Large Motion Model for Unified Multi-Modal Motion Generation – Supplementary Material –

A MotionVerse

In this section, we offer additional details about the construction of **MotionVerse** benchmark.

A.1 Dataset Preprocess

Based on the characteristics of each dataset, we employ different preprocessing methods. To avoid overlap between the training and test sets of different datasets, we **excluded** sequences from the training set of each benchmark that intersect with any test set motion sequences. Below, we provide detailed explanations of the processing methods for each dataset. After aligning each dataset to SMPL-X joints and processing them into TOMATO format using the same set of scripts, they are decomposed into 10 body parts. Therefore, our main focus will be on how each dataset is aligned to SMPL-X 3D joints and which body parts are included in each dataset.

HumanML3D. The HumanML3D dataset is annotated from two sources: AMASS data and HumanAct12. The former provides native SMPL-X annotations, while the latter offers 22 keypoint annotations based on the SMPL format. Additionally, the MotionX dataset provides facial motion data corresponding to each action sequence in HumanML3D. Therefore, there are two overall annotation formats. For data from AMASS, it includes all 10 body parts and does not require keypoint mapping. We use SMPL-X model to convert the SMPL-X beta parameters and theta parameters into 3D coordinates. For data from HumanAct12, it includes 7 body parts (excluding face, left hand, and right hand).

KIT-ML. We located the AMASS data corresponding to KIT-ML and utilized this portion of the AMASS data to generate the motion sequences for KIT-ML. Since there is no additional face motion data available, it comprises a total of 9 body parts.

Motion-X. Motion-X provides key points based on the SMPL-X format and facial expressions based on FLAME. Here, we don't need to perform additional key point conversion, and it includes all 10 body parts.

BABEL. BABEL is also annotated based on AMASS. Since there is no face motion available, we only consider its 9 body parts.

UESTC. For the UESTC dataset, we follow the processing method used in ACTOR. We use the SMPL parameters estimated from VIBE as the raw data. We use default betas parameters to obtain the 3D coordinates of each joint. Since

we do not consider global orientation and global translation during evaluation, and due to the significant noise in the estimation from VIBE, we do not consider four body parts: left hand, right hand, face expression, and global configuration.
HumanAct12. For the HumanAct12 dataset, we employ the pre-processing method used in HumanML3D. Here, HumanAct12 does not include three body parts: left hand, right hand, and facial expression.
NTU-RGB-D 120. For the NTU-RGBD 120 dataset, it comes with native 3D keypoint annotations, but due to their poor accuracy, we only consider the inherent motion captured by these keypoints. Regarding the spine, we use an interpolation method to map the keypoint data from NTU-RGBD 120 to the SMPLX format for the spine. Finally, we only consider four body parts: spine, left hand, right hand, and head.
AMASS. AMASS provides annotations based on the SMPL-X format. We use the provided beta and theta parameters to obtain the corresponding 3D keypoints. Here, we do not consider the body part of facial expression.
3DPW. 3DPW provides SMPL parameters, allowing us to obtain the 3D keypoint positions in the SMPL format. Since the motion prediction task involved in 3DPW does not consider global translation, we only consider six body parts: spine, left arm, right arm, left leg, right leg, and head.
Human3.6M. Similar to 3DPW, we obtain keypoint sequences using SMPL parameters, which are ultimately converted into six body parts: spine, left arm, right arm, left leg, right leg, and head.
TED-Gesture++. TED-Gesture++ only provides keypoints for the upper body, so we consider only the spine, left arm, right arm, and head as the five body parts. For the spine, we utilize interpolation to obtain a keypoint set that conforms to SMPL-X.
TED-Expressive. The keypoint annotations of TED-Expressive++ are almost identical to SMPL-X. We directly selected the corresponding keypoints and removed the redundant parts. It includes all body parts except for facial expressions.
Speech2Gesture-3D. Similar to TED-Expressive, we directly selected the corresponding keypoints and removed the redundant parts. It includes all body parts except for facial expressions.
BEAT. The keypoint set of BEAT completely covers the keypoints of SMPL-X and provides facial expression, so we consider all body parts and discard the keypoints that do not exist in SMPL-X.
AIST++. AIST++ provides annotations based on SMPL parameters, corresponding to 7 body parts excluding face expression, left hand, and right hand.
MPI-INF-3DHP. Similar to 3DPW, we obtain keypoint sequences using SMPL parameters, which are ultimately converted into six body parts: spine, left arm, right arm, left leg, right leg, and head.

A.2 Motion Translator

During evaluation, there are three types of estimation. The first type is based on H3D vectors, primarily used in the Text2Motion datasets. For this type,

we train an MLP to convert our frame representations into the corresponding representations required for evaluation. The second type is based on keypoint sequences, without considering global translation and global orientation, such as in the UESTC evaluation. Here, we also directly train an MLP for mapping. The third type considers global translation and global orientation, and is based on keypoint sequence evaluation. In this case, we first convert our representations into the keypoint sequence format and then train an MLP for mapping.

B Large Motion Model

To facilitate a deeper understanding of the LMM approach, this chapter provides additional technical details.

B.1 Diffusion Model

This paper utilizes the Denoising Diffusion Probabilistic Model (DDPM) [8], a probability generative model based on the Markov chain. Its essence lies in two intertwined processes: the forward diffusion process and the reverse diffusion process.

The forward diffusion process systematically injects noise into the original distribution, progressively disrupting the data's initial distribution. Starting from the original distribution $x_0 \sim q(x_0)$, noise is added over T steps to generate x_1, x_2,..., x_T. This process employs an efficient, tractable noise addition method, with Gaussian perturbation being a classic approach. The specific formula is:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)=\mathcal{N}\left(\mathbf{x}_t ; \sqrt{1-\beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right), \tag{1}$$

where β controls the amount of noise added. In the context of motion generation tasks, x can be considered a series of poses. To streamline the forward process, the noise-added result at any step can be approximately calculated from x_0: $\mathbf{x}_t=\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}$, where $\alpha_t=1-\beta_t$ and $\bar{\alpha}_t=\prod_{s=1}^{t} \alpha_t$.

The reverse diffusion process is the inverse operation of adding noise, aiming to restore the original distribution from a noisy distribution. Given the difficulty and critical nature of this process, we employ deep learning models to learn the denoising process, defined as: $q(x_{t-1} \mid x_t) = M(\mathbf{x}, \mathrm{m}, \mathrm{c})$. During the model training phase, the supervisory objective is to minimize the difference between the predicted distribution $\hat{x}_0$ and the ground truth x_0.

B.2 ArtAttention

Following a similar approach to FineMoGen, we incorporate the temporal aspect to account for the influence of motion sequences and other condition signals across different time intervals. Specifically, we introduce the notion of time

explicitly into this process. Formally, we present the following approximation for refining temporal features:

$$\mathbf{Y}_{k,i} \approx \mu_i(x_k) = \sum_{j=1}^{N_g} \mathbf{G}'_{i,j}(x_k) \cdot \mathbf{G}^*_{i,j}(x_k) \tag{2}$$

where x_k represents the time position of kth element in the motion sequence. $\mathbf{G}'_{i,j}(x)$ indicates the time-varied signal we derive from the feature vector $\mathbf{G}_{i,j}$ and $\mathbf{G}^*_{i,j}(x_k)$ denotes the relative significance of this template for the kth position. $\mathbf{G}_{i,j}$ is the j-th global template in the i-th attention head. We construct $\mathbf{G}^*_{i,j}(x_k)$ as:

$$\mathbf{G}^*_{i,j}(x_k) = \frac{e^{-(x_k - \mathbf{G}^t_{i,j})^2/\sigma^2}}{\sum_{l\in[1,N_g]} e^{-(x_k - \mathbf{G}^t_{i,l})^2/\sigma^2}}, \tag{3}$$

In this setup, the jth global template of the ith group is considered as a set of signals propagating outward from the temporal center $\mathbf{G}^t_{i,j}$. As for $\mathbf{G}'_{i,j}(x)$, we consider its Taylor expansion at $\mathbf{G}^t_{i,j}$:

$$\mathbf{G}'_{i,j}(x) = \sum_{n=0}^{k} \frac{\mathbf{G}^{(n)}_{i,j}}{n!}(x - \mathbf{G}^t_{i,j})^n. \tag{4}$$

We use linear projections to process the original $\mathbf{G}_{i,j}$ and acquire all coefficients $\mathbf{G}^t_{i,j}, \mathbf{G}^{(n)}_{i,j}, n \in [0, k]$. We perceive a global template as an anchor with its initial state defined as $\mathbf{G}^{(0)}_{i,j}$, velocity as $\mathbf{G}^{(1)}_{i,j}$, acceleration as $\mathbf{G}^{(2)}_{i,j}$ and so on. Therefore we name this method a kinetic modelling on the latent feature space.

Moreover, to integrate influences from all signals, we adopt the square of the time difference as a metric to assess the significance of each global template. We employ a Softmax operation to standardize their weights. An immediate benefit of this modeling strategy is its flexibility in appending a new stage subsequent to the current one. This can be achieved by adjusting a bias term in $\mathbf{G}^t_{i,j}$ accordingly, facilitating our method to execute zero-shot temporal combination.

B.3 Stylization Block

The primary function of the stylization block is to inject the information of the timestamp t into the features, thereby informing the model about the current step in the reverse process. This enhancement aids in improving the model's denoising capability. The stylization block injects information about frame rate, dataset name, and current timestep into the feature representation. Drawing inspiration from FineMoGen, we convert the timestamp t into a vector $\mathbf{e_t}$. In each stylization block, $\mathbf{e_t}$ undergoes two linear transformations to generate two features $\mathbf{e_w} \in \mathbb{R}^{H\times D}$ and $\mathbf{e_b} \in \mathbb{R}^{H\times D}$. Every pose feature θ inputted into this module is optimized as $\theta' = \theta \cdot \mathbf{e_w} + \mathbf{e_b}$, where $(\cdot)$ denotes Hadamard product.

C Experiments

C.1 Implementation Details

Batch Formation Overall, we determine the sampling probability of motion sequences in each dataset based on the quality of the dataset and the diversity of the actions.

1. **Text-to-Motion (40%)**: In the task of text-to-motion, there is a wide variety of motion types, mostly consisting of high-quality motion capture data from AMASS, which is beneficial for the model. Therefore, the sampling proportion is set relatively high at 40%. Within this category, HumanML3D provides high-quality and semantically rich motions, accounting for 15%; Motion-X, although lower in quality, offers high diversity in motions, also at 15%; KIT-ML and BABEL each contribute 5%.
2. **Unconditional Motion Generation (25%)**: We primarily focus on the AMASS dataset, where the motion quality is generally high, aiding the model in learning motion priors. Hence, we set a relatively high proportion for this dataset.
3. **Action-to-Motion (10%)**: We uniformly sample sequences from the HumanAct12, UESTC, and NTU-RGBD 120 datasets.
4. **Speech-to-Gesture (10%)**: As BEAT is selected as the test set, we assign it half of the weight. The remaining portion is evenly distributed among TED-Gesture++, TED-Expressive, and Speech2Gesture-3D.
5. **Music-to-Dance (5%)**: For Music2dance, there is only one AIST++ dataset, which accounts for all the weight.
6. **Motion Imitation (10%)**: During training, we exclude the 3DPW dataset and only consider MPI-INF-3DHP and H36M, with both datasets equally sharing the weight.

Model	#Latent Dim	#Layers	#Experts	#Params
LMM-Tiny	64	4	16	90M
LMM-Small	64	8	16	160M
LMM-Base	128	12	16	410M
LMM-Large	128	20	32	760M

Table 1: Model card.

Model Card. Tab. 1 shows the hyperparameter of each variant.
Mask strategy. Considering that larger models have stronger capabilities to fit motions, to enhance the control ability of conditions, we use mask probabilities of 0.1, 0.2, 0.3, and 0.4 for LMM-Tiny, LMM-Small, LMM-Base, and LMM-Large, respectively.

C.2 More Quantitative Results

Table 2: Quantitative results on the KIT-ML test set.

Methods	R Precision↑ Top 1	Top 2	Top 3	FID↓	MM Dist↓	Diversity↑	MM↑
Real motions	$0.424^{\pm.005}$	$0.649^{\pm.006}$	$0.779^{\pm.006}$	$0.031^{\pm.004}$	$2.788^{\pm.012}$	$11.08^{\pm.097}$	-
Guo *et al.* [5]	$0.370^{\pm.005}$	$0.569^{\pm.007}$	$0.693^{\pm.007}$	$2.770^{\pm.109}$	$3.401^{\pm.008}$	$10.91^{\pm.119}$	$1.482^{\pm.065}$
T2M-GPT [18]	$0.416^{\pm.006}$	$0.627^{\pm.006}$	$0.745^{\pm.006}$	$0.514^{\pm.029}$	$3.007^{\pm.023}$	$10.921^{\pm.108}$	$1.570^{\pm.039}$
MDM [12]	-	-	$0.396^{\pm.004}$	$0.497^{\pm.021}$	$9.191^{\pm.022}$	$10.847^{\pm.109}$	$\mathbf{1.907^{\pm.214}}$
MotionDiffuse [19]	$0.417^{\pm.004}$	$0.621^{\pm.004}$	$0.739^{\pm.004}$	$1.954^{\pm.062}$	$2.958^{\pm.005}$	$\underline{11.10}^{\pm.143}$	$0.730^{\pm.013}$
ReMoDiffuse [20]	$0.427^{\pm.014}$	$0.641^{\pm.004}$	$0.765^{\pm.055}$	$\underline{0.155}^{\pm.006}$	$2.814^{\pm.012}$	$10.80^{\pm.105}$	$1.239^{\pm.028}$
FineMoGen [21]	$\underline{0.432}^{\pm.006}$	$0.649^{\pm.005}$	$0.772^{\pm.006}$	$0.178^{\pm.007}$	$2.869^{\pm.014}$	$10.85^{\pm.115}$	$1.877^{\pm.093}$
MoMask [4]	$\mathbf{0.433^{\pm.007}}$	$\mathbf{0.656^{\pm.005}}$	$\mathbf{0.781^{\pm.005}}$	$0.204^{\pm.011}$	$\mathbf{2.779^{\pm.022}}$	-	$1.131^{\pm.043}$
LMM-Tiny	$0.419^{\pm.018}$	$0.627^{\pm.014}$	$0.748^{\pm.019}$	$0.817^{\pm.015}$	$2.904^{\pm.022}$	$10.85^{\pm.087}$	$1.607^{\pm.110}$
LMM-Small	$0.421^{\pm.015}$	$0.634^{\pm.021}$	$0.755^{\pm.017}$	$0.471^{\pm.017}$	$2.851^{\pm.021}$	$10.94^{\pm.101}$	$1.625^{\pm.114}$
LMM-Base	$0.428^{\pm.015}$	$0.648^{\pm.017}$	$0.769^{\pm.017}$	$0.239^{\pm.015}$	$2.810^{\pm.018}$	$11.05^{\pm.097}$	$1.804^{\pm.130}$
LMM-Large	$0.430^{\pm.015}$	$\underline{0.653}^{\pm.017}$	$\underline{0.779}^{\pm.014}$	$\mathbf{0.137^{\pm.023}}$	$\underline{2.791}^{\pm.018}$	$\mathbf{11.24^{\pm.103}}$	$\underline{1.885}^{\pm.127}$

Text-to-Motion. We observed that compared to its performance on HumanML3D, LMM-Large performs slightly worse on KIT-ML, which could be related to the proportion of the two datasets in batch formation. However, overall, KIT-ML also achieves accuracy comparable to the state-of-the-art, especially achieving a new state-of-the-art in terms of FID.

Table 3: Quantitative results for Action-conditioned Motion Generation. As for UESTC dataset, we report FID on the test split. MM: MultiModality.

Methods	HumanAct12 FID↓	Accuracy↑	Diversity→	MM→	UESTC FID↓	Accuracy↑	Diversity→	MM→
Real motions	$0.020^{\pm.010}$	$0.997^{\pm.001}$	$6.850^{\pm.050}$	$2.450^{\pm.040}$	$2.79^{\pm.29}$	$0.988^{\pm.001}$	$33.34^{\pm.320}$	$14.16^{\pm.06}$
Action2Motion [6]	$0.338^{\pm.015}$	$0.917^{\pm.003}$	$6.879^{\pm.066}$	$2.511^{\pm.023}$	-	-	-	-
ACTOR [11]	$0.12^{\pm.00}$	$0.955^{\pm.008}$	$6.84^{\pm.03}$	$2.53^{\pm.02}$	$23.43^{\pm2.20}$	$0.911^{\pm.003}$	$31.96^{\pm.33}$	$14.52^{\pm.09}$
INR [2]	$0.088^{\pm.004}$	$0.973^{\pm.001}$	$6.881^{\pm.048}$	$2.569^{\pm.040}$	$15.00^{\pm.09}$	$0.941^{\pm.001}$	$31.59^{\pm.19}$	$14.68^{\pm.07}$
MotionDiffuse [19]	$\underline{0.07}^{\pm.00}$	$\mathbf{0.992^{\pm.13}}$	$\mathbf{6.85^{\pm.02}}$	$\underline{2.46}^{\pm.02}$	$\underline{9.10}^{\pm.437}$	$\underline{0.950}^{\pm.000}$	$\underline{32.42}^{\pm.214}$	$14.74^{\pm.07}$
LMM-Tiny	$0.105^{\pm.00}$	$0.992^{\pm.008}$	$6.819^{\pm.025}$	$\mathbf{2.457^{\pm.018}}$	$20.16^{\pm1.78}$	$0.917^{\pm.002}$	$30.80^{\pm.228}$	$\mathbf{14.29^{\pm.066}}$
LMM-Small	$0.094^{\pm.00}$	$0.963^{\pm.008}$	$6.827^{\pm.028}$	$2.498^{\pm.022}$	$14.28^{\pm1.14}$	$0.922^{\pm.002}$	$31.25^{\pm.231}$	$\underline{14.42}^{\pm.067}$
LMM-Base	$0.087^{\pm.00}$	$\underline{0.985}^{\pm.007}$	$\underline{6.848}^{\pm.030}$	$2.551^{\pm.022}$	$10.36^{\pm0.60}$	$0.948^{\pm.000}$	$32.39^{\pm.236}$	$14.65^{\pm.065}$
LMM-Large	$\mathbf{0.065^{\pm.00}}$	$\mathbf{0.992^{\pm.008}}$	$6.871^{\pm.031}$	$2.560^{\pm.019}$	$\mathbf{9.01^{\pm0.54}}$	$\mathbf{0.952^{\pm.000}}$	$\mathbf{32.58^{\pm.254}}$	$14.81^{\pm.064}$

Action-to-Motion. On the action-conditioned motion generation task, each LMM-Large model achieves the best performance in terms of both FID and Accuracy. Additionally, due to exposure to more data, it exhibits higher diversity and multimodality. However, because of the nature of the action-to-motion task, an increase in both aspects does not necessarily indicate better performance.

Speech-to-Gesture. In MotionVerse, we introduce multiple speech-to-gesture datasets, and overall, LMM-Large performs impressively on the BEAT dataset as well.

Motion Imitation We evaluate our method on the test set of 3DPW, and obtain PA-MPJPE scores of 95.7, 91.2, 76.3, and 71.5 for LMM-Tiny, LMM-Small, LMM-Base, and LMM-Large, respectively. For reference, the PA-MPJPE scores for HMR and VIBE are 81.3 and 51.9, respectively. The performance for

Table 4: Quantitative results on Speech-to-Gesture on the BEAT dataset.

Methods	FGD↓	SRGR↑	BeatAlign↑
Seq2Seq [17]	261.3	0.173	0.729
Speech2Gesture [3]	256.7	0.092	0.751
MultiContext [16]	176.2	0.195	0.776
Audio2Gesture [9]	223.8	0.097	0.766
CaMN [10]	123.7	0.239	0.783
TalkShow [15]	91.0	-	0.840
GestureDiffuCLIP [1]	85.17	-	-
CoG [14]	**45.87**	**0.308**	**0.931**
LMM-Tiny	92.51	0.142	0.825
LMM-Small	86.94	0.169	0.836
LMM-Base	57.18	0.228	0.879
LMM-Large	47.95	0.277	0.913

video-conditioning is relatively low; we will focus on addressing this issue in future work.

Table 5: Quantitative results of motion prediction on the Human3.6M test set for different time steps (ms). We report the MPJPE error in *mm*.

Method	Human3.6M							
	80	160	320	400	560	720	880	1000
siMLPe [7]	9.6	21.7	46.3	57.3	75.7	90.1	101.8	109.4
GCNext [13]	**9.3**	**21.5**	45.5	56.4	74.7	88.9	100.8	108.7
LMM-Tiny	14.8	28.6	48.3	59.2	79.3	93.6	105.9	112.0
LMM-Small	14.1	27.4	47.2	58.1	78.1	91.5	103.4	110.3
LMM-Base	12.9	25.9	44.9	55.0	74.8	87.6	99.5	107.1
LMM-Large	11.8	23.6	**43.7**	**53.1**	**73.6**	**85.0**	**96.9**	**104.6**

Motion Prediction Similar to the conclusion we found in 3DPW and AMASS dataset, LMM-Large performs worse than the existing work in short-term prediction and better than these work in long-term prediction.

Table 6: Quantitative results of conditional motion completion on the HumanML3D test set. We report the MPJPE error in *mm*. We use LMM-Large for all experiments.

Condition	First 25 frames	Last 25 frames	avg-MPJPE
No	Yes	No	63.8
No	Yes	Yes	59.1
Yes	Yes	No	54.7
Yes	Yes	Yes	51.9

Conditional Motion Completion To facilitate the conditional motion completion task, we selected motion sequences from the HumanML3D test set with lengths ranging from 80 to 150 frames. We experimented with various settings and observed that the difficulty of motion inbetweening is significantly lower than motion prediction. Furthermore, introducing text conditions proved advantageous in reducing prediction errors.

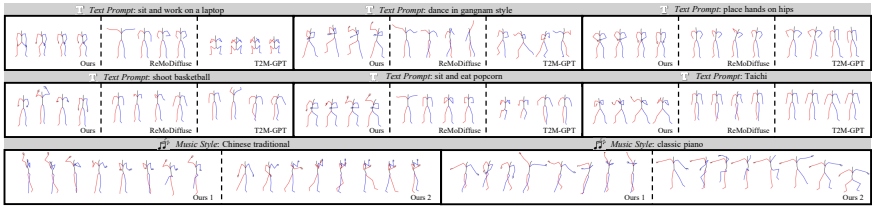

Fig. 1: Visualization Results. Some Qualitative comparations between LMM, ReMoDiffuse and T2M-GPT. We also show two music-to-dance sequences.

C.3 More Ablation Study

Table 7: More Ablation Study.

Setting	Type	HumanML3D			3DPW		
		Top 1	FID	MModality	80	400	1000
Using 'all'	S	0.493	0.428	2.127	16.5	46.5	83.6
No Pretraining	S	0.512	0.144	1.530	17.9	48.4	87.6
Single dataset	S	0.514	0.145	1.462	18.6	50.1	94.3
Ours	S	0.505	0.227	1.761	16.2	45.9	82.8
Using 'all'	B	0.502	0.187	2.619	14.3	44.4	74.2
No Pretraining	B	0.515	0.141	2.305	16.8	45.0	76.3
Single dataset	B	0.516	0.143	2.218	17.5	48.2	81.7
Ours	B	0.511	0.138	2.426	14.1	44.1	73.6
Using 'all'	L	0.519	0.107	**2.915**	13.3	42.9	69.0
No Pretraining	L	0.521	0.075	2.536	15.9	43.9	74.2
Single dataset	L	0.520	0.086	2.417	16.5	44.7	75.9
Ours	L	**0.525**	**0.040**	2.683	**13.1**	**42.4**	**68.0**

1) Using 'all' label. Using the 'all' label has varying effects on different types of tasks. For conditional generation tasks, since metrics like FID are calculated using contrastive learning models trained only on this dataset, generating actions outside the dataset distribution, even if they meet given requirements, can result in worse numerical indicators. Conversely, for motion prediction tasks, the impact on numerical indicators is relatively minor. *2) Pretraining & multi-task learning.* For motion prediction, introducing pretraining and multi-task learning enables the model to acquire better motion priors, resulting in more accurate predictions. For conditional motion generation, despite a significant increase in the number of parameters, LMM-Base does not show a noticeable improvement in accuracy compared to LMM-Small. In addition, the potential of LMM-Large has not been fully unleashed. However, the effectiveness of larger models is better realized after incorporating these two techniques.

C.4 More Visualization Result

We provide more comparison in Fig. 1. Our synthesized are more consistent with given prompts while with higher motion quality. In addition, we provide four dance sequences with two different types of music. We will supply more results in the final version, including qualitative comparison on each task and generated motions under multiple conditions.

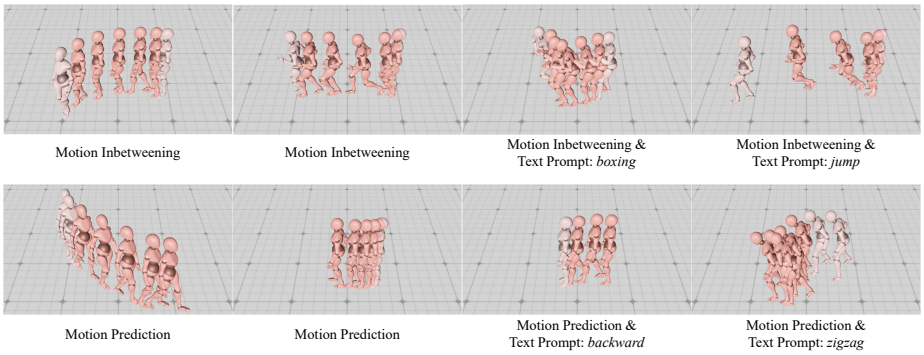

Fig. 2: Visualization results.. Visualization results on Motion Inbetweening and Motion Prediction, without and with text prompt control.

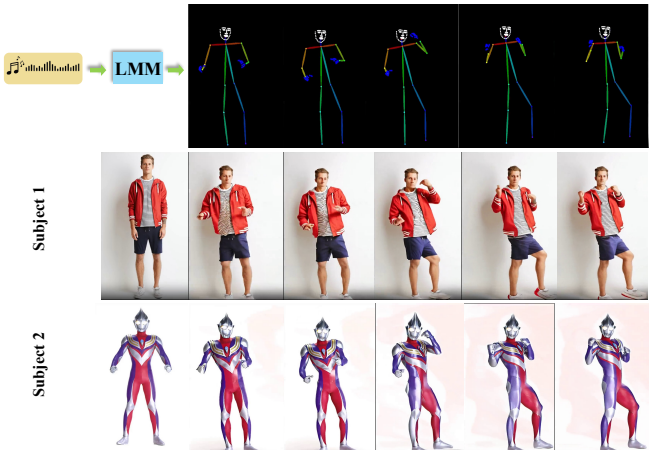

Fig. 3: Video Generation with our synthesized motion sequence. After generating a sequence of motions conditioned on music by our LMM-Large, we map the 3D keypoints to a 2D plane, serving as guidance for video generation.

Fig. 2 shows examples about motion predicion and motion in-betweening. The left four images are generated without other conditions and LMM generates the most probable sequences according to the given clues. The right four images are generated with specific text prompts. Although some spatial conditions are not highly consistent with the textual description, our model can still synthesize reasonable sequences.

C.5 More Application

In Figure 3, we show two videos that are generated based on our synthesized motion sequences. As a vital application direction, users can leverage our large motion model to customize their desired motion data by providing personalized condition signals, such as text commands or accompanying music. With the

assistance of off-the-shelf motion-guided video generation technology, users can freely create videos for their favorite characters.

D Limitation and Broader Impact

Limitation. The intermediate representation we propose can only address scenarios where entire body parts are missing, but it struggles to effectively handle cases where individual keypoints within a body part are missing. Our method of using motion translators introduces additional noise in downstream tasks, leading to a decrease in motion quality. A more flexible approach to motion representation and modeling needs to be explored and researched. Additionally, due to practical limitations in memory, our model needs to employ zero-shot methods for long-sequence motion generation, which may pose challenges for users in practical applications.

Boarder Impact. The ability to generate natural human motion under flexible condition signals can highly enhance productivity. However, it may also be misused for malicious activities such as creating deceptive deepfake videos or generating realistic-looking but false evidence in legal cases.

References

1. Ao, T., Zhang, Z., Liu, L.: Gesturediffuclip: Gesture diffusion model with clip latents. ACM Trans. Graph.
2. Cervantes, P., Sekikawa, Y., Sato, I., Shinoda, K.: Implicit neural representations for variable length human motion generation. In: European Conference on Computer Vision. pp. 356–372. Springer (2022)
3. Ginosar, S., Bar, A., Kohavi, G., Chan, C., Owens, A., Malik, J.: Learning individual styles of conversational gesture. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3497–3506 (2019)
4. Guo, C., Mu, Y., Javed, M.G., Wang, S., Cheng, L.: Momask: Generative masked modeling of 3d human motions. arXiv preprint arXiv:2312.00063 (2023)
5. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5152–5161 (2022)
6. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2motion: Conditioned generation of 3d human motions. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2021–2029 (2020)
7. Guo, W., Du, Y., Shen, X., Lepetit, V., Alameda-Pineda, X., Moreno-Noguer, F.: Back to mlp: A simple baseline for human motion prediction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4809–4819 (2023)
8. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
9. Li, J., Kang, D., Pei, W., Zhe, X., Zhang, Y., He, Z., Bao, L.: Audio2gestures: Generating diverse gestures from speech audio with conditional variational autoencoders. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11293–11302 (2021)
10. Liu, H., Zhu, Z., Iwamoto, N., Peng, Y., Li, Z., Zhou, Y., Bozkurt, E., Zheng, B.: Beat: A large-scale semantic and emotional multi-modal dataset for conversational gestures synthesis. In: European Conference on Computer Vision. pp. 612–630. Springer (2022)
11. Petrovich, M., Black, M.J., Varol, G.: Action-conditioned 3d human motion synthesis with transformer vae. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10985–10995 (2021)
12. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-or, D., Bermano, A.H.: Human motion diffusion model. In: The Eleventh International Conference on Learning Representations (2022)
13. Wang, X., Cui, Q., Chen, C., Liu, M.: Gcnext: Towards the unity of graph convolutions for human motion prediction. arXiv preprint arXiv:2312.11850 (2023)
14. Xu, Z., Zhang, Y., Yang, S., Li, R., Li, X.: Chain of generation: Multi-modal gesture synthesis via cascaded conditional control. arXiv preprint arXiv:2312.15900 (2023)
15. Yi, H., Liang, H., Liu, Y., Cao, Q., Wen, Y., Bolkart, T., Tao, D., Black, M.J.: Generating holistic 3d human motion from speech. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 469–480 (2023)
16. Yoon, Y., Cha, B., Lee, J.H., Jang, M., Lee, J., Kim, J., Lee, G.: Speech gesture generation from the trimodal context of text, audio, and speaker identity. ACM Transactions on Graphics (TOG) **39**(6), 1–16 (2020)
17. Yoon, Y., Ko, W.R., Jang, M., Lee, J., Kim, J., Lee, G.: Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In: 2019

International Conference on Robotics and Automation (ICRA). pp. 4303–4309. IEEE (2019)
18. Zhang, J., Zhang, Y., Cun, X., Huang, S., Zhang, Y., Zhao, H., Lu, H., Shen, X.: T2m-gpt: Generating human motion from textual descriptions with discrete representations. arXiv preprint arXiv:2301.06052 (2023)
19. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
20. Zhang, M., Guo, X., Pan, L., Cai, Z., Hong, F., Li, H., Yang, L., Liu, Z.: Remodiffuse: Retrieval-augmented motion diffusion model. In: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023. pp. 364–373 (2023)
21. Zhang, M., Li, H., Cai, Z., Ren, J., Yang, L., Liu, Z.: Finemogen: Fine-grained spatio-temporal motion generation and editing. Advances in Neural Information Processing Systems **36** (2024)