UC Santa Barbara
(tsu-juifu,william)@cs.ucsb.edu

Xin Eric Wang
UC Santa Cruz
xwang366@ucsc.edu

Abstract

Despite having promising results, style transfer, which requires preparing style images in advance, may result in lack of creativity and accessibility. Following human instruction, on the other hand, is the most natural way to perform artistic style transfer that can significantly improve controllability for visual effect applications. We introduce a new task—language-driven image style transfer (LDIST)—to manipulate the style of a content image, guided by a text. We propose contrastive language visual artist (CLVA) that learns to extract visual semantics from style instructions and accomplish LDIST by the patch-wise style discriminator. The discriminator considers the correlation between language and patches of style images or transferred results to jointly embed style instructions. CLVA further compares contrastive pairs of content image and style instruction to improve the mutual relativeness between transfer results. The transferred results from the same content image can preserve consistent content structures. Besides, they should present analogous style patterns from style instructions that contain similar visual semantics. The experiments show that our CLVA is effective and achieves superb transferred results on LDIST.

1 Introduction

Style transfer [1][2][3][56] adopts appearances and visual patterns from another reference style images to manipulate a content image. Artistic style transfer has a considerable application value for creative visual design, such as image stylization and video effect [112][113][114][115][116]. However, it requires preparing collections of style image in advance. It even needs to redraw new references first if there is no expected style images, which is impractical due to an additional overhead. Language is the most natural way for humans to communicate. If a system can follow textual descriptions and automatically perform style transfer, we can significantly improve accessibility and controllability.

In this paper, we introduce language-driven image style transfer (LDIST). As shown in Fig.1, LDIST treats a content image and an instruction as the input, and the style transferred result is manipulated from a style description. It should preserve the structure of car scene from the content image and simultaneously modifies the style pattern that corresponds to “horizontally lined texture, blues.” Our LDIST task is different from the general language-based image-editing (LBIE) [25][26][30][28], which usually alters objects or properties of object in an image. The main challenge of LDIST is to express visual semantics from language as style patterns. For example, not only color distributions (“blue” or “green”) but also texture patterns (“horizontally lined” or “veined, bumpy”) should be presented in transferred results. More importantly, it requires linking concrete objects with their visual concepts, such as “zebra” with “black-and-white stripe” or “pineapple” with “bumpy yellow.”

We present contrastive language visual artist (CLVA), including language visual artist (LVA) and contrastive reasoning (CR), to perform style transfer conditioning on guided texts. LVA preserves content structures from content images (C) and extracts visual semantics from style instructions
Figure 1: The introduced language-driven image style transfer (LDIST) task. LDIST allows manipulating colors and textures of a content image (C), guided by a style instruction (X).

(\lambda) to carry out LDIST. With the patch-wise style discriminator, it considers patches of style image and transferred result with language to jointly embed style instructions. Furthermore, CR compares contrastive pairs to obtain an additional improvement, where relative content images or style instructions should present similar content structures or style patterns.

To evaluate LDIST, we conduct a new dataset built upon DTD\textsuperscript{2}[4], which provides texture images with textual descriptions (\lambda) as reference styles. We also collect numerous wallpapers that present diverse scenes as content images (C). The experiments show that our CLVA is effective for LDIST and achieves superb transferred results on both automatic metrics and human evaluation. Besides, using natural instructions improves the controllability of style transfer and can support partial semantic editing of reference style. In summary, our contributions are four-fold:

- We introduce LDIST that follows natural language to accomplish artistic style transfer;
- We present CLVA, which learns to align visual semantics from style instructions with style images, to provide sufficient style patterns for LDIST;
- For evaluation, we prepare a new dataset containing diverse scenes as content images and descriptions of texture as style instructions;
- Extensive experiments and qualitative examples demonstrate that our CLVA outperforms baselines on both automatic metrics and human evaluation.

2 Related Work

Artistic Style Transfer Style transfer [1,43,45,47,49,50,52,54,56] is to redraw an image with a specific style. In general, style transfer can be divided into two categories: photorealistic and artistic. Photorealistic style transfer [57,59,60,61,62] aims at applying reference styles on scenes without hurting details and satisfying contradictory objectives. By contrast, artistic style transfer [63,64,65,66,67,69,70,71,72,73,74,75,76,77,78,79,80,81,82] captures style concepts from reference and modifies color distributions and texture patterns of content images. However, it requires preparing numerous style images in advance, which limits practicality of style transfer. We introduce LDIST that allows following textual descriptions to perform artistic style transfer. By understanding diverse semantics of the natural language, LDIST improves the accessibility of humans for creative visual effect (VFX) design.

Language-based Image Editing The general task of LDIST is language-based image editing (LBIE), which also uses language to edit input images. With rule-based instructions and predefined semantic labels, they [17,18] first carry out LBIE but under limited practicality. Inspired by text-to-image generation [19,20,21], previous works [22,23,24,25,26,27,30] perform LBIE by conditional GAN. Following multi-turn manipulation from humans, iterative LBIE (ILBIE) [28,29] edits images step-by-step. LBIE usually modifies objects and properties of objects in the image. In contrast, LDIST aims at preserving the scene structure from the content image and manipulating the color distribution and the texture pattern from the style instruction.

Contrastive Representation Learning Contrastive representation learning has been widely used in self-supervised learning by maximizing mutual information [32,34,35,36,39]. For computer vision, they involve data augmentation [37,38,31] or patch-wise comparison [33,40] to provide contrastive loss. For language, both positive and negative contexts are sampled to learn sentence representation [41,42]. Regarding vision-and-language research, we are the first to apply contrastive learning into image editing, where our CLVA considers contrastive visual-text pairs to improve LDIST.
3 Language-Driven Image Style Transfer (LDIST)

3.1 Task Definition

We introduce the language-driven image style transfer (LDIST) task to manipulate colors and textures of a content image (C), guided by a style instruction (X). As illustrated in Fig. 1, a system is required to extract not only color distribution but also texture patterns from style instructions (X). For training, we have pairs of style image (S) with style instruction (X) to learn the correlation between visual patterns and language semantics. During testing, only style instructions (X) are provided for LDIST to carry out style transfer purely relied on language.

3.2 Overview of CLVA

We present contrastive language visual artist (CLVA) in Fig. 2. Language visual artist (LVA) extracts content structures from content images (C) and visual patterns from style instructions (X) to carry out LDIST. LVA adopts the patch-wise style discriminator (D) to connect extracted visual semantics to patches of paired style image (PS in Fig. 2). Contrastive reasoning (CR) allows comparing contrastive pairs of content image and style instruction (C1 − X1, C2 − X1, and C2 − X2). In this way, it should present consistent content structures from the same content images (C2) or analogous style patterns from related style images (S1 and S2), despite using different style instructions (X1 and X2).

3.3 Language Visual Artist (LVA)

To tackle LDIST, language visual artist (LVA) first adopts visual encoder (GE) to extract the content feature (hC) and the style feature (hS) for an image and text encoder (ϕ) to extract the style instruction feature (hVX) from an instruction. hC ∈ R^h×w×c is a spatial tensor containing the content structure feature, and hS ∈ R^s is a vector representing the global style pattern but without spatial information. hVX ∈ R^s embeds into the same space of hS to reflect the extracted visual semantic. Then, visual decoder (GD) produces transferred result (O) from hC and hVX, which performs style transfer for content images by style instructions:

$$
\begin{align*}
  \hat{O} &= G_D(h_C, h_{VX}), \\
  h_C &= G_E(C), \\
  h_{VX} &= \phi(X), \\
  h_S &= \text{concat}(h_C, h_{VX}).
\end{align*}
$$

In particular, visual decoder (GD) applies self-attention [98, 28, 29] along the channel side to fuse content features (hC) and style features (hS), where hS is concatenated with hC along each spatial
where Crop is to randomly crop an image (\(G\) visual decoder (3.4 Contrastive Reasoning (CR)) the LV A inference to acquire cross results for pairs of content image and style instruction: style patterns from transferred results of contrastive pair. A contrastive pair consists of two different content images. As shown in Fig. 2, contrastive reasoning (CR) compares content structures or content structure. Moreover, related style instructions can apply analogous style patterns to all kinds inputs, which meets the goal of extract content features (\(C\)) and the style matching loss (\(L_{\text{sm}}\)). To address the connection between language semantics from \(X\) and visual semantics from \(S\), we introduce the patch-wise style discriminator (\(D\)). Inspired by texture synthesis [107, 108], images with analogous patch patterns should appear perceptually similar texture patterns.\(D\) tries to recognize the correspondence between an image patch (\(P\)) and a style instruction (\(X\)):

\[
P_{\hat{O}}, P_{S} = \text{Crop}(\hat{O}), \text{Crop}(S), \\
L_{\text{pd}} = \log(1 - D(P_{\hat{O}}, X)), \\
L_{D} = \log(1 - D(P_{\hat{O}}, X)) + \log(D(P_{S}, X)),
\]

where Crop is to randomly crop an image (1/8 on each side) into patches. With the patch-wise style loss (\(L_{\text{pd}}\)), we aim at generating transferred results that are correlated with style instructions. On the other hand, by the discriminator loss (\(L_{D}\)), \(D\) learns to distinguish that patches from style images (\(P_{S}\)) are true cases, and patches from transferred results (\(P_{\hat{O}}\)) should be false cases. This adversarial loss [105,106] enforces that style patterns from style instructions are presented similarly with style images in transferred results, which jointly embeds the extracted visual semantic between each other.

**Content Matching and Style Matching** To further enhance the alignment with inputs, inspired by cycle consistency [100,101,102,103,104,105,106], we consider the content matching loss (\(L_{\text{cm}}\)) and the style matching loss (\(L_{\text{sm}}\)) of transferred results (\(\hat{O}\)). We adopt visual encoder (\(G_{E}\)) again to extract content features (\(h_{O}^{c}\)) and style features (\(h_{S}^{s}\)) of \(\hat{O}\), where \(h_{O}^{c}\) and \(h_{S}^{s}\) should correlate with content features (\(h_{O}^{c}\)) of content image and style features (\(h_{S}^{s}\)) of reference style image (\(S\)):

\[
(h_{O}^{c}, h_{O}^{s}), (\ldots, h_{S}^{s}) = G_{E}(\hat{O}), G_{E}(S), \\
L_{\text{cm}}, L_{\text{sm}} = ||h_{O}^{c} - h_{O}^{c}||_{2}, ||h_{S}^{s} - h_{S}^{s}||_{2}.
\]

Therefore, transferred results are required to align with content structures and style patterns from inputs, which meets the goal of LDIST.

### 3.4 Contrastive Reasoning (CR)

The above LVA process considers LDIST with a single pair of content image and style instruction. However, a content image should be able to transfer to various styles while preserving the same content structure. Moreover, related style instructions can apply analogous style patterns to all kinds of content images. As shown in Fig. 2, contrastive reasoning (CR) compares content structures or style patterns from transferred results of contrastive pair. A contrastive pair consists of two different content images (\(C_{1}\) and \(C_{2}\) in Fig. 2) with two reference styles (\(\{S_{1}, X_{1}\}\) and \(\{S_{2}, X_{2}\}\)). We follow the LVA inference to acquire cross results for pairs of content image and style instruction:

\[
(h_{C_{1}}^{c}, \ldots), (h_{C_{2}}^{c}, \ldots) = G_{E}(C_{1}), G_{E}(C_{2}), \\
h_{X_{1}}, h_{X_{2}} = \phi(X_{1}), \phi(X_{2}) \\
\hat{O}_{C_{1} \rightarrow X_{1}}, \hat{O}_{C_{1} \rightarrow X_{2}}, \hat{O}_{C_{2} \rightarrow X_{1}}, \hat{O}_{C_{2} \rightarrow X_{2}} = G_{D}(h_{C_{1}}^{c}, h_{S}^{s}), G_{D}(h_{C_{2}}^{c}, h_{S}^{s}), G_{D}(h_{C_{1}}^{c}, h_{X_{2}}^{c}), G_{D}(h_{C_{2}}^{c}, h_{X_{2}}^{c}), G_{D}(h_{C_{2}}^{c}, h_{X_{1}}^{c}), G_{D}(h_{C_{2}}^{c}, h_{X_{2}}^{c}).
\]

**Consistent Matching** The transfer results should present similar content structures (\(\hat{O}_{C_{2} \rightarrow X_{1}}\) and \(\hat{O}_{C_{2} \rightarrow X_{2}}\)) or analogous style patterns (\(\hat{O}_{C_{1} \rightarrow X_{1}}\) and \(\hat{O}_{C_{2} \rightarrow X_{2}}\)) if using the same content image (\(C_{2}\) or
When style images are related, it has to align style features to certain extent even if paired style consistent matching.

Algorithm 1 Language Visual Artist (LVA)

1: $G_E, G_D$: Visual Encoder, Visual Decoder
2: $\phi$: Text Encoder
3: $D$: Patch-wise Style Discriminator
4: while TRAIN_VLA do
5:    $C, \{S, X\}$ $\leftarrow$ Sampled content/style
6:    $h_C^C, h_C^S$ $\leftarrow$ $G_E(C)$
7:    $\hat{O}$ $\leftarrow$ $G_D(h_C^C, h_C^S)$
8:    $L_{rec}$ $\leftarrow$ Reconstruction loss $\triangleright$ Eq. 2
9:    $h_X^C$ $\leftarrow$ $\phi(\hat{O})$
10:   $O$ $\leftarrow$ $G_D(h_X^C, h_X^S)$
11:   $P_S, P_O$ $\leftarrow$ Crop($S$), Crop($O$)
12:   $L_{psd}$ $\leftarrow$ Patch-wise style loss $\triangleright$ Eq. 3
13:   $L_{cm}$ $\leftarrow$ Content matching loss $\triangleright$ Eq. 4
14:   $L_{sm}$ $\leftarrow$ Style matching loss $\triangleright$ Eq. 3
15:   $L_G$ $\leftarrow$ $L_{rec} + L_{psd} + L_{cm} + L_{sm}$
16:   Update $G_E, G_D, \phi$ by minimizing $L_G$
17:   $L_D$ $\leftarrow$ Discriminator loss for $D$ $\triangleright$ Eq. 7
18:   Update $D$ by maximizing $L_D$
19: end while

the same style instruction ($X_1$):

$$\begin{align*}
(h_{C_{1-x_1}}^C, h_{C_{1-x_1}}^S), (h_{C_{2-x_2}}^C, h_{C_{2-x_2}}^S) & = G_E(\hat{O}_{c_1-x_1}), G_E(\hat{O}_{c_1-x_2}), \\
(h_{C_{2-x_1}}^C, h_{C_{2-x_1}}^S), (h_{C_{1-x_2}}^C, h_{C_{1-x_2}}^S) & = G_E(\hat{O}_{c_2-x_1}), G_E(\hat{O}_{c_2-x_2}), \\
L_{c-c} & = ||h_{C_{1-x_1}}^C - h_{C_{1-x_2}}^C||_2 + ||h_{C_{2-x_1}}^C - h_{C_{2-x_2}}^C||_2, \\
L_{c-s} & = ||h_{C_{1-x_1}}^S - h_{S_{2-1}}^S||_2 + ||h_{C_{1-x_2}}^S - h_{S_{2-x_2}}^S||_2,
\end{align*}$$

where consistent matching of content structure ($L_{c-c}$) or style pattern ($L_{c-s}$) is aligned by content features or style features of transferred results, extracted by visual encoder.

Relative Matching Apart from consistent matching, distinct style instructions, which imply corresponding visual semantics, should still present relative style patterns in transferred results. For example, we can only discover “red, repetitive, floral” literally from $X_2$. However, if comparing reference style images ($S_1$ and $S_2$), we can perceive that they share a similar texture pattern and link the visual concept of “lacelike” from $X_2$ to “smooth, soft, fabric” from $X_1$. We define relative matching ($L_{c-s}$) with the cosine similarity (CosSim) between reference style images as the weight:

$$\begin{align*}
L_{r-c} & = ||h_{C_{1-x_1}}^C - h_{C_{1-x_2}}^C||_2 + ||h_{C_{2-x_1}}^C - h_{C_{2-x_2}}^C||_2, \\
L_{r-s} & = ||h_{C_{1-x_1}}^S - h_{S_{2-1}}^S||_2 + ||h_{C_{1-x_2}}^S - h_{S_{2-x_2}}^S||_2, \\
L_{c-s} & = ||h_{C_{1-x_1}}^S - h_{S_{2-1}}^S||_2 + ||h_{C_{1-x_2}}^S - h_{S_{2-x_2}}^S||_2, \\
\end{align*}$$

When style images are related, it has to align style features to certain extent even if paired style instructions are different. Otherwise, $L_{c-s}$ will be close to 0 and ignore this unrelated style pair. The overall contrastive reasoning loss ($L_{crt}$) considers both consistent matching and relative matching:

$$L_{crt} = L_{c-c} + L_{c-s} + L_{r-c}.$$  \hspace{1cm} (7)

3.5 Learning of CLVA

For each epoch of training, we first train with the LVA process and then CR for our CLVA. Algo. 2 presents the learning process of VLA with the patch-wise style discriminator ($D$). We consider the reconstruction loss ($L_{rec}$) to preserve content structure and the patch-wise style loss ($L_{psd}$) between style instruction and visual pattern of transferred results. Both content matching loss ($L_{cm}$) and style matching loss ($L_{sm}$) are applied to enhance the matching with the inputs. We minimize $L_G$, the total
We also conduct human evaluation in Sec. 4.2 to investigate the quality from the human aspect.

Apart from visual similarity, we also consider the correlation between style instructions and which presents diverse scenes as content images (• WCT • NST style features for training and directly serves as the provided style during testing. Images and style images, as the compared baselines. Style instructions are jointly embedded with typical arbitrary style transfer methods, where a single trained model should support any content structure consistency and style relativeness from contrastive transferred results. Therefore, the overall optimization of CLVA can be summarized as:

$$L_G = L_{rec} + L_{psd} + L_{cm} + L_{um},$$

$$\min_G \max_D L_G + L_D + L_{cr},$$

(8)

4 Experiments

4.1 Experimental Setup

Dataset To evaluate our contrastive language visual artist (CLVA), we build a new dataset for language-driven image style transfer (LDIST). We collect 14,924 wallpapers from WallpapersCraft\(^1\) which presents diverse scenes as content images (C). Each content image is resized to 512x384 in our experiment. For style instructions, the DTD\(^2\) \([4]\) dataset supplies 5,368 pairs of texture image and natural description. We consider texture images \([6]\) as style images (S) and related descriptions as style instructions (X), illustrated in Fig. 3. We randomly sample 2,500 pairs of unseen content image and unseen style instruction, which are not accessible for training, to evaluate the generalizability of LDIST during testing. Note that both style images and style instructions appear for training, but only style instructions are provided during testing to perform style transfer guided by language.

Evaluation Metrics To support large-scale evaluation, we treat style transfer results from style images as the semi-ground truth (Semi-GT) \([4, 15, 16]\) by FastStyleTransfer \([87]\). We apply the following metrics to compare the visual similarity between LDIST results and Semi-GT:

- **Mean-squared Error (MSE)**: As the most traditional metric of visual similarity, MSE is calculated from the mean pixel difference. A lower MSE means a higher pixel similarity with Semi-GT;
- **Structural Similarity Index Measure (SSIM)**: SSIM \([8]\) compares images in different aspects, including luminance, contrast, and structure. A higher SSIM has a higher structural similarity;
- **Frechet Activation Distance (FAD)**: Inspired from IS \([9]\) and FID \([10]\), FAD is computed by the mean L2 distance of the activations from the Inception V3 \([11]\) feature. As a distance metric, a lower FAD represents that LDIST results and Semi-GT are more relevant.

Apart from visual similarity, we also consider the correlation between style instructions and LDIST results. Following GODIVA \([12]\), we adopt CLIP \([13]\) that provides visual-text matching:

- **Vision-and-Language Similarity (VLS)**: VLS calculates the cosine similarity between the joint embedding of the instruction (X) and the transferred result (O) from CLIP;
- **Relative Similarity (RS)**: Since VLS only provides an absolute score of semantic matching, we regard the relative VLS with the Semi-GT (O) as RS to reduce the influence from CLIP;

$$VLS(O, X) = \text{CosSim}(\text{CLIP}(O), \text{CLIP}(X)), \quad \text{RS}(O, C) = \frac{VLS(O, X)}{VLS(O, X)}. \quad \text{(9)}$$

We also conduct human evaluation in Sec. 4.2 to investigate the quality from the human aspect.

Baselines Since being a brand new task, there is no existing baseline on our LDIST. We consider typical arbitrary style transfer methods, where a single trained model should support any content images and style images, as the compared baselines. Style instructions are jointly embedded with style features for training and directly serves as the provided style during testing.

- **NST** \([1]\): By adopting the gram matrix as the style feature, NST requires numerous iterations of optimization between the content loss and the style loss to acquire the transferred result;
- **WCT** \([2]\): WCT applies an encoder-decoder architecture for style transfer in a single pass, where the whitening transform and the coloring transform is to match the covariance of the style feature;

\(^1\)WallpapersCraft: [https://wallpaperscraft.com/](https://wallpaperscraft.com/)
Automatic Metrics (vs. Semi-GT) Human Evaluation (↑)

| Method | MSE (↓) | SSIM (↑) | FAD (↓) | VLS (↑) | RS (↑) | vs. Instruction | vs. Style |
|--------|---------|----------|---------|---------|--------|----------------|----------|
| NST [1] | 0.05515 | 12.114 | 0.12932 | 21.973 | 96.853 | 491 | 490 |
| WCT [2] | 0.03543 | 36.097 | 0.09567 | 22.265 | 98.078 | 534 | 547 |
| AdaIN [3] | 0.03498 | 57.180 | 0.09380 | 22.475 | 98.261 | 594 | 597 |
| CLVA  | 0.03279 | 60.217 | 0.08489 | 22.641 | 98.769 | 631 | 616 |

Table 1: Testing results on automatic metrics and human evaluation (accumulated ranking points).

- AdaIn [3]: AdaIn is also a feed-forward-based method that incorporates the adaptive instance normalization to fuse the content feature and the style feature.

Implementation Detail: Visual encoder ($G_E$) contains 4 layers of downsampling ResBlock [95] to extract content feature and style feature. Particularly, style feature is acquired from a dense layer after average pooling to represent a global style pattern without spatial information. To understand style instructions, text encoder ($\phi$) first adopts RoBERTa [96, 97] for a general language representation, and then a dense layer to jointly embed with style feature. During self-attention in visual decoder ($G_D$), to prevent excessive memory usage, we first shrink the size of the CNN channel down to 64 [98], but expand into the same input size 256 as the output. $G_D$ is built upon 4 upsampling ResBlocks and a convolution layer with output feature 3 to produce RGB results. The patch-wise style discriminator ($D$) follows a similar architecture, where self-attention with a dense layer determines the correlation between instruction feature and patch style feature [28, 29] from $\phi$ and $G_E$. We adopt Adam [99] to optimize the entire CLVA with learning rate 3e-4 for $L_G$, 1e-4 for $L_D$, and 3e-5 for $L_C$.

4.2 Quantitative Results

Automatic Evaluation: Table 1 shows the automatic evaluation results on LDIST. For NST [1], although style instructions are jointly embedded with gram matrices, it cannot provide sufficient style features and leads to poor transferred results (low 12.1 SSIM and high 0.129 FAD). WCT [2] overpasses NST but still relies on the predefined transforms, which makes the generalizability quite limited. Different from NST and WCT, AdaIN leads to workable results (57.2 SSIM and 0.09 FAD) by the learnable adaptive normalization with style instructions. Our CLVA, which adopts contrastive reasoning to compare pairs of content image and style instruction, achieves the best LDIST results with the highest 60.2 SSIM and the lowest 0.08 FAD. A similar trend can be found on the visual-text matching evaluation. NST and WCT are both subject to limited style features from style instructions and result in lower VLS and RS. On the other hand, with the patch-wise discriminator between style images and transferred results from style instructions, our CLVA can obtain LDIST results that are more correlated with style instructions (the highest 22.6 VLS and the highest 98.8 RS).

Human Evaluation: Apart from automatic metrics, we also investigate the quality of LDIST results from the human aspect. Table 1 demonstrates human evaluation between baselines and our CLVA. We randomly sample 75 results from pairs of content image and style instruction. MTurkers from Amazon Mechanical Turk [2] (AMT) rank the correlation of the LDIST result from each method between the style instruction (vs. Instruction) or the style image (vs. Style), where rank 1 gets 4 points, rank 2 gets 3 points, and so on. Each example is assigned to 3 different MTurkers to avoid evaluation bias. At first, our CLVA acquires the highest points (631 points in total) regarding style instructions (vs. Instruction), which indicates that transferred results from CLVA are the most corresponding for humans. Concerning style images (vs. Style), we still obtain the highest 616 points and represent that CLVA is not only corresponding to style instructions but also correlated with style patterns. Last but not least, we discover that human evaluation results follow a similar trend to the results of automatic metrics. It shows that our conducted automatic metrics are aligned with the human aspect and can provide a reasonable large-scale evaluation for LDIST.

4.3 Ablation Study

We conduct an ablation study to present the effect of each component in Table 2. At row (a), our CLVA is purely pretrained on AdaIn results as a warm-up. It slightly overpasses AdaIn, which shows that our self-attention fusion between content images and style instructions has stronger generalizability than

---

2Amazon Mechanical Turks: [https://www.mturk.com/](https://www.mturk.com/)
Figure 3: Visualization of baselines and our CLVA on language-driven image style transfer (LDIST).

Table 2: Ablation study with reconstruction ($L_{rec}$), patch-wise style discriminator ($L_{psd}$), content matching ($L_{cm}$), style matching ($L_{sm}$), and contrastive reasoning ($L_{cst}$) of CLVA on automatic metrics.

| Ablation Settings | Automatic Metrics (vs. Semi-GT) |
|-------------------|----------------------------------|
| $L_{rec} + L_{psd}$ | $L_{cm}$ | $L_{sm}$ | $L_{cst}$ |
| **(a)** | ☑ | ☐ | ☐ | ☑ | ☐ | ☐ | ☑ | ☐ | ☐ | ☑ | ☐ | ☐ | 0.03522 | 57.353 | 0.09263 | 22.552 | 98.437 |
| **(b)** | ☑ | ☑ | ☐ | ☐ | ☑ | ☑ | ☐ | ☑ | ☐ | ☑ | ☑ | ☐ | 0.03448 | 57.593 | 0.09161 | 22.567 | 98.474 |
| **(c)** | ☑ | ☑ | ☑ | ☐ | ☑ | ☑ | ☐ | ☑ | ☐ | ☑ | ☑ | ☐ | 0.03280 | 58.285 | 0.08944 | 22.589 | 98.554 |
| **(d)** | ☑ | ☑ | ☑ | ☐ | ☑ | ☑ | ☐ | ☑ | ☐ | ☑ | ☑ | ☐ | 0.03534 | 59.026 | 0.08788 | 22.602 | 98.566 |
| **(e)** | ☑ | ☑ | ☑ | ☐ | ☑ | ☑ | ☐ | ☑ | ☐ | ☑ | ☑ | ☐ | 0.03395 | 59.098 | 0.08732 | 22.604 | 98.597 |
| **(f)** | ☑ | ☑ | ☑ | ☐ | ☑ | ☑ | ☑ | ☑ | ☐ | ☑ | ☑ | ☐ | 0.03279 | 60.217 | 0.08489 | 22.641 | 98.769 |

Table 3: Inference time cost.

| Method | BS=1 | BS=50 |
|--------|------|-------|
| NST    | 40.40 | 1.035 |
| WCT    | 0.159 | 4.796 |
| AdaIN  | 0.010 | 0.336 |
| CLVA   | **0.009** | **0.268** |

Inference Efficiency Table 3 shows the inference time (in second) about running LDIST on GPU (NVIDIA TITAN X) with different batch sizes (BS). NST, which relies on numerous iterations, executes with low efficiency. Despite being an encoder-decoder architecture, the coloring transform in WCT takes an overhead when searching the correlation. In contrast, both AdaIN and our CLVA go through a feed-forward process. CLVA carries out with over 100 FPS and even achieves 180 FPS with parallelization of 50 examples.

Qualitative Results Fig. 3 demonstrates the visualization results of baselines and our CLVA on LDIST, along with the semi-ground truth (Semi-GT). For NST, the pixels are broken and lead to poor style quality of transferred results. WCT merely picks color information from style instructions, where it only transfers color distribution but not style patterns. AdaIN produces better results, but the objects in content images sometimes become vague. With the patch-wise style discriminator and
Figure 5: Visualization of CLVA results on diverse pairs of content image and style instruction. Contrastive reasoning, our CLVA captures not only color distribution but also texture patterns that are correlated with style instructions. For example, the first row of the second content image, though our CLVA result is not that matching with the Semi-GT, it presents a corresponding pink appearance with a soft texture as the assigned "pinkish pillow covers" in the style instruction. Fig. 5 illustrates more visualization results on diverse pairs of content image and style instruction, produced by CLVA.

Fine-grained Control via Partial Semantic Editing Furthermore, LDIST allows fine-grained control of transferred styles via partial semantic editing. As shown in Fig. 4, we can easily modify the language prompts to control the style semantic. For example, we can manipulate the color distribution from "green" to "orange" or the texture pattern from "veined, bumpy" to "fabric, metallic". Then, the related semantics of LDIST results would be changed correspondingly by the edited style instructions.

5 Conclusion and Discussion

We introduce language-driven image style transfer (LDIST) to manipulate colors and textures of a content image by a style instruction. We propose contrastive language visual artist (CLVA) that adopts the patch-wise style discriminator and contrastive reasoning to jointly learn between style images and style instructions. The experiments show that CLVA outperforms baselines on both automatic metrics and human evaluation, and using guided language can improve accessibility without preparing style images. LDIST also supports partial semantic editing, which makes visual applications like image/video effect more controllable for humans. Though our work benefits creative visual applications, there may be a “fake as real” doubt for those manipulated images. To mitigate this issue, we can apply techniques from image forensics [88, 89, 90] to detect the authenticity of an image. Regarding guided instructions, for example, hate speech detection [91, 92, 93, 94] can help to filter out malicious texts and prevent from producing controversial results with ethics concerns.

Please visit project website for more visualization results: https://ai-sub.github.io/ldist/
A  Used Dataset on LDIST

We build a new dataset to evaluate language-based image style transfer (LDIST). As shown in Fig. 6, we collect 14,924 wallpapers from WallpapersCraft [7] as content images (C), including diverse scenes like building, animal, or island. We apply DTD² [4] that provides 5,368 pairs of texture image (S) and textual description (X) as reference styles, such as striped, smeared, or paisley.

B  Human Evaluation

We investigate the quality of LDIST results from the human aspect through Amazon Mechanical Turk (AMT). Fig. 7 illustrates the screenshots of the human ranking task. MTurkers rank the correlation of the LDIST result from each method between the style instruction (vs. Instruction) or the style image (vs. Style). Each MTurker rewards $1.0 and takes a mean of 17 minutes for 6 ranking tasks.

References

[1] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A Neural Algorithm of Artistic Style. arXiv:1508.06576, 2015.
[2] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal Style Transfer via Feature Transforms. NeurIPS, 2017.
[3] Xun Huang and Serge Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. ICCV, 2017.
[4] Chenyun Wu, Mikayla Timm, and Subhransu Maji. Describing Textures using Natural Language. ECCV, 2020.
[5] Panos Achlioptas, Maks Ovsjanikov, Kilichbek Haydarov, Mohamed Elhoseiny, and Leonidas Guibas. ArtEmis: Affective Language for Visual Art. arXiv:2101.07396, 2021.
[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing Textures in the Wild. CVPR, 2014.
[7] WallpapersCraft. https://wallpaperscraft.com
[8] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncell. Image Quality Assessment: From Error Visibility to Structural Similarity. TIP, 2004.
11

[9] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved Techniques for Training GANs. NeurIPS, 2016.

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two-Time-Scale Update Rule Converge to a Local Nash Equilibrium. NeurIPS, 2017.

[11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. CVPR, 2016.

[12] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. GODIVA: Generating Open-Domain Videos from nAtural Descriptions. arXiv:2104.14806, 2021.

[13] Lei Shi, Kai Shuang, Shije Geng, Peng Su, Zhengkai Jiang, Peng Gao, Zuohui Fu, Gerard de Melo, and Sen Su. Contrastive Visual-Linguistic Pretraining. arXiv:2007.13135, 2020.

[14] Ali Al-Sarraf, Bok-Suk Shin, Zezhong Xu, and Reinhard Klette. Ground Truth and Performance Evaluation of Lane Border Detection. ICCVG, 2014.

[15] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. GODIVA: Generating Open-Domain Videos from nAtural Descriptions. arXiv:2104.14806, 2021.

[16] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. GODIVA: Generating Open-Domain Videos from nAtural Descriptions. arXiv:2104.14806, 2021.
[34] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning Deep Representations by Mutual Information Estimation and Maximization. ICLR, 2019.

[35] Sindy Löwe, Peter O’Connor, and Bastiaan S. Veeling. Putting An End to End-to-End: Gradient-Isolated Learning of Representations. NeurIPS, 2019.

[36] Ishan Misra and Laurens van der Maaten. Self-Supervised Learning of Pretext-Invariant Representations. CVPR, 2020.

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748, 2018.

[38] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Multiview Coding. ECCV, 2020.

[39] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination. CVPR, 2018.

[40] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive Learning for Unpaired Image-to-Image Translation. ECCV, 2020.

[41] Lajanugen Logeswaran and Honglak Lee. An Efficient Framework for Learning Sentence Representations. ICLR, 2018.

[42] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. ICML, 2020.

[43] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying Neural Style Transfer. IJCAI, 2017.

[44] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and Controllable Neural Texture Synthesis and Style Transfer Using Histogram Losses. arXiv:1701.08893, 2017.

[45] Shaohua Li, Xinxing Xu, Liqiang Nie, and Tat-Seng Chua. Laplacian-Steered Neural Style Transfer. ACMMM, 2017.

[46] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic Style Transfer for Videos. GCPR, 2016.

[47] Chuan Li and Michael Wand. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. CVPR, 2016.

[48] Alex J. Champandard. Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks. arXiv:1603.01768, 2016.

[49] Yi-Lei Chen and Chiu-Ting Hsu. Towards Deep Style Transfer: A Content-Aware Perspective. BMVC, 2016.

[50] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The Contextual Loss for Image Transformation with Non-Aligned Data. ECCV, 2018.

[51] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual Attribute Transfer through Deep Image Analogy. SIGGRAPH, 2017.

[52] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling Perceptual Factors in Neural Style Transfer. CVPR, 2017.

[53] Gantugs Atarsaikhan, Brian Kenji Iwana, Atsushi Narusawa, Keiji Yanai, and Seiichi Uchida. Neural Font Style Transfer. ICDAR, 2017.

[54] Carlos Castillo, Soham De, Xintong Han, Bharat Singh, Abhay Kumar Yadav, and Tom Goldstein. Son of Zorn’s Lemma: Targeted Style Transfer Using Instance-aware Semantic Segmentation. ICASSP, 2017.

[55] Ahmed Selim, Mohamed Elgharib, and Linda Doyle. Painting Style Transfer for Head Portraits using Convolutional Neural. SIGGRAPH, 2016.

[56] Yongchong Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural Style Transfer: A Review. arXiv:1705.04058, 2017.

[57] Lvmmin Zhang, Yi Ji, and Xin Lin. Style Transfer for Anime Sketches with Enhanced Residual U-net and Auxiliary Classifier GAN. ACCPR, 2017.

[58] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep Photo Style Transfer. CVPR, 2017.

[59] Roey Mechrez, Eli Shechtman, and Lihi Zelnik-Manor. Photorealistic Style Transfer with Screened Poisson Equation. BMVC, 2017.

[60] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A Closed-form Solution to Photorealistic Image Stylization. ECCV, 2018.

[61] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic Style Transfer via Wavelet Transforms. ICCV, 2019.
Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping Autoencoder for Deep Image Manipulation. NeurIPS, 2020.

Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-Time Neural Style Transfer for Videos. CVPR, 2017.

Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Characterizing and Improving Stability in Neural Style Transfer. ICCV, 2017.

Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent Online Video Style Transfer. ICCV, 2017.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. ECCV, 2016.

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. ICML, 2016.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved Texture Networks: Maximizing Quality and Diversity in Feed-forward Stylization and Texture Synthesis. CVPR, 2017.

Chuan Li and Michael Wand. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. ECCV, 2016.

Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stereoscopic Neural Style Transfer. CVPR, 2018.

Shuhui Jiang and Yun Fu. Fashion Style Generator. IJCAI, 2017.

Ming Lu and Hao Zhao and Anbang Yao and Feng Xu and Yurong Chen and Li Zhang. Decoder Network over Lightweight Reconstructed Feature for Fast Semantic Style Transfer. ICCV, 2017.

Samaneh Azadi, Matthew Fisher, Vladimir Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-Content GAN for Few-Shot Font Style Transfer. CVPR, 2018.

Xiao-Chang Liu, Ming-Ming Cheng, Yu-Kun Lai, and Paul L. Rosin. Depth-Aware Neural Style Transfer. SNPAR, 2017.

Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer. CVPR, 2017.

Yongcheng Jing, Yang Liu, Yezhou Yang, Zunlei Feng, Yizhou Yu, Dacheng Tao, and Mingli Song. Stroke Controllable Fast Style Transfer with Adaptive Receptive Fields. ECCV, 2018.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A Learned Representation For Artistic Style. ICLR, 2017.

Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. StyleBank: An Explicit Representation for Neural Image Style Transfer. CVPR, 2017.

Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Diversified Texture Synthesis with Feed-forward Networks. CVPR, 2017.

Hang Zhang and Kristin Dana. Multi-style Generative Network for Real-time Style Transfer. arXiv:1703.06953, 2017.

Tian Qi Chen and Mark Schmidt. Fast Patch-based Style Transfer of Arbitrary Style. NeurIPS WS, 2016.

Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the Structure of a Real-Time, Arbitrary Neural Artistic Stylization Network. BMVC, 2017.

Dmytro Kotovenko, Artiom Sanakoyeu, Pingchuan Ma, Sabine Lang, and Björn Ommer. A Content Transformation Block For Image Style Transfer. CVPR, 2019.

Artiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Björn Ommer. A Style-Aware Content Loss for Real-time HD Style Transfer. ECCV, 2018.

Nicholas Kolkin, Jason Salavon, and Greg Shakhnarovich. Style Transfer by Relaxed Optimal Transport and Self-Similarity. CVPR, 2019.

Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning Linear Transformations for Fast Arbitrary Style Transfer. CVPR, 2019.

TensorFlow Hub. Fast Style Transfer for Arbitrary Styles. magenta/arbitrary-image-stylization. 2021.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. CNN-Generated Images are Surprisingly Easy to Spot...for Now. CVPR, 2020.

Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting Fake News: Image Splice Detection via Learned Self-Consistency. ECCV, 2018.
[90] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging Frequency Analysis for Deep Fake Image Recognition. ICML, 2020.

[91] Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. Deep Learning Models for Multilingual Hate Speech Detection. ICML-PKDD, 2020.

[92] Xiaolei Huang, Linzi Xing, Franck Dernoncourt, and Michael J. Paul. Multilingual Twitter Corpus and Baselines for Evaluating Demographic Bias in Hate Speech Recognition. LREC, 2020.

[93] Nilofar Sani Samghabadi, Parth Patwa, Srinivas PYKL, Prerana Mukherjee, Amitava Das, and Thamar Solorio. Aggression and Misogyny Detection using BERT: A Multi-Task Approach. LREC, 2020.

[94] Bidisha Samanta, Niloy Ganguly, and Soumen Chakrabarti. Improved Sentiment Detection via Label Transfer from Monolingual to Synthetic Code-Switched Text. ACL, 2019.

[95] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. ICCV, 2015.

[96] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692, 2019.

[97] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP, 2019.

[98] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. PMLR, 2019.

[99] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. ICLR, 2015.

[100] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning Correspondence from the Cycle-Consistency of Time. ICCV, 2019.

[101] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. ICCV, 2017.

[102] Lin Wu, Yang Wang, and Ling Shao. Cycle-Consistent Deep Generative Hashing for Cross-Modal Retrieval. TIP, 2018.

[103] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Cycle-Consistent Deep Generative Hashing for Cross-Modal Retrieval. CVPR, 2019.

[104] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. MirrorGAN: Learning Text-to-image Generation by Redescription. CVPR, 2019.

[105] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. ICCV, 2017.

[106] Yingce Xia, Di He, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual Learning for Machine Translation. NeurIPS, 2016.

[107] Wenqi Xian, Patorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. TextureGAN: Controlling Deep Image Synthesis with Texture Patches. CVPR, 2018.

[108] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture Synthesis Using Convolutional Neural Networks. NeurIPS, 2015.

[109] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. NeurIPS, 2014.

[110] Pegah Salehi, Abdullah Chalechale, and Maryam Taghizadeh. Generative Adversarial Networks (GANs): An Overview of Theoretical Model, Evaluation Metrics, and Recent Developments. arXiv:2005.13178, 2020.

[111] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. Science, 2006.

[112] Nathan Somavarapu, Chih-Yao Ma, and Zsolt Kira. Frustratingly Simple Domain Generalization via Image Stylization. arXiv:2006.11207, 2020.

[113] Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi. Image Super-Resolution by Neural Texture Transfer. CVPR, 2019.

[114] Wenjing Wang, Shuai Yang, Jizheng Xu, and Jiaying Liu. Consistent Video Style Transfer via Relaxation and Regularization. TIP, 2019.

[115] Chang Gao, Derun Gu, Fangjun Zhang, and Yizhou Yu. ReCoNet: Real-time Coherent Video Style Transfer Network. arXiv:1807.01197, 2018.

[116] Haozhi Huang, Hao Wang, Wenhao Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-Time Neural Style Transfer for Videos. CVPR, 2017.