A Contrastive Unified Encoding Framework for Sticker Style Editing

Zhihong Ni[1], Chengze Li[2], Hanyuan Liu[3,4], Xueting Liu[2], Tien-Tsin Wong[3,5], Zhenkun Wen[1], and Huisi Wu[†,1]

[1]Shenzhen University, Shenzhen, Guangdong, China
[2]Saint Francis University, Hong Kong SAR, China
[3]The Chinese University of Hong Kong, Hong Kong SAR, China
[4]The City University of Hong Kong, Hong Kong SAR, China
[5]Monash University, Australia

Abstract

Stickers are widely used in digital communication to enhance emotional and visual expressions. The conventional process of creating new sticker pack images involves time-consuming manual drawing, including meticulous color coordination and shading techniques for visual harmony. Learning the visual styles of distinct sticker packs would be critical to the overall process; however, existing solutions usually learn this style information within a limited number of style "domains", or per image. In this paper, we propose a contrastive learning framework that allows the style editing of an arbitrary sticker based on one or a number of style references with a continuous manifold to encapsulate all styles across sticker packs. The key to our approach is the encoding of styles into a unified latent space so that each sticker pack correlates with a unique style latent encoding. The contrastive loss ensures identical style latents within the same sticker pack, while distinct styles diverge. Through exposure to diverse sticker sets during training, our model crafts a consolidated continuous latent style space with strong expressive power, fostering seamless style transfer, interpolation, and mixing across sticker sets. Experiments show compelling style transfer results, with both qualitative and quantitative evaluations confirming the superiority of our method over existing approaches.

CCS Concepts

• ***Applied computing*** → ***Media arts;***

1. Introduction

Stickers have gained immense popularity as a form of visual expression on numerous digital platforms. Users use stickers as a lively and captivating method to personalize information, interactions, and digital content, infusing their exchanges with emotion, humor, and creativity. Traditionally, designing new sticker pack images requires complete manual creation, which is tedious. Artists may spend considerable time designing and aligning the color themes and shading techniques carefully for the overall visual harmony and balance. Clearly, there is an opportunity and substantial benefit for machine learning-powered automation in this area. It would be highly valuable if there is a solution for an artist to easily transfer the *styles*, including shading palettes and textures, to other stickers. Such a solution will provide an efficient way to preview and align the visual styles for multiple stickers and also offer a more intuitive means of gaining inspiration by producing crossovers for stickers with varying appearances and feels.

Several existing solutions may have the potential to realize such an objective. For example, style transfer methods [GEB16, HB17, KSLO19, LLH*21, CWZ*21, WZDB22, ZTD*22, WZZ*23] were proposed to transfer the textural compositions of the style reference to the content image. However, the notion of *style* of stickers differs fundamentally from traditional photographs and paintings. Generic style transfer methods typically focus on transferring local texture patterns and do not capture the higher-level style semantics such as color themes and shading techniques that are necessary for stickers. When applied to stickers, these methods usually result in distorted or inconsistent colors and are difficult to maintain the color and shading consistency for semantic continuous regions of the content image. Unsupervised image-to-image translation [LBK17, HLBK18, CCK*18, LHM*19, CUYH20, BCU*21, MTL*22] may also be used for reference-based image style editing. These methods typically decompose images into two latent manifolds: a *content* space, representing shapes and structures that contribute to subject recognition, and a *style* space, encoding colors, textures, and other properties related to visual styles. High-quality style editing and transfer can be learned through cycle

† Corresponding author

DOI: 10.2312/pg.20241304

consistency [ZPIE17] and adversarial learning [GPAM*14]. However, these methods typically do not naturally support shared style spaces, that is, multiple manifolds, often referred to as *visual domains* [LBK17], are required to be constructed to encode different types of visual styles. This requires large-scale data for each visual domain to support a comprehensive latent space. In our task, the number of such domains could be extensive as we may have thousands of sticker packs. On the other hand, the small number of stickers per set discourages the training of complete style manifolds. Besides, the recent diffusion-based style editing models [LvdWH*23,ZHT*23,SSK*23] have shown high-quality image generation capabilities; but similarly to the other existing methods, they cannot be directly applied to our specific problem to learn the notion of styles for stickers with harmonious shadings and awareness to high-level sticker content semantics.

In this paper, we aim to edit the style of an arbitrary sticker based on one or a number of style references with a continuous manifold to encapsulate all styles across sticker packs. Our key insight is based on the fact that stickers from the same pack share the same visual style, while different sticker packs exhibit different visual styles. This observation is mostly true, as within the same sticker pack the author usually uses the same color schemes, similar shading styles for sticker content semantics. This observation naturally establishes a shared style latent manifold for all sticker packs, which we can effectively learn via contrastive learning. The contrastive supervision of the style space also straightforwardly disentangles the style-specific information from the content. Specifically, we introduce a novel deep learning framework that applies this contrastive design to existing unsupervised image-to-image solutions, facilitating effective and comprehensive learning of the content and style latent spaces via a contrastive style encoder and a content encoder. A contrastive style loss is implemented and applied for learning in the style manifold. Additionally, we design a generator model to reassemble the decomposed latents back to the raster sticker image and apply several reconstruction losses for cycle consistency in both the image and the latent domains. We also propose a style-guided discriminator that reads style latents as additional conditions to enforce realistic and stylistically consistent generator outputs. We have collected more than 200 sets of stickers with different styles for training and evaluation purposes. Our experiment achieves compelling visual results in several tasks, including style editing of arbitrary sticker, sticker sketch coloring, and sticker style interpolation. Furthermore, the results demonstrate that our model can extend its performance to sticker packs that were not part of the training data. We also evaluate our framework quantitatively, which shows that our method outperforms baseline approaches on style translation efficacy and quality metrics. We summarize our contributions in this work as follows:

- We propose a novel contrastive learning framework that allows the style editing of an arbitrary sticker based on one or a number of style references.
- With a large number of sticker packs, our framework learns a continuous manifold to encapsulate all styles across sticker packs, allowing arbitrary style transfer, modification, or interpolation for any sticker.

2. Related Work

2.1. Unsupervised Style Translation

Image-to-image translation involves representation learning and mapping across the so-called *image domains* which represents image groups of similar characteristics. CycleGAN [ZPIE17] first employed cycle consistency, combining with adversarial learning [GPAM*14] for stable unsupervised image translation between domains. UNIT [LBK17] and MUNIT [HLBK18] further learned to decouple the feature representations into *contents* and *styles* for interpretable cross-domain learning, with or without multi-model capabilities. While the learning and generation capability of these methods are decent, they struggled with our complex task for a vast number of sticker packs, necessitating new networks for each sticker pack pair. To handle translation for multiple domains with a single network, [CCK*18] introduced domain-conditioned generation and discrimination. This method was further extended by FUNIT [LHM*19], which learned a style latent to represent the appearances of images. The class condition could be further computed from the class average. While FUNIT has the potential to be applied to our problem, its representation learning does not guarantee a comprehensive style space to support a large number of classes with only a few instances in each class, as will be shown in the evaluation section. TUNIT [BCU*21] further clustered pseudo-domain labels to eliminate the idea of image domains. However, when applied to our task, its representation learning fails to create stable clusters and generate very poor results. [MTL*22] proposed a unified attribute space that achieves continuous and diverse translation across visual domains, but its signed attribute vector can be extremely complex for more than 100 explicit visual domains and may also suffer from the curse of dimensionality upon training and inference. Generally, defining each sticker pack into an individual visual domain causes extensive complexity in model training. While FUNIT and TUNIT relaxed the strict barrier between visual domains, their supervision may lead to incomplete learning of visual appearances. In sharp contrast, we explore the visual similarity and dissimilarities across sticker packs to leverage contrastive learning for efficient and comprehensive learning, without being burdened by the definition of visual domains.

2.2. Style Transfer

Style transfer involves altering the style of an image while preserving its content structure. [GEB16] introduced convolutional neural networks to style transfer via optimization in feature spaces. AdaIN [HB17] achieved arbitrary style transfer by re-adjusting channel-wise feature statistics. However, the Gram matrix commonly used in their learning objectives only captures global feature correlations but lacks awareness to image local structures and semantics. When applied to stickers, this often results in a lack of semantic coherence, leading to obvious distorted and inharmonious textures. AdaAttN [LLH*21] introduced attention and normalization modules to better control local features, yet it still struggled with color and texture distortion issues. [CWZ*21] introduced contrastive learning into style transfer by minimizing differences in style translation within the same content or style image and vice versa. CCPL [WZDB22] took advantage of contrastive learning within the image and video neighbor patches before and after style

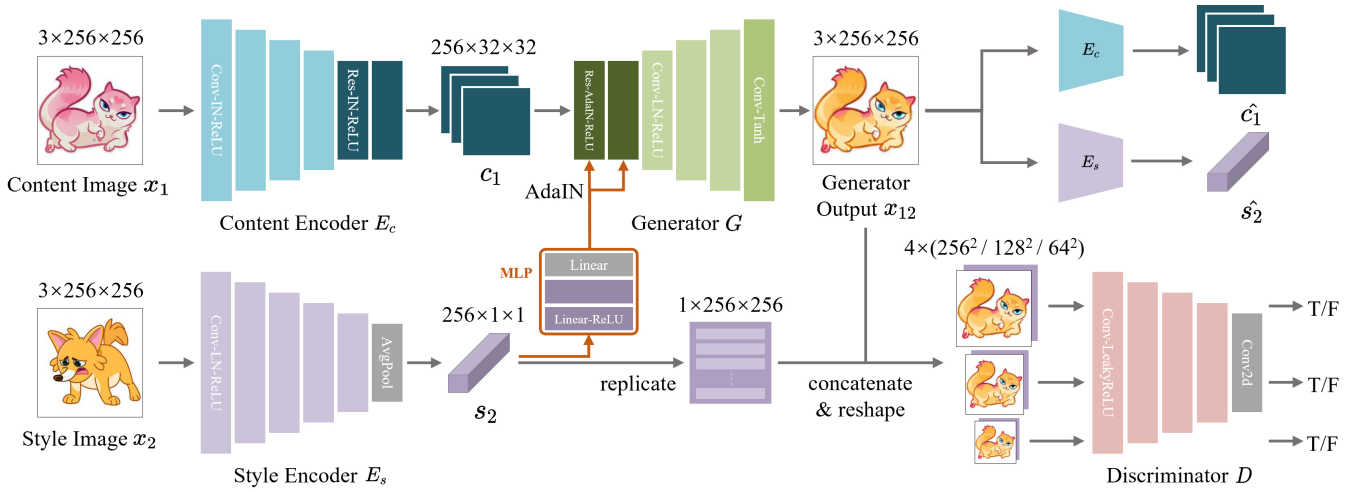

Figure 1: *Model overview. Our framework consists of four modules: (1) a style encoder E_s to extract a style latent; (2) a content encoder E_c to extract a content latent; (3) a generator model G, to reassemble an output image from a tuple of content and style lantents; (4) a style-guided multiscale discriminator D that enforces the generator output to be aligned with its designated input styles. For reconstruction supervision, x_1 and x_2 will be the same and the computational graph will conclude at the generator output.*

transfer for improved quality and consistency for video style transfer. CAST [ZTD*22, ZTD*23] shared a similar idea with latent disentanglement of image-to-image translation methods and used contrastive learning to support comprehensive learning on latent style space. While these methods achieve better style understanding thanks to the contrastive objectives, they are mainly tailored for style transfer, which translates and discriminates only against the artistic or realistic domain. Moreover, their contrastive learning of visual styles is estimated per image, omitting style similarities within a single sticker pack. MicroAST [WZZ*23] introduced a mini-batch contrastive loss to handle relationships between various styles within a training batch, but they still fail to consider style similarities within a single sticker pack, which could be much more challenging due to the amount and diversity in appearances. While color transfer methods, such as DNCM [KLZ*23], can manipulate the overall color tone of images to some extent, the underlying deterministic color mapping models cannot recognize visual semantics and propagate satisfactory coloring for local details. Besides, even though diffusion models show potential in generating high-quality images, they are not directly applicable to our specific problem and require customization in the denoise diffusion pipeline for style transfers. Tailored diffusion-based style transfer model including [LvdWH*23, ZHT*23, SSK*23] have difficulties in maintaining structure or handling contrast with these methods, as they primarily focus on general style transfer. This focus differs from our goal of achieving visual harmony and smooth shading for stickers based on a reference. Furthermore, diffusion-based methods typically incur significantly higher computational costs. In practice, these methods still lead to unrealistic color distribution and color bleeding artifacts, which will be examined in more detail in the evaluation section.

2.3. Contrastive Learning

Contrastive learning has been widely applied in self-supervised representation learning. It focuses on learning representations by distinguishing between similar and dissimilar instances, thereby mapping similar instances closer in the feature space while pushing apart dissimilar ones. This approach has been utilized in various applications, including image dehazing [WQL*21], object detection [SLC*21], and image generation [KP20, LGC*21]. In this work, we aim at the practical uses of contrastive learning to learn the styles of stickers effectively. A similar prior work CUT [PEZZ20] applied a patch-based contrastive loss to enforce consistency between input and output image patches. However, the focus on patch-level consistency may overlook broader stylistic coherence across the entire sticker. IEST [CWZ*21] uses feature statistics including mean and standard deviation as style priors Contrastive supervision is employed to pull images with similar style statistics closer in the feature space while pushing images with different styles apart, thereby associating images that share the same style. Nonetheless, its reliance on feature statistics from pre-trained networks may not fully capture the unique and diverse styles present in sticker packs.

3. Method

3.1. Overview

We aim at a learning-based framework capable of re-rendering any sticker image in a target reference style. In this work, we define the *style* as a characteristic attribute of the sticker packs, which remain consistent within the same sticker pack and diverge across different sticker packs. This specific formulation of style supports a contrastive and disentangled feature representation of stickers, where any possible styles of all sticker packs will be encoded in a uni-

form latent space S. Furthermore, as the style latent s is constant for the stickers in the same sticker pack, the remainder will naturally fall into the content latent c to encode all style-independent information of stickers. For supervision, we apply contrastive learning on s, where we minimize the distance of the decomposed style latent s for stickers from the same set and maximize the distance for stickers from different sets. When a large number of sticker packs are fed during training, contrastive learning of decomposition automatically supports a unified style space S, allowing style transfer, alteration, or interpolation of arbitrary stickers.

3.2. Network Architecture

As shown in Fig. 1, our framework consists of four modules: (1) a style encoder E_s to extract the style latent s_2 of the style input x_2; (2) a content encoder E_c to extract the content latent c_1 of the content input x_1; (3) a generator model G, to reassemble an output image x_{12} from c_1 and s_2; (4) a style-guided adversarial discriminator D which receives latent style s_2 as an additional condition to supervise G to produce a high quality sticker according to s_2.

3.2.1. Style Encoder

We devise the style encoder E_s to generate the style latent s for any sticker image. Specifically, we construct a contrastive supervision to learn a continuous unified style space S using the style encoder. We design the encoder E_s using five convolutional blocks. After the convolutions, we apply global average pooling on the two spatial dimensions to obtain a 256-dimensional vector as the style latent s. We use ReLU activation and layer normalization [BKH16] for stable training. To enable contrastive learning in the S manifold, we use a metric based on cosine similarity for supervision. We explain the objectives in the following subsection.

3.2.2. Content Encoder

We employ a content encoder E_c to extract the content latent c of an input image. The content encoder consists of four convolutional blocks with feature processing and condensing. Subsequently, two residual blocks are placed to enhance the representation power of the features. Throughout the content encoder, we apply instance normalization to discard style-related information [HB17] and activate normalized features with the ReLU function. The extracted content latent c is with a resolution of $256 \times 32 \times 32$. Ideally, it is expected to represent the structural semantics of the input, and we will explain the disengtanglement in the next subsection.

3.2.3. Generator

The generator G is a style-conditioned deconvolutional network that reads the content c and the style s to construct a raster image output. When c and s are extracted from the same input, G can be seen as a reconstruction network; while c and s come from different images, G can be seen as a style translation network. Specifically, G reads from c and first uses two residual AdaIN blocks [HB17] to enable style conditioning from s, where s is passed with a three-layer MLP. Following these residual blocks, there are three upsampling deconvolutional blocks. These upsampling blocks utilize deconvolutions with layer normalizations and ReLU activations. The last convolutional block, without the inclusion of a normalization layer, incorporates a Tanh activation function to generate the $3-$ channel color image.

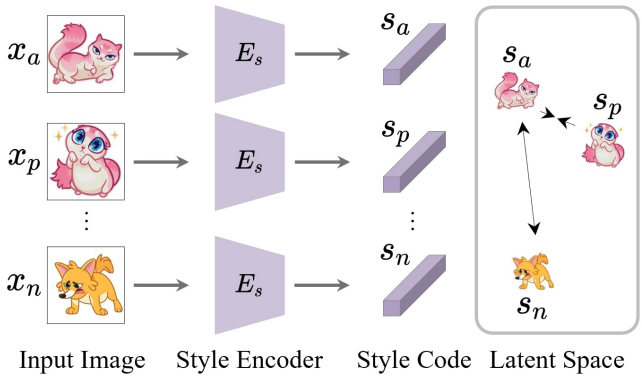

Figure 2: *Our style encoder E_s extracts a style latent s_a from an anchor image x_a. The positive sample x_p should belong to the same sticker pack as x_a, and vice versa for the negative sample x_n.*

3.2.4. Style-guided Discriminator

We employ a multi-scale discriminator model D, similarly as stated in [WLZ*18], to guide the generator G toward a high-quality output. The key design of our discriminator D is to incorporate the conditional GAN mechanism to receive additional information from a style latent s extracted by the style encoder. Ideally, as the contrastive learning of the sticker pack style space S approaches convergence, the discriminator model D will effectively learn from the style space S, forcing the generator G to comply with the style input s. This allows our generator to generate high-quality output with a consistent style of the style input. Specifically, the discriminator model receives three different input scales: 256^2, 128^2, and 64^2. At the 256^2 scale, the style latent s is first spatially replicated to construct a 256^2 sized matrix. After that, the replicated latent is concatenated with the generator output channel-wise, to form a 4-channel discriminator input. For the other two scales, the concatenation is similar, with the generator output bilinearly reshaped and the latent reshaped via a linear weight. The discriminator is a patchGAN model [ZPIE17] containing 4 convolutional blocks, with LeakyReLU activations with $\alpha = 0.2$. For different scales, the output shape will also be different. It is also worth noting that normalization is not used in the model D.

3.3. Objectives

We train the overall framework within the scope of three different learning tasks: contrastive learning, style crossover, and image reconstruction . We will explain the training objectives for each task as follows.

3.3.1. Contrastive Learning of the Style Manifold

We aim to learn a unified and continuous latent style space S capturing the styles of stickers and aligning the style latents per sticker pack. As shown in Fig. 2, given an anchor image x_a sampled from

Figure 3: *Qualitative comparisons on seen style images (top 3 rows) and unseen style images (bottom 2 rows).*

all stickers, we randomly select an image from the same sticker pack as the positive sample x_p and an image from a different pack as the negative sample x_n. We minimize the dissimilarity between the style of the anchor sample s_a and the style of the positive sample s_p, while maximizing the dissimilarity with the style of the negative sample s_n. Here, we use cosine similarity to estimate the dissimilarity and construct the contrastive objective as follows:

$$\begin{aligned}\mathcal{L}_c = \mathbb{E}_{x_a,x_p,x_n}[&\mathrm{sim}^2\left(E_s\left(x_a\right),E_s\left(x_n\right)\right)\\&+\max\left(1-\mathrm{sim}\left(E_s\left(x_a\right),E_s\left(x_p\right)\right),0\right)^2],\end{aligned} \tag{1}$$

where $\mathrm{sim}(x,y)=\frac{x\cdot y}{\|x\|\|y\|}$ is vector cosine similarity.

3.3.2. Style Crossover

Inpired by [HLBK18], we design the style crossover task for latent-level reconstruction and cycle consistency. Specifically, after sampling an input tuple x_1 and x_2 from the dataset, we extract their content and style latents, namely c_1, c_2, s_1 and s_2. We then generate two style crossover images $x_{12}=G(c_1,s_2)$ and $x_{21}=G(c_2,s_1)$ by shuffling the latents. After that, we decompose these crossover images again and ensure the cycle consistency of the decomposed latents. We first define the cycle consistency objectives:

$$\begin{aligned}\mathcal{L}_{\mathrm{cyc}}^{c_1}&=\mathbb{E}_{x_1,x_2}\left[\|E_c(G(E_c(x_1),E_s(x_2)))-E_c(x_1)\|_1\right],\\ \mathcal{L}_{\mathrm{cyc}}^{s_2}&=\mathbb{E}_{x_1,x_2}\left[\|E_s(G(E_c(x_1),E_s(x_2)))-E_s(x_2)\|_1\right].\end{aligned} \tag{2}$$

These L_1 cycle consistency objectives are also applied to c_2 and s_1 as $\mathcal{L}_{\mathrm{cyc}}^{c_2}$ and $\mathcal{L}_{\mathrm{cyc}}^{s_1}$ in a similar way. In addition, we also employ adversarial loss for a higher fidelity. Here, we use the LSGAN objective [MLX*17]:

$$\begin{aligned}\mathcal{L}_{\mathrm{adv}}^{x_{12}}=\mathbb{E}_{x_1,x_2}&\left[\mathbf{0}-D\left(G(E_c(x_1),E_s(x_2)),E_s(x_2)\right)\right]^2\\&+\mathbb{E}_{x_2}\left[\mathbf{1}-D\left(x_2,E_s(x_2)\right)\right]^2,\end{aligned} \tag{3}$$

where $D(\cdot,\cdot)$ receives from an image and a style latent as input to produce multi-scale patchGAN predictions. $\mathbf{0}$ and $\mathbf{1}$ are the matrix of all 0-s or 1-s with the corresponding patchGAN output scale. This adversarial objective also happens on x_{21} as $\mathcal{L}_{\mathrm{adv}}^{x_{21}}$.

3.3.3. Image Reconstruction

For a single image, we also want to ensure pixel-wise identical reconstruction after decomposing the input with E_c and E_s and then reassemble them again with G. Here we shall use the previous sampled image x_1 and x_2 from the dataset to achieve so:

$$\mathcal{L}_{\mathrm{rec}}^{x_1}=\mathbb{E}_{x_1}\left[\left\|G\left(E_c(x_1),E_s(x_1)\right)-x_1\right\|_1\right]. \tag{4}$$

We can also compute a similar loss $\mathcal{L}_{\mathrm{rec}}^{x_2}$. Note that we do not involve adversarial learning for reconstruction.

3.3.4. Overall Objectives

We jointly train our model, including encoders, generators, and discriminators, to optimize the following combined objectives:

$$\begin{aligned}\min_{E_c,E_s,G}\max_{D}\mathcal{L}_{\mathrm{total}}=&(\mathcal{L}_{\mathrm{adv}}^{x_{12}}+\mathcal{L}_{\mathrm{adv}}^{x_{21}})+\mathcal{L}_c+\lambda_x(\mathcal{L}_{\mathrm{rec}}^{x_1}\\&+\mathcal{L}_{\mathrm{rec}}^{x_2})+\lambda_c(\mathcal{L}_{\mathrm{cyc}}^{c_1}+\mathcal{L}_{\mathrm{cyc}}^{c_2})+\lambda_s(\mathcal{L}_{\mathrm{cyc}}^{s_1}+\mathcal{L}_{\mathrm{cyc}}^{s_2}),\end{aligned} \tag{5}$$

where $\lambda_x=3$, $\lambda_s=3$ and $\lambda_c=1$ are loss multipliers.

Methods	Seen		Unseen	
	FID↓	IS↑	FID↓	IS↑
AdaIN	171.01	2.355±0.109	166.41	2.544±0.117
FUNIT	150.11	2.789±0.117	152.51	2.808±0.109
AdattN	159.63	2.464±0.089	152.18	2.648±0.117
CCPL	153.47	2.639±0.129	151.32	2.747±0.079
CAST	162.45	2.563±0.165	156.18	2.607±0.117
DNCM	-	-	163.80	2.847±0.152
MicroAST	162.37	2.622±0.089	158.06	2.687±0.073
Ours	**141.17**	**2.809±0.124**	**133.46**	**2.911±0.104**

Table 1: *Quantitative comparison on sticker style transfer.*

Methods	MAE ↓
AdaIN	0.0962
FUNIT	0.0785
AdaAttN	0.0741
CCPL	0.0801
CAST	0.0652
DNCM	-
MicroAST	0.0607
Ours	**0.0548**

Table 2: *Quantitative comparison on the reconstruction of stickers from the decomposed content and style latents.*

4. Experiments and Discussions

4.1. Experimental Settings

4.1.1. Implementation Details

Our model is implemented based on PyTorch and trained on a NVIDIA RTX 3090 GPU. In the training process, we used the Adam optimizer [KA*15] with β_1 set to 0.5, β_2 set to 0.999, and learning rate set to 0.0001. We selected a batch size of 8 to train our model for 35 epochs.

4.1.2. Datasets

Our dataset comprises over 200 sticker sets sourced from the internet, each resized to a consistent 256 × 256 dimensions. We construct two primary datasets for evaluation. Initially, we build the *unseen-style* dataset by leaving 20 packs withheld from model training. Subsequently, around 30% of the remaining sticker images formed the *seen-style* dataset. To ensure the distribution alignment of the seen-style dataset with the remaining 70% used for training and validation, the data splitting occurred at the image level rather than the pack level. We individually perform data augmentation on the training, validation, and test dataset. In all, the training and validation dataset contains 59,080 images, sharing the same distribution as the seen-style dataset of 3,692 images; while the unseen-style dataset contains 1,380 images.

4.2. Evaluation and Comparison

We compare qualitatively and quantitatively with various state-of-the-art approaches. We include the style transfer methods AdaIN [HB17], AdaAttN [LLH*21], CCPL [WZDB22], CAST [ZTD*22]. MicroAST [WZZ*23]. We also compare with the color transfer method DNCM [KLZ*23] and the unsupervised image translation method FUNIT [LHM*19] as our competitors. In this evaluation, we mainly experiment with the style transfer tasks, as it is the main use case of our proposed methodology. We achieve this by extracting the latents from a pair of content and style images and generating the stylized output through the generator. This evaluation is conducted on both the seen-style dataset and the unseen-style dataset.

4.2.1. Qualitative Evaluation

We demonstrate the qualitative comparison results on Fig. 3. We observe that DNCM may be successful in manipulating the rough color tint of the image, but they usually cannot achieve correct color and shading compositions after color transfer (1st, 2nd, 3rd, and 5th rows). AdaIN achieves arbitrary style transfer by readjusting channel-wise feature statistics. However, the Gram matrix used to estimate feature correlation often produces stickers with mixed styles. This leads to unclear content details and color bleeding (1st, 3rd, and 5th rows). Although FUNIT retains the structure of the content image through consistency constraints, it cannot guarantee a comprehensive style space to support a much larger number of classes. In some cases, the style distribution may be difficult to achieve noticeable style transfer (1st, 3rd, and 5th rows). AdaAttN is limited to the internal style statistics of a single artistic image. This may cause color distortions in the results (1st, 3rd, and 5th rows). CCPL defines a relaxed constraint on local image patches. Without global content consistency constraints, it may exhibit unrealistic color distributions (1st, 3rd, and 4th rows) and unnecessary halos or artifacts around sticker edges (3rd and 4th rows). CAST treats each image as an individual style, omitting style similarities within a single sticker pack. This can lead to color distortions in some cases (2nd and 3rd rows) and may cause disappearance of internal structure lines (3rd and 5th rows). MicroAST faces the same problem of omitting style similarities within a single sticker pack, which leads to unrealistic color distributions and distortions (1st, 3rd, and 5th rows).

4.2.2. Quantitative Evaluation

We also perform a quantitative evaluation on the style transfer task. For both the seen-style dataset and the unseen-style dataset, we randomly select 40 content images and 40 style images from different sticker packs to form 1,600 style transfer outputs. We employ two metrics: FID [HRU*17] to estimate the distribution gap against real sticker images and IS [SGZ*16] to estimate the overall quality of style transfer. Note that for the color transfer model DNCM is not learning-based and thus we skip its evaluation on seen style stickers. In addition, we conduct a quantitative evaluation on the reconstruction of stickers by generating a sticker back from its decomposed content and style latents without other modifications. For this task, we select 1,000 images from the test dataset and calculate the mean absolute error (MAE) between the original and reconstructed images. We show the statistics in Table 1 and Table 2. Based on the statistics, our method outperforms the other methods in all metrics in all datasets. We believe the superiority mainly comes from

Figure 4: *Ablation study results on various loss functions.*

Methods	Realism ↑	Style Obedience ↑
AdaIN	0.437	1.378
FUNIT	5.567	3.647
AdattN	2.765	2.773
CCPL	2.647	3.744
CAST	2.206	3.609
DNCM	5.453	3.130
MicroAST	3.055	3.659
Ours	**5.869**	**6.059**

Table 3: *User study statistics on style transfer.*

our learning scheme of the latent style space, providing a thorough understanding of sticker style semantics. Thus, we observe a much lower FID than all existing methods. Furthermore, our conditional GAN design also improved generation quality, resulting in a higher margin of IS compared to all methods except FUNIT, which also leverages another variant of the conditional GAN design. The quantitative results on reconstruction also demonstrate the effectiveness of our model in preserving both content and style semantics.Additionally, because our method is GAN-based, it achieves faster performance compared to diffusion-based methods.

4.2.3. User Study

We also conduct a user study to evaluate the style transfer performance of our model from a subjective perspective. First, we evaluate the realism of the style transfer results. Specifically, we randomly select 20 content-style pairs from our test dataset to create 20 different style transfer outputs with our method and the competitors. Notably, participants are solely tasked with ranking the realism, without access to the input images. The participants evaluate the quality of results from eight different methods, presented in a randomized sequence. The user-assigned rankings are then transformed into scores ranging from 0 to 7, with higher scores denoting superior outcomes. After that, we also evaluate the obedience of the reference style by presenting another 20 style transfer results together with the input content and style images. The ranking is conducted in a similar way. This user study involves 30 participants, and we present the average ratings in Table 3. According to the statistics, our approach is the most favorable among our participants.

Ablation	Seen		Unseen	
	FID↓	IS↑	FID↓	IS↑
W/o $\mathcal{L}^s_{\text{cyc}}$	144.62	2.745±0.098	140.11	2.883±0.115
W/o $\mathcal{L}^c_{\text{cyc}}$	146.08	2.866±0.187	148.87	3.063±0.154
W/o $\mathcal{L}_{\text{rec}}$	160.27	2.618±0.119	161.38	2.706±0.061
W/o model D	149.65	2.865±0.098	145.29	2.964±0.089

Table 4: *Statistics of the ablation study on model design.*

4.2.4. Ablation Study

We further conduct an ablation study by removing specific models or loss terms. We illustrate the style transfer outputs with different model designs in Fig. 4. When we disable cycle consistency $\mathcal{L}^c_{\text{cyc}}$ on the content latent, the model fails to preserve the structural information in the output. Similarly, the absence of style latent cycle consistency $\mathcal{L}^s_{\text{cyc}}$ leads to inconsistent shading. Notably, the image reconstruction loss $\mathcal{L}_{\text{rec}}$, commonly employed in image-to-image translation methods, proves crucial in our context. Its omission causes the GAN model to hallucinate unexpected details and textures that lack correspondence in the content image. The significance of the contrastive loss $\mathcal{L}_c$ is evident. Its absence results in an unconstrained style space. This, in turn, leads to unsuccessful decoupling in both content and style spaces, thus impeding style transfer. Moreover, our style-guided discriminator model is of great importance. Without it, the generator risks generating vibrant style transfers, yielding disharmonious and subdued color palettes in the output. The absence of style guidance in the style generator is also notable. Without such guidance, the generator loses its ability to faithfully adhere to style references, potentially leading to mode collapse during style learning. We proceed with a quantitative evaluation of the ablation study, and the results are summarized in Table 4. For brevity, we omit assessments related to contrastive style encoding and the style-guided discriminator, as these models converge into failure modes during the training process.

4.3. Applications

4.3.1. Sticker Sketch Coloring

Our approach can be extended to transfer the color palettes of stickers to arbitrary sketch images without further training. We illustrate

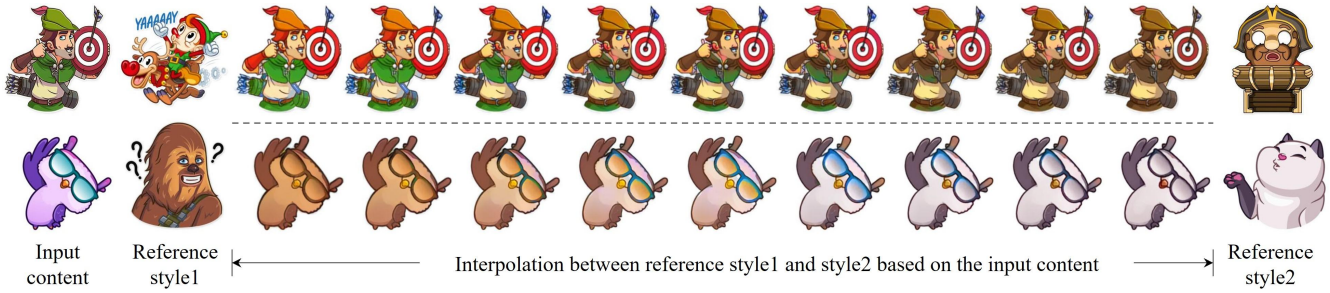

Figure 5: *Sticker style interpolation. The leftmost images are the content, with their styles linearly interpolated between references.*

Figure 6: *Application on sticker sketch coloring.*

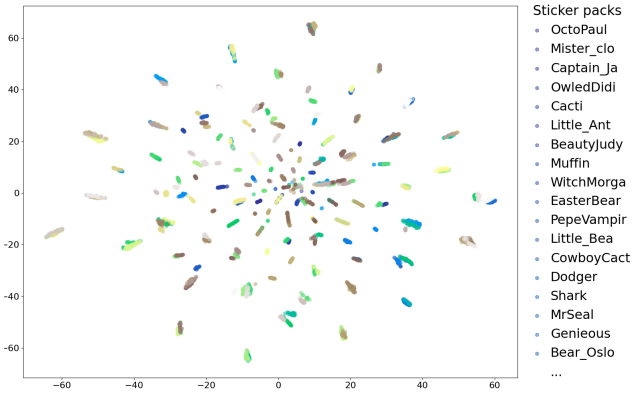

Figure 7: *2-D representation of the style latent using t-SNE for 200 sticker packs. Stickers from the same pack are represented by the same color. Similar packs, measured by cosine similarity, are positioned closer together. Note that only a subset of the sticker categories is listed in the legend due to space limitations.*

our results in Fig. 6, where the sketch images are used as content images. Our model shows decent generalization ability on these out-of-distribution sketch images, where the reference color styles are propagated with visual harmony and correspondence to the content images.

4.3.2. Sticker Style Interpolation

As stated above, contrastive learning enables a comprehensive uniform style space S. Interestingly, we observe good transitions in this style space, enabling a continuous interpolation between different sticker styles. We show two pairs of examples in Fig. 5, where the style transfer is performed by gradually reweighting the interpolation between a pair of style latents. This allows us to seamlessly translate the style from one reference sticker to the other for better adaptation and creativity.

4.4. Exploration of the Style Latents

4.4.1. Latent Space Visualization

In Fig. 7, we first explore the latent space learned by the style encoder. We use t-SNE to visualize the latent style in a two-dimensional space. It can be seen that stickers from the same pack are grouped together in the style latent space. This demonstrates that our contrastive learning approach can effectively learn the stylistic features of different sticker packs, mapping stickers of similar styles to nearby regions. The smooth, continuous manifold structure of the latent space indicates that the encoder successfully captures meaningful style representations. This enables the interpolation and generation of new sticker styles within the space. It is worth noting that as we optimize the style latent space with the cosine-based metric. Thus, the polar coordinate is more representative of the visual styles than the Euclidean distances in this 2D space.

4.4.2. Random Style Sampling

To further evaluate the effectiveness of our approach, we evaluate whether the contrastive learning manages to support a comprehensive style latent space, where most of the space should be able to render visually pleasant stickers. To do so, we draw random style latents with Gaussian sampling with the mean and variance computed from all stickers across 200 packs in our training set. Even though the learned latents are not strictly following a Gaussian distribution, this visualization may still be able to reveal the internals

Figure 8: *Random sticker generation from a given content image. The styles are sampled from an estimated gaussian distribution of all sticker pack styles.*

Figure 9: *Random sticker generation from a given content image. The styles are sampled from an estimated gaussian distribution of one sticker pack styles.*

of the style latent space. As shown in Fig. 8, random styles will also lead to visually pleasant outputs, further proving the effectiveness of contrastive learning of the style encoder.

4.4.3. Within-Set Style Sampling

We also perform the same task *within a specific sticker set*, that is, to sample a random sticker from the Gaussian sampling with statistics within a single sticker set. To evaluate whether the visual styles are stable around a specific sticker set, we also perform the truncation trick [KLA*20] with different cutoff thresholds on the random style latent to investigate the variations in the generated visual styles. As shown in Fig. 9, the learned styles are mostly stable, indicating good localities of our learned latents targeting a specific sticker style.

4.5. Limitation

Our method achieves satisfactory results when reference images exhibits practical color palettes and shading styles, even if they deviate from the existing dataset, due to the encoder's generalization. However, for styles significantly outside the expected distribution (e.g., overly saturated colors, shading-free sketches), our results

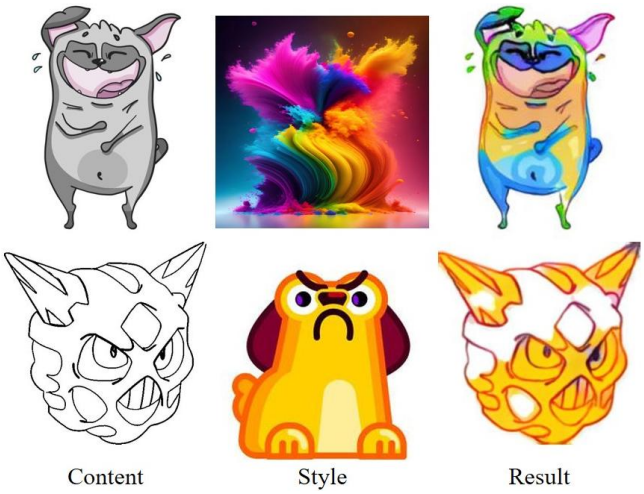

Figure 10: *Failure cases occurred when the style images had overly saturated colors, or when shading-free sketches were used as the content image.*

may suffer from poor texture integration or reduced style fidelity. The generalization may also break when the content contains no grayscale information. Suppose the content image is lacking gradients (e.g., containing solid black or white blobs); it may be difficult for our generator to propagate the target style palette to the input. This is due to the lack of relevant data in our dataset, which limits the ability to encode such styles with our style encoder. In the future, we may discover the potential to solve this problem by introducing more data in our training, as well as studying an advanced mechanism in better decoupling the content and styles.

5. Conclusion

In this work, we propose a novel framework for sticker style editing and translation. We utilize a continuous manifold to encapsulate all styles across sticker packs, encoding the styles into a uniform latent space. Identical style latents within the same sticker pack, while distinct styles diverge. We construct a simple and effective contrastive supervision that minimizes the distance of the style latent of the same sticker pack and maximizes the distance of the style latent of different sticker packs. Extensive experimental results show that our proposed method has superior style transfer results compared to state-of-the-art methods. In the future, we hope to extend our method to other visual tasks.

Acknowledgements

This work was supported in part by grants from the National Natural Science Foundation of China (No. 62273241), the Natural Science Foundation of Guangdong Province, China (No. 2024A1515011946), the Major Project of the New Generation of Artificial Intelligence (No. 2018AAA0102900), the Grant for Marshall Laboratory of Biomedical Engineering (SZU), the Grant for Shenzhen Institute of Artificial Intelligence and Robotics for Society, and the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. UGC/FDS11/E02/23).

References

[BCU*21] Baek K., Choi Y., Uh Y., Yoo J., Shim H.: Rethinking the truly unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14154–14163. 1, 2

[BKH16] Ba J. L., Kiros J. R., Hinton G. E.: Layer normalization. *arXiv preprint arXiv:1607.06450* (2016). 4

[CCK*18] Choi Y., Choi M., Kim M., Ha J.-W., Kim S., Choo J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 8789–8797. 1, 2

[CUYH20] Choi Y., Uh Y., Yoo J., Ha J.-W.: Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 8188–8197. 1

[CWZ*21] Chen H., Wang Z., Zhang H., Zuo Z., Li A., Xing W., Lu D., et al.: Artistic style transfer with internal-external learning and contrastive learning. *Advances in Neural Information Processing Systems 34* (2021), 26561–26573. 1, 2, 3

[GEB16] Gatys L. A., Ecker A. S., Bethge M.: Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2414–2423. 1, 2

[GPAM*14] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.: Generative adversarial nets. *Advances in Neural Information Processing Systems 27* (2014). 2

[HB17] Huang X., Belongie S.: Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 1501–1510. 1, 2, 4, 6

[HLBK18] Huang X., Liu M.-Y., Belongie S., Kautz J.: Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 172–189. 1, 2, 5

[HRU*17] Heusel M., Ramsauer H., Unterthiner T., Nessler B., Hochreiter S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems 30* (2017). 6

[KA*15] Kinga D., Adam J. B., et al.: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (2015), vol. 5, San Diego, California;, p. 6. 6

[KLA*20] Karras T., Laine S., Aittala M., Hellsten J., Lehtinen J., Aila T.: Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8110–8119. 9

[KLZ*23] Ke Z., Liu Y., Zhu L., Zhao N., Lau R. W.: Neural preset for color style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 14173–14182. 3, 6

[KP20] Kang M., Park J.: Contragan: Contrastive learning for conditional image generation. *Advances in Neural Information Processing Systems 33* (2020), 21357–21369. 3

[KSLO19] Kotovenko D., Sanakoyeu A., Lang S., Ommer B.: Content and style disentanglement for artistic style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 4422–4431. 1

[LBK17] Liu M.-Y., Breuel T., Kautz J.: Unsupervised image-to-image translation networks. *Advances in Neural Information Processing Systems 30* (2017). 1, 2

[LGC*21] Liu R., Ge Y., Choi C. L., Wang X., Li H.: Divco: Diverse conditional image synthesis via contrastive generative adversarial network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 16377–16386. 3

[LHM*19] Liu M.-Y., Huang X., Mallya A., Karras T., Aila T., Lehtinen J., Kautz J.: Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 10551–10560. 1, 2, 6

[LLH*21] Liu S., Lin T., He D., Li F., Wang M., Li X., Sun Z., Li Q., Ding E.: Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 6649–6658. 1, 2, 6

[LvdWH*23] Li S., van de Weijer J., Hu T., Khan F. S., Hou Q., Wang Y., Yang J.: Stylediffusion: Prompt-embedding inversion for text-based editing. *arXiv preprint arXiv:2303.15649* (2023). 2, 3

[MLX*17] Mao X., Li Q., Xie H., Lau R. Y., Wang Z., Paul Smolley S.: Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 2794–2802. 5

[MTL*22] Mao Q., Tseng H.-Y., Lee H.-Y., Huang J.-B., Ma S., Yang M.-H.: Continuous and diverse image-to-image translation via signed attribute vectors. *International Journal of Computer Vision 130*, 2 (2022), 517–549. 1, 2

[PEZZ20] Park T., Efros A. A., Zhang R., Zhu J.-Y.: Contrastive learning for unpaired image-to-image translation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16* (2020), Springer, pp. 319–345. 3

[SGZ*16] SALIMANS T., GOODFELLOW I., ZAREMBA W., CHEUNG V., RADFORD A., CHEN X.: Improved techniques for training gans. *Advances in Neural Information Processing Systems 29* (2016). 6

[SLC*21] SUN B., LI B., CAI S., YUAN Y., ZHANG C.: Fsce: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 7352–7362. 3

[SSK*23] SINHA A., SUN B., KALIA A., CASANOVA A., BLANCHARD E., YAN D., ZHANG W., NELLI T., CHEN J., SHAH H., ET AL.: Text-to-sticker: Style tailoring latent diffusion models for human expression. *arXiv preprint arXiv:2311.10794* (2023). 2, 3

[WLZ*18] WANG T.-C., LIU M.-Y., ZHU J.-Y., TAO A., KAUTZ J., CATANZARO B.: High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 8798–8807. 4

[WQL*21] WU H., QU Y., LIN S., ZHOU J., QIAO R., ZHANG Z., XIE Y., MA L.: Contrastive learning for compact single image dehazing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 10551–10560. 3

[WZDB22] WU Z., ZHU Z., DU J., BAI X.: Ccpl: contrastive coherence preserving loss for versatile style transfer. In *European Conference on Computer Vision* (2022), Springer, pp. 189–206. 1, 2, 6

[WZZ*23] WANG Z., ZHAO L., ZUO Z., LI A., CHEN H., XING W., LU D.: Microast: towards super-fast ultra-resolution arbitrary style transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2023), vol. 37, pp. 2742–2750. 1, 3, 6

[ZHT*23] ZHANG Y., HUANG N., TANG F., HUANG H., MA C., DONG W., XU C.: Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2023), pp. 10146–10156. 2, 3

[ZPIE17] ZHU J.-Y., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 2223–2232. 2, 4

[ZTD*22] ZHANG Y., TANG F., DONG W., HUANG H., MA C., LEE T.-Y., XU C.: Domain enhanced arbitrary image style transfer via contrastive learning. In *ACM SIGGRAPH 2022 Conference Proceedings* (2022), pp. 1–8. 1, 3, 6

[ZTD*23] ZHANG Y., TANG F., DONG W., HUANG H., MA C., LEE T.-Y., XU C.: A unified arbitrary style transfer framework via adaptive contrastive learning. *ACM Transactions on Graphics* (2023). 3