Adversarial Training Inspired Self-attention Flow for Universal Image Style Transfer

Kaiheng Dang[1], Jianhuang Lai[1,2,3(✉)], Junhao Dong[1], and Xiaohua Xie[1,2,3]

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
{dangkh,dongjh8}@mail2.sysu.edu.cn, {stsljh,xiexiaoh6}@mail.sysu.edu.cn
[2] Guangdong Key Laboratory of Information Security Technology, Guangzhou, China
[3] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Guangzhou, China

Abstract. Flow-based model receives more and more attention and has been recently applied to image style transfer. While these methods can achieve splendid performance, there remains a problem that the stacked convolutions are inefficient and cannot focus on valuable features. Starting with training an adversarial robust model, we find that no matter in the perceptual loss network or the transfer model, robust features are beneficial for performing better universal style transfer (UST) results. Based on this initial conclusion, we improve the current Glow model by applying self-attention mechanism with three different blocks using ViT, non-local and involution, respectively. Designed feature extraction blocks can capture more valuable deep features with fewer parameters, making Glow more effective and efficient in UST. Our improved Glow can generate artistic images that look nicer and more stable. Both visual results and quantitative metrics are compared to prove that our improvement makes Glow more suitable for UST.

Keywords: Image style transfer · Flow-based model · Adversarial robust feature

1 Introduction

Image style transfer task aims to synthesize two images, a content image and a style image, into a single one with the former's global content and the latter's artistic effects. Recent years have witnessed the substantial development of neural style transfer. Several excellent methods have been published after the initial successful attempt of Gatys [1]. Similar approaches appeared a lot that used feed-forward networks and iterative optimization [2–4]. Universal style transfer

This work was supported in part by the NSFC under Grant 62076258 and in part by the Key-Area Research and Development Program of Guangzhou under Grant 202007030004.

C. Wallraven et al. (Eds.): ACPR 2021, LNCS 13189, pp. 476–489, 2022.
https://doi.org/10.1007/978-3-031-02444-3_36

(UST) can handle the generalization ability and perform good results for arbitrary style and content. The most representative methods include AdaIN [5], WCT [6], Avatar-Net [7] and Linear Transformation [8]. These methods explore the second-order statistical transformation from style image features onto content image features via different transformations.

A recent work named ArtFlow [9] proposed an unbiased style transfer framework based on Glow [10]. With perfect mathematical support, flow-based models can generate confidential image results in many image generation tasks. ArtFlow contains a chain of revertible operators proposed by Glow, including activation normalization layers, invertible 1×1 convolutions, and affine coupling layers. A simple reverse operation can be performed to reconstruct the image since the flow-based model is reversible.

Despite DNNs superior performance, there exists tailored examples to disturb DNNs called adversarial examples [12,16,30]. These examples are inputs to machine learning models that deliberately add some subtle interference by attackers imperceptible in human vision. The discussion of adversarial examples [31] has shown us non-robust features dominates in the style transfer mission. Specifically, VGG-based networks perform poorly in adversarial training tasks yet outperform other networks like ResNet and Inception regarding style transfer.

Although there is no doubt that we can obtain beautiful transfer results with the powerful flow-based model, there is still some weakness of the framework. Researchers do not attend flow-based models for many years due to their weak feature representation ability. To improve the feature extraction ability, we start by exploring the relationship between robust features and the style transfer model based on Glow. Experiments show adversarial robust features are not only useful in iterative optimization methods but also can work well in UST. Based on the finding, we try to improve the expression of the flow-based model to make it capture more valuable features in transfer image style. We have an attempt with vision transformer first because [22] finds that ViTs has a better performance than convolution layers in the adversarial training mission. Then we further design two blocks with non-local [24] and involution [25], respectively. Both visual results and quantitative comparisons show our improved Glow can generate more excellent images. There are two main contributions of this work:

1. We confirm the effectiveness of adversarial robust features in UST by performing experiments. Robust features are helpful both as loss calculating and transfer features. The conclusion will broaden the road of future study of relative areas.
2. We improve the performance of flow-based model by replacing the current convolution layers. The original feature extraction block contains a simple stack of convolutions, which needs extra parameters and will not necessarily capture useful information. We design feature extraction blocks with self-attention, which use fewer parameters and focus on significant features. Experiments are conducted to prove our redesigned Glow is capable of capturing more valuable features for image style transfer.

2 Related Works

Image Style Transfer. Traditional methods can paint high-quality images yet may take much time, which means they have to trade-off between quality and costs. This problem has been a hindrance until Gatys [1] first introduce the neural network to extract deep features and represent image styles by Gram matrix. The iterative optimization process has a high computational cost. Numerous neural style transfer methods emerge then, which can be roughly divided into three categories. One style per model method [2,13,14] trains feed-forward neural networks to minimize the same feature reconstruction loss and style loss. Multiple-style per model methods [3,15,32] represent several styles with a single model, which can perform multiple image style transfer. Universal style transfer methods [3,5,6,8,17,18] aim to improve the generalization ability of neural style transfer by matching statistical variables like mean and variance, generating excellent results for arbitrary style and content images.

Flow-Based Model. The flow-based model was first proposed by the work of NICE [11], which extracts high dimension features with a stack of affine coupling layers. It has not been pay much attention to because of its weak feature expression capability, which is the consequence of ensuring reversibility. Subsequent work of Glow [10] improves flow with flexible reversible 1×1 convolution, increasing the performance of the flow-based model in an extensive range. Recent proposed flow-based models [19] are capable of synthesizing high-quality images and realistic speech data. ArtFlow has just been made public using the architecture of Glow, which can handle content leak problems and is capable of performing unbiased image style transfer.

Adversarial Examples. Existing models achieve good results except for a particular case which is named adversarial examples. These examples may cause the model to give an erroneous output with high confidence. Andrew et al. [20] does some experiments and proposes that adversarial examples are due to non-robust features that are highly predictive but imperceptible to humans. This conclusion arises many works in various fields. Wang et al. [21] rethink the difference of architectural between VGG and ResNet and their performance in the style transfer task, further proposing a simple solution to improve the robustness of ResNet.

3 Method

3.1 Robust Features and Style Transfer

We first state the initial conclusion about the relationship between robust features and image style transfer. The discussion [31] about robust features [20] gives us hints that VGG is more suitable in image style transfer tasks, but other networks like the most popular ResNet cannot work very well without tricks.

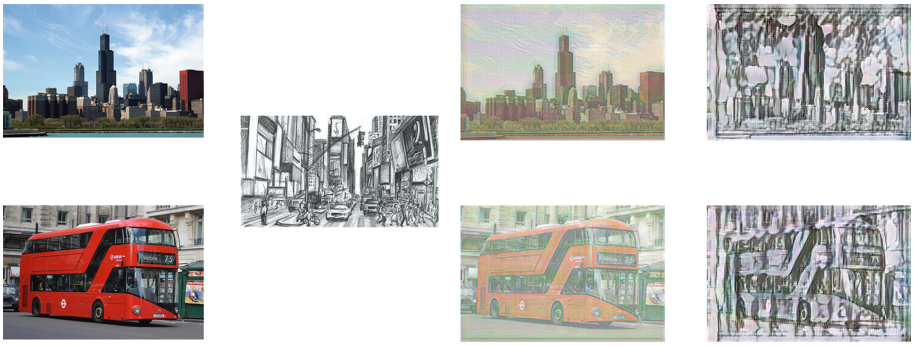

Fig. 1. Glow-based Style transfer results with non-robust and robust ResNet-50. Robust one generate more reasonable results.

VGG is far behind compared to other models like ResNet, Inception-v3 and DenseNet when it comes to the adversarial training tasks. This phenomenon can easily conclude that it is just because VGG is unable to capture non-robust features as efficient as other architectures that make it capable of performing confidential style transfer outputs. [31] does some quick experiments and shows the first four layers of VGG are almost as robust as the layers of robust ResNet.

[21] conducts more experiments and finds the residual connection is unsuitable in style transfer and adds a simple trick on loss function to create a more uniform distribution of activations, which is beneficial to produce good style transfer with ResNet. Although this is useful, we find the trick can only be used for iterative optimization methods, which can only transfer one pair of style and content once, similar to Gatys [1], and cannot work well when it comes to universal style transfer. We attempt to apply the solution to ArtFlow and only get noisy results, with the loss and gradient values being strange.

To expand the current conclusion to a universal case, we first train ArtFlow with a robust perceptual loss network [2] to verify that robust feature is still working for UST. Figure 1 shows the results of ArtFlow using standard and robust ResNet-50 as the perceptual loss network. The transferring is not really working well with standard ResNet-50. Nevertheless, the outputs become far better with robust cases, which indicates that a robust network can indeed capture features that are useful to style transfer.

We further consider that since robust features are more critical in evaluating the distance of features from loss networks, it is more reasonable to perform the transfer with robust features than non-robust ones. Table 1 shows quantitative metrics of robust and non-robust ArtFlow.

Algorithm 1. Affine coupling with reverse.

Input: input feature tensor x_{in} **Output:** output feature tensor y_{out}

$x_a, x_b = split(x_{in})$
$(logs, t) = NN(x_b)$
$s = exp(logs)$
$y_a = s \odot x_a + t$
$y_b = x_b$
$y_{out} = concat(y_b, y_a)$
return y_{out}

3.2 Glow Architecture

ArtFlow [9] introduces the flow-based model to solve the content leak problem of style transfer mission, whose overall architecture is the same with Glow [10], including a chain of three reversible transformations, i.e., affine coupling, invertible 1×1 convolution, and Actnorm [10]. Different from the widely used auto-encoder methods, the flow-based model can perform as both encoder and decoder. The following are detailed descriptions of the main reversible transformations of the network.

Actnorm. Early used batch normalization (BN) is subject to the batch size, which may add noise and cause performance to degrade. Actnorm is then proposed for activation normalization, which performs an affine transformation of the activations using a scale and bias per channel. Parameters are initialized to make the activations have zero mean and unit variance, which will output the initial minibatch of data. Actnorm performs per channel as:

$$y_{i,j} = \omega \odot x_{i,j} + b \tag{1}$$

where i, j denote the position on the feature tensor. ω and b are the scale and bias and are learnable in training, which is similar to BN.

Invertible 1×1 Convolution. Since affine coupling layers only process half of the features, it is necessary to permute the channels of the feature maps. Instead of fixed permutation in flow-based models before, Glow uses a learnable invertible 1×1 convolution. This convolution part is the main reason for the performance increase of the flow-based model. The operation can be represented by:

$$y_{i,j} = W x_{i,j} \tag{2}$$

where $W \in \mathcal{R}^{c \times c}$ is the weight matrix with c being the channel dimension of the feature tensor.

Affine Coupling Layers. The essential part of the flow-based model is the expressive reversible transformation named affine coupling proposed by Dinh

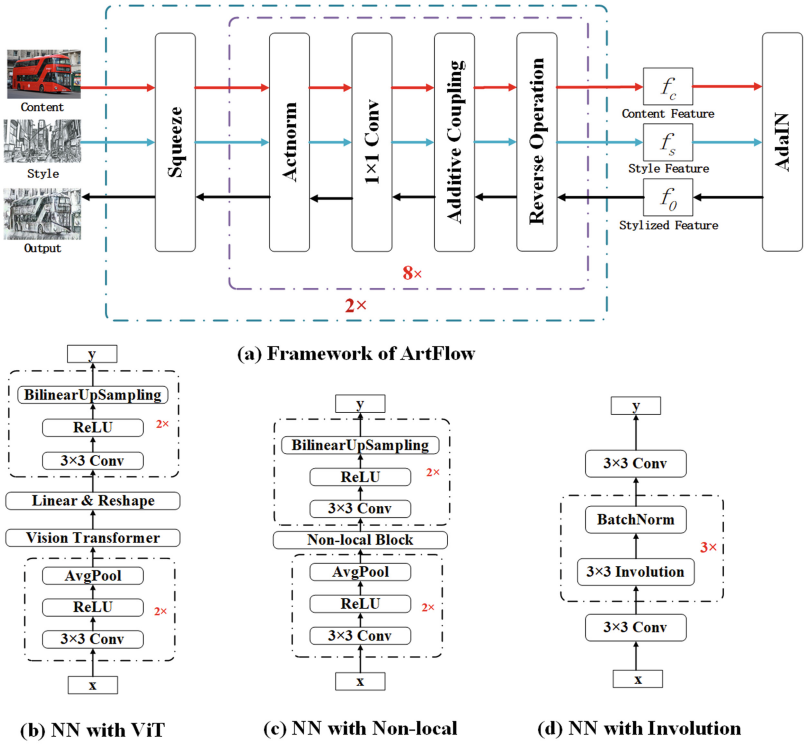

Fig. 2. (a) The overall framework of ArtFlow used to perform Style Transfer, adding the reverse operation. (b) NN with ViT uses average pooling to downsample the feature to reduce the calculation. Linear and Reshape layers transform the tokens back into features tensors. (c) NN with Non-local is similar to the previous one and doesn't need Linear since the shape will not change through the non-local block. (d) NN with Involution performs better with a convolution layer to increase the number of channels first and remains the same count of channels inside. A BatchNorm layer helps handle outliers.

et al. [11]. Roughly speaking, an affine coupling layer splits the input tensor into two parts along the channel dimension. The first part unchanged to be the first half of the output tensor, and the second part does affine transformation using the transformed result of the first part.

Reverse Operation. Inspired by the conclusion of [21] that residual connections may interfere style transfer task, we make a slight change of the affine coupling layer. Although the sophisticated 1×1 convolution is indeed able to learn an appropriate permutation of the input, it is well known that a good initialization can speed up the model convergence and get better results. To reduce the tendency of delivering the same half of the tensor multiple times directly, we add a simple reverse operation to exchange the two parts of the output of the

Algorithm 2. Pseudo PyTorch code of involution.

Input: $x \in \mathcal{R}^{B \times C \times H \times W}$ **Output:** $y \in \mathcal{R}^{B \times C \times H \times W}$

```
# K: kernal size, s: stride, r: reduction ratio
# initialization
o = AvgPool2d(s, s) if s > 1 else Identity()
reduce = Conv2d(C, C//r, 1)
span = Conv2d(C//r, K*K, 1)
unfold = Unfold(K, padding=K//2, s)
weight1 = Parameter((1,1,H,W))
weight2 = Parameter((1,1,H,W))
# forward
weighted_sum = ReLU(BN(mul(weight1, x))).sum(dim=(2,3)) # B, C
weighted_sum = weighted_sum.unsqueeze().unsqueeze()# B, C, 1, 1
weighted_sum = mul(weighted_sum, weight2) # B, C, H, W
x_unfolded = unfold(x).view(B, C, K*K, H, W)
kernel = reduce(o(x+weighted_sum)) # B, C//r, H, W
kernel = span(ReLU(BN(kernel))).view(B, 1, K*K, H, W)
y = mul(kernel, x_unfolded).sum(dim=2) # B, C, 1, H, W
return y.squeeze()
```

affine coupling layer. The affine coupling with reverse is summarized in Algorithm 1. An additive coupling layer is a simplified case with $s = 1$, which is the one exactly used in ArtFlow.

3.3 Improve Feature Extraction

There is no doubt that robust features are beneficial to performing more wonderful image style transfer results, but adversarial training is very time-consuming. To increase the performance while holding the efficiency, it is a better idea to improve the architecture of the network. As we can see from the modules, affine coupling layer consists of the only feature extraction Neural Network (NN) in Glow since the 1×1 convolution is for feature shuffling. Aiming to capture robust features, we need to use a more suitable structure. Shao et al. [22] has recently published a work about the adversarial robustness of ViTs [23]. It can be inferred from their experiments that ViTs possess better adversarial robustness compared with convolutional neural networks, which raises an assumption that self-attention is playing an essential role in this question.

As shown in Fig. 2, we design three different neural network blocks for the affine coupling block to increase the feature extraction ability, using vision transformer, non-local [24] and involution [25], respectively. To be clear, non-local is a widely used attention mechanism in the computer vision area, which is a lighter weight module than ViT. Involution is a neural network operator whose kernel parameters are shared along the channel dimension, which is different from convolution, whose kernel remains the same along pixels. The kernels are transform results of the vectors along the channel dimension with a kernel generation

function. The involution operator can be a general form of self-attention by replacing the generation function. To let each channel receive the global information, which is important in style expression, we add a global weighted sum along the channel dimension to the channels. Furthermore, we use one more weight matrix to learn the importance of the global information to the current channel. The global information we add only need $2 \times H \times W$ more parameters and can obtain much promotion. We accept the group number, reduction ratio and dilation to be all 1. Algorithm 2 is the pseudo-PyTorch code of involution we apply. Experiments show that involution is actually capable of capturing helpful features.

3.4 Loss Function

Gatys [1] propose the Gram matrix to represent the style of an image and soon becomes the general criterion of style transfer. The perceptual loss [2] further extends the usage with a loss network, which brings up the development of Universal Style Transfer. Loss networks, usually VGG-19, maps an image into a set of feature maps $\{F^l(x_0)\}_{l=1}^{L}$ where F^l is the mapping from the image to the activations of the l^{th} layer. Suppose the activation to be $\mathcal{R}^{C_l \times W_l \times H_l}$ and can also be reshaped into a matrix $F^l(x_0) \in \mathcal{R}^{C_l \times M_l}$, where $M_l = W_l \times H_l$. The Gram matrix $G^l \in \mathcal{R}^{C_l \times C_l}$ is computed by the inner product between the feature maps in layer l:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} \tag{3}$$

then with x_s representing the style image and x the output image, the style loss can be measured by $\mathcal{L}_{style}$, as:

$$\mathcal{L}_{style}(x_s, x) = \sum_{l=1}^{L} \frac{\omega_l}{4C_l^2 M_l^2} ||G^l(F^l(x)) - G^l(F^l(x_s))||_2^2 \tag{4}$$

where $\omega_l \in \{0, 1\}$ are factors using to choose which layers will contribute to the style loss. Content loss $\mathcal{L}_content$ is a simple mean square error as:

$$\mathcal{L}_{content}(x_c, x) = \frac{1}{2} ||F^l(x) - F^l(x_c)||_2^2 \tag{5}$$

where x_c is the content image and x the output. l here represents the feature used to measure the content distance, usually the deepest layer. The total loss function $\mathcal{L}_{total}$ is a weighted sum of style loss and content loss as:

$$\mathcal{L}_{total}(x, x_c, x_s) = \lambda_{content} \mathcal{L}_{content}(x, x_c) + \lambda_{style} \mathcal{L}_{style}(x, x_s) \tag{6}$$

It is necessary to clarify that VGG-19 is used as the perceptual loss network. There is no fixed statement about which layers to use. According to experiments of [8], we adopt the combination of four outputs of the first ReLU layer of the first four VGG blocks as *relu1_1*, *relu2_1*, *relu3_1*, *relu4_1*, respectively. As for

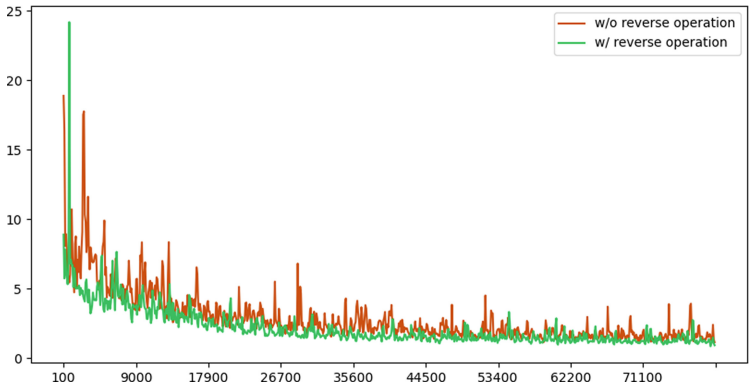

Fig. 3. Training Loss of Glow with and without the reverse operation. Applying reverse can accelerate the training procedure and makes it more stable.

ResNet used in the comparison experiment of Sect. 4.2, we refer to the approach of [31] and choose layers of *relu2_3*, *relu3_4*, *relu4_6*, *relu5_3* considering the fair comparison with VGG.

4 Experiments

In this section, we explain the experiment details of three main terms. We first prove the adversarial training has a positive influence on universal style transfer. Then we conduct a fast experiment of the effect of the Reverse Operation. Moreover, extensive experiments are performed to show the enhancement of the improved Glow.

4.1 Experiment Settings

Datasets. Following the existing image style transfer methods, we use the MS-COCO dataset [26] as our content images and the WikiArt dataset [27] as style images. The input images are resized to 512×512 and then randomly cropped to 256×256. In the experiment of adversarial training, we follow the current study to train the Glow on cifar-10 [28], then use the pretrained model to transfer the style of our test set.

Network Structure. We adopt the structure of ArtFlow [9] using two Glow blocks, with each block containing eight combinations of the three reversible transformations. The author has discussed that additive coupling is sufficient for style transfer and is more stable, which is the same with our attempts, no matter which NN we use.

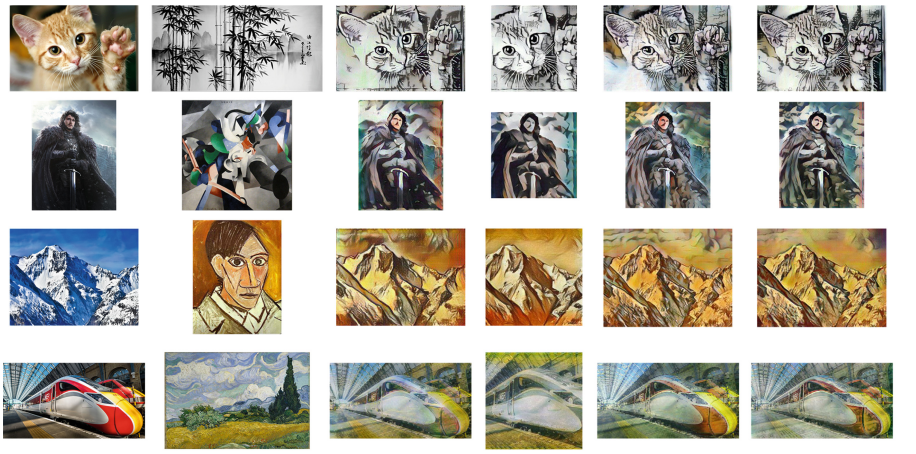

Fig. 4. Visual results compared to original ArtFlow. Our ViT block requires the content feature to be in certain sizes, so we randomly crop the input content image. Our improved ArtFLow generates more beautiful and stable images, especially the details and textures. Please zoom in to confirm.

Training. We implement all experiments on the PyTorch framework. Standard training on cifar-10 takes about 15 h for 250 epochs on an RTX 2080Ti GPU. Adversarial training needs 4 h for one epoch, and the loss is usually becoming stable after 40 epochs. We adopt the widely used TRADES [29] to perform adversarial training with step size, epsilon, number of perturbation iterations to be 0.003, 0.031, 7, respectively. For the training of Glow, the loss weights are set to 0.1 for $\lambda_{content}$ and 1 for λ_{style} based on previous work experience. Adain is used as the style transfer module because of its simplicity and effectiveness. We perform 100000 iterations using Adam with the initial learning rate of 1e−4 and decay of 5e−5. The original architecture takes about 22 h with a batch size of 4 on an RTX 2080Ti GPU or 21 h with a batch size of 2 on a GTX 1080Ti GPU.

Metrics. Visual results are first compared to show the superiority of our methods. We choose different kinds of style and content images as the test set. A good style transfer result should remain more overall content and generate vivid hues and detailed textures. In addition, we also make quantitative comparisons. The perceptual loss value of the test set is a common metric among image synthesis tasks. We use the content loss to measure the content preservation and the Gram matrix loss to measure the style transfer ability. The efficiency is important as well, so we compare the transfer speed and the model size.

Table 1. Quantitative evaluation comparisons. The first two rows are the result of Sect. 4.2 training on cifar-10. The rest are the results of Sect. 4.3. Transfer time is evaluated on 256×256 images using an TITAN RTX GPU.

Models	Style loss	Content loss	NLL loss	Time (s)	Model size (MB)
Standard Glow	11.1	1.924	3.39	0.144	74.38
Robust Glow	8.5	1.997	3.49	0.144	74.38
ArtFlow [9]	3.905	3.199	3.39	**0.144**	74.38
Glow+ViT (ours)	4.631	**2.902**	3.42	0.221	239.36
Glow+nonlocal (ours)	3.55	3.003	3.44	0.185	51.62
Glow+Involution (ours)	**3.110**	2.939	3.52	0.157	**34.46**

4.2 Comparing Adversarial and Standard Networks

We first show that robust features are still working when it comes to universal style transfer. Since adversarial training is very time-consuming, we directly use the pretrained robust ResNet. The Glow used in this part remains the same with ArtFlow. As shown in Fig. 1, the first row are the results of standard ResNet-50, and the second row is from robust ResNet-50. Obviously, standard ResNet-50 is not really performing style transfer, yet the robust one makes better performance.

Then we compare the robust Glow and the standard Glow. With the training setting stated before, we use the pretrained Glow models to transfer the style of images. The pretrained model is not able to perform reasonable image results. However, from the loss comparison, we can see the robust model extracts more useful features for style transfer, leading to a lower Gram matrix loss.

4.3 Improved Glow

Ablation Experiment. Firstly, we prove that training will be accelerated with the simple reverse operation. We use the original Glow to perform this part of experiments. Figure 3 shows training procedures of the two cases, one of which uses the reverse operation. It is clear that with the reverse operation, training loss descends faster, which confirms our assumption that the parameters of the 1×1 convolution in the network are trained to have similar behaviour.

Secondly, we demonstrate that with a more suitable design for NN, Glow can obtain more excellent style transfer results. Visual comparisons are shown in Fig. 4. The attention mechanism can enrich details of the image results, and the textures are described better. The designed block with involution achieves relatively better performance than others, with generated images being rich in details and seems stable in the meantime. Quantitative comparisons are made with the testing loss aforementioned. We also compare the negative-log-likelihood loss when training Glow models with cifar-10. NLL is the most common loss function to train flow-based models and can show their classification ability. As we can see in Table 1, our blocks make the classification a little bit worth but

(a) Content (b) ArtFlow[9] (c) 128-Involution (d) 64-Involution (e) 64-Involution*

Fig. 5. Detail comparisons. (c) Our involution block with 128 dimensions of hidden layers and is the one used in the previous section. (d) The same structure of (c) with 64-dimension hidden layers. (e) Pure involution without the convolutions in Fig. 2. Model sizes (MB) are 34.46, 10.86, 4.20, respectively.

facilitate style transfer, which also confirms our point of view that style-transfer-useful features may have some degrees of difference with those in recognition tasks. ViT gets a similar score with original convolutions. Our blocks with non-local and involution obtain lower style and content loss, indicating the model transfers more artistic effects while preserving the global content. The model size is smaller since our blocks can capture more valuable features. Using the involution block reduces the scale more than twice. The time cost rises a little, owing to the time-consuming calculations of self-attention.

Detail Comparison. We compare the details of our involution block of different sizes. As shown in Fig. 5, our involution block can generate more textures of the corresponding artistic effect, which benefits both content and style loss. After further comparisons, we can find that as the parameters decrease (from c to e), the performance of colour begins to degrade first, and then the textures (notice the purple part of the ear). This phenomenon indicates that our involution block has a more powerful ability to capture the global stroke of the style image, which is more complex than capturing colours. The promotion is due to the self-attention mechanism and the weighted sum we add, both of which are able to increase the overall awareness.

5 Conclusions

In this paper, we first explore the relationship between adversarial robust features and universal image style transfer. Although standard ResNet-50 is not suitable to be the perceptual loss network in UST, using an adversarial robust ResNet-50 makes things different and generates confidential results. Experiments prove robust features are helpful not only during loss calculating but also in the transfer procedure. Based on the conclusion, we improve the existing Glow model by enhancing the original feature extraction block with self-attention mechanism, making it perform more pleasing and more stable style transfer results. Three different blocks are used with ViT, non-local and involution, respectively. Our block with involution gets the best results while significantly reducing the model size.

References

1. 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, pp. 2414–2423 (2016)
2. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
3. Zhang, H., Dana, K.: Multi-style generative network for real-time transfer. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11132, pp. 349–365. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11018-5_32
4. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: StyleBank: an explicit representation for neural image style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1897–1906 (2017)
5. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1501–1510 (2017)
6. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. arXiv preprint arXiv:1705.08086 (2017)
7. Sheng, L., Lin, Z., Shao, J., Wang, X.: Avatar-Net: multi-scale zero-shot style transfer by feature decoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8242–8250 (2018)
8. Li, X., Liu, S., Kautz, J., Yang, M.H.: Learning linear transformations for fast image and video style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3809–3817 (2019)
9. An, J., Huang, S., Song, Y., Dou, D., Liu, W., Luo, J.: ArtFlow: unbiased image style transfer via reversible neural flows. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 862–871 (2021)
10. Kingma, D.P., Dhariwal, P.: Glow: generative flow with invertible 1×1 convolutions. arXiv preprint arXiv:1807.03039 (2018)
11. Dinh, L., Krueger, D., Bengio, Y.: NICE: non-linear independent components estimation. arXiv preprint arXiv:1410.8516 (2014)
12. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: DeepFool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582 (2016)
13. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.S.: Texture networks: feed-forward synthesis of textures and stylized images. In: ICML, June 2016, vol. 1, no. 2, p. 4 (2016)
14. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Improved texture networks: maximizing quality and diversity in feed-forward stylization and texture synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6924–6932 (2017)
15. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. arXiv preprint arXiv:1610.07629 (2016)
16. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
17. Gu, S., Chen, C., Liao, J., Yuan, L.: Arbitrary style transfer with deep feature reshuffle. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8222–8231 (2018)

18. Shen, F., Yan, S., Zeng, G.: Neural style transfer via meta networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8061–8069 ((2018))
19. Ho, J., Chen, X., Srinivas, A., Duan, Y., Abbeel, P.: Flow++: improving flow-based generative models with variational dequantization and architecture design. In: International Conference on Machine Learning, pp. 2722–2730. PMLR (May 2019)
20. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. arXiv preprint arXiv:1905.02175 (2019)
21. Wang, P., Li, Y., Vasconcelos, N.: Rethinking and improving the robustness of image style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 124–133 (2021)
22. Shao, R., Shi, Z., Yi, J., Chen, P.Y., Hsieh, C.J.: On the adversarial robustness of visual transformers. arXiv preprint arXiv:2103.15670 (2021)
23. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
24. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803 (2018)
25. Li, D., et al.: Involution: inverting the inherence of convolution for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12321–12330 (2021)
26. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
27. Nichol, K.: Painter by numbers, wikiart (2016). https://www.kaggle.com/c/painter-by-numbers
28. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
29. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning, May 2019, pp. 7472–7482. PMLR (2019)
30. Dong, J., Xie, X.: Visually maintained image disturbance against deepfake face swapping. In 2021 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (July 2021)
31. Nakano: A discussion of 'Adversarial Examples Are Not Bugs, They Are Features': adversarially robust neural style transfer. Distill (2019). https://distill.pub/2019/advex-bugs-discussion/
32. Ma, J., Yu, W., Liang, P., Li, C., Jiang, J.: FusionGAN: a generative adversarial network for infrared and visible image fusion. Inf. Fus. **48**, 11–26 (2019)