Appendix

A. Implementation details

Mapping network. Our mapping network consists of an MLP with 10 output branches, where 10 indicates the number of multiple dance genres. Two fully connected layers are shared among all domains, followed by two specific fully connected layers for each domain. In Table 5, we report the dimensions of the latent code, the hidden layer, and the style code to 1024, 512, and $512 \times T$, where T is equal to the length of the future motions to be predicted. After style code representation, we feed the obtained style code into the transformer decoder as key and value.

Transformer architecture. Our transformer model is identical to the GPT2 architecture [51] and we vary its capacity mainly through varying amounts of blocks and hidden dimensions (see Table 9).

Dataset All the experiments are performed on the AIST++ [38] dataset which contains 3D human dance motions paired with music. Above all, this dataset is the largest 3D human dance dataset obtained from basic and advanced choreographies and has 10 different dance genres which are 0: break, 1: pop, 2: lock, 3: middle hip-hop, 4: LA style hip-hop, 5: house, 6: waack, 7: krump, 8: street jazz, and 9: ballet jazz. Since the dance motions are equally distributed among all dance genres in AIST++, we carefully organize training and test subsets to ensure both have equal dance genres distributions.

Library. The experimentation pipeline is implemented using PyTorch [48], Pytorch-Lightning [15], and Hydra [60]. The code is publicly released at: . Furthre, we integrate the differentiable SMPL-X [49] which is the differentiable SMPL layer using the PyTorch.

B. Additional experiments

Influence of the diversity loss. As explained in Section 3.2, there is a trade-off between diversity and realism depending on the weighting parameter λ_{div} of the diversity loss. Here, we empirically show the influence of the weighting parameter quantitatively. Table 6 presents results for several value of λ_{div} and we observe that is best performance at $\lambda = 0.5$. We use this value in all our experiments.

Influence of the batch size. As explained in Section 3.3, we find that the proposed model works sensitively to batch size. We conduct quantitative comparisons by discretely increasing the batch size with a fixed learning rate. As shown in Table 7, the best performance is recorded at Batch size $=$ 10, which is used in all our experiments.

Influence of the predicted length. After training our model with different predict lengths, we observe how long the model is possible to generate plausible motion sequences by specifying the predicted length at inference time. As shown in Table 8, the best performance is obtained at Predict length $=$ 60 (60) of training (testing). We observe that shorter generations produce incomplete dance motions.

Influence of the model size. As shown in Table 9, we enlarge the transformer block and embedded dimensions for both generator and discriminator. That comes with a consistent and remarkable gain for all metrics without any extra hyperparameter tuning. Further, we observe that our setting is better and more stable to generate diverse dance motions and could obtain minor gain from bigger models.

C. Additional comparisons

Motion-music consistency. We further visualize two examples of beat alignment between music and generated dance. As shown in Figure 7, our results show that the kinematic beats coherence with the musical beats, which agrees with common sense that dancers make movements following the musical beats.

Additional quantitative and qualitative result. Figure 8 and Figure 9 demonstrate the diversity of our generated motions for additional dance genres for the two perspectives identical with Section 4.3.

Qualitative comparison. For a more comprehensive evaluation, we compare our results with SOTA baselines qualitatively. We use the same music beat and seed motions to generate dance motions of 600 frames for all models. Compared to the baselines producing similar motions repeatedly, our model generates diverse and realistic dance motions, as shown in Figure 10.

D. Limitation

We observed that our model often adhere to the style of the seed motion when we synthesis dance motions from one domain to the other domain, although the style-focussing term emphasizes the importance of the embedded style code. This is probably due to the strong interaction between the generated and input seed motions. The output sequence of our model is very short 60 frames (1 seconds). To generate a long-range sequence, we utilize auto-regressive manner where the model predicts future motions based on past motions. Therefore, the seed motion is essential for our framework. Future work will explore seed motion free dance generation, which might become possible with further progress in sequence-to-sequence motion generation predicting long-range duration.

Type	Layer	Activation	Output shape
Shared	Latent z	-	1024
Shared	Linear	GELU	512
Shared	Linear	GELU	512
Unshared	Linear	GELU	512
Unshared	Linear	-	$512 \times T$

Table 5. Details of mapping network architecture.

	$\text{FID}_g \downarrow$	$\text{Dist}_{m,g} \uparrow$	BeatAlign ↑
$\lambda_{div} = 0.1$	30.48	4.11	**0.253**
$\lambda_{div} = 0.5$	**29.52**	**6.93**	0.246
$\lambda_{div} = 0.8$	63.10	5.27	0.142
$\lambda_{div} = 1.0$	67.94	6.02	0.110

Table 6. We investigate a proper trade-off between diversity and motion realism using a weighting parameter of diversity loss. We set the weight λ_{div} to 0.5 in our training.

	$\text{FID}_k \downarrow$	$\text{Dist}_{m,k} \uparrow$	BeatAlign ↑
Batch size = 10	**29.52**	**6.93**	**0.246**
Batch size = 20	31.32	6.29	0.231
Batch size = 30	35.95	3.10	0.184
Batch size = 40	39.29	3.58	0.139

Table 7. We find that our transformer GAN is sensitive to the batch size during training and conduct several experiments using different batch sizes. We set this hyperparameter to 10 in our training.

Training	Testing	$\text{FID}_k \downarrow$	$\text{Dist}_{m,k} \uparrow$	BeatAlign ↑
Predict length = 30	Predict length = 10	38.61	3.50	0.143
Predict length = 30	Predict length = 30	32.52	4.38	0.155
Predict length = 30	Predict length = 60	31.09	6.13	0.205
Predict length = 60	Predict length = 10	33.58	5.86	0.131
Predict length = 60	Predict length = 30	30.17	6.92	0.227
Predict length = 60	Predict length = 60	**29.52**	**6.93**	**0.246**

Table 8. We evaluate the influence of the predicted length. For the fixed model, we observe that the best performance is recorded when the model is trained and tested with Predict length = 60.

Generator Depth	Discriminator Depth	Dim	$\text{FID}_g \downarrow$	$\text{Dist}_{m,g} \uparrow$
4	4	384	33.57	6.23
8	4	384	32.38	6.04
8	8	512	30.14	6.51
12	8	512	**29.52**	**6.93**

Table 9. We scale up the model size of MNET. Here, Depth is the number of transformer block and Dim represents the embedded dimension of transformer.

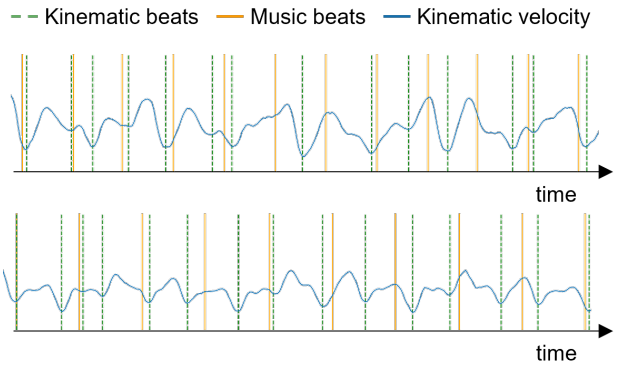

Figure 7. We visualize additional examples of beat alignment between music and generated dance. The top is a short clip of Krump while the bottom is a short clip of hip-hop.

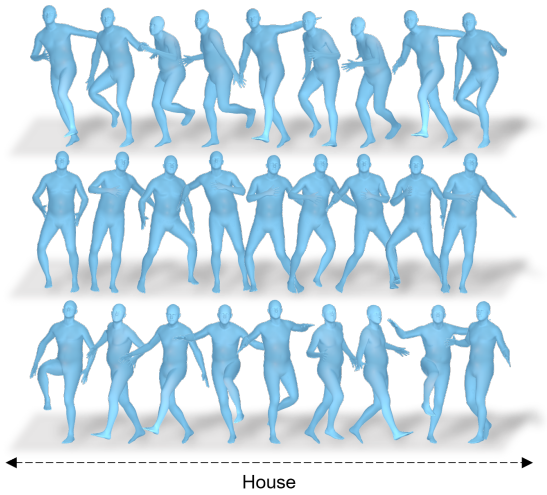

Figure 8. We visualize additional qualitative results. The results are guided by latent code.

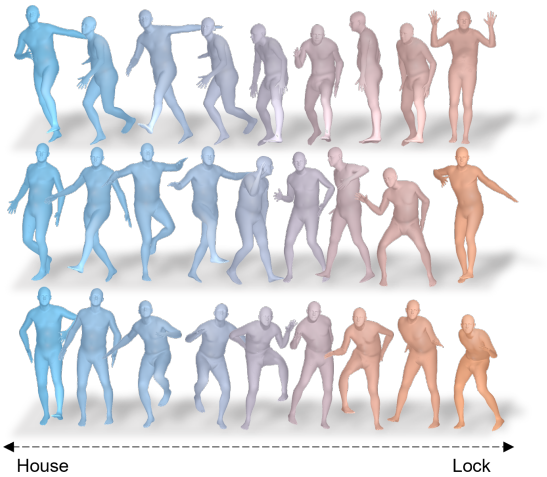

Figure 9. We visualize additional qualitative results. The results are guided by style code.

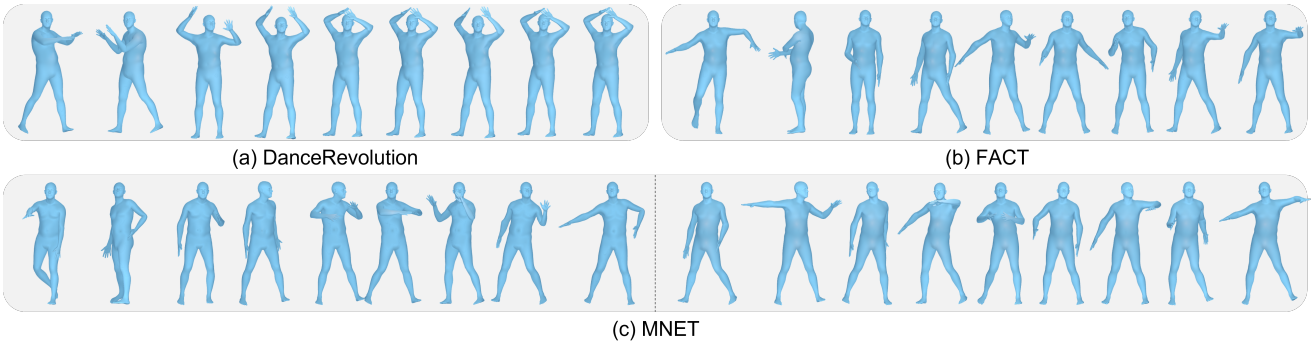

Figure 10. We compare our results with DanceRevolution [21] and FACT [38]. Compared to the baseline models that return the repeated motions, MNET generates favorable and dynamic motions in a variety of ways.