Received: 30 April 2021 | Accepted: 6 May 2021

DOI: 10.1002/cav.2003

SPECIAL ISSUE PAPER

WILEY

Learning a deep motion interpolation network for human skeleton animations

Chi Zhou[1,2] | Zhangjiong Lai[2] | Suzhen Wang[2] | Lincheng Li[1,2] | Xiaohan Sun[1] | Yu Ding[1,2]

[1]Zhejiang University, Zhejiang, China

[2]Virtual Human Group, Netease Fuxi AI Lab, Zhejiang, China

Correspondence
Yu Ding, Virtual Human Group, Netease Fuxi AI Lab, Zhejiang, China.
Email: dingyu01@corp.netease.com

Abstract

Motion interpolation technology produces transition motion frames between two discrete movements. It is wildly used in video games, virtual reality and augmented reality. In the fields of computer graphics and animations, our data-driven method generates transition motions of two arbitrary animations without additional control signals. In this work, we propose a novel carefully designed deep learning framework, named deep motion interpolation network (DMIN), to learn human movement habits from a real dataset and then to perform the interpolation function specific for human motions. It is a data-driven approach to capture overall rhythm of two given discrete movements and generate natural in-between motion frames. The sequence-by-sequence architecture allows completing all missing frames within single forward inference, which reduces computation time for interpolation. Experiments on human motion datasets show that our network achieves promising interpolation performance. The ablation study demonstrates the effectiveness of the carefully designed DMIN.[1]

KEYWORDS

image inpainting, motion control, motion interpolation, animation, deep learning

1 | INTRODUCTION

Keyframe interpolation is a vital technology in human motion animation. In video games, manually authoring animation sequences with motion frames is highly time-consuming processes. In this work, we propose a data-driven transition generation method to generate animation from the given motion frames automatically. Common rule-based methods to interpolate sparse keyframes include linear interpolation (Lerp) and spherical linear interpolation (Slerp). Slerp does, in fact, perform great arc interpolation on the quaternion sphere. Quaternion includes four dimensions, that is x, y, z, w. According to Dam et al.,[1] each of the methods results in an interpolation curve defined as follows. Given q_0, q_1 from a quaternion set H, the interpolation $\gamma : H \times H \times [0, 1] \rightarrow H$ must satisfy the constraints:

$$\begin{aligned}\gamma(q_0, q_1, 0) &= q_0,\\ \gamma(q_0, q_1, 1) &= q_1.\end{aligned} \tag{1}$$

[1]The research work was performed during an internship at Virtual Human Group, Netease Fuxi AI Lab.

Traditional rule-based interpolation methods view every joint of skeletons as an independent variable. It does not considerate the influence and interrelation of connected joints. For example, the motion of human's arms is likely to influence the motion of human's hands because these two joints are connected. Slerp interpolates between two keyframes, ignoring information of other frames, such as the overall rhythm of the motions. To address this problem, we learn the patterns of motions from real motion capture dataset and propose a data-driven method with a supervised deep neural network.

Concretely, we propose a neural network architecture mainly for human motion interpolation tasks (see Figure 1 for detail). First, we convert two human motions into a motion image with mask. Second, we use deep motion interpolation network (DMIN) to complete the image. Our network relies on the generative adversarial network[2] consisting of a generator and a discriminator. The generator has a hourglass architecture that could capture information at different scales. Inspired by this, we propose a hourglass structured network that extracts multiscale information of a motion to improve the effect of inpainting and we propose BPE loss to reduce jitter in the boundary. In our quaternion-based mocap datasets, experiments show that our hourglass GAN with AdaIN and swish architecture achieves promising results. We represent joint rotation as quaternion and use forward kinematics to calculate the position of all joints. However, the forward kinematics will result in error accumulation of the end-effectors so that left and right foot will be jittery. Therefore, the resulting animated character may not accurately follow the target end-effector positions, which degrades the quality of motions. Thus for leg joints, we use DMIN to complete global positions and use inverse kinematics plugin in unity to fine-tune the interpolated motions.

Our contributions are summarized as follows:

- Viewing the motion interpolation task as the image inpainting task, We propose an hourglass-like generator structure with AdaIN blocks to capture multiple scale information and reduce mutations in interpolation transition regions.
- Our motion interpolation architecture is sequence-by-sequence and could complete all missing frames within single forward inference, instead of frame-by-frame iteration, which reduces computation time for interpolation.
- For lower body joints, we use DMIN to complete global positions and use inverse kinematics to fine-tune the interpolated motions to force the character follow the target positions.
- Our subjective evaluation shows the interpolated motions are more natural compared with Slerp results.

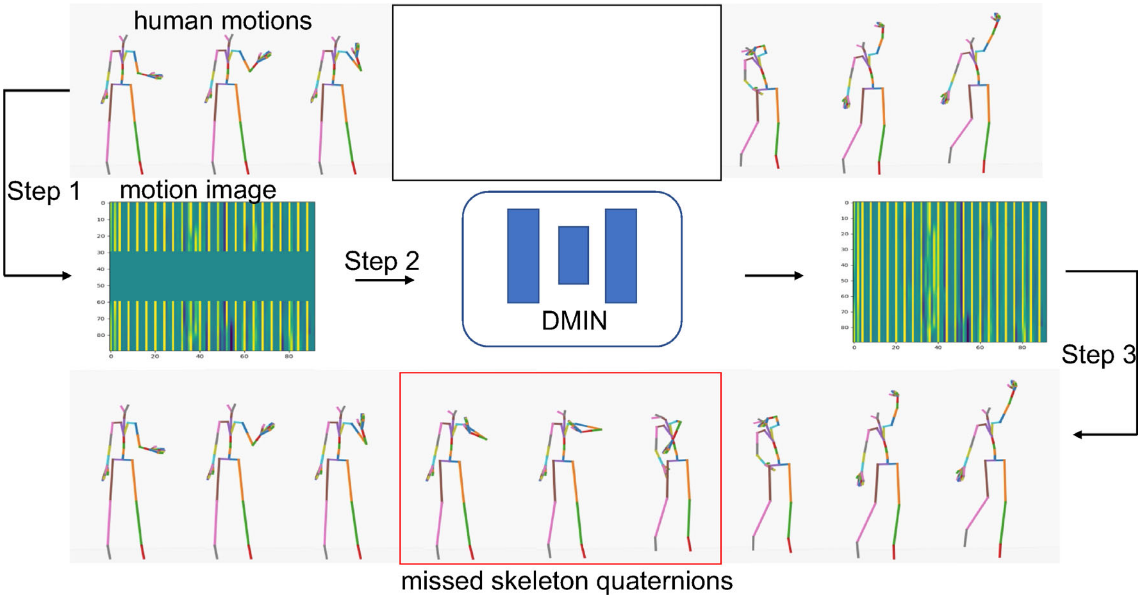


FIGURE 1 Illustration of our DMIN. Step 1: Convert human motions into an image. Step 2: Complete the motion image using DMIN. Step 3: Convert completed motion image back to human motions. Thus we get the missed skeleton quaternions

2 | RELATED WORK

There have been multiple studies about motion control, motion prediction, motion interpolation, and image inpainting.

2.1 | Motion control

Motion control means that temporally dense external signals, usually user-defined, are used to drive the generation of an animation. Motion graphs[3,4] aim at building a graph of motions based on short clips, which was later expanded to meshes.[5] Motion graphs allow one to produce motions by traversing nodes and edges that map to character states or motions segments from a dataset. Motion graphs requires preloading of motion clips in memory to meet real-time needs.

Many deep learning techniques train a network offline to mitigate the requirement. Holden et al.[6] use root trajectories or end-effectors' positions as control signals and train feed-forward convolutional neural networks to build a motion editing framework. Henter et al.[7] introduce a new class of probabilistic, generative, and controllable motion-data models based on normalizing flows. These data-driven methods help reduce artifacts and show impressive results, especially for human locomotion. Motion control techniques require temporally dense external signals to constrain result animation. With this requirement, it cannot be applied to motion interpolation directly.

2.2 | Motion prediction

Human motion prediction aims at generating future frames of human motion based on an observed sequence of skeletons. It is often addressed by training recurrent neural networks (RNNs). Fragkiadaki et al.[8] propose an encoder-recurrent-decoder model for recognition and prediction of human body pose. Pavllo et al.[9] propose a quaternion-based RNN model to predict rotations with absolute position loss. Their loss function takes joint rotations as input and runs forward kinematics to compute the position of each joint to ease error accumulation. Hernandez et al.[10] propose a generative adversarial network (GAN) to forecast 3D human motion given a sequence of past 3D skeleton poses represented as position coordinates. Given past context, these carefully designed neural network show impressive results on long-term prediction of human motion. The difference between motion prediction and interpolation is that motion prediction lacks of spatial constraints of last frame.

2.3 | Motion interpolation

Motion interpolation arises in many situations such as keyframe animation. There are different traditional interpolation methods based on parameterizations. In our work, we represent rotation as quaternion other than Euler angles because Euler angle suffers from discontinuities and singularities and gimbal lock according to Dam et al..[1] Quaternion expresses rotation as a rotation angle about a rotation axis. This is a more natural way to perceive rotation than Euler angles. Commonly we use Slerp between two keyframes. Given $q_0, q_1 \in H$ and $h \in [0, 1]$, we can formulate Slerp as following functions.

$$\begin{aligned} cos(\Omega) &= q_0 \cdot q_1, \\ Slerp(q_0, q_1, h) &= \frac{q_0 sin((1-h)\Omega) + q_1 sin(h\Omega)}{sin(\Omega)}. \end{aligned} \tag{2}$$

For human skeletons, Slerp is commonly used to construct smooth animation curves. This traditional method views joint as independent variant. However, we take the influence and interrelation between connected joints into consideration. For example, the motion of human's arms is likely to influence the motion of human's hands. We use a deep neural network to complete the missing frames with image inpainting idea. With large receptive field of the network, the rotations of every joint are decided by whole joints of preceding and succeeding motions.

Harvey et al.[11] propose a recurrent transition networks based architecture to generate transition animation from past frames along with a target keyframe, which they called ERD-quaternion velocity network (ERD-QV). When generating multiple transitions from several keyframes, the model is simply applied in series, using its last generated frames as past context for the next sequence. They apply encoder–recurrent–decoder (ERD) networks and propose time-to-arrival embeddings to allow robustness to variable lengths of in-betweening, especially for long-term motions and achieve promising results for locomotion.

ERD-QV runs an autoregressive manner and predict transition motions frame-by-frame with a iteration method, which require computational resources with increasing of lengths of in-betweening. Instead, we format the in-betweening as an image inpainting problem and "inpainting" multiple missing frames within a single forward inference simultaneously in a sequence-by-sequence manner, which will reduce computation time for interpolation.

2.4 | Image inpainting

Motion interpolation can be viewed as completing missing motion frames within a sequence, which is similar to image inpainting. Image inpainting aims at completing missing area of an image. Iizuka et al.[12] firstly propose a GAN based model with global and local discriminator. Yu et al.[13] propose a learnable dynamic feature selection mechanism. Hong et al.[14] propose a u-net structure with fusion blocks. The fusion block is a learnable block implementing pixel level fusion in the transition region, which lead to smooth transition in mask boundary. In this work, image inpainting is employed to interpolate rotations of human motion.

3 | METHODS

The task studied in this work is to perform human motion interpolation with neural network. Given two skeleton motions M_1 with size $N_1 \times K$, M_2 with size $N_3 \times K$, where N_1, N_3 represents number of motions' frames, K represents quaternions' dimensions of all joints. We estimate the root position and the pose with a same inpainting architecture for generating transition frames of animation end-to-end. That is to say, we have an image of size $(N_1 + N_2 + N_3) \times (K + 3)$, where number 3 represents the normalized 3D root position. Middle N_2 rows are missing. We interpolate N_2 frames between these motions. The main module of DMIN is a generator–discriminator architecture with hourglass structure, as shown in Figure 2.

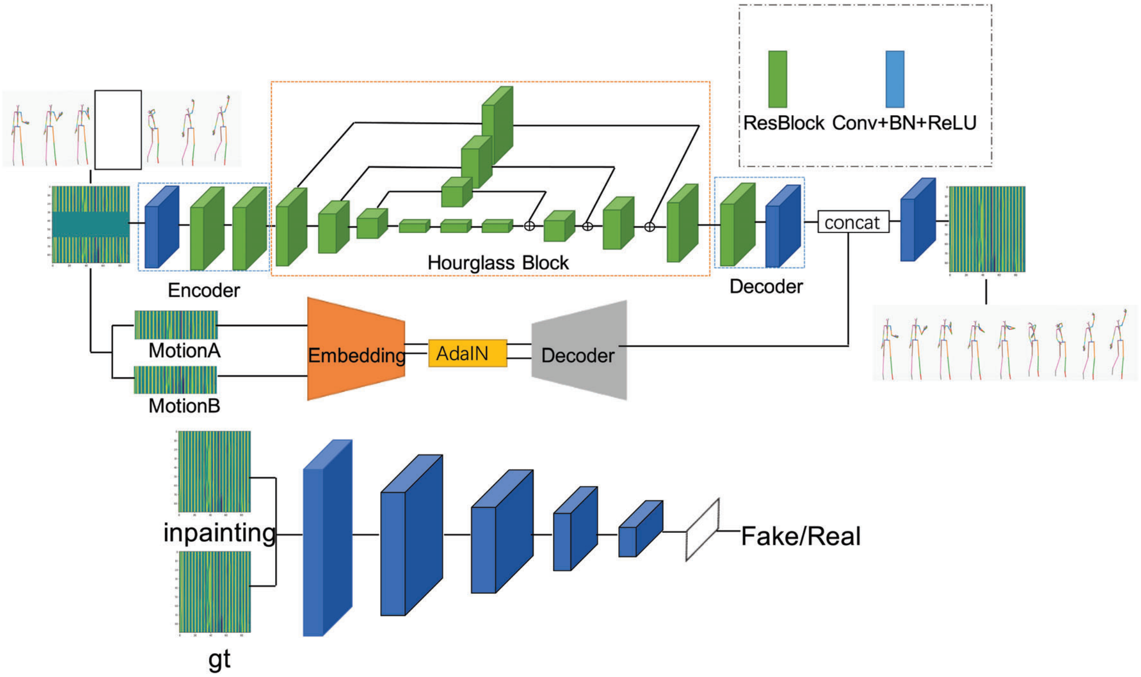


FIGURE 2 Overview of our deep motion interpolation network (DMIN). It consists of a generator and a discriminator. Generator is an encoder-transformer-decoder structure with a hourglass block. We also apply a second branch to extract the common features of two motions and an AdaIN layer to combine these features to guide the inpainting task. The discriminator is a fully convolution network named PatchGAN trained to distinguish real motions from completed ones

3.1 | Network architectures

We first represent motions as motion images so as to utilize the idea of image inpainting method. Figure 1 shows that the motion's image is similar to the texture image. The main components of DMIN are as follows.

Generator. The generator is a multiscale hourglass network. Newell et al.[15] used a hourglass network for human pose estimation. To better extract multiscale information, we design an hourglass network that has multiscale feature maps range from coarse to fine in Figure 2.

Hourglass. The proposed hourglass network takes multiscale information into account. Features are processed across all scales and consolidated to best capture the various spatial relationships. Similar to u-net[16] that pools down to a low resolution, then unsamples and combine features across multiple resolutions using concat operation, the hourglass network does downsample and upsample operation and combine multiple scales features with residual structure. While local evidence is essential for inpainting features like motion of hands, feet, and the head, global information is also required to determine the global speed and gesture of the full body. The relationship of connected joints could be best recognized at different scales in the image. We put the hourglass into an encoder–decoder structure in Figure 2. The encoder and decoder are composed of ResBlock and convolution operations. The hourglass network also makes use of residual blocks. We introduce a motion encoder before motion image being fed to the hourglass block. And the decoder transforms the output of hourglass network back to the completed image.

AdaIN. The interpolation motions depends on the former motion M_1 and the latter motion M_2. Not only the body posture of M_1 and M_2, but also their movement speed influences the result of interpolation. Huang et al.[17] propose adaptive instance normalization (AdaIN) layer to achieve arbitrary style transfer of images. We use AdaIN layer too that aligns the mean and variance of the embedding features of those two motions. We use three Conv-ReLU layers as the embedding and Conv-ReLU-nearest neighbor upsampling layers as the decoder in AdaIN branch. We concatenate AdaIN's output with hourglass network's output to produce final inpainting result, where the embedding of former and latter motion could guide the training of inpainting network. AdaIN receives a content embedding E_1 of M_1 and E_2 of M_2, and simply aligns the channelwise mean and variance of E_1 to match those of E_2.

$$AdaIN(E_1, E_2) = \sigma(E_2)\left(\frac{E_1 - \mu(E_1)}{\sigma(E_1)}\right) + \mu(E_2). \tag{3}$$

Swish. Ramachandran et al.[18] propose Swish activation function. They discovered activation function, $f(x) = x\dot{s}igmoid(\beta x)$, which they named Swish, tend to work better than ReLU on deeper models across a number of challenging datasets. We add Swish into every residual block in hourglass network.

Discriminator. We design a global discriminator to determine if it is an inpainting image or an ground truth image. GAN manages to truly model the data distribution of ground truth motions to generate natural-looking results. The discriminator is a fully convolution network. It includes stacked convolutions, batch normalization, and leaky ReLU layers in Figure 2. PatchGAN only penalizes structure at the scale of patches. This discriminator tries to classify if each $N \times N$ patch in an image is real or fake.

3.2 | Loss functions

We define two main losses including reconstruction loss and adversarial loss between inpainting result and ground truth image. Reconstruction loss forces the generator to preserve the information of original input. Adversarial loss forces the interpolated motions more natural. We define Q as original motion image, Q_{out} as inpainting network's output. M as mask image which is a binary image that indicates the image completion mask (1 for a pixel to be completed).

Mask loss. Mask loss computes the L1 distance between inpainting motion and the ground truth in the mask area.

$$L_{mask} = ||Q \odot M - Q_{out} \odot M||_1. \tag{4}$$

This loss improves the similarity of output image and the ground truth.

Overlap loss. Besides L1 loss in mask area, we define overlap loss of N frames near the mask area. Let $M_{overlap}$ be the 0-1 indicator matrix indicating the overlap area.

$$L_{overlap} = ||Q \odot M_{overlap} - Q_{out} \odot M_{overlap}||_1. \tag{5}$$

This loss forces the network to produce a natural transition of inpainting motion.

Smooth loss. To smooth the inpainting motions, we propose motion smooth loss. The smooth loss is sum of L1 distance between $t+1$ frame's quaternions q_{t+1} and t frame's quaternions q_t in mask area.

$$L_{smooth} = \sum_t \|q_{t+1} - q_t\|_1. \tag{6}$$

BPE loss. In inpainting task, Hong et al.[14] produced a novel evaluation metric named boundary pixels error (BPE) loss. They propose that pixels in unknown region that near the boundary have very small variance while these pixels play the most important role in structure and texture transition. In our human motion datasets, small variance near boundary will seriously affect the visual performance. BPE only considers pixels error near the boundary. For boundary area b, which is n pixels narrow band adjacent to the boundary of unknown region, BPE is defined as follows.

$$L_{BPE} = \|b \odot (Q - Q_{out})\|_1. \tag{7}$$

SSIM loss. Wang et al.[19] propose structural similarity as an image quality assessment method. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. Zhao et al.[20] compare SSIM loss with L1, L2 loss in image reconstruction task. SSIM loss compares local region of target pixel between reconstructed and original images. L1 loss could not suppress small noise in the inpainting result, which lead to jitter of human motion. We compute SSIM loss between inpainting result Q_{out} and ground truth Q. Ablation study in Table 1 shows the effect of SSIM loss.

$$L_{SSIM} = 1 - SSIM(Q, Qout). \tag{8}$$

Global rotation loss. We represent human pose at every frame as a quaternion vector. To preserve 3d position's information, we propose global rotation loss. Original quaternions are the local rotations of the child joints to their parent joints. Using forward kinematic algorithm, we could compute global rotations of each joint. Here our loss function takes as input local joint rotations and runs forward kinematics to compute the global rotations of each joint. We can then compute the L1 loss between each predicted global joint rotations Q_{gt_global} and the reference pose Q_{global}.

$$L_{grot} = \|Q_{gt\_global} - Q_{global}\|_1. \tag{9}$$

Gan loss. We use a generator–discriminator architecture. The generator completes the missing frames of motions and output the completed motion image. The discriminator takes inpainting motion and ground truth motion as inputs and determine if they are real or fake. We refer generator as G, discriminator as D. The gan loss are computed as follows.

$$L_D = \mathbb{E}_{Q_{out}}[log(1 - D(G(Q \odot M)))] + \mathbb{E}_Q[log(D(Q))], \tag{10}$$

$$L_G = \mathbb{E}_{Q_{out}}[log(D(G(Q \odot M)))]. \tag{11}$$

Final loss. The final loss L is a weighted sum of previous losses.

$$L = \lambda_{mask}L_{mask} + \lambda_{overlap}L_{overlap} + \lambda_{smooth}L_{smooth} + \lambda_{PBE}L_{PBE} + \lambda_{SSIM}L_{SSIM} + \lambda_{grot}L_{grot} + \lambda_D L_D + \lambda_G L_G, \tag{12}$$

where λ denotes the weight of losses, respectively.

We can define minimax problem:

$$G^* = \underset{G}{argmin}\max_D L. \tag{13}$$

3.3 | Adaptive inpainting

Traditional motion interpolation like Slerp between two keyframes must identify the number of interpolation frames. The number of frames is manually adjusted by experts after visualization. The interpolation frame number is related to

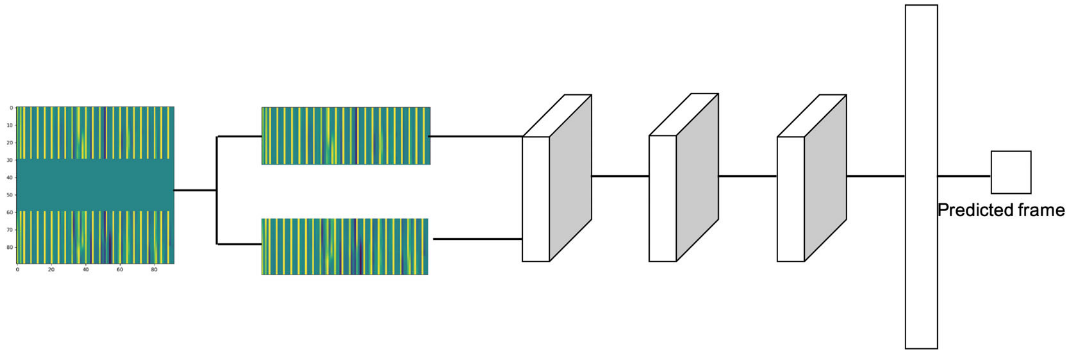


FIGURE 3 Adaptive inpainting network (AIN). It takes motions as input and predicts the number of interpolation frames

the speed of former and latter motions and distance between manifolds of two motions. To obtain the finest interpolation frame number, we propose a simple convolution neural network to predict the interpolation number of frames given two motions. Benefit from the fully convolutional generator and inpainting mechanism, we can change the inpainting area along the frame dimension.

To the best of our knowledge, we are the first to propose adaptive interpolation method with neural network. Figure 3 show the structure of adaptive inpainting network (AIN). AIN has three conv-bn-relu blocks with a fully connected layer.

3.4 | Fine-tuning with inverse kinematics

Previously we represent motions as joints' rotations and using DMIN to complete the missing motions. Then we adopt the Biped IK in Unity plugin FINAL IK to fine-tune our inpainting motions. It uses ccd algorithm to solve the ik problem.

We complete rotations and positions of all joints by DMIN, respectively. For upper body joints, we assign predicted rotations as final rotations. For lower body joints, we extract predicted foot positions to solve the IK problem and fine-tune the rotations. Benefit from the image inpainting model, we could change the inpainting area by adjusting the mask area. Thus we could complete two legs with order and improve the quality of final motions.

4 | EXPERIMENTS

4.1 | Datasets and implementation details

All experiments in this work are implemented using Pytorch on Nvidia RTX 2080Ti. The weights of losses $\lambda_{mask}, \lambda_{overlap}, \lambda_{smooth}, \lambda_{PBE}, \lambda_{SSIM}, \lambda_{grot}, \lambda_D, \lambda_G$ are set to 6, 1, 6, 1, 6, 6, $1e-4$, 1 respectively. The generator is trained by the Adam optimizer with weight decay regularization. The learning rate of is initialized to $5e-4$, which drops with a rate of 0.5 every 10k batches. The learning rate of discriminator is set to $2e-5$. The mini-batch size is set to 32 during training. We train the model for 100k batches with early stopping. The parameters of the network is initialized using uniform initialization method.

Our human mocap datasets are used to evaluate deep motion interpolation network. One is motion capture dataset for gesture animation, another is for gait animation. Gesture dataset contains 614 human motions with the number of frames range from 50 to 180 and Gait dataset contains 13 human motions with 3k to 7k frames. We split dataset into 90% train data and 10% test data. Each motion is represented by quaternion rotations of skeleton. Here we use human skeleton with 23 joints. To represent valid rotations, We perform postprocessing to normalize interpolated quaternion to a unit vector. For the motions whose frame number less than 96, we pad the motion before first frame and after last frame. For the motion whose frame number greater than 96, we simply randomly extract 96 consecutive frames. For quantitative comparison, we fix the number of interpolation frames to 30. We could also use AIN to predict the number of frames.

We first quantitatively and qualitatively evaluate our model on our datasets. Then we conduct ablation studies to demonstrate the effect of each component in our DMIN. The comparison between Slerp and our DMIN shows that ours generate more natural motions. We use subjective evaluation to evaluate visual quality of inpainting results. We also apply previous loss function to measure the distance of generated motions and their targets.

4.2 Quantitative results

We compare our inpainting network with Slerp. Table 1 summarizes Slerp and ours for quantitative comparison. For the situation when we interpolation between two truth motions, we find that Slerp has lower l1 loss. However, this l1 loss could not directly reflect the visual quality. Despite lower loss, Slerp generates motion sequences with same angular velocity. It only considers two keyframes, does not take the concrete motions into account, thus the motions are less natural. We can see the result of subjective evaluation in quantitative results. We also compare Slerp with our model on LaFAN1[11] datasets. Results are showed in the supplementary video.

Ablation study. We conduct an ablation study to verify the impact of each component of our proposed network and the influence of different loss function together with the GAN loss. As shown in Table 1, we calculate losses on test dataset without Swish activation, AdaIN layers and change our hourglass structure to u-net structure. These losses could be reduced with proposed modules. We also evaluate the influence on test dataset without proposed loss function such as GAN loss, SSIM loss, BPE loss, smooth loss, and global rotation loss. Quantitative results show that Slerp generate more smooth transition but the qualitative results and video show that our method generate more natural results, which reflect the limitations of these smooth losses to some extent.

Speed performance. We conduct speed performance comparison with ERD-QV in Table 2. The in-betweening length is set to 15, 30, 60 frames and we record CPU model inference time. Our model generates transition frames within a single forward inference, thus the computation time is frame-independent and could reduce computation time for interpolation.

4.3 Qualitative results

To verify the effectiveness of the proposed method, we conduct a subjective evaluation. A user study online was carried out, by comparing four conditions including 1. Slerp, 2. DMIN (inpainting rotations by DMIN), 3. DMIN+IK (inpainting rotations and positions with inverse kinematics fine-tuning), 4. ground truth. We sample 10 examples from test data of gait dataset and use Slerp, DMIN and DMIN+IK interpolation to generate rotations. For comparison of interpolation result of gesture, in test data of gesture dataset, we randomly sample 12 motions and use Slerp and DMIN to interpolate between six preceding and six succeeding motions to see the effectiveness of interpolation of different motions. A virtual character is driven with the outputs animations and we record them in videos. Finally, 10 videos of gait animation and six videos of gesture animation were prepared. Each video is demonstrated on one webpage. Thirty one participants were invited in the user study. After watching the video, they were instructed to use 5-point Likert scale to rate the naturalness of the

TABLE 1 Testing losses for deep motion interpolation network with different modules, different losses and Slerp method

Method	Origin	W/o Swish	W/o AdaIN	u-net	W/o GAN	W/o SSIM	W/o BPE	W/o smooth	W/o grot	Slerp
Mask	**0.022**	0.022	0.022	0.025	0.022	0.023	0.022	0.023	0.022	0.019
Overlap	0.0045	0.0050	0.0052	**0.0013**	0.0075	0.0069	0.0081	0.0092	0.0060	/
Smooth	0.0033	0.0034	0.0033	0.0036	**0.0031**	0.0032	0.0032	/	0.0033	0.0018
BPE	**0.0080**	0.0088	0.0090	0.011	0.012	0.013	/	0.011	0.0094	0.0037
SSIM	0.0061	0.0055	0.0056	0.0064	0.0056	/	0.0053	**0.0052**	0.0056	0.0046
Grot	**0.0093**	0.0099	0.0095	0.0097	0.012	0.030	0.029	0.027	/	0.0067

TABLE 2 Model inference time comparison

Method	15 frames	30 frames	60 frames	CPU info
ERD-QV	0.30 s	0.31 s	0.31 s	E5-1650 @ 3.20 GHz
Ours	**0.17 s**	**0.17 s**	**0.17 s**	I7-8700K @ 3.70 GHz

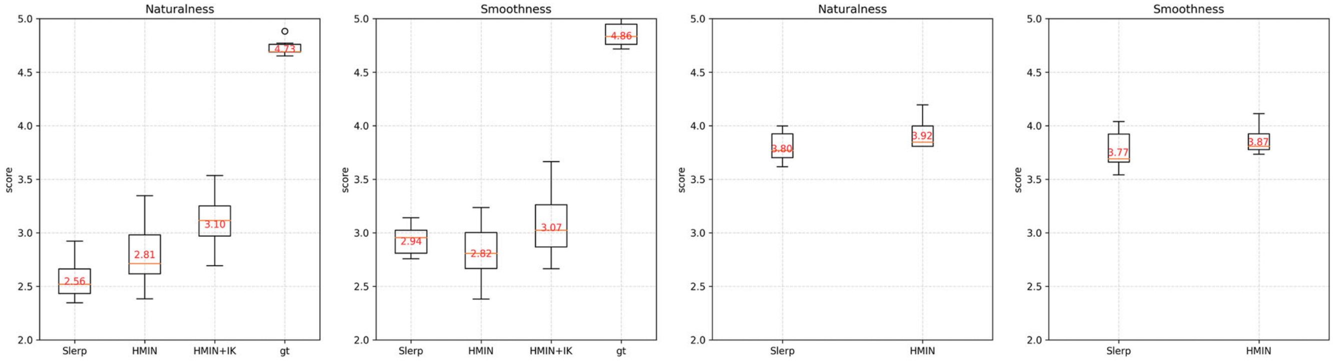


FIGURE 4 Rated values of different methods in the user study. First row shows the results of Slerp, DMIN, DMIN+IK, and ground truth in gait dataset. Second row shows the results of Slerp and DMIN in gesture dataset

TABLE 3 Subjective evaluation result

	DMIN+IK	
	Naturalness	**Smoothness**
Slerp	27.7025**	1.3662
DMIN	4.9094*	3.2961
Ground truth	349.4553**	279.1573**

Note: The table shows results of pair ANOVA. The values in the table represent the F-score.
*$p < .05$, **$p < .0001$.

animation and smoothness of the animation. The term of naturalness refers to the quality of visual animations (whether the steering is reasonable, whether the footsteps are sliding). The smoothness means whether the transition is smooth and whether the rhythm of generated motion is consistent with the given motions.

Figure 4 shows the rated results. Table 3 shows the results of ANOVA. The test shows that our DMIN+IK outperforms Slerp, DMIN in terms of naturalness and smoothness. This validates the proposed method. For gesture of upper body especially the hands, we randomly extract two motion sequences from arbitrary motion A and motion B and we use Slerp or DMIN to interpolate these sequences. These motions have completely different pose and speed thus the Slerp could not handle these problems, leading to the unnaturalness of the motions. Subjective evaluation in Figure 4 shows that DMIN generates more natural and smooth motions compared with Slerp.

As shown in Figure 5, we give an example interpolated by Slerp and DMIN. This example shows that the inpainting motions are more natural. The Slerp generates motions with same angular velocity. The joints' rotation speed of the Slerp motion remains unchanged, which makes the motion look stiff. Videos demo can be found in supplementary materials.

To give a more clear comparison between our inpainting method and Slerp, as shown in Figure 6, we plot rotation curve of left arm joints in interpolation motion. The rotation curve show that the Slerp does not considerate on the tendency or speed of former and latter motions.

4.4 | Adaptive inpainting results

To predict the number of interpolation frames, we mask the ground truth motions and divide the frames into 5 categories, including 5, 25, 45, 65, 85 frames and we view AIN as a classifier. We use cross-entropy loss and train AIN on gait animation by the Adam optimizer with the learning rate 1e−4. Final classification accuracy in test data is 62.5%.

4.5 | IK fine-tuning

For gait dataset. We use ccd algorithm to fine-tune the inpainting local rotations and the subjective evaluation shows the improvement with IK. Furthermore, in Figure 7, we display an example with IK fine-tuning. The result shows that IK fine-tuning could reduce drift of foot and improve the quality of interpolation. Before the fine-tuning, the movements of foots are slightly noisy. After the fine-tuning, the foot follow the ground truth well, and the jitter is reduced.

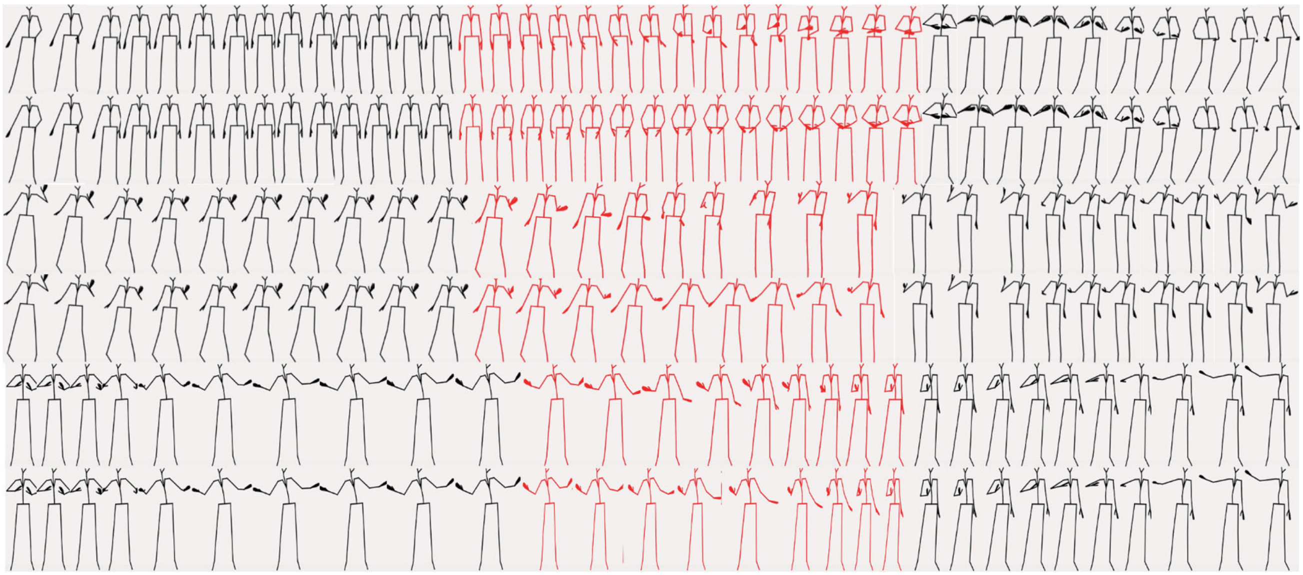


FIGURE 5 Three examples of interpolation. The sequences in row 1, 3, 5 correspond to Slerp and the sequences in row 2, 4, 6 correspond to DMIN. The dark sequences correspond to ground truth. The red sequences correspond to interpolated motions

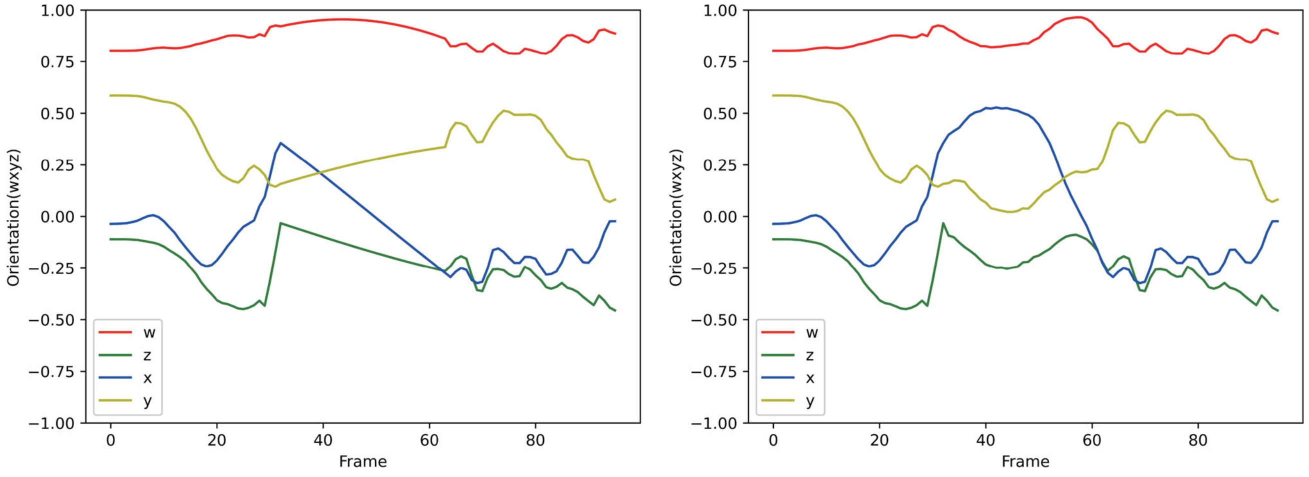


FIGURE 6 Rotation curve of motion interpolated by Slerp and DMIN methods, including four axis, x, y, z, w

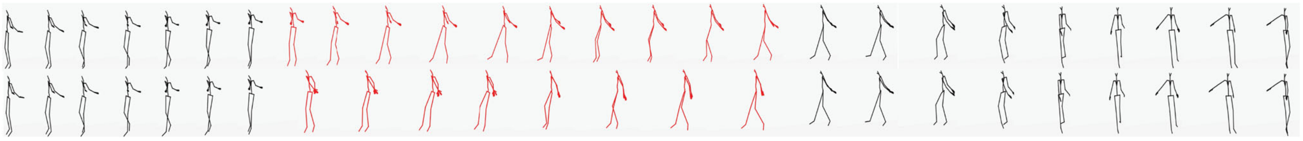


FIGURE 7 Interpolation results of motion with and without inverse kinematics fine-tuning with DMIN methods. Motions on first row are ik fine-tuning results and motions on second row are raw inpainting results. The foot follow the ground truth well, and the jitter is reduced after ik fine-tuning

5 | CONCLUSIONS

In this work, we propose DMIN, a novel deep learning architecture for human motion interpolation. We view joints' rotations and positions of human motion as motion images and complete missing area by DMIN. The unknown motion frames can be generated in a single model inference rather than frame-by-frame iteration, which reduces computation time. The subjective evaluation shows that the motions generated by DMIN are more natural compared with Slerp. Slerp only considers two keyframes. It does not considerate on the influence of global motion and global speed, ignoring the overall rhythm of the motions and it can only be applied to FK and cannot solve the problem of sliding of foot. Our method could capture feature of global speed and produce more natural transition given two motions. Previous human motion modeling tasks like motion prediction shows that the forward kinematics will result in the jitter of foot joints. We use inverse kinematics to fine-tune the inpainting quaternions and achieve better results.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

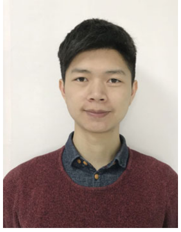
Chi Zhou https://orcid.org/0000-0001-7358-2062

REFERENCES

1. Dam Erik B, Koch Martin, Lillholm Martin. Quaternions, interpolation and animation. Vol 2. København: Datalogisk Institut, Københavns Universitet; 1998.
2. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. Advances in neural information processing systems. Cambridge: MIT Press; 2014. p. 2672–80.
3. Kovar L, Gleicher M, Pighin F. Motion graphs. ACM SIGGRAPH 2008 classes. New York, NY, United States: Association for Computing Machinery; 2008. p. 1–10.
4. Heck R, Gleicher M. Parametric motion graphs. Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games. Seattle, Washington; 2007. p. 129–36.
5. Casas D, Tejera M, Guillemaut JY, Hilton A. 4D parametric motion graphs for interactive animation. Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. Costa Mesa, CA; 2012. p. 103–10.
6. Holden D, Saito J, Komura T. A deep learning framework for character motion synthesis and editing. ACM Trans Graph. 2016;35(4):1–11.
7. Henter GE, Alexanderson S, Beskow J. Moglow: probabilistic and controllable motion synthesis using normalising flows. ACM Trans Graph. 2020;39(6):1–14.
8. Fragkiadaki K, Levine S, Felsen P, Malik J. Recurrent network models for human dynamics. Proceedings of the IEEE International Conference on Computer Vision. Santiago, Chile; 2015. p. 4346–54.
9. Pavllo D, Grangier D, Auli M. Quaternet: a quaternion-based recurrent model for human motion. arXiv preprint arXiv:180506485; 2018.
10. Hernandez A, Gall J, Moreno-Noguer F. Human motion prediction via spatio-temporal inpainting. Proceedings of the IEEE International Conference on Computer Vision. Seoul, Korea; 2019. p. 7134–43.
11. Harvey FG, Yurick M, Nowrouzezahrai D, Pal C. Robust motion in-betweening. ACM Trans Graph. 2020;39(4):60–1.
12. Iizuka S, Simo-Serra E, Ishikawa H. Globally and locally consistent image completion. ACM Trans Graph. 2017;36(4):1–14.
13. Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS. Free-form image inpainting with gated convolution. Proceedings of the IEEE International Conference on Computer Vision. Seoul, Korea; 2019. p. 4471–80.
14. Hong X, Xiong P, Ji R, Fan H. Deep fusion network for image completion. Proceedings of the 27th ACM International Conference on Multimedia. Nice, France; 2019. p. 2033–42.
15. Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation. Proceedings of the European Conference on Computer Vision. New York, NY: Springer; 2016. p. 483–99.
16. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention. New York, NY: Springer; 2015. p. 234–41.
17. Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalization. Proceedings of the IEEE International Conference on Computer Vision. Venice, Italy; 2017. p. 1501–10.
18. Ramachandran P, Zoph B, Le QV. Searching for activation functions. arXiv preprint arXiv:171005941; 2017.
19. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process. 2004;13(4):600–12.
20. Zhao H, Gallo O, Frosio I, Kautz J. Loss functions for neural networks for image processing. arXiv preprint arXiv:151108861; 2015.

AUTHOR BIOGRAPHIES

Chi Zhou received the master's degree in mathematics from Zhejiang University, Zhejiang, China, in 2021. His research interests include computer graphics and computer vision.



Zhangjiong Lai received the MS degree in 2019 from the Department of Computer Science at Zhejiang University. He is now working in NetEase, Inc, Hangzhou, China. His research interests include human motion analysis and synthesis.

Suzhen Wang is currently the assistant researcher in Netease Fuxi AI Lab. He has received his master degree in 2019 from Zhejinag University. His research interests include deep learning and its applications on multimodal learning and talking face generation. He has published several papers in AAAI and IJCAI.

Lincheng Li received the B.S. and Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2011 and 2017 respectively. He is currently a researcher in Netease Fuxi AI Lab, Hangzhou, China. His research interests include computer vision, pattern recognition, and image and video processing.

Xiaohan Sun received the master's degree in mathematics from Zhejiang University in 2020. His research interests include computer graphics and computer vision.

Yu Ding is currently an artificial intelligence expert at Netease Fuxi AI Lab, Hangzhou, China. His research interests include deep learning, image and video processing, animation analysis and generation, talking-head generation, multimodal computing, affective computing, nonverbal communication (face, gaze, and gesture), and embodied conversational agent. He received Ph.D. degree in Computer Science (2014) at Telecom Paristech in Paris (France).

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Zhou C, Lai Z, Wang S, Li L, Sun X, Ding Y. Learning a deep motion interpolation network for human skeleton animations. Comput Anim Virtual Worlds. 2021;e2003. https://doi.org/10.1002/cav.2003