UNPAIRED MOTION STYLE TRANSFER WITH MOTION-ORIENTED PROJECTION FLOW NETWORK

Yue Huang, Haoran Mo, Xiao Liang, Chengying Gao[*]

Sun Yat-sen University
{huangy346,mohaor,liangx66}@mail2.sysu.edu.cn, mcsgcy@mail.sysu.edu.cn

ABSTRACT

Existing motion style transfer methods trained with unpaired samples tend to generate motions with inconsistent content or inconsistent number of frames when compared with the source motion. Moreover, due to the limited training samples, these methods perform worse in unseen style. In this paper, we propose a novel unpaired motion style transfer framework that generates complete stylized motions with consistent content. We introduce a motion-oriented projection flow network (M-PFN) designed for temporal motion data, which encodes the content and style motions into latent codes and decodes the stylized features produced by adaptive instance normalization (AdaIN) into stylized motions. The M-PFN contains dedicated operations and modules, e.g., Transformer, to process the temporal information of motions, which help to improve the continuity of the generated motions. Comparisons with the state-of-the-art methods show that our method effectively transfers the style of the motions while retaining the complete content and has stronger generalization ability in unseen style features.

Index Terms— motion generation, style transfer, flow network, AdaIN

1. INTRODUCTION

Generating various stylized motions can greatly help produce realistic and expressive character animation which plays a fundamental role in diverse applications such as human animation, games, robotics, etc. While using motion capture (MoCap) technologies to obtain different styles of motions is time-consuming and requires expensive equipment, style transfer based on existing motions is a more economical and feasible direction.

Motion style is an abstract concept that is difficult to be described by precise mathematical definitions. Recently, data-driven methods (e.g., deep learning) have shown strong ability in representing style with latent codes [1–5], but most of them rely on paired samples. The acquisition of such paired samples is tedious and requires actors to perform several motions with almost the same steps and different styles.

Training motion style transfer models with unpaired samples alleviates the need for paired data. The model proposed by Aberman et al. [6] transfers the style from videos to motions, but tends to generate motions with *inconsistent content*. Wen et al. [7] aim at synthesizing high-quality stylized motions, but suffer from issues of producing *incomplete motion content* and requiring *more style motion frames than the content ones* due to its heavy dependence on explicit context information. To address these issues, we propose an unpaired motion style transfer framework that is able to transfer styles between motion clips of arbitrary length and generate complete stylized motions with consistent content.

In the framework, we introduce a motion-oriented projection flow network (M-PFN) which is modified from the projection flow network (PFN) [8] to adapt to the temporal motion sequences. As shown in Fig. 1, the M-PFN first encodes the input content and style motion clips through its forward propagation procedure, and extracts their latent features in a lossless manner. The two latent codes are fused in an adaptive instance normalization (AdaIN) layer [9] to be a stylized feature vector, which is subsequently passed to the reverse propagation of M-PFN to reconstruct the stylized motion. With the reversible transformation of the M-PFN, the framework is able to ensure the consistency of the content of the generated motions.

The PFN [8] utilizes convolutions to perform expressive transformation for image data, however, they do not work well with the temporal information of motion data and may cause discontinuous results when transferring style to the entire sequence directly. To overcome this issue, our motion-oriented PFN (M-PFN) replaces the convolution modules with Transformer [10], which has been proven to have superior ability in processing sequential data [11] and is able to pay more attention to long-term and global information. Furthermore, the self-attention mechanism implicitly models the relationship between each frame pair, which offers sufficient context information and guarantees the completeness of the generated frames. We also replace the squeeze operations in PFN with interpolation ones to maintain the temporal dimen-

[*] The corresponding author is Chengying Gao. This work was supported by the Natural Science Foundation of Guangdong Province, China (Grant No. 2022A1515011425, 2021A1515012242) and the Natural Science Foundation of China (Grant No. 61972433).

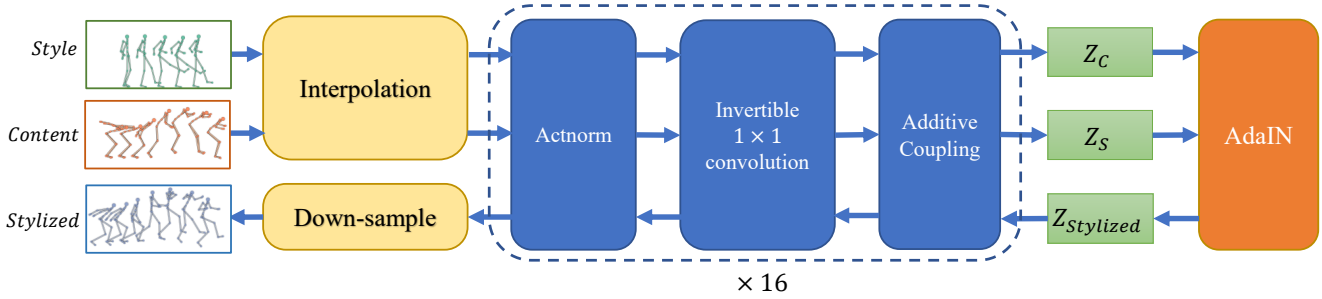


Fig. 1. Network architecture of our proposed motion style transfer framework. It includes a motion-oriented projection flow network (M-PFN) for extracting latent features of motions, and an adaptive instance normalization (AdaIN) layer for style transfer of latent features. In addition, there is an interpolation operation for pre-process and post-process of motions.

sion for the motions.

We evaluate our approach through comprehensive ablation studies and comparisons with state-of-the-art methods. The experiments show that our method is able to generate visually pleasing stylized motions while maintaining the completeness and consistency of the content motions. In addition, while the training samples are limited, the results demonstrate that our approach generalizes better to unseen motion styles compared to the existing methods.

In summary, we make the following major contributions:

- A novel unpaired motion style transfer framework that transfers styles between motion sequences of arbitrary length and generates complete stylized motions with consistent content.
- A motion-oriented projection flow network (M-PFN) designed for temporal motion data, which employs Transformer to model the long-term information and an interpolation operation to maintain the temporal dimension.
- In depth comparisons with existing approaches, which demonstrate the effectiveness and the generalization ability in unseen styles of our approach.

2. RELATED WORK

2.1. Motion Style Transfer

Data-driven motion style transfer approaches can be divided into two streams: supervised and unsupervised approaches. Supervised algorithms [1–5] rely on paired samples during training, but the collection of paired samples needs a huge amount of user labour and expensive equipment.

Recently unsupervised approaches have been introduced and make it possible to train motion style transfer models with unpaired samples. Aberman et al. [6] propose to transfer style from videos to motions based on adaptive instance normalization (AdaIN) [9]. The convolutional encoder which is not reversible tends to lose necessary information and generates motions with inconsistent content. Wen et al. [7] introduce a motion style transfer method based on generative flow network [12] for synthesizing high-quality stylized motions, but requires more style motion frames than the content ones to provide sufficient context information. Moreover, it has an inclination to produce incomplete motion content due to the lack of context information in the first few frames of the input content motions.

Compared with the two methods above, our approach is able to produce stylized motions with consistent content and complete output frames. Furthermore, there is no constraint on the length of the input motion sequences.

2.2. Image Style Transfer

Most motion style transfer algorithms [6, 7] adopt ideas from the field of image style transfer because of the intrinsic correlations between them. Image style transfer has been broadly studied these years [8, 9, 13, 14]. Gatys et al. [13] show that the style and content of images can be represented by statistics of deep features extracted from a pretrained classification network. Ulyanov et al. [14] propose an instance normalization (IN) layer and show that the style can be manipulated by modifying the second order statistics (mean and variance) of channels of intermediate layers. Inspired by this idea, Huang et al. [9] propose an adaptive instance normalization (AdaIN) layer to modify the statistics of deep features of the input content with the encoded style code. We integrate the AdaIN layer into our framework that is required to generate a stylized feature vector based on the encoded content and style features.

In order to overcome the issue of content leakage of images and improve the consistency of content, An et al. [8] propose an ArtFlow framework containing a projection flow network (PFN), which includes squeeze operations to reduce the resolution of features while keeping the feature information, and 2D convolution-based modules to process the spatial features. We exploit the idea of the PFN, but replace the squeeze and 2D convolution operations designed for 2D im-

ages with ones for temporal motion data, *i.e.,* sequence interpolation and Transformer networks [10].

3. METHOD

3.1. Overview

Given a content motion clip m^s with content m and style s, and a style motion clip n^t with content n and style t, our motion style transfer framework aims at generating a stylized motion sequence m^t with content m and style t. The architecture of the proposed motion style transfer framework is illustrated in the Fig. 1. It consists of a motion-oriented projection flow network (M-PFN) and an unbiased style transfer module AdaIN. Given a content input m^s and a style input n^t, we first perform an interpolate operation on the motion frames to increase the time dimension. Then the interpolated sequences from style and content domains are fed into the M-PFN for lossless feature extraction, and the M-PFN produces latent codes z_c and z_s for content and style, respectively. Next, with the two extracted latent codes, the AdaIN layer performs unbiased style transfer and generates the latent code $z_{stylized}$ for the stylized motion. Finally, the $z_{stylized}$ is reconstructed to a stylized motion m^t via the reverse propagation procedure of the M-PFN.

3.2. Motion-oriented Projection Flow Network (M-PFN)

3.2.1. Original Projection Flow Network (PFN)

The original Artflow [8] is designed for image data, and use squeeze operations in the proposed PFN to reduce the spatial size of 2D feature maps and expand their channel dimension. Apart from the squeeze operations, PFN contains a chain of three transformations, i.e., Actnorm, invertible 1×1 convolution and additive coupling. Each module is reversible to ensure that the information is lossless in the forward and reverse propagation process. The Actnorm layer is designed to assign zero mean and unit variance to each channel of features to facilitate subsequent calculations. The invertible 1×1 convolution layer is used to permute the channel dimensions of feature maps, so that each dimension can affect all the other dimensions. The additive coupling layer is a special case of affine coupling proposed by Dinh et al [15]. It performs nonlinear transformations to one part of the input, and the remaining part is kept unchanged. Thus, the reversible transformation of the flow is computationally efficient.

3.2.2. Motion-oriented Modifications

When the squeeze operations which reduce the spatial size of image data are applied to temporal type motion data, the time dimension is compressed, which leads to shaking in the generated motions. Therefore, we replace the squeeze operations with interpolation ones to increase the time dimension of motion features, which benefits the training of style features. Specifically, we use the nearest neighbor interpolation to double the frames before inputting the motion sequences to the M-PFN.

The additive coupling layer in the original PFN uses several convolutional modules for processing images. Given that convolutions tend to fail at modeling the temporal information, we replace them with Transformer encoding blocks [10] to capture the global and long-term sequence information of motions. The computation of the additive coupling layer is then formulated as follow:

$$
\begin{aligned}
z_1, z_2 &= \mathrm{split}(x), && (1)\\
z_2' &= \mathrm{Transformer}\left(\mathrm{concat}\left(z_1, x_{rot}\right)\right) + z_2, && (2)\\
y &= \mathrm{concat}\left(z_1, z_2'\right), && (3)
\end{aligned}
$$

where x is the input features and y output ones. The $\mathrm{split}(\cdot)$ and $\mathrm{concat}(\cdot)$ are two functions for splitting and concatenation. The x_{rot} denotes the complete motion sequence including the root information as the control signal. We concatenate it with half of the input features z_1 in order to help extract deeper style features in the Transformer. We employ the architecture of the original Transformer encoder [10], which consists of multi-head self-attention, layer normalization and feed forward networks.

3.3. Loss Function

Given the motion-oriented projection flow network G, the content input m^s and the style input n^t, we use the following loss functions to train the network:

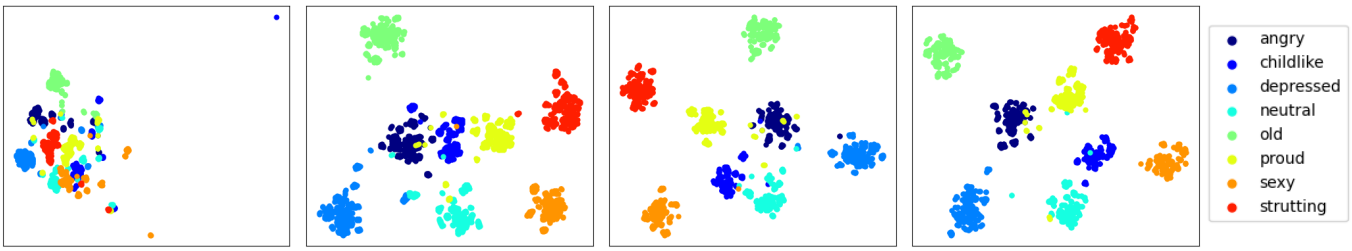
Content Loss and Style Loss. We followed the content loss L_c and style loss L_s used in the AdaIN [9]:

$$
\begin{aligned}
L_c &= \left\| G\left(G^{-1}(p)\right) - p \right\|_2, && (4)\\
L_s &= \left\| \mu\left(G\left(G^{-1}(p)\right)\right) - \mu\left(G\left(n^t\right)\right) \right\|_2 + && (5)\\
&\quad \left\| \sigma\left(G\left(G^{-1}(p)\right)\right) - \sigma\left(G\left(n^t\right)\right) \right\|_2,
\end{aligned}
$$

where $G(\cdot)$ and $G^{-1}(\cdot)$ represent the forward and backward propagation of M-PFN, respectively. p is the output of AdaIN. μ, σ represent the mean and standard deviation of the feature. The M-PFN is trained to balance between L_c and L_s, and thus it can avoid reconstructing the original motions.

Style Triplet Loss. In order to further promote the clustering of different style features in the latent space, we adopt the style triplet loss proposed by Aberman et al. [6] which exploits the style labels. At each iteration, the loss is calculated by three motions m^s, w^s and x^t of styles s and t. The calculation is as follow:

$$
\begin{aligned}
L_{tri} &= [\| f\left(G\left(m^s\right)\right) - f\left(G\left(w^s\right)\right) \| - && (6)\\
&\quad \left\| f\left(G\left(m^s\right)\right) - f\left(G\left(x^t\right)\right) \right\| + \delta]_+,
\end{aligned}
$$



(a) Squeeze (b) w/o squeeze (c) Interpolation (d) Interpolation+Transformer (Ours)

Fig. 2. t-SNE visualization of the encoded style information from motions with different styles. We evaluate the performance of four methods, namely, with squeeze operation (a), without squeeze operation (b), with interpolation (c) and using Transformer after interpolation (d).

where $f(\cdot)$ represents the style features which concatenates the mean and standard deviation of the output of M-PFN forward propagation. δ is a constant margin.

Motion Reconstruction Loss. When the content and style input are the same motion, we calculate the L1 norm between the output and input as the motion reconstruction loss:

$$L_r = \left\| G^{-1}\left(G\left(m^s \mid m^s\right)\right) - m^s \right\|_1 \tag{7}$$

Unit Quaternion Loss. This loss encourages the quaternions of the generated results to be a unit quaternion which promotes the authenticity of the generated motions.

$$L_{qt} = \frac{1}{T}\left\| m^s - 1 \right\|_2 \tag{8}$$

The final loss function is given by a combination of the aforementioned loss terms:

$$L = \omega_c L_c + \omega_s L_s + \omega_{tri} L_{tri} + \omega_r L_r + \omega_{qt} L_{qt}, \tag{9}$$

where $\omega_c, \omega_s, \omega_{tri}, \omega_r$ and ω_{qt} are scalars, representing the weights of different loss functions.

4. EXPERIMENTS

4.1. Dataset and Implementation Details

We use the dataset captured by Xia et al. [2] which contains motion sequences labeled with 8 styles. We take 10% of the dataset as the test set, and the remaining as the training set. All the motion sequences in the training set are downsampled to 30 frames per-second, and trimmed into short overlapping clips of 32 frames with an overlap of 8. The pre-processed data contains about 1500 motion clips.

16 flows are stacked in the M-PFN. We use 2 heads for the multi-head self-attention of the Transformer in the additive coupling layer. δ in Eq.(6) is set to 5. We set $\omega_c = 3, \omega_s = 3, \omega_{tri} = 0.1, \omega_r = 4, \omega_{qt} = 1$ in Eq.(9). The Adam optimizer is adopted during training. We train a total of 300k epochs with a batch size of 32.

4.2. Ablation Experiments

Interpolation vs. Squeeze. We compare the performance between the squeeze operations in the original PFN [8] and our proposed interpolation method by visualizing the encoded style information of the motions. We project the statistics (mean and variance) of the latent code output from the forward propagation of M-PFN into a 2D space by using t-distributed stochastic neighbor embedding (t-SNE), and colorize them according to their style labels. Figure 2 (a)-(c) show the comparison results of replacing the squeeze operation with an interpolation one. It can be seen that when applying squeeze operation, the latent codes of different styles are indistinguishable, which might lead to a worse decoupling of content and style during motion style transfer. After removing the squeeze operation, the latent code information of different styles start to be separated although there is still overlap. After employing our proposed interpolation operation, the clusters have an even better separation, which demonstrates the efficacy of the interpolation method.

Table 1. Content consistency before and after using Transformer. A lower value is better.

	Content consistency (↓)
Without Transformer	46.14
With Transformer	**31.96**

Transformer vs. Convolution. We replace the convolution modules in the original PFN [8] with Transformer [10] to help better model the temporal motion data. Figure 2-(d) shows the incorporated Transformer further improves the clusters when compared with Figure 2-(c). We also conduct quantitative evaluation on the content consistency with the metrics proposed by Wen et al. [7], which is able to measure the similarity of contents from the input content motion and the output motion regardless of their styles (we recommend the audiences to refer to their paper for more details). The quantitative results in Table 1 show that the content consistency improves by a large margin (about 30%) after incorporating Transformer. Please refer to the supplemental video for the visual results of the motion style transfer.

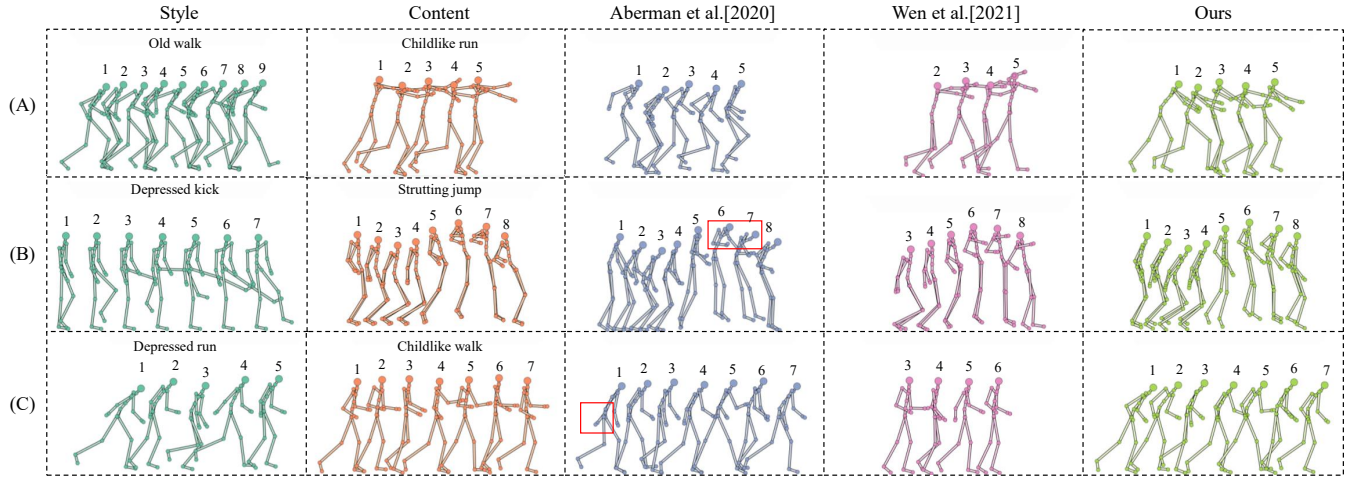


Fig. 3. Qualitative comparisons with existing approaches. We visualize a sub-sequence from the entire motion sequence for brevity by choosing one frame every K frames. For walking samples we use $K = 10$, and $K = 4$ or 6 for the rest. The frame numbers of the output motions should be aligned with the input content motions. The phrases are from the original dataset.

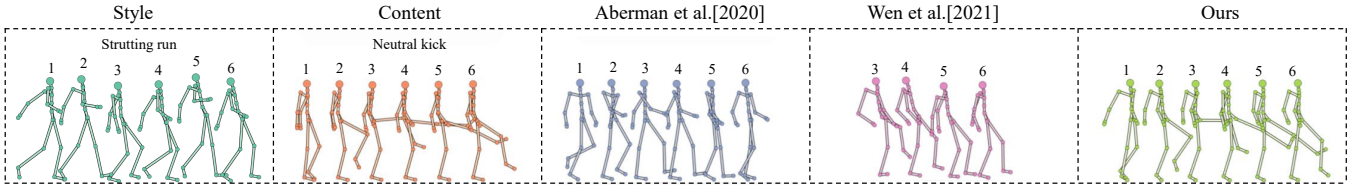


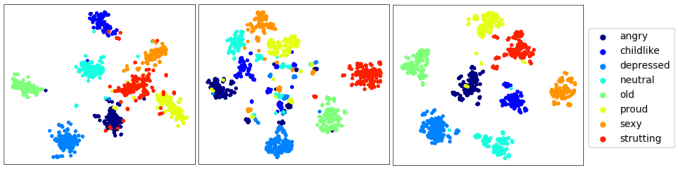
Fig. 4. Motion style transfer of unseen style. The "strutting" style is unseen during training.

4.3. Comparison with Existing Methods

Motion style transfer. We compare with two representative methods on unpaired motion style transfer proposed by Aberman et al. [6] and Wen et al. [7]. The comparison results are shown in Fig. 3. Model from Aberman et al. [6] is able to produce complete motions as the input content motions, but suffers from inconsistent content. For example, in (A), the "run" type motion tends to lift arms, but results from Aberman et al. follow the pose of the style input and put down the arms. In contrast, our generated motions still lift arms while seeming "older" with a bent spine.

Wen et al. [7] require more style frames than the content ones, so we truncate the content input for their model when the input motions do not meet this requirement such as (B) and (C). Although their model generates consistent content (e.g., the lifting arms in (A)), the performance of style transfer is worse, which can seen from (A) where the generated motions are not "old" enough and (C) where the generated motions are not "depressed". Moreover, it fails to produce the same number of frames as the content motions. Our approach works with all these cases with arbitrary length of input style motions and content motions, and is able to generate complete motion frames. Furthermore, our framework produces well-stylized motions (e.g., the "depressed" appearance in (C)) while maintaining consistent content. We show the dynamic results in the supplemental video.

Unseen Styles. We additionally evaluate the performance on unseen styles to measure the generalization ability. We remove all the motions labeled by "strutting" during training and test with motions of this label. Figure 4 shows Aberman et al. [6] transfer the style to some extent, albeit with the same issue of inconsistent content of "kick". The unreasonable arm poses from the results of Wen et al. [7] indicate their poor ability to transfer unseen styles. In contrast, our model performs well and generates "strutting" kicking.



(a) Aberman et al. (b) Wen et. al. (c) Ours

Fig. 5. t-SNE visualization of the encoded style information for unseen styles. "strutting" serves as the unseen styles during training.

We also evaluate the performance on unseen styles by visualizing the encoded style information. As can be seen in the

Fig. 5, our model successfully clusters the samples with the new style. The model of Aberman et al. [6] classifies part of the "strutting" motions into styles "proud" or "angry". Wen et al. [7] separate the unseen label "strutting" well, but suffer from a worse clustering for all types of motions because they do not utilize style labels during training. To quantitatively evaluate the performance on unseen styles, we follow Wen et al. and calculate the Silhouette Coefficient (SCoeff) and Calinski-Harabaz Index (CHI) of the encoded style information. The results in Table 2 are in line with the visual results above, showing that our approach generalizes better on unseen styles even when trained on limited samples. We show more results in the supplemental video.

Table 2. Quantitative evaluation about unseen style. The evaluation is performed on all the styles. For both metrics, a higher value is better.

Method	SCoeff (↑)	CHI (↑)
Aberman et al.	0.649	3499.01
Wen et. al	0.306	653.02
Ours	**0.659**	**4358.83**

5. CONCLUSION

In this paper, we propose an unpaired motion style transfer framework that is able to transfer style from one motion to another while maintaining the content consistency. We introduce a motion-oriented projection flow network tailored for motion sequences, in which Transformer is employed and an interpolation operation is exploited to improve the modeling ability of temporal data. Comprehensive experiments corroborate the efficiency of our approach as well as the generalization ability in unseen styles. While our approach mainly focuses on motion style transfer, we did not evaluate our framework on motions from different skeletons. This belongs to another direction in the field of motion generation, i.e., motion retargeting, and we envision a combination of this direction with our approach for motion style transfer in the future.

6. REFERENCES

[1] Graham W Taylor and Geoffrey E Hinton, "Factored conditional restricted boltzmann machines for modeling motion style," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 1025–1032.

[2] Shihong Xia, Congyi Wang, Jinxiang Chai, and Jessica Hodgins, "Realtime style transfer for unlabeled heterogeneous human motion," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–10, 2015.

[3] Harrison Jesse Smith, Chen Cao, Michael Neff, and Yingying Wang, "Efficient neural networks for real-time motion style transfer," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 2, no. 2, pp. 1–17, 2019.

[4] Daniel Holden, Jun Saito, and Taku Komura, "A deep learning framework for character motion synthesis and editing," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–11, 2016.

[5] Yuzhu Dong, Andreas Aristidou, Ariel Shamir, Moshe Mahler, and Eakta Jain, "Adult2child: Motion style transfer using cyclegans," in *Motion, Interaction and Games*, pp. 1–11. 2020.

[6] Kfir Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen, "Unpaired motion style transfer from video to animation," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 64, 2020.

[7] Yu-Hui Wen, Zhipeng Yang, Hongbo Fu, Lin Gao, Yanan Sun, and Yong-Jin Liu, "Autoregressive stylized motion synthesis with generative flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2021, pp. 13612–13621.

[8] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo, "Artflow: Unbiased image style transfer via reversible neural flows," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2021.

[9] Xun Huang and Serge Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*, 2019.

[12] Diederik P Kingma and Prafulla Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in Neural Information Processing Systems (NeurIPS)*, p. 10236–10245, 2018.

[13] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[14] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[15] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio, "Density estimation using real nvp," *International Conference on Learning Representations (ICLR)*, 2017.