Style-ERD: Responsive and Coherent Online Motion Style Transfer

Tianxin Tao[1] Xiaohang Zhan Zhongquan Chen[2] Michiel van de Panne[1]
[1] University of British Columbia [2] University of California, Davis
{taotianx,van}@cs.ubc.ca xiaohangzhan@outlook.com czqchen@ucdavis.edu

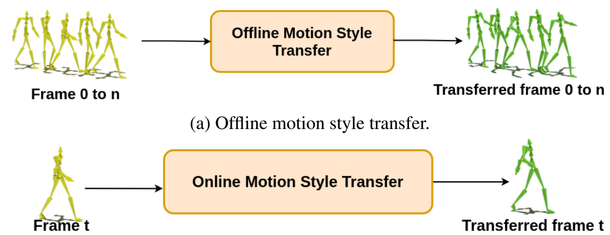
Abstract

Motion style transfer is a common method for enriching character animation. Motion style transfer algorithms are often designed for offline settings where motions are processed in segments. However, for online animation applications, such as real-time avatar animation from motion capture, motions need to be processed as a stream with minimal latency. In this work, we realize a flexible, high-quality motion style transfer method for this setting. We propose a novel style transfer model, Style-ERD, to stylize motions in an online manner with an Encoder-Recurrent-Decoder structure, along with a novel discriminator that combines feature attention and temporal attention. Our method stylizes motions into multiple target styles with a unified model. Although our method targets online settings, it outperforms previous offline methods in motion realism and style expressiveness and provides significant gains in runtime efficiency.

1. Introduction

Animators commonly seek to create stylized motions to express the characters' personalities or emotions, thus making characters more lifelike. Since many computer animation techniques are based on motion capture data, the variety and diversity of the motion data play an essential role in the quality of the resulting animation. However, a capture-everything approach scales poorly if there is a need to capture every style, *e.g.*, *childlike* or *depressed*, for every motion type. Hence, animators usually capture motion in a neutral style and then stylize them by hand, which is again laborious. This motivates automated methods for stylizing existing motions according to desired target-style labels.

In this work, we develop a novel motion style transfer framework capable of stylizing streaming input motion data for online applications, which we define as *Online Motion Style Transfer*. As shown in Fig. 1, current motion style transfer methods [1,5,17,19,38,54] with deep learning models require a motion segment as input, and produce a transferred motion segment as output, and where the segment has



(a) Offline motion style transfer.

(b) Online motion style transfer.

Figure 1. a) Offline motion style transfer processes motion segments, while b) online motion style transfer processes motions in stream.

a minimum duration of 1 s. While these methods make significant progress on a difficult problem, with a subset being described as being real-time [47, 56], they still suffer from startup latency caused by waiting for the multiple frames required as input. For *online motion style transfer*, only the current frame is processed by the model, which enables the direct processing of the stream of motion data. We believe such transfer methods are more suitable for many novel applications requiring streaming motion data. For example, in animating a human avatar, motion is captured online to animate the virtual avatar in real-time, and the streaming motion data needs to be processed with minimal latency. Online motion style transfer can also be easily incorporated into the workflows of real-time motion systems, such as games, interactive exhibitions, and augmented reality with minimal additional latency.

Motion style transfer exists as a long-standing research problem due to several difficulties, among many: (1) lack of a standardized qualitative style representation for motions, (2) difficulty in handling and generating temporally correlated data, (3) a lack of temporally registered motion data in different styles. Several approaches [20, 46, 56, 59] aim to solve this problem with manually designed models. However, they often fail to generalize well to large motion datasets with various styles. Researchers have developed more scalable methods with the rapid progress of deep learning machinery [1, 6, 17, 36]. However, only a few of them can transfer the motion to multiple target styles with a unified model [1, 38]. On top of the aforementioned challenges, *online* motion style transfer poses more difficulties because style and content are ill-defined and unrecognizable within one frame, yielding low-quality transfer results. Current offline motion style transfer methods are commonly conditioned on multiple input frames in order to understand the motion semantics, thus realizing better transfer at the cost of introducing non-trivial latency.

To accomplish high-quality, efficient motion style transfer with minimal latency, we embed knowledge regarding the previous frames in the memory of the motion transfer module in order to infer and track the style and content. The transfer module is thus aware of the context of content and style even when only presented with the current frame. We adapt the Encoder-Recurrent-Decoder framework to the online motion style transfer task by designing novel recurrent residual connections to capture features for each style. We name this novel architecture as *Style-ERD*. In *Style-ERD*, we enable each residual connection to learn its own initial hidden state h_0 conditioned on the style and content label. The learned hidden states are vital to the responsiveness of the style transfer results. In addition, to produce temporally coherent motions, we design a new discriminator with feature and temporal attention, *FT-Att Discriminator*, to supervise the post-transfer style. As a result, our deep learning model demonstrates a strong capability to perform the desired motion style transfer efficiently and with minimal latency.

The contributions of this work are as follows: (1) We introduce the online motion style transfer problem and aim to stimulate future research into this area to facilitate real-time animation applications. (2) We present a novel framework, *Style-ERD*, as well as a new supervision module, *FT-Att Discriminator*, achieving the goal of style transferring motion with minimum latency. Our style transfer framework provides a $5\times$ reduction in compute time, as compared with the current state-of-the-art approach. (3) Our method can transfer the motion into its stylistic counterpart with high fidelity, showing better style transfer as compared to offline methods.

2. Related Work

2.1. Motion Synthesis and Control

Motion synthesis has been a long-standing research problem in computer animation and computer vision. Common data-driven methods are designed around motion graphs and search algorithms [3, 25–28, 33, 43, 52], or Principle Component Analysis (PCA) [4, 50]. These methods are typically non-parametric in nature, in which case they demand large-and-complete datasets with limited ability to generalize .

Recently, deep learning methods have also been applied to motion synthesis, motivated by their potential for scalability, generalization, and compute efficiency. Holden *et al*. [18] introduced a feedforward neural network model named Phase-Functioned Neural Networks (PFNN) with special weight blending mechanism. Further work built upon the weight blending mechanism of PFNN and improved its generalizability [60], interaction [48] and responsiveness [49]. Recurrent networks, including those enabled by Long Short-Term Memory (LSTM) and variations thereof, are another widely adopted structure for motion generation. Fragkiadak *et al*. [7] uses the Encoder-Recurrent-Decoder (ERD) framework to predict the next pose given the current pose. Martinez *et al*. [35] proposed a residual sequence-to-sequence architecture to learn differences between successive poses. The ERD framework was further extended to generate animation conditioned on keyframes [13, 14]. Given a physics-based simulator, optimal control techniques [24, 41, 51] and reinforcement learning approaches [39, 40, 57, 58] also tackle the problem of motion generation with reference trajectories.

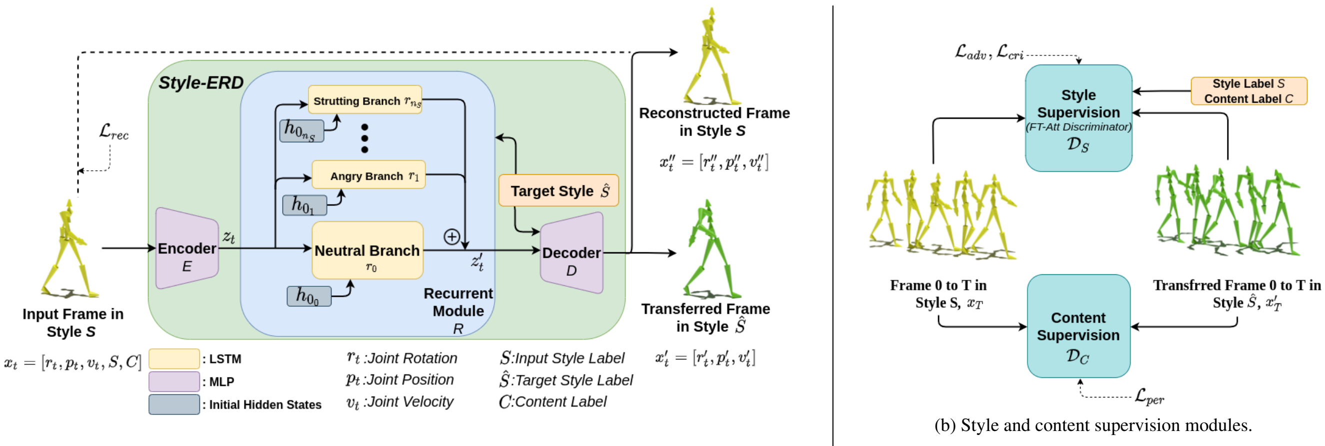
2.2. Image Style Transfer

Style transfer for images was explored in [9] through features extracted by convolutional neural networks. Johnson *et al*. [22] proposed a specific form of perceptual loss to accelerate the process. Later, instance normalization (IN) was proposed to normalize the style of image [53], allowing the style to be manipulated by varying the mean and variance of IN [21]. Recently, impressive progress has been made towards enhancing picture quality [45, 61], enabling user control [10, 42], improving runtime efficiency [8, 29, 30] and accomplishing arbitrary style transfer [12, 21, 55]. In our work, the content supervision module is inspired by perceptual loss [22] and the style normalization effects introduced via IN [53].

2.3. Motion Style Transfer

In early work, Hsu *et al*. [20] proposed the use of a linear time-invariant model to represent motion style variance. Shapiro *et al*. [46] deployed Independent Component Analysis (ICA) to separate motions into different style components which can be adjusted to form stylized motions. Spectral domain features have also been used to capture the style differences between motions in a way that is largely invariant to the content [59]. Xia *et al*. [56] proposed a local mixture of autoregressive models to extract the complex relationships between styles of motion.

However, these models suffer from scalability issues and can still be slow for runtime use, motivating a recent focus on neural network methods. Holden *et al*. [17, 19] propose a style transfer framework composed of a pretrained motion manifold to supervise content and Gram matrices to represent style of the motion. Aberman *et al*. [1] adopt the



(b) Style and content supervision modules.

(a) Style transfer module: *Style-ERD*.

Figure 2. Framework overview. (a) The input frame is encoded, forwarded in the recurrent module, and finally decoded to original motion format in joint rotations. (b) At training time, the output motion segment is required to achieve the target style while preserving the content.

AdaIN layer applied in image style transfer to the motion style transfer task, and show that content can be stripped of style by IN. Inspired by that, Park *et al*. [38] replace the 1D temporal convolution structure with a spatial-temporal graph convolution. A residual model can also be used to extract the style ingredients of the motion [36]. Wen *et al*. [54] propose a style transfer framework with generative flow.

Inspired by [1, 17], our framework adopts the two-way constraints on style and content. In contrast to those previous deep learning models, our framework can operate with single input frames while delivering high-quality motion in a target style and being considerably more compute efficient in an online setting.

3. Methodology

Our goal is to develop an online motion style transfer algorithm with high-quality transfer and minimal latency. In particular, we seek to reduce the number of input frames needed at each timestep to synthesize the current frame of the target style. However, with fewer input frames, the style transfer model may err in interpreting the style and content. We therefore leverage a recurrent model to maintain relevant estimates of the style and content. Our framework consists of three components: a style transfer module, a style supervision module and a content supervision module. An overview of our method is displayed in Fig. 2.

Inspired by the ERD framework [7], we name the style transfer module as *Style-ERD*. It is characterized by multiple recurrent residual connections and by hidden states that have learned initial values which are conditioned on the input. The novel recurrent residual connections play a key role in the success of our method because the memory of past frames provides style and content information regarding the current frame while the residuals capture the features of each style. The *Style-ERD* model achieves the goal of style transfer from each single frame input in real-time.

The style transfer module (*Style-ERD*) delivers poor style transfer when used with only a reconstruction loss. Conditioning on style and content before and after the transfer task can boost the style transfer effects. Both supervision modules take multiple frames of motion as input. This multi-frame input to supervision modules does not hinder the online property of our method since the supervision modules are unused at inference time. We propose a novel attention mechanism that spans both feature space and temporal space of the feature maps in the style discriminator, *FT-Att Discriminator*, to enable the style transfer module to avoid mode-collapse issues that would otherwise preclude modeling the desired variety of style and content. The content supervision module adopts the idea of perceptual loss [22] with features that focus on the content of motion.

3.1. Architecture

Motion Transfer Module. Our style transfer module, *Style-ERD*, consists of three parts: an encoder E to compress the input frame x_t, a recurrent module R consisting of residual connections to learn the offsets of different styles, and a decoder D to map the latent code back to the transferred motion frame x_t' represented by joint rotations.

The input frame x_t contains joint rotations in unit quaternions $r_t \in \mathbb{R}^{4\times J}$, joint positions offset by the root $p_t \in \mathbb{R}^{3\times J}$, and linear joint velocity $v_t \in \mathbb{R}^{3\times J}$ at timestep t, where J is the number of joints. Additionally, the encoder is conditioned on the style label S and content label C of the input frame x_t while the decoder is conditioned on the target style label, $\hat{S}$. Both style and content labels are represented by one-hot vectors.

The encoder consists of a two-layer MLP (multilayer

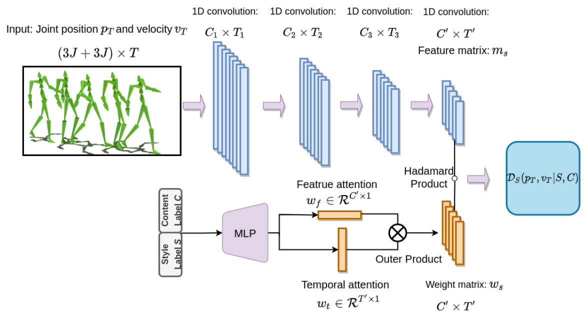


Figure 3. *FT-Att Discriminator* structure. The discriminator forms the weight matrix via an outer product between two attention vectors, then applies the attention matrix on the extracted features via a Hadamard product.

perceptron) to compress the input to a low-dimensional space, z. We choose to compress the input for two reasons: (1) A low-dimensional latent space can simplify capturing an abstracted style representation; and (2) With this low-dimensional bottleneck and the given training tasks, the encoder can normalize the style of the input frame to neutral.

The recurrent module is designed as a stack of LSTM layers, *i.e.*, $R = [r_0, r_1, \dots, r_{n_S}]$, with each one learning a style offset of one specific style with respect to the neutral style. Here, we assign the recurrent branch r_0 to learn the features of the neutral style, which serves as the basis for all other style offsets. Then, the residual value computed by the target style branch $r_{\hat{S}}(z_t)$ is added to the neutral branch output, $r_0(z_t)$. Thus, the operation of our recurrent module can be expressed as: $z'_t = r_0(z_t) + r_{\hat{S}}(z_t)$

In addition, it is challenging to perform style transfer on the first few frames because the memory of the LSTM layers may not have seen enough information to infer the necessary style information. Common ways to initialize hidden states include setting the hidden states to zeros [31] or random noise [63], and treating the initial hidden states as parameters for the network to learn [13]. In order to enhance the performance on the first few frames, we propose learning multiple initial states $h_0 = [h_{0_0}, h_{0_1}, \dots, h_{0_{n_S}}]$ conditioned on the style label S and the content label C. Specifically, assume there are n_C different content labels; the neutral branch r_0 learns n_C initial hidden states simultaneously and selects the one corresponding to the content label. Similarly, each style branch learns its own initial hidden state for its corresponding style.

The conditional decoder D expands the latent code z'_t back to joint rotations in quaternions and linear joint velocities through four MLP layers, further conditioned on the target style label $\hat{S}$. Joint positions are also computed via forward kinematics of joint rotations.

Style Supervision Module. We propose a novel discriminator $\mathcal{D}_S$ with attention mechanism, *FT-Att Discriminator*, to supervise the style transfer task. Fig. 3 shows the structure of *FT-Att Discriminator*. Unlike the style transfer module, our discriminator receives a segment of T ($T = 24$) frames as input to infer the style of the input motion. Each frame is represented by the concatenation of joint positions offset by the root $p_t \in \mathbb{R}^{3 \times J}$ and linear joint velocities $v_t \in \mathbb{R}^{3 \times J}$ as Aberman *et al*. [1] found positions are more representative for styles than rotations.

The discriminator attempts to distinguish generated motion from real motion samples according to its style label S and content label C. We adopt a 1D temporal convolution structure similar to [1, 32] to extract the 2D feature matrix $m_s \in \mathbb{R}^{C' \times T'}$ but add novel attention modules conditioned on style and content. The intuition behind the attention modules is that the discriminator should judge motion style according to the desired style and its content by weighing the features unevenly. The attention modules are comprised of MLP layers with style and content labels as input, and outputs feature attention vector $w_f \in \mathbb{R}^{C'}$ and temporal attention vector $w_t \in \mathbb{R}^{T'}$. We then compute an outer product between the feature attention w_f and temporal attention w_t to form a weight matrix $w_s \in \mathbb{R}^{C' \times T'}$: $w_s = w_f \bigotimes w_t$. Finally, the weight matrix w_s is applied to the feature matrix m_s via a Hadamard product, *i.e.*, element-wise multiplication. Thus, given the feature map m_s and weight matrix w_s, the output of the discriminator can be expressed as:

$$\mathcal{D}_S(p_T, v_T|S, C) = \sum_{i=0}^{C'} \sum_{j=0}^{T'} (m_s \circ w_s)[i, j] \tag{1}$$

Content Supervision Module. At the same time as transferring style, we expect the content of the motion to be unaltered. We apply perceptual loss [22] based on a pre-trained content-classification network $\mathcal{D}_C$ to preserve content. The classification network follows the same convolution layers as the discriminator while accepting joint rotations, joint positions and velocities as input. Inspired by the style normalization effects of IN [1, 21], each convolution layer is followed by IN such that the classification network focuses on the motion content and disregards the style.

3.2. Training

The training process is analogous to the training of the standard Generative Adversarial Network [11]. As a generator, the proposed style transfer module *Style-ERD* is trained to reconstruct the input frame and to stylize the frame to the target style to fool the discriminator, while the objective of the *FT-Att Discriminator* is to distinguish the transferred motion from real data samples. We add perceptual loss and further adopt a gradient penalty in the discriminator to improve the overall training process. For simplicity and clarity, we use the notation (.)' to indicate attributes of the transferred results.

Reconstruction. Given a motion input x_t and a target style label $\hat{S}$ identical to the original style, the motion transfer module should output an identical frame $x''_t = [r''_t, p''_t, v''_t]$. This reconstruction task can be viewed as an auxiliary task to learn disentangled style variance for each residual branch. The reconstruction loss is applied over the joint rotations in quaternions r_t, translational joint positions p_t and velocities v_t:

$$\mathcal{L}_{quat}(r_t, r''_t) = \left\| cos^{-1}(r_t \cdot r''_t) \right\|^2, \tag{2}$$

$$\mathcal{L}_{rec_t} = \mathcal{L}_{quat}(r_t, r''_t) + \frac{1}{2} \left\| p_t - p''_t \right\|^2 + \left\| v_t - v''_t \right\|^2, \tag{3}$$

where $\mathcal{L}_{quat}$ denotes the quaternion difference represented by the angle between two rotations in radians. More details about $\mathcal{L}_{quat}$ can be found in the supplementary material.

Style Transfer. We adopt the Least Squares Generative Adversarial Networks (LSGAN) [34] framework to train the *FT-Att Discriminator*. We assume that neutral style motion serves as a common basis for other styles. At training time, we set the target styles of all neutral motion to be any other existing style in the dataset, while motions in other styles except neutral should be transferred to the neutral style. With these training objectives, we expect the encoder E to normalize the input motion to a neutral style. Therefore, the adversarial loss is applied to manipulate the style of the motion by fooling the critic. At the same time, the critic is trained to distinguish the fake generated motion from the real motion sample:

$$\mathcal{L}_{adv} = \left\| \mathcal{D}_s(p'_T, v'_T | \hat{S}, C) \right\|^2, \tag{4}$$

$$\begin{aligned} \mathcal{L}_{cri} = & \left\| \mathcal{D}_s(p_T, v_T | S, C) - 1 \right\|^2 \\ & + \left\| \mathcal{D}_s(p'_T, v'_T | \hat{S}, C)) + 1 \right\|^2. \end{aligned} \tag{5}$$

Gradient Penalty. GAN training is known to suffer from instability and convergence issues, with multiple approaches proposed to address this issue [15, 16, 23, 37, 44]. In this work, we apply a gradient penalty on the real samples to prevent the discriminator from creating a non-zero gradient orthogonal to the data manifold when the generator produces the true data distribution [37]:

$$\mathcal{L}_{gp} = \left\| \nabla_{\hat{x}} \mathcal{D}_s(\hat{x})|_{\hat{x}=(p_T, v_T | S, C)} \right\|^2 \tag{6}$$

Perceptual Loss. In order to preserve content before and after the transfer, we add a perceptual loss [22] $\mathcal{L}_{per}$ to the generator with a pretrained multi-class content-classification network $\mathcal{D}_C$. The perceptual loss encourages the convolution feature maps ϕ extracted by the classification network before and after the transfer to match:

$$\mathcal{L}_{per} = \left\| \phi - \phi' \right\|^2. \tag{7}$$

The final loss applied to the motion style transfer module (generator) is a weighted sum of reconstruction, adversarial, perceptual loss while a gradient penalty is added to the discriminator loss:

$$\mathcal{L}_{gen} = \sum_{t=0}^{T} \mathcal{L}_{rec_t} + w_{adv} \mathcal{L}_{adv} + w_{per} \mathcal{L}_{per}, \tag{8}$$

$$\mathcal{L}_{dis} = \mathcal{L}_{cri} + w_{gp} \mathcal{L}_{gp}, \tag{9}$$

where we set $w_{adv} = 1$, $w_{per} = 0.1$ and $w_{gp} = 128$.

4. Experiments

We test our framework based on the dataset provided by Xia *et al*. [56]. We first compare the style transfer results with previous offline style transfer methods [1, 38]. We adopt the quantitative Fréchet Motion Distance (FMD) metric proposed in [38] to evaluate the quality of the transferred results, which is a variant of the Fréchet Inception Distance (FID) [16]. We trained a denoising autoencoder as the feature extractor for FMD. Additionally, we perform qualitative evaluations on the transferred results according to three criteria: style expressiveness, temporal consistency and content preservation. Second, we measure the runtime efficiency of our method, to assess feasibility for use in real-time online applications. We also demonstrate that insufficient input frames can degrade the results of offline style transfer methods [1, 38], and that our approach works well in the online transfer setting, with minimal latency. We then experiment with interpolation of the style ingredients of the transfer module. Finally, we test the generalization property of our methods. For better visualization and comparison, we refer readers to our supplementary video and material.

4.1. Dataset

Our method is trained and tested on the dataset supplied by Xia *et al*. [56]. The motion clips cover $n_S = 7$ different styles and $n_C = 5$ distinct types of content. We downsample the original 120fps motion data to 60fps, which results in around 1500 motion clips in total. Each motion clip spans between 28 to 80 frames. To ensure that the model is agnostic to the choice of initial frame, we further split each motion clip into multiple windows of $T = 24$ frames, with an overlap of 4 frames. The motion clips are assigned randomly into the training and test sets without any overlap.

4.2. Style Transfer Quality

Tab. 1 lists the FMD scores for our method and alternative approaches [1, 38]. Our method achieves the lowest FMD, which is indicative of its transferred results being closer to real stylized motion samples, in a probabilistic sense, than the alternative methods. Fig. 4 shows three sets of style transfer results using our method and those proposed by Aberman *et al*. [1] and Park *et al*. [38]. In addition

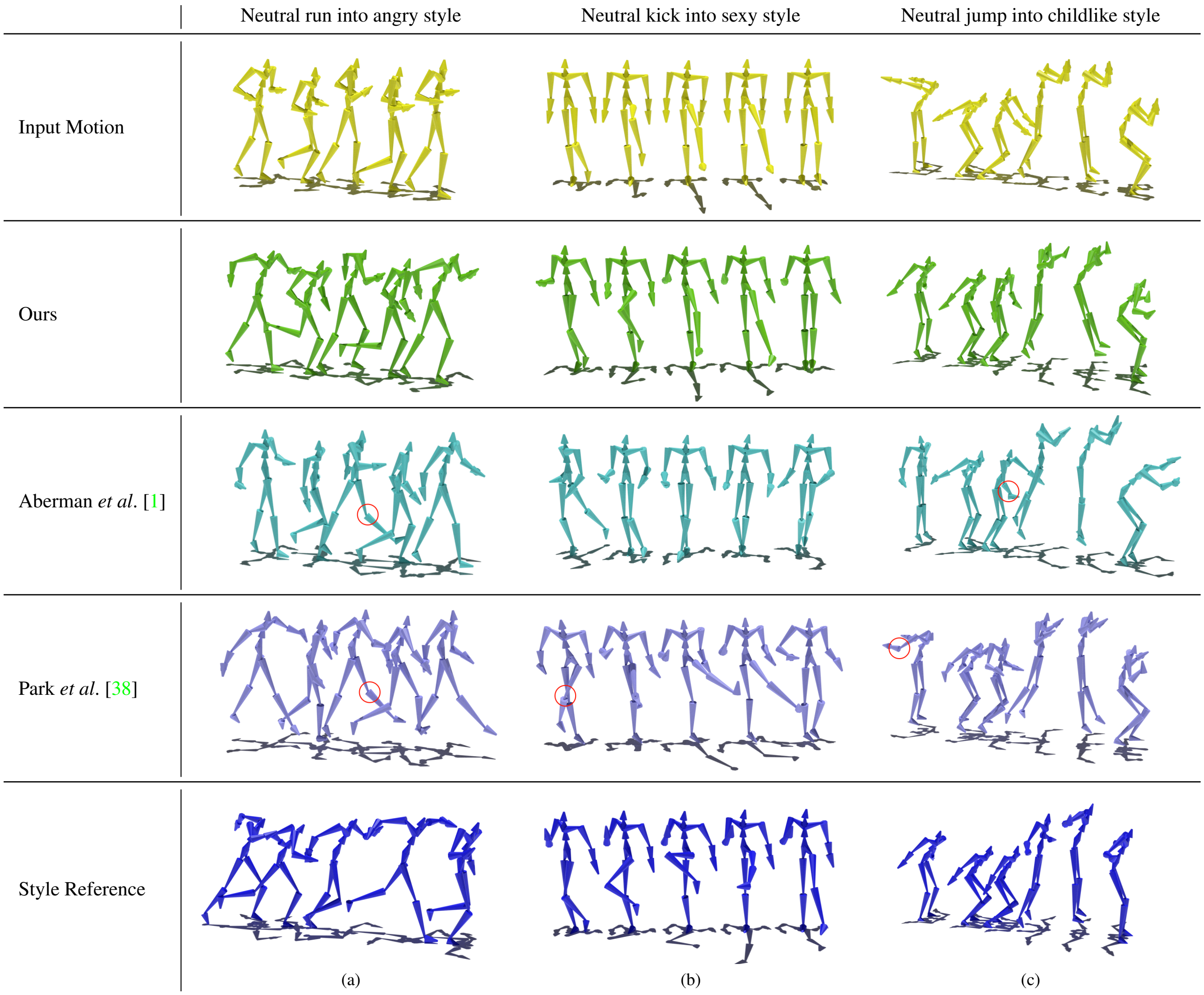


Figure 4. Style transfer comparison: show the input motion, style transfer results produced by our method, the methods in Aberman *et al*. [1] and in Park *et al*. [38]. Example artifacts are circled in red. Style reference is the existing input motion content in the target style, as provided by the test set. Ideally, transferred results should resemble the style references, with the motion content remaining unchanged.

Method	FMD↓	Ablation study	FMD↓
Aberman *et al*. [1]	563.41	Ours w/ $\mathcal{L}_{gen} = \mathcal{L}_{rec}$	285.17
Park *et al*. [38]	190.94	Ours w/ $\mathcal{L}_{gen} = \mathcal{L}_{rec} + \mathcal{L}_{adv}$	75.40
Ours	**61.95**	Ours w/ $\mathcal{L}_{gen} = \mathcal{L}_{rec} + \mathcal{L}_{per}$	380.78
		Ours w/o attention	382.57
		Ours w/o learnt initial states	309.70

Table 1. Quantitative evaluations on our method and existing motion style transfer methods, as well as the ablation study results.

to the above objective measures, the motions transferred with our method subjectively resemble the existing motions in the target styles while Aberman *et al*. [1]'s method, which is an unsupervised model, may exhibit a limited change of style (see Fig. 4c). Furthermore, our approach produces consistent style effects throughout the motion cycle, while results produced by [38] may contain artifacts at the beginning and the end of the motion (see Fig. 4b). In terms of content preservation, the content of the transferred motion by our method is easily recognizable and remains identical to the input. In contrast, the motions produced by [1] and [38] exhibit a degree of content variation after the style transfer (see Fig. 4a and Fig. 4b). Our framework can also stylize heterogeneous action sequences, *i.e*., one motion clip with multiple content types. Results are included in the supplementary material.

4.3. Efficiency

We evaluate our method in an online setting based on two criteria: (1) startup latency and (2) runtime. Startup latency is important in an online motion transfer task be-

Method	Runtime (ms) Online	L=60	L=90	L=120	Input Length L (frames)
Aberman *et al.* [1]	26.82	27.02	27.56	27.74	32
Park *et al.* [38]	10.21	10.20	10.24	**10.25**	64
Xia *et al.* [56]	18.10	217.2	325.8	434.4	5
Ours	**1.73**	**7.47**	**10.18**	13.22	**1**

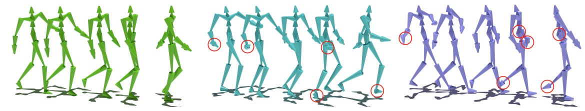
Table 2. Runtime measurements in an online setting and designed input frame number. Input length indicates the ideal length of input segment, which relates to latency.

cause the delay can directly impact user interaction. In Tab. 2, we list the runtime of our method and alternative approaches [1,38,56], and the designed input length at training. Although those methods are not restricted to a particular length of input, we find they often have degraded motion transfer when used with fewer input frames. Thus, the designed input frames serve as an indicator for latency in a streaming application. In Fig. 5, the motion transfer results by [1, 38] contain noticeable artifacts on the rotation of the limbs when only 10 frames are supplied as content input. On the other hand, our framework can produce good motion transfer results from the very beginning due to the learned initial hidden states and recurrent structure.

We also measure the runtime for style transfer, per frame, in an online setting and provide additional measurements on multiple frames as reference. The runtime performance is tested on a PC with NVIDIA GeForce GTX 1060 GPU (6GB), with the exception of the measurements for [56], for which use their documented results. Since previous methods [1, 38] are not designed for online purposes, we pad the current frame with the past 32 frames in the online runtime tests. In terms of runtime, our method significantly outperforms the alternative transfer methods, providing a speedup of a factor of 5 as compared to the second fastest runtime of Park *et al.* [38], in the online setting. A shorter input window enables lower latency of handling streaming motion data to produce realistic transfer results. Despite the method introduced by Xia *et al.* [56] requiring the second shortest input window, their algorithm operates much slower than ours with the default k-nearest neighbors (kNN) implementation. Our method also achieves the fastest runtime when transferring short motion segments. However, the sequential nature slows down the runtime of our method when used in conjunction with a long input sequence, although this is not our focus. In conclusion, our method outperforms previous style transfer methods by a large margin in both runtime and startup latency. Our method is sufficiently fast for current online streaming applications at up to 120 Hz.

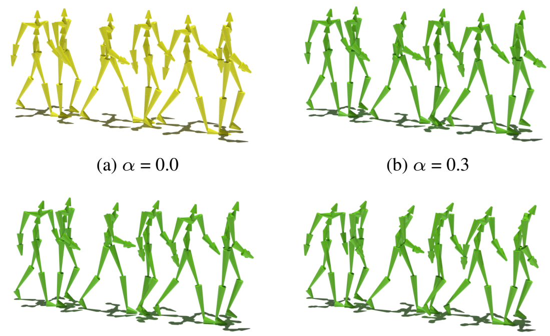
4.4. Style Interpolation

We also experiment with style interpolation. More specifically, we set the latent code z'_t to be either a linear combination of two target styles $\hat{S}, \tilde{S}$: $z'_t = r_0(z_t) + \alpha * r_{\hat{S}}(z_t) + (1-\alpha) * r_{\tilde{S}}(z_t)$, or a scaled value of a designated target style: $z'_t = r_0(z_t) + \alpha * r_{\hat{S}}(z_t)$, where α is a scalar in $[0, 1]$. As shown in Fig. 6, motions with different style intensity can be produced by adjusting the coefficient α of a single target style. In the supplementary material, we demonstrate generating motions in a mixture of styles.



(a) Our method (b) Aberman *et al.* [1] (c) Park *et al.* [38]

Figure 5. Style transfer results of the first ten frames. Task: Neutral walk into proud style. Noticeable artifacts of awkward joint rations are circled in red.



(a) α = 0.0 (b) α = 0.3 (c) α = 0.6 (d) α = 1.0

Figure 6. Style interpolation. Task: Neutral walk into depressed style. $\alpha = 0$ is equivalent to reconstruction of input motion.

4.5. Generalizability

Motion capture data naturally contains significant variability due to motion speeds, the actor skeleton, *etc.* The style transfer algorithm is likely to encounter out-of-distribution motion data. To examine robustness to motion pertubations, we test our framework and the approaches proposed in [1, 38] on retargetted public motion data from *Mixamo* [2]. As shown in Fig. 7, our method can successfully transfer the unseen motion data to the desired style, while the two alternative methods contain apparent artifacts.

5. Ablation Studies

We conduct extensive ablation studies to verify the relevance of multiple components in our framework, including each term in the generator loss $\mathcal{L}_{gen}$, the attention mechanism in *FT-Att Discriminator*, and learned initial hidden states. Quantitative measurements based on FMD are listed in Tab. 1, which reveal the necessity of the modules in our framework. Qualitative visualization for each ablation experiment is included in the supplementary material.

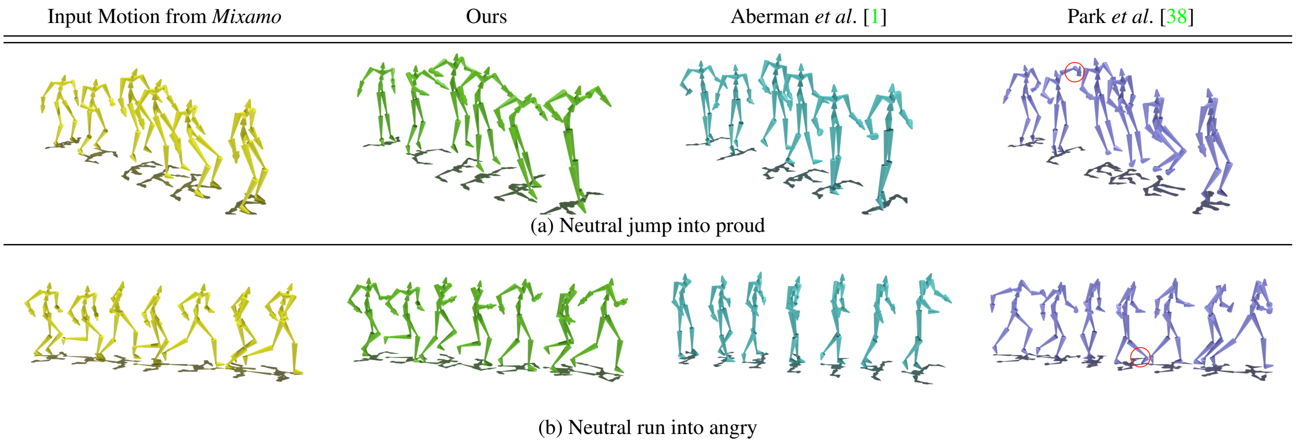


Figure 7. Generalization test: style transfer results for our methods and alternative methods [1, 38] on motion data from *Mixamo*. *Mixamo* motion data is unseen at training time and has different motion patterns. Some obvious artifacts are circled in red.

Which terms matter in the loss function? The residual model has previously been proposed with a reconstruction loss to generate stylized motion by Mason *et al*. [36]. Therefore, we evaluate the need for the supervision modules, and set the training objective $\mathcal{L}_{gen}$ to be a pure regression task using $\mathcal{L}_{rec}$ in Eq. 3. As a result, we find that *Style-ERD* fails to produce motions in the target style when only trained on the reconstruction objective. Instead, the transferred motion is minimally different from the input motion.

Since the attention module in *FT-Att Discriminator* depends on both content and target style labels, the discriminator can theoretically supervise both style and content of the motion. Thus, we remove the perceptual loss from the generator loss, *i.e*., $\mathcal{L}_{gen} = \mathcal{L}_{rec} + \mathcal{L}_{adv}$. We find that some style transfer results experience obvious content changes after transfer without the content supervision module.

We also attempt to only remove the adversarial loss $\mathcal{L}_{adv}$ from the generator loss, *i.e*., $\mathcal{L}_{gen} = \mathcal{L}_{rec} + \mathcal{L}_{per}$. Without the adversarial loss, the *Style-ERD* model behaves similarly to an autoencoder, which can only reconstruct the input.

Can we discard the attention mechanism in the discriminator? We replace the proposed *FT-Att Discriminator* with a standard multi-class discriminator used in [1, 32]. The multi-class discriminator has a shared convolutional feature extractor and individual heads for each style discrimination task. We find that the full model produces results with more expressive styles while the baseline discriminator fails to capture some style features and contains artifacts.

Do learnable initial hidden states help? Instead of setting the hidden states as learnable parameters, the initial hidden states of LSTM layers are set to zero as done in [31]. We find that the transfer results can exhibit a temporal shift, especially for short motion clips. For example, the start of a motion can be transferred to the middle stage of the motion sequence.

6. Conclusion

In summary, we introduce a novel style transfer model, *Style-ERD*, with the Encoder-Recurrent-Decoder structure as a solution to the online motion style transfer problem. In our style-modeling framework, the memory module encapsulates the style and content context of the past frames. We incorporate learnable initial hidden states conditioned on the input to enhance the responsiveness of our method. Furthermore, we propose a new discriminator *FT-Att Discriminator* with attention on both feature and temporal dimensions to supervise the style of the output. Our method enables stylizing the input frame with minimal delay while significantly accelerating the transfer process in an online setting. Compared with previous methods, our *Style-ERD* model is able to produce more realistic style transferred motions while being robust to perturbations in the input.

Limitations Our method still requires paired motion data in different styles. This could possibly be addressed by adopting the cycle consistency idea [62]. Also, our model is conditioned on content labels, which could be unavailable. As with other approaches, we still rely on inverse kinematics based cleanup to remove minor foot sliding. As future work, we are also interested in optimizing our framework to quickly adapt to unseen styles in a few-shot by matrix decomposition as shown in [36].

Societal Impact The motion data is biased towards one default skeleton, ignoring the needs of minority groups, *e.g*., children, elders and people with physical disabilities. This can be mitigated by motion retargetting in the preprocessing step. Motion style transfer could be used for creating human animations with the intention to mislead.

References

[1] 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)*, 39(4):64–1, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[2] Adobe's Mixamo. https://www.mixamo.com, Accessed: 2021-11-10. 7

[3] Okan Arikan and David A Forsyth. Interactive motion generation from examples. *ACM Transactions on Graphics (TOG)*, 21(3):483–490, 2002. 2

[4] Jinxiang Chai and Jessica K Hodgins. Performance animation from low-dimensional control signals. In *ACM SIGGRAPH 2005 Papers*, pages 686–696. 2005. 2

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

[6] Han Du, Erik Herrmann, Janis Sprenger, Noshaba Cheema, Somayeh Hosseini, Klaus Fischer, and Philipp Slusallek. Stylistic locomotion modeling with conditional variational autoencoder. In *Eurographics (Short Papers)*, pages 9–12, 2019. 1

[7] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015. 2, 3

[8] Wei Gao, Yijun Li, Yihang Yin, and Ming-Hsuan Yang. Fast video multi-style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3222–3230, 2020. 2

[9] 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*, pages 2414–2423, 2016. 2

[10] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3985–3993, 2017. 2

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 4

[12] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. Arbitrary style transfer with deep feature reshuffle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8222–8231, 2018. 2

[13] Félix G Harvey and Christopher Pal. Recurrent transition networks for character locomotion. In *SIGGRAPH Asia 2018 Technical Briefs*, pages 1–4. 2018. 2, 4

[14] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020. 2

[15] Tamir Hazan, George Papandreou, and Daniel Tarlow. Adversarial perturbations of deep neural networks. 2017. 5

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[17] Daniel Holden, Ikhsanul Habibie, Ikuo Kusajima, and Taku Komura. Fast neural style transfer for motion data. *IEEE computer graphics and applications*, 37(4):42–49, 2017. 1, 2, 3

[18] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017. 2

[19] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 1, 2

[20] Eugene Hsu, Kari Pulli, and Jovan Popović. Style translation for human motion. In *ACM SIGGRAPH 2005 Papers*, pages 1082–1089. 2005. 1, 2

[21] 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*, pages 1501–1510, 2017. 2, 4

[22] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2, 3, 4, 5

[23] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 5

[24] Nam Hee Kim, Hung Yu Ling, Zhaoming Xie, and Michiel van de Panne. Flexible motion optimization with modulated assistive forces. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 4(3):1–25, 2021. 2

[25] Lucas Kovar and Michael Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics (ToG)*, 23(3):559–568, 2004. 2

[26] Lucas Kovar, Michael Gleicher, and F Pighin. Motion graphs. *ACM Trans. Graphics*, 21(3):473–482, 2002. 2

[27] Jehee Lee, Jinxiang Chai, Paul SA Reitsma, Jessica K Hodgins, and Nancy S Pollard. Interactive control of avatars animated with human motion data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 491–500, 2002. 2

[28] Jehee Lee and Kang Hoon Lee. Precomputing avatar behavior from human motion data. *Graphical models*, 68(2):158–174, 2006. 2

[29] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision*, pages 702–716. Springer, 2016. 2

[30] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast arbitrary style transfer. *arXiv preprint arXiv:1808.04537*, 2018. 2

[31] Zimo Li, Yi Zhou, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. *arXiv preprint arXiv:1707.05363*, 2017. 4, 8

[32] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 10551–10560, 2019. 4, 8

[33] Wan-Yen Lo and Matthias Zwicker. Real-time planning for parameterized human motion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 29–38, 2008. 2

[34] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017. 5

[35] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017. 2

[36] Ian Mason, Sebastian Starke, He Zhang, Hakan Bilen, and Taku Komura. Few-shot learning of homogeneous human locomotion styles. In *Computer Graphics Forum*, volume 37, pages 143–153. Wiley Online Library, 2018. 1, 3, 8

[37] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018. 5

[38] Soomin Park, Deok-Kyeong Jang, and Sung-Hee Lee. Diverse motion stylization for multiple style domains via spatial-temporal graph-based generative model. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 4(3):1–17, 2021. 1, 2, 3, 5, 6, 7, 8

[39] Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. Learning predict-and-simulate policies from unorganized human motion data. *ACM Transactions on Graphics (TOG)*, 38(6):1–11, 2019. 2

[40] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 2

[41] Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. *arXiv preprint arXiv:2008.06389*, 2020. 2

[42] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*, 2017. 2

[43] Alla Safonova and Jessica K Hodgins. Construction and optimal search of interpolated motion graphs. In *ACM SIGGRAPH 2007 papers*, pages 106–es. 2007. 2

[44] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242, 2016. 5

[45] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Bjorn Ommer. A style-aware content loss for real-time hd style transfer. In *proceedings of the European conference on computer vision (ECCV)*, pages 698–714, 2018. 2

[46] Ari Shapiro, Yong Cao, and Petros Faloutsos. Style components. In *Graphics interface*, volume 2006, pages 33–39, 2006. 1, 2

[47] 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*, 2(2):1–17, 2019. 1

[48] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019. 2

[49] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)*, 39(4):54–1, 2020. 2

[50] Jochen Tautges, Arno Zinke, Björn Krüger, Jan Baumann, Andreas Weber, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bernd Eberhardt. Motion reconstruction using sparse accelerometer data. *ACM Transactions on Graphics (ToG)*, 30(3):1–12, 2011. 2

[51] Emanuel Todorov and Weiwei Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 300–306. IEEE, 2005. 2

[52] Adrien Treuille, Yongjoon Lee, and Zoran Popović. Near-optimal character animation with continuous control. In *ACM SIGGRAPH 2007 papers*, pages 7–es. 2007. 2

[53] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 2

[54] 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*, pages 13612–13621, 2021. 1, 3

[55] Xiaolei Wu, Zhihao Hu, Lu Sheng, and Dong Xu. Styleformer: Real-time arbitrary style transfer via parametric style composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14618–14627, 2021. 2

[56] Shihong Xia, Congyi Wang, Jinxiang Chai, and Jessica Hodgins. Realtime style transfer for unlabeled heterogeneous human motion. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015. 1, 2, 5, 7

[57] Zhaoming Xie, Hung Yu Ling, Nam Hee Kim, and Michiel van de Panne. Allsteps: Curriculum-driven learning of stepping stone skills. In *Computer Graphics Forum*, volume 39, pages 213–224. Wiley Online Library, 2020. 2

[58] Zhiqi Yin, Zeshi Yang, Michiel Van De Panne, and KangKang Yin. Discovering diverse athletic jumping strategies. *ACM Transactions on Graphics (TOG)*, 40(4):1–17, 2021. 2

[59] M Ersin Yumer and Niloy J Mitra. Spectral style transfer for human motion between independent actions. *ACM Transactions on Graphics (TOG)*, 35(4):1–8, 2016. 1, 2

[60] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018. 2

[61] Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, and Jimei Yang. Multimodal style transfer via

graph cuts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5943–5951, 2019. 2

[62] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 8

[63] Hans-Georg Zimmermann, Christoph Tietz, and Ralph Grothmann. Forecasting with recurrent neural networks: 12 tricks. In *Neural Networks: Tricks of the Trade*, pages 687–707. Springer, 2012. 4

Style-ERD: Responsive and Coherent Online Motion Style Transfer Supplementary Material

A. Implementation Details

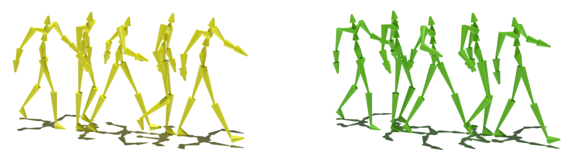
Our method is implemented in PyTorch and trained on a PC with an 8-core AMD Ryzen 7 CPU with a single NVIDIA GeForce GTX 1060 GPU. The overall training process takes about 10 hours to complete. Both the generator and discriminator are optimized by Adam optimizer [2] with learning rates of 10^{-4} and 5×10^{-5} respectively. Our model is trained for 2000 epochs with a batch size of 16. We adopt the default initialization offered by PyTorch for all the neural networks in our model.

The encoder in the style transfer module consists of two MLP layers with $(64, 32)$ neurons and ReLU activation. The latent code z has a dimension of 32. The residual module comprises 6 LSTM layers for the neutral branch r_0 and 4 LSTM layers for other branches $[r_1, \ldots, r_{n_S}]$. The decoder is conditioned on target style labels and comprises 4 MLP layers with $(58, 86, 128, 184)$ neurons and ReLU activation. For the discriminator, we find that replacing ReLU with LeakyReLU accelerates the training.

We experiment with two different loss functions on the quaternion values: L2-norm and $\mathcal{L}_{quat}$ in Eq. (2). Though the four coefficients of a quaternion are continuous and smooth, the rotation value is not evenly-spaced. Optimizing in this not evenly-spaced space with L2-norm can lead to jerky motion, which can be solved by adopting $\mathcal{L}_{quat}$.

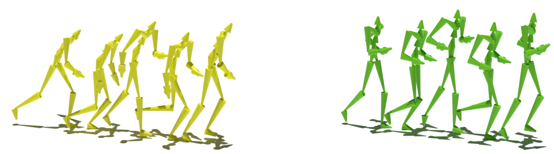
We pre-train a denoising autoencoder with 1D convolution layers as the feature extractor for Frechet Motion Distance (FMD). At training time, the joint rotations are sampled from the dataset, and we add random noise sampled from a normal distribution, $\mathcal{N}(0, 0.03)$ to the joint rotations. The denoising autoencoder is trained to reconstruct the noisy joint rotation input with $\mathcal{L}_{quat}$ in Eq. (2) as the loss function. After the training, we use the activation of the last convolution layer in the encoder as features for the FMD score.

We plan to publish our code, which is currently included in the supplementary material package for reviewing purposes.



(a) Input: Angry walk (b) Output: Strutting walk

Figure 1. Style transfer with non-neutral input motion. Our model is not restricted by the input motion style.



(a) Input: Depressed run (b) Output: Neutral run

Figure 2. Transfer non-neutral input to neutral style.

B. Discussions

B.1. Style Transfer with Unpaired Data

We design our framework to perform motion style transfer with paired motion data. For example, to transfer a neutral walking motion to angry, the dataset should contain an angry walking motion as reference, which does not need to be temporally registered. However, some researchers have demonstrated style transfer with unpaired motion data is also feasible [1]. Such style transfer with unpaired data is defined as heterogeneous style transfer. Exploring style transfer with unpaired data remain as a future work since it lowers the requirement on the dataset and enables to learn more diverse style transfer effects.

B.2. Choice of Input Style

Most qualitative results in this work use input motion in neutral style. Style transfer starting from neutral motion is a more practical problem because more neutral motion exists in the available database. Nevertheless, our framework is not restricted by the input style to be neutral. Fig. 1 shows our method can successfully transfer input motion in angry style to motion in strutting style. Given the training pro-

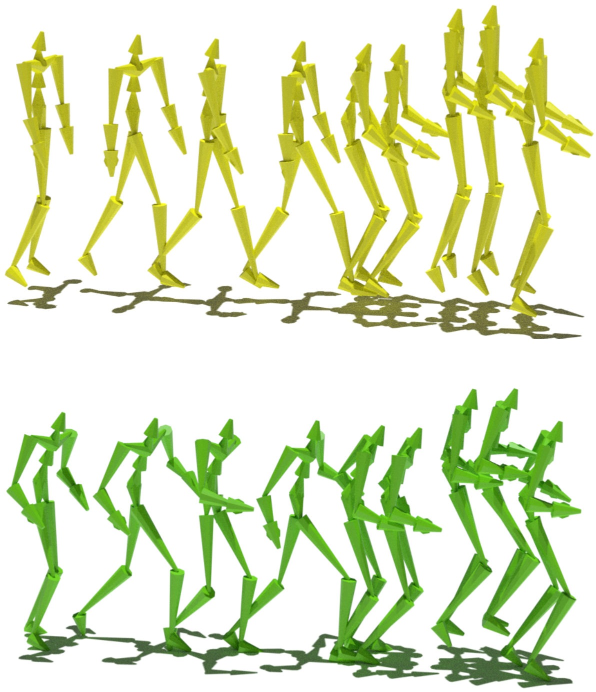
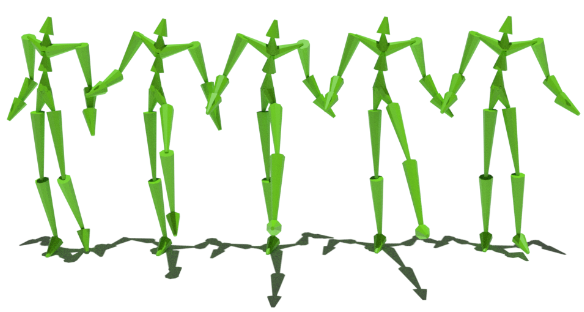


Figure 3. Style transfer on heterogeneous motion sequence. Task: Neutral walk and jump motion to old.

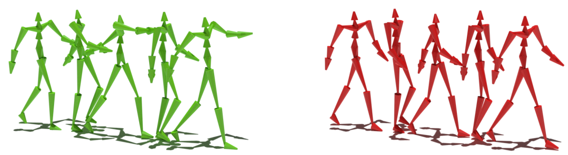
(a) Neutral run to old and depressed (α = 0.5)



(b) Neutral kick to childlike and strutting (α = 0.5)

Figure 4. Mixture of two existing styles.

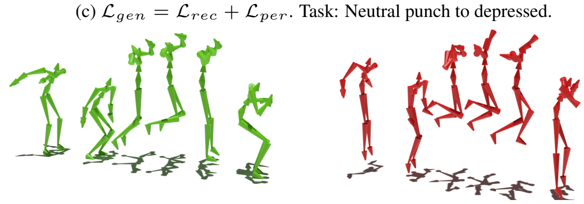
cess, transferring non-neutral motion to another style other than neutral is an unseen task at inference time. Moreover,



(a) $\mathcal{L}_{gen} = \mathcal{L}_{rec}$. Task: Neutral walk to proud.

(b) $\mathcal{L}_{gen} = \mathcal{L}_{rec} + \mathcal{L}_{adv}$. Task: Neutral run to angry.

(c) $\mathcal{L}_{gen} = \mathcal{L}_{rec} + \mathcal{L}_{per}$. Task: Neutral punch to depressed.



(d) Attention mechanism of the discriminator. Task: Neutral jump to sexy.

(e) Learnable initial hidden states. Task: Neutral punch to childlike.

Figure 5. Ablation Studies. Left (green): results produced by the full framework, Right (red): ablation experiment results.

we also experiment with transferring stylized motion to its neutral counterpart, and show the results in Fig. 2. Such results reveal that the encoder E in the style transfer module *Style-ERD* can normalize input motion to neutral style as expected.

When the input motion is not clearly depicted with a style label, the motion can usually be considered neutral for the transfer algorithm and produces satisfying style transfer results. For example, the *Mixamo* data we used in the generalization test does not own a specific style label. We treat it as neutral motion and feed it to the *Style-ERD* module to generate the motions shown in Fig. 7 of the paper. Alternatively, a style-classification model can be trained to predict the style of the input motion. Therefore, *Style-ERD* can accept the style prediction of the classification model as the input style label for the transfer task.

C. Extra Visualization Results

In this section, we show the results that are omitted in the paper, including style transfer results on heterogeneous motion sequence in Fig. 3, mixture of two styles in Fig. 4, and ablation study results in Fig. 5.

References

[1] 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)*, 39(4):64–1, 2020. 1

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1