Two-Person Interaction Augmentation with Skeleton Priors

Supplementary Material

1. More Details on Dataset

One instance of the nine motions (Judo, Face-to-back, Turn-around, Hold-body, Around-the-back, Back-flip, Big-ben, Noser and Chandelle) was captured from different subjects by different systems. Therefore, we have two skeletons, with 25 joints and 24 bones (Judo, Face-to-back, Turn-around and Hold-body), and 17 joints and 16 bones (Around-the-back, Back-flip, Big-ben, Noser and Chandelle), shown in Fig. 1. The motions in D1 and D2 are shown in Fig. 2 and Fig. 3.

For each captured motion, we vary bones with scales within [0.75, 1.25] with a 0.05 spacing, where the original skeleton is used as the template skeleton and labeled as scale 1. An exhaustive permutation of all possible scaling is impractical. Therefore, we only use full-body uniform scaling and single-bone scaling on the upper-body bones which are heavily involved in interactions. We manually specify the skeleton variations and use InteractionMesh [3] to generate motions.

InteractionMesh is an optimization framework where the required input is the original motion and the scaled target skeleton. InteractionMesh make a mesh structure by connecting every pair of points between two characters, called interaction mesh. When adapting the motion for a desired scaled skeleton, it minimizes the Laplacian energy, i.e. a deformation energy term of the interaction mesh, to keep the spatial relations as much as possible for every pair of joints. Using InteractionMesh, instead of hiring more actors, allows us to: (1) have exact control over the bone lengths; (2) explore atypical skeleton/body sizes, e.g. left arm longer than right arm. However, the optimization process is sensitive to initialization and weight tuning of the object function. For each skeleton variation, we manually conduct several rounds of optimizations and visually inspect the quality of the generated motion, until it become satisfactory.

Admittedly, compared with the only dataset for interactions [2], the number of interactions in our dataset is smaller (9 vs 16), but our emphasis is the diversity of body sizes. Overall, we have 9 base motions, a total of 967 body variations with 351045 frames, which is larger than [2] in terms of the number of sequences and frames.

2. Additional Results and Details

2.1. Detailed experiments

The full comparison results of different methods for both retargeting and generation are shown in Tab. 1-Tab. 9.

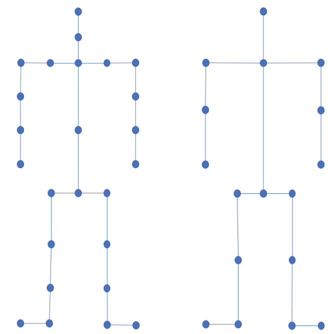

Figure 1. Two skeletons in our dataset. Left: 25 joints, Right: 17 joints

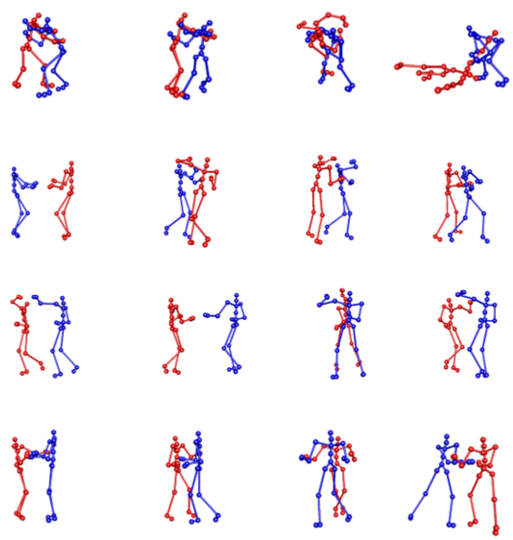

Figure 2. The base motion in D1(M1-M4). From top to bottom: Judo, Face-to-back, Turn-around and Hold-body.

2.2. Skeletal Visualization vs Body Visualization.

Skeletal visualization is widely adopted in existing research (e.g. character animation, motion prediction, activity recognition, etc.), but we do notice a recent trend of showing body shapes with skeletal motions. Theoretically, it is possible to generate body meshes e.g. via SMPL [6]. However, for our problem, this is not the case because generating/adapting body meshes for varying bone lengths is non-trivial and is itself an entirely different topic. Not only is there no body geometry in the data we used, but the motion contains rich contacts between characters. Therefore, generated body meshes could easily lead to penetration so

Metric	F-CNNs	F-GCNs	M-CNNs	M-GCNs	Ours
E_r	0.673/2.569	0.834/5.642	0.517/2.771	0.681/3.447	0.298/1.840
	0.821/4.225	1.684/8.018	1.206/4.060	1.372/3.530	0.463/3.571
	1.251/6.031	3.604/6.858	1.363/4.823	1.622/5.166	0.942/4.744
	1.521/6.455	4.002/6.840	1.684/5.690	1.812/6.110	1.130/4.912
E_b	0.093/0.355	0.120/0.484	0.109/0.835	0.136/0.650	0.072/0.270
	0.127/1.097	0.135/1.067	0.164/1.014	0.158/1.270	0.102/0.506
	0.234/1.214	0.262/1.42	0.273/1.449	0.278/1.516	0.189/0.763
	0.448/1.368	0.403/1.653	0.428/1.506	0.418/1.834	0.305/1.053
JPD	4.078	4.358	6.654	6.877	3.008
	4.938	4.877	6.821	7.239	4.248
	6.674	7.034	8.345	8.003	4.443
	7.894	8.234	8.861	8.642	4.754
FID	3.574/4.928	6.784/14.304	5.421/11.483	3.136/8.09	2.134/3.734
	4.841/8.103	7.541/24.021	7.412/11.838	4.158/9.046	3.824/4.122
	6.854/7.112	7.984/25.080	8.025/15.867	4.278/10.846	4.033/4.109
	7.931/8.761	12.841/26.721	9.541/16.207	6.418/10.654	4.214/4.524
E_b	0.254/0.350	0.365/0.569	0.315/0.549	0.228/0.518	0.176/0.184
	0.621/0.925	0.421/0.825	0.512/1.480	0.862/0.926	0.285/0.423
	0.687/1.763	1.654/2.276	1.862/2.820	1.923/1.947	0.532/0.452
	1.325/3.081	2.284/3.022	2.684/3.424	2.047/5.267	0.737/0.769
JPD	7.542	8.844	6.543	7.832	3.421
	8.043	9.641	7.965	8.239	4.304
	8.821	10.632	8.517	9.632	4.903
	9.852	12.245	9.786	10.985	5.067

Table 1. Comparison on Judo retargeting (top) and generation (bottom). XX/XX are Character A/B results. All results are per joint results. The four rows in each cell are results of Random, Cross-scale, Cross-interaction and Cross-scale-interaction respectively.

manually created meshes are needed. Furthermore, since we sample different bone lengths, manual creation of body geometry for every scaled skeleton would be required, as naive non-uniform scaling on the body mesh designed for a template skeleton would easily cause mesh deformation artefacts or contact breach. Methods such as SMPL might help but with no guarantee, because arbitrary bone scaling easily leads to out-of-distribution skeletons deviating from their training data. We tested SMPL and show one such example in Fig. 4. But this does not mean our motion quality is low. The motion quality can be visually inspected in the video.

2.3. Generation Diversity

Our model contains 3 learned Gaussian distributions and therefore is intrinsically stochastic. We show a Judo motion sampled multiple times (in different colors) using the same skeleton in Fig. 5 (zoom-in for better visualization). While there are motion diversity, we do realize that the motions do not visually show big variations. Note that this is due to the fact that the skeleton is exactly the same for all motions, and more importantly the key interaction features such as contacts need to be maintained in different samples. These contacts implicitly act as constraints for augmentation. However, as shown before, when the bone sizes change, bigger diversities can be seen.

2.4. Generalizability on Reduced Training Samples

Since high-quality interaction motion is hard to capture and data augmentation is not easy, it is highly desirable if augmentation can work on as few training samples as possible. To test this, we choose Face-to-back (M2) and Big-ben (M7) under Cross-scale-interaction, and reduce the training samples to 24, to 12 and 6. More specifically, when using the scale [0.75, 0.85] and [1.15, 1.25] of M2 as the testing data, we randomly select 24, 12 and 6 training samples from the scale [0.95, 1.05] of M3-M4 for training. Similarly, when choosing the scale [0.75, 0.85] and [1.15, 1.25] of M7 as the testing data, we randomly select 24, 12 and 6 training samples from the scale [0.95, 1.05] of M8-M9 for training. Note this is a very challenging setting.

Tab. 10 shows a quantitative comparison. Note metrics have different scales and cross-metric comparison is not meaningful. Unsurprisingly, all metrics become worse when the number of training samples decreases. However, the increase of errors is slow compared with the correspond-

Metric	F-CNNs	F-GCNs	M-CNNs	M-GCNs	Ours
E_r	0.227/0.328	0.445/1.474	0.102/0.232	0.424/2.760	0.058/0.076
	0.234/0.335	0.544/1.554	0.124/0.372	1.732/3.714	0.263/0.425
	0.297/0.451	0.548/1.573	0.156/0.434	1.988/3.876	0.352/0.990
	0.725/1.812	0.641/1.785	0.921/1.932	4.412/5.689	0.630/1.472
E_b	0.009/0.018	0.040/0.107	0.035/0.048	0.120/0.727	0.002/0.006
	0.022/0.053	0.082/0.241	0.056/0.078	0.312/0.739	0.012/0.024
	0.054/0.081	0.103/0.357	0.841/0.959	0.327/0.884	0.089/0.085
	0.245/0.432	0.584/0.633	1.294/2.230	0.972/1.064	0.045/0.217
JPD	0.599	0.517	0.330	0.465	0.104
	0.658	0.505	0.414	0.302	0.241
	0.892	0.703	0.678	0.526	0.625
	1.284	1.724	1.595	0.951	0.845
FID	2.637/7.218	21.238/37.530	4.825/14.917	1.379/2.782	1.134/2.304
	3.118/8.745	20.457/35.483	5.215/14.551	1.751/3.451	1.824/2.904
	3.331/8.286	25.844/38.517	5.466/19.651	2.154/3.756	2.533/3.621
	5.542/9.844	26.723/40.425	7.983/24.842	2.831/4.237	2.814/3.698
E_b	0.032/0.046	0.186/0.200	1.157/1.965	0.125/0.753	0.001/0.009
	0.043/0.062	0.267/0.352	1.305/2.021	0.163/0.847	0.018/0.028
	0.107/0.108	0.349/0.514	2.687/2.984	0.195/0.954	0.053/0.141
	0.342/0.504	0.652/0.721	3.864/4.030	0.642/1.231	0.073/0.213
JPD	1.017	3.916	2.469	0.369	0.101
	1.157	4.148	2.672	0.454	0.645
	1.872	6.216	2.896	0.648	1.004
	1.904	7.385	4.542	2.034	1.317

Table 2. Comparison on Face-to-back retargeting (top) and generation (bottom). XX/XX are on Character A/B. All results are per joint results. The four rows in each cell are results of Random, Cross-scale, Cross-interaction and Cross-scale-interaction respectively.

ing experiments in retargeting and generation part , showing our method has high data efficiency. We show more results in the video.

2.5. Extrapolating to Large Unseen Scales

There is one example of Turn-around on 0.65 and 1.3 in the Fig. 6 , which shows that our model can extrapolate to larger skeletal variations when trained only using data on scales [0.95, 1.05]. More examples can be found in the video.

3. Methodology Details

3.1. ST-GCN Layers

Spatio-temporal Graph Convolutions (ST-GCNs) are widely used in analyzing human motions. Our construction of it is inspired by [5]. Given $q = \{q^0, \dots, q^T\} \in \mathbb{R}^{T\times N\times 3}$, where T is frame number of a motion, N is the number of joints and each joint location is represented by it 3D coordinates, we first construct a graph adjacency matrix $A_n \in \mathbb{R}^{n\times n}$ of the skeleton, indicating the connectivity between joints. The spatial graph convolution of a layer can be represented as:

$$X^t_{i+1} = ReLU(A_n X^t_i W_i + X^t_i U_i) \in \mathbb{R}^{n\times h_i} \tag{1}$$

where the subscript of X is the layer index, t is a frame and h_i is the latent dimension of the layer. W_i and U_i are trainable network weights. Further the temporal convolution can be achieved by using standard 2D convolution on X. In addition, we also add one Batch Normalization layer and a ReLU layer before the 2D convolution and one more Batch Normalization layer and one Dropout layer after the 2D convolution. After combining the spatial and temporal convolution, we have one ST-GCN layer.

3.2. G-GRU Layers

Graph Gated Recurrent Unit Network, or G-GRU is based on standard GRU network [1], which is a Recurrent Neural Network which can model time-series data. Traditional GRU networks do not consider structured data such as graphs. A combination of GRU and Graph Neural Network

Metric	F-CNNs	F-GCNs	M-CNNs	M-GCNs	Ours
E_r	0.454/0.874	0.622/1.121	0.334/1.244	1.735/2.714	0.398/1.754
	0.534/0.925	0.751/1.334	0.453/1.348	1.956/2.819	0.263/2.863
	0.796/2.071	0.728/1.127	0.879/2.941	2.001/2.771	0.352/2.936
	1.296/2.842	1.121/2.254	1.641/3.263	2.942/3.234	0.530/3.326
E_b	0.020/0.038	0.075/0.082	0.363/0.473	0.320/0.773	0.003/0.037
	0.050/0.079	0.098/0.112	0.383/0.536	0.334/0.801	0.028/0.104
	0.112/0.135	0.102/0.133	0.349/0.551	0.503/0.978	0.059/0.119
	0.234/0.524	0.221/0.508	0.641/0.897	0.842/1.235	0.105/0.155
JPD	3.359	2.291	3.765	2.155	2.274
	3.507	3.814	4.202	2.261	2.948
	3.741	4.001	4.268	3.054	3.147
	4.542	6.123	4.964	4.637	3.493
FID	6.806/7.702	9.037/10.487	9.830/11.495	4.407/8.824	3.214/7.932
	7.023/8.112	10.148/12.046	11.049/16.839	4.466/8.847	3.854/9.258
	7.214/8.849	12.645/20.984	12.057/18.213	5.121/9.157	3.708/9.716
	8.678/10.845	13.412/23.582	14.325/21.842	6.051/9.821	3.923/9.803
E_b	0.413/0.454	0.315/0.445	0.940/1.986	0.332/0.776	0.006/0.054
	0.464/0.457	0.486/0.781	1.001/2.068	0.348/0.816	0.025/0.163
	0.516/0.604	0.715/1.033	1.104/2.211	0.401/0.849	0.052/0.202
	0.605/0.840	1.254/1.930	1.529/2.842	0.645/1.731	0.137/0.169
JPD	3.678	4.120	4.399	3.206	2.134
	3.845	4.368	4.501	4.025	2.872
	4.008	4.808	4.815	4.419	3.095
	4.845	5.614	5.325	5.004	3.317

Table 3. Comparison on Turn-around retargeting (top) and generation (bottom). XX/XX are on Character A/B. All results are per joint results. The four rows in each cell are results of Random, Cross-scale, Cross-interaction and Cross-scale-interaction respectively.

can overcome this shortcoming [5]:

$$r^t = \sigma(r_{input}(X^t)) + r_{hidden}(A_s H^t W),$$
$$u^t = \sigma(u_{iput}(X^t)) + u_{hidden}(A_s H^t W),$$
$$c^t = tanh(c_{input}(X^t)) + r^t \odot c_{hidden}(A_s H^t W),$$
$$H^{t+1} = u^t H^t + (1 - u^t) \odot c^t \quad (2)$$

where r_{input}, u_{input}, c_{input}, r_{hidden}, u_{hidden} and c_{hidden} are trainable functions. X^t is the input, H^t is the hidden state at t and W is trainable weights. A_s is the adjacency matrix.

3.3. Network Implementation and Training Details

The network implementation details of ST-GCN1 and G-GRU1, including network layer configurations and architecture, are shown in Tab. 11 and Tab. 12. The network details of ST-GCN2, ST-GCN3 and G-GRU2 are shown are Tab.13 - Tab. 16.

For training, we use a batch size 32 and Adam as the optimizer (learning rate = 0.001) for all our experiments. We train our model on a Nvidia Geforce RTX2080 Ti Graphics Card. The average training time for different models is 243 minutes with training epoch = 50, and the inference time = 0.323s per motion.

4. Alternative Architectures

We use a frame-based Convolution Neural Networks (CNNs) and a frame-based Graph Convolution Networks (GCNs) as the encoders (MLP1, ST-GCN1-3) and decoders (MLP2, G-GRU1-2) in all three VAEs denoted as F-CNNs and F-GCNs. In addition, we also use motion-based CNNs (M-CNNs) and GCNs (M-GCNs). The M-CNNs follow the architecture in [4]. For M-GCNs, we mirror the GCN encoders in ST-GCN1, ST-GCN2 and ST-GCN3, and use them as the decoders. Due to the limited data, we did not choose architectures that require large amounts of data such as Transformers, Flows or Diffusion models.

Totally, there are four baseline networks: Frame-based CNNs (F-CNNs), Frame-based GCNs (F-GCNs), Motion-based CNNs (M-CNNs) and Motion-based GCNs (M-GCNs). The detailed architectures of them are given in Tab. 17, Tab. 18, Tab. 19, and Tab. 20, respectively. Numerically, our current setting significantly outperforms all the other alternatives by as much as 66.99% in E_r, 49.42% in E_b (retargeting), 56.25% in JPD (retargeting), 72.17% in FID, 74.82% in E_b (generation) and 61.32% in JPD (gener-

Metric	F-CNNs	F-GCNs	M-CNNs	M-GCNs	Ours
E_r	0.230/0.258	0.504/1.178	0.077/0.289	0.339/0.860	0.098/0.284
	0.257/0.291	0.604/1.258	0.125/0.291	0.458/0.909	0.163/0.779
	0.304/0.345	0.771/1.541	0.201/0.294	0.517/0.931	0.252/0.982
	0.651/0.837	1.204/1.976	0.604/0.849	0.915/1.677	0.430/1.364
E_b	0.007/0.014	0.049/0.055	0.045/0.059	0.127/0.569	0.003/0.031
	0.015/0.041	0.051/0.059	0.057/0.169	0.199/0.605	0.007/0.151
	0.049/0.064	0.074/0.098	0.099/0.203	0.232/0.771	0.012/0.196
	0.184/0.251	0.142/0.194	0.204/0.531	0.671/0.949	0.025/0.199
JPD	0.617	2.076	0.807	1.685	0.264
	0.824	2.148	0.814	1.694	0.418
	0.835	2.548	1.215	1.805	0.589
	1.542	4.287	2.674	3.004	0.624
FID	3.585/8.344	20.815/24.261	0.721/3.867	0.322/3.513	0.214/2.944
	3.748/9.424	21.784/28.454	0.915/2.245	1.751/2.158	0.854/3.442
	3.982/9.458	22.511/30.368	1.052/2.244	1.981/2.752	0.712/4.584
	4.874/12.828	23.074/29.241	2.452/3.657	3.642/3.777	0.923/5.265
E_b	0.056/0.101	0.504/0.591	0.097/0.519	0.137/0.587	0.006/0.054
	0.077/0.125	0.607/0.614	0.128/0.684	0.252/0.640	0.025/0.149
	0.098/0.130	0.701/0.848	0.157/0.745	0.425/0.672	0.052/0.170
	0.249/0.341	1.204/1.899	0.531/1.112	0.822/1.054	0.067/0.191
JPD	1.105	2.217	1.304	1.707	0.297
	1.235	2.148	1.365	1.735	1.071
	1.442	3.331	1.317	1.844	1.347
	2.140	4.640	2.384	2.896	1.915

Table 4. Comparison on Hold-body retargeting (top) and generation (bottom). XX/XX are on Character A/B. All results are per joint results. The four rows in each cell are results of Random, Cross-scale, Cross-interaction and Cross-scale-interaction respectively.

ation).

References

[1] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 3

[2] Wen Guo, Xiaoyu Bie, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Multi-person extreme motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13053–13064, 2022. 1

[3] Edmond S. L. Ho, Taku Komura, and Chiew-Lan Tai. Spatial relationship preserving character motion adaptation. *ACM Trans. Graph.*, 29(4), 2010. 1

[4] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph.*, 35(4), 2016. 4

[5] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 214–223, 2020. 3, 4

[6] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 1

Metric	F-CNNs	F-GCNs	M-CNNs	M-GCNs	Ours
E_r	1.586/1.614	2.277/4.159	3.120/4.384	1.759/4.045	1.153/2.797
	1.662/4.770	2.282/4.342	3.252/4.844	2.201/4.349	1.851/3.497
	1.976/4.824	2.044/4.157	3.924/4.121	2.471/4.174	2.252/3.882
	2.782/5.452	3.451/5.735	4.812/6.328	3.421/5.418	3.453/4.275
E_b	0.141/0.225	0.088/0.226	0.030/0.063	0.003/0.031	0.001/0.005
	0.156/0.267	0.135/0.287	0.105/0.161	0.010/0.061	0.003/0.029
	0.225/0.305	0.197/0.334	0.210/0.370	0.017/0.120	0.018/0.050
	0.647/0.812	0.729/0.964	0.748/0.792	0.079/0.234	0.055/0.079
JPD	1.844	3.841	3.525	0.307	0.398
	2.217	3.428	3.191	0.941	0.837
	2.618	3.627	3.224	1.715	1.672
	3.542	4.521	4.751	2.642	2.114
FID	0.349/0.746	5.142/8.672	1.824/1.971	0.337/0.584	0.214/1.166
	0.662/0.997	5.771/9.071	2.054/2.642	0.417/0.742	0.854/1.712
	1.087/1.671	6.041/9.817	2.511/2.912	0.661/0.942	1.212/1.650
	1.574/1.942	8.452/10.122	2.981/3.514	0.967/1.345	1.723/2.071
E_b	0.170/0.317	0.565/0.953	0.105/0.251	0.006/0.040	0.006/0.014
	0.204/0.391	0.642/1.074	0.287/0.354	0.038/0.084	0.025/0.037
	0.396/0.504	0.699/1.611	0.487/0.515	0.051/0.191	0.042/0.058
	0.925/1.213	1.077/1.921	1.073/1.258	0.254/0.293	0.047/0.141
JPD	2.485	4.253	3.671	0.345	0.604
	2.671	4.506	3.851	1.414	1.157
	3.211	5.011	4.514	1.892	1.894
	4.359	5.824	5.942	2.487	2.268

Table 5. Comparison on Around-the-back motion retargeting (top) and generation (bottom). XX/XX are on Character A/B. All results are per joint results. The four rows in each cell are results of Random, Cross-scale, Cross-interaction and Cross-scale-interaction respectively.

Figure 3. The base motion in D2 (M5-M9). From top to bottom: Around-the-back, Back-flip, Big-ben, Noser and Chandelle.

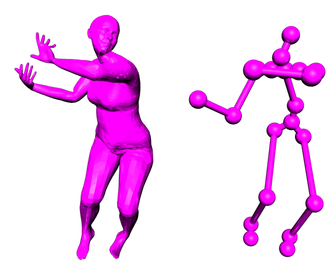

Figure 4. SMPL results on our skeleton. Left: the SMPL generated mesh. Right: the skeleton we captured in Judo motion for Character A. Due to the skeleton differences, e.g. different number of joints and different lengths of bones, severe distortion (both hands and left foot) exists in the body shape.

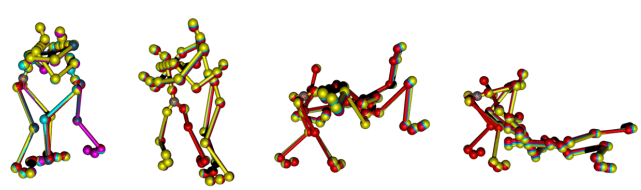

Figure 5. Generation diversity. Judo motion sampled multiple times, shown by different colors.

Metric	F-CNNs	F-GCNs	M-CNNs	M-GCNs	Ours
E_r	1.402/4.493	2.015/3.483	2.641/3.783	1.501/3.142	1.541/2.215
	1.427/4.025	2.421/3.214	2.453/2.924	1.701/3.542	2.511/4.719
	1.481/4.406	2.812/4.082	2.125/3.421	1.412/3.199	3.052/4.974
	3.214/6.643	3.542/6.547	3.895/6.852	4.624/8.954	3.453/6.299
E_b	0.051/0.063	0.122/0.272	0.031/0.092	0.001/0.031	0.002/0.014
	0.071/0.094	0.228/0.309	0.077/0.108	0.008/0.071	0.005/0.039
	0.081/0.104	0.320/0.481	0.334/0.471	0.014/0.191	0.012/0.050
	0.171/0.307	0.422/0.554	0.445/0.575	0.089/0.201	0.041/0.111
JPD	0.637	4.123	4.241	0.480	0.495
	1.734	4.187	4.651	1.712	1.163
	2.794	4.914	4.987	2.923	2.320
	4.045	5.514	5.612	3.818	3.762
FID	0.283/0.806	5.849/6.246	2.421/3.841	0.305/0.512	0.424/0.952
	0.310/0.884	5.244/6.121	2.451/3.874	0.540/0.917	0.878/1.562
	0.711/0.976	6.018/6.924	3.084/4.312	0.749/1.034	0.912/2.134
	1.874/1.854	6.684/7.896	4.548/4.845	1.342/1.837	1.027/2.307
E_b	0.112/0.389	0.398/1.662	0.248/0.745	0.003/0.064	0.006/0.020
	0.162/0.401	0.407/1.823	0.425/0.945	0.008/0.118	0.015/0.041
	0.227/0.454	0.487/2.132	0.504/1.003	0.031/0.216	0.022/0.056
	0.421/0.645	0.722/2.972	0.924/1.781	0.135/0.421	0.037/0.129
JPD	1.510	5.204	4.312	1.613	0.624
	1.601	5.405	4.894	1.819	1.273
	2.718	5.827	5.181	3.003	2.024
	4.248	5.922	6.247	4.252	3.941

Table 6. Comparison on Back-flip motion retargeting (top) and generation (bottom). XX/XX are on Character A/B. All results are per joint results. The four rows in each cell are results of Random, Cross-scale, Cross-interaction and Cross-scale-interaction respectively.

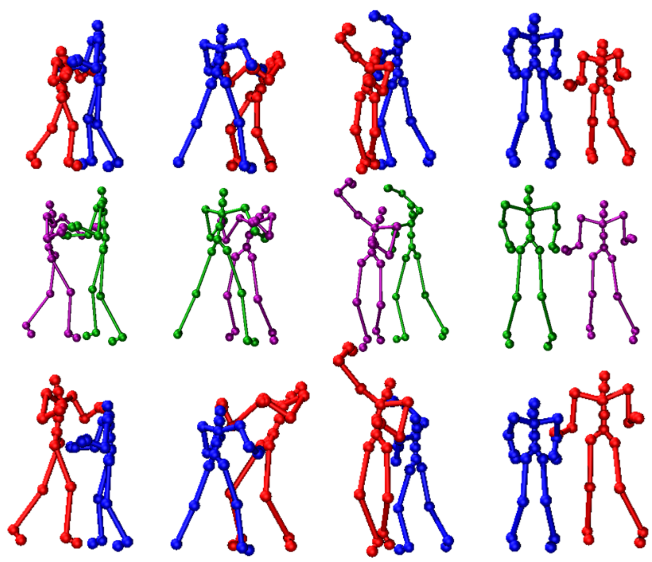

Figure 6. Large-scale extrapolation results. The skeleton of the red character is changed. The motion is Hold-body on scale 0.65 (top), original scale (mid) and scale 1.3 (bottom).

Metric	F-CNNs	F-GCNs	M-CNNs	M-GCNs	Ours
E_r	1.691/3.948	2.371/4.434	3.013/3.212	1.926/4.337	1.621/3.871
	3.802/8.071	3.427/8.724	3.412/8.894	4.016/9.711	2.054/8.354
	3.914/9.221	3.624/9.217	4.642/10.945	5.174/10.915	2.752/8.544
	5.544/10.221	4.052/9.826	4.952/10.954	5.204/11.065	2.453/9.061
E_b	0.049/0.134	0.159/0.316	0.031/0.140	0.001/0.033	0.003/0.009
	0.108/0.227	0.207/0.375	0.099/0.204	0.004/0.091	0.012/0.022
	0.172/0.271	0.271/0.405	0.123/0.306	0.017/0.194	0.022/0.036
	0.211/0.294	0.301/0.534	0.325/0.452	0.036/0.205	0.031/0.101
JPD	0.956	3.569	3.356	0.737	0.495
	0.917	3.453	3.541	1.571	1.163
	2.127	4.485	3.941	2.584	2.320
	2.354	4.755	4.842	2.948	3.762
FID	0.301/1.816	6.833/8.771	0.504/2.051	0.472/0.520	0.407/0.883
	0.651/1.971	7.661/8.875	0.571/2.364	0.841/1.117	0.841/1.465
	0.806/2.011	8.054/9.404	0.604/2.781	0.894/1.199	0.934/2.050
	1.068/2.325	8.725/9.891	1.262/2.934	1.288/1.824	0.939/2.011
E_b	0.072/0.216	1.084/1.497	0.105/0.310	0.0017/0.0606	0.004/0.026
	0.109/0.312	1.400/1.425	0.184/0.412	0.018/0.107	0.009/0.051
	0.206/0.337	1.701/1.832	0.208/0.577	0.037/0.401	0.017/0.067
	0.332/0.521	2.054/2.641	0.355/0.851	0.109/0.484	0.022/0.116
JPD	2.072	5.972	3.451	0.895	0.702
	2.424	6.422	3.411	1.718	2.140
	2.672	7.051	4.823	1.896	2.320
	2.791	7.455	6.271	2.942	2.759

Table 7. Comparison on Big-ben motion retargeting (top) and generation (bottom). XX/XX are on Character A/B. All results are per joint results. The four rows in each cell are results of Random, Cross-scale, Cross-interaction and Cross-scale-interaction respectively.

Metric	F-CNNs	F-GCNs	M-CNNs	M-GCNs	Ours
E_r	1.256/4.045	1.853/3.997	2.645/4.241	2.736/4.360	0.953/3.591
	1.864/4.689	2.907/5.084	2.756/5.601	2.862/5.336	1.638/4.610
	2.362/4.898	3.015/5.915	3.808/6.032	2.991/5.617	1.937/4.841
	2.623/5.185	3.530/6.240	4.810/6.937	3.244/6.054	2.223/5.719
E_b	0.088/0.105	0.141/0.353	0.286/0.422	0.144/0.216	0.002/0.010
	0.582/0.698	0.164/0.391	0.231/0.488	0.189/0.271	0.009/0.033
	0.612/0.706	0.200/0.446	0.409/0.521	0.217/0.595	0.017/0.063
	0.620/0.705	0.220/0.450	0.426/0.554	0.237/0.607	0.031/0.175
JPD	3.557	3.792	5.451	5.670	0.402
	3.804	3.984	5.669	6.265	0.964
	4.602	4.205	6.324	6.618	1.534
	5.552	5.434	6.729	6.887	2.341
FID	0.624/4.578	10.764/12.042	3.011/11.084	0.831/2.471	0.297/1.055
	2.642/3.637	11.684/14.587	3.512/12.986	1.986/3.076	0.685/2.013
	3.186/5.804	15.545/22.688	4.336/14.745	3.957/4.225	0.907/2.435
	4.169/7.804	17.550/23.821	5.306/16.075	4.580/4.904	1.274/4.656
E_b	0.176/0.278	0.121/0.350	0.334/1.147	0.186/0.286	0.004/0.020
	0.532/0.758	0.225/0.421	0.418/1.379	0.167/0.399	0.009/0.067
	0.685/0.721	0.345/0.584	0.514/1.536	0.231/0.763	0.017/0.097
	0.688/0.842	0.385/0.604	0.595/1.623	0.243/0.789	0.052/0.154
JPD	1.116	2.207	2.914	2.843	0.634
	2.513	4.741	5.045	5.157	1.374
	4.895	5.068	6.861	6.854	1.862
	5.542	6.068	7.861	7.560	2.675

Table 8. Comparison on Noser retargeting (top) and generation (bottom). XX/XX are Character A/B results. All results are per joint results. The four rows in each cell are results of Random, Cross-scale, Cross-interaction and Cross-scale-interaction respectively.

Metric	**F-CNNs**	**F-GCNs**	**M-CNNs**	**M-GCNs**	**Ours**
E_r	0.754/4.548	1.696/4.302	0.857/4.241	1.733/4.346	0.735/3.733
	0.930/5.288	1.957/4.553	1.263/4.914	1.869/4.866	1.328/4.542
	1.513/5.585	2.013/4.9014	1.881/4.937	1.909/5.670	1.863/4.649
	2.062/6.018	2.415/6.001	2.384/6.237	2.841/6.287	2.197/5.049
E_b	0.018/0.102	0.086/0.203	0.082/0.222	0.043/0.196	0.003/0.007
	0.064/0.203	0.184/0.334	0.258/0.547	0.134/0.220	0.008/0.012
	0.127/0.259	0.222/0.446	0.299/0.687	0.205/0.335	0.015/0.031
	0.156/0.372	0.252/0.474	0.394/0.697	0.229/0.385	0.034/0.094
JPD	2.645	3.762	4.552	4.850	0.403
	3.512	3.874	5.589	5.125	0.934
	4.214	4.978	6.872	6.051	1.674
	4.985	5.541	7.085	6.452	2.971
FID	0.587/2.584	6.542/7.255	3.214/6.211	0.610/2.714	0.384/0.884
	1.524/3.450	7.225/8.254	3.5124/7.986	1.226/3.274	0.571/2.253
	2.269/4.804	10.542/12.457	3.303/9.524	2.957/4.545	0.694/2.990
	3.542/5.274	13.275/17.681	4.656/12.865	4.033/5.125	1.250/4.458
E_b	0.076/0.278	0.071/0.357	0.124/0.254	0.128/0.208	0.006/0.012
	0.142/0.305	0.122/0.402	0.361/0.537	0.146/0.409	0.015/0.071
	0.285/0.421	0.205/0.408	0.484/0.596	0.223/0.658	0.017/0.085
	0.435/0.527	0.321/0.568	0.590/0.605	0.338/0.763	0.022/0.206
JPD	4.436	4.258	5.454	4.954	0.561
	5.452	5.751	6.592	6.334	1.259
	5.494	6.006	6.881	8.454	1.903
	6.899	8.158	7.461	9.046	2.842

Table 9. Comparison on Chandelle retargeting (top) and generation (bottom). XX/XX are Character A/B results. All results are per joint results. The four rows in each cell are results of Random, Cross-scale, Cross-interaction and Cross-scale-interaction respectively.

	Training samples	E_r	E_b	**JPD**	FID	E_b	**JPD**
M2	24	1.158	0.142	0.892	3.485	0.167	1.428
	12	1.347	0.186	0.963	3.676	0.192	1.667
	6	1.657	0.224	1.305	3.983	0.258	2.017
M7	24	5.861	0.082	3.035	1.897	0.094	3.923
	12	6.025	0.104	3.879	1.957	0.123	4.343
	6	6.254	0.173	4.241	2.124	0.205	4.587

Table 10. Result with limited training samples. Here is the result of Face-to-back (M2) and Big-ben (M7).

Layer index	Output channels	Dimension	Layer	Stride
Input	/	[32,T,n,4]	/	/
1	32	[32,T,n,32]	ST-GCN	1
2	64	[32,T/2,n,64]	ST-GCN	2
3	128	[32,T/4,n,128]	ST-GCN	2
4	256	[32,T/8,n,256]	ST-GCN	2
5	256	[32,T/8,n,256]	ST-GCN	1
6	256	[32,1,n,256]	Temporal Averaging	/
7	262	[32,1,n,262]	Concatenation with $\hat{q}_B^0$ and $\hat{q}_B^T$	/
8	256	[32,1,n,256]	Dense	

Table 11. Detailed architecture of ST-GCN1. T is the motion length. n is the number of joints.

Layer Index	Input	Dimension	Layer
1	Hidden state at time t	[32,1,n,256]	/
2	B_s, $\hat{q}_B^0$, $\hat{q}_B^T$,and $\triangle\bar{q}_B^t$	[32,1,n,10]	Concatenation
3	output of 1, 2	[32,1,n,256]	G-GRU
4	output of 3	[32, 1, n, 256]	Dense
5	output of 4	[32, 1, n, 256]	Dense
6	output of 5	[32,1,n,3]	Dense

Table 12. Detailed architecture of G-GRU1. It takes as input z, $\hat{q}_B^0$ and $\hat{q}_B^T$ and outputs $\triangle\bar{q}_B$. n is the number of joints.

Layer Index	Output channels	Dimension	Layer	Stride
Input	/	[32,T,n,3]	/	/
1	32	[32,T,n,32]	ST-GCN	1
2	64	[32,T/2,n,64]	ST-GCN	2
3	128	[32,T/4,n,128]	ST-GCN	2
4	256	[32,T/8,n,256]	ST-GCN	2
5	256	[32,T/8,n,256]	ST-GCN	1
6	256	[32,1,n,256]	Temporal Averaging	/

Table 13. Detailed architecture of ST-GCN2. T is the motion length and n is the number of joints.

Layer Index	Output channels	Dimension	Layer	Stride
Input	/	[32,T,n,8]	/	/
1	16	[32,T,n,16]	ST-GCN	1
2	16	[32,T/2,n,16]	ST-GCN	2
3	16	[32,T/4,n,16]	ST-GCN	2
4	16	[32,T/8,n,16]	ST-GCN	2
5	16	[32,T/8,n,16]	ST-GCN	1
6	16	[32,1,n,16]	Temporal Averaging	/

Table 14. Detailed architecture of ST-GCN3. T is the motion length and n is the number of joints.

Layer Index	Output channels	Dimension	Layer	Stride
Input	278	[32,1,n,278]	Concatenation of outputs from the $\triangle q_A$ and q'_B branches, $\hat{q}_A^0$ and $\hat{q}_A^T$	/
1	256	[32,1,n,256]	Dense	/
2	256	[32,1,n,256]	Dense	/

Table 15. Detailed architecture after the ST-GCN2 and ST-GCN3. n is the number of joints. The network finally outputs z.

Layer Index	Input	Dimension	Layer
1	Hidden state at time t	[32,1,n,256]	/
2	encoded q'_B, $\hat{q}_A^0$, $\hat{q}_A^T$,and $\triangle\bar{q}_A^t$	[32,1,n,10]	Concatenation
3	output of 1, 2	[32,1,n,256]	G-GRU
4	output of 3	[32, 1, n, 256]	Dense
5	output of 4	[32, 1, n, 256]	Dense
6	output of 5	[32,1,n,3]	Dense

Table 16. Detailed architecture of G-GRU2. It takes as the first input z, encoded q'_B, $\hat{q}_A^0$ and $\hat{q}_A^T$ and outputs $\triangle\bar{q}_A$. n is the number of joints.

Index	Output channels	Feature Shape	Operation	Stride
Input	/	[32,n,3]	/	/
1	32	[32,n,32]	Conv	1
2	64	[32,n/2,64]	Conv and Maxpooling	1
3	128	[32,n/4,128]	Conv and Maxpooling	1
4	256	[32,n/8,256]	Conv and Maxpooling	1
5	260	[32,n/8,260]	Concatenate B_s and $\hat{q}_B$	/
6	256	[32,n/8,256]	Dense	/
Index	Output channels	Feature Shape	Operation	Stride
7	/	[32,n/8,260]	Concatenate B_s and $\hat{q}_B$	/
8	256	[32,n/8,256]	Dense	/
9	256	[32,n/4,256]	ConvTranspose	2
10	128	[32,n/2,128]	ConvTranspose	2
11	32	[32,n,32]	ConvTranspose	2
Output	3	[32,n,3]	Dense	/
Index	Output channels	Feature Shape	Operation	Stride
Input	/	[32,n,3]	/	/
1	32	[32,n,32]	Conv	1
2	64	[32,n/2,64]	Conv and Maxpooling	1
3	128	[32,n/4,128]	Conv and Maxpooling	1
4	256	[32,n/8,256]	Conv and Maxpooling	1
5	264	[32,n/8,264]	Concatenate encoding $\hat{q}_A$ and q'_B	1
6	256	[32,n/8,256]	Dense	/
Index	Output channels	Feature Shape	Operation	Stride
7	/	[32,n/8,264]	Concatenate encoding $\hat{q}_A$ and q'_B	/
8	256	[32,n/8,256]	Dense	/
9	128	[32,n/4,128]	ConvTranspose	2
10	64	[32,n/2,64]	ConvTranspose	2
11	32	[32,n,32]	ConvTranspose	2
Output	3	[32,n,3]	Dense	/

Table 17. F-CNNs detailed architecture in Character B (top) and Character A (bottom)

Index	Output channels	Feature Shape	Operation	Stride
Input	/	[32,n,3]	/	/
1	32	[32,n,32]	GCN	1
2	64	[32,n,64]	GCN	1
3	84	[32,n,84]	Concatenate encoding B_s and $\hat{q}_B$	1
4	128	[32,n,128]	GCN	1
5	256	[32,n,256]	GCN	1
Index	Output channels	Feature Shape	Operation	Stride
6	/	[32,n,276]	Concatenate encoding B_s and $\hat{q}_B$	/
7	256	[32,n,256]	GCN	1
8	128	[32,n,128]	GCN	1
9	64	[32,n,64]	GCN	1
10	32	[32,n,32]	GCN	1
11	3	[32,n,3]	GCN	1
Output	3	[32,n,3]	Dense	1
Index	Output channels	Feature Shape	Operation	Stride
Input	/	[32,n,3]	/	/
1	32	[32,n,32]	GCN	1
2	64	[32,n,64]	GCN	1
3	80	[32,n,80]	Concatenate encoding q'_B and $\hat{q}_A$	1
4	128	[32,n,128]	GCN	1
5	256	[32,n,256]	GCN	1
Index	Output channels	Feature Shape	Operation	Stride
6	/	[32,n,272]	Concatenate encoding q'_B and $\hat{q}_A$	/
7	256	[32,n,256]	GCN	1
8	128	[32,n,128]	GCN	1
9	64	[32,n,64]	GCN	1
10	32	[32,n,32]	GCN	1
11	3	[32,n,3]	GCN	1
Output	3	[32,n,3]	Dense	/

Table 18. F-GCNs detailed architecture in Character B (top) and Character A (bottom)

Index	Output channels	Feature Shape	Operation	Stride
Input	/	[32,T,n,4]	/	/
1	16	[32,T,n,16]	Conv and Maxpooling	1
2	32	[32,T,n,32]	Conv and Maxpooling	1
3	64	[32,T,n,64]	Conv and Maxpooling	1
Index	Output channels	Feature Shape	Operation	Stride
4	/	[32,T,n,65]	Concatenate B_s	/
5	64	[32,T,n,64]	Dense	/
6	32	[32,T,n,32]	ConvTranspose	1
7	16	[32,T,n,16]	ConvTranspose	1
Output	3	[32,T,n,3]	ConvTranspose	1
Output	3	[32,T,n,3]	Dense	/
Index	Output channels	Feature Shape	Operation	Stride
Input	/	[32,T,n,16]	Concatenate encoding $\hat{q}_A$ and q'_B	/
1	32	[32,T,n,32]	Conv and Maxpooling	1
2	64	[32,T,n,64]	Conv and Maxpooling	1
Index	Output channels	Feature Shape	Operation	Stride
3	/	[32,T,n,72]	Concatenate encoding $\hat{q}_A$ and q'_B	/
4	64	[32,T,n,64]	Dense	/
5	32	[32,T,n,32]	ConvTranspose	1
6	16	[32,T,n,16]	ConvTranspose	1
Output	3	[32,T,n,3]	Dense	/

Table 19. M-CNNs detailed architecture in Character B (top) and Character A (bottom)

Index	Output channels	Feature Shape	Operation	Stride
Input	/	[32,T,n,4]	/	/
1	32	[32,T,n,32]	ST-GCN	1
2	64	[32,T,n,64]	ST-GCN	1
3	128	[32,T,n,128]	ST-GCN	1
4	128	[32,T,n,128]	Dense	1
Index	Output channels	Feature Shape	Operation	Stride
5	/	[32,T,n,129]	Concatenate B_s	/
6	128	[32,T,n,128]	ST-GCN	1
7	64	[32,T,n,64]	ST-GCN	1
8	32	[32,T,n,32]	ST-GCN	1
9	16	[32,T,n,16]	ST-GCN	1
Output	3	[32,T,n,3]	ST-GCN	1
Index	Output channels	Feature Shape	Operation	Stride
Input	/	[32,T,n,3]	/	/
1	32	[32,T,n,32]	ST-GCN	1
2	64	[32,T,n,64]	ST-GCN	1
3	128	[32,T,n,128]	ST-GCN	1
4	144	[32,T,n,144]	Concatenate encoding q'_B	1
5	128	[32,T,n,128]	Dense	1
Index	Output channels	Feature Shape	Operation	Stride
6	/	[32,T,n,144]	Concatenate encoding q'_B	/
7	128	[32,T,n,128]	ST-GCN	1
8	64	[32,T,n,64]	ST-GCN	1
9	32	[32,T,n,32]	ST-GCN	1
Output	3	[32,T,n,3]	Dense	/

Table 20. M-GCNs detailed architecture in Character B (top) and Character A (bottom)