Self-Intersection-Aware 3D Human Motion Generation Using an Efficient Human Sphere Proxy

Pascal Herrmann[1]
pascal.herrmann@de.bosch.com
Maarten Bieshaar[1]
maarten.bieshaar@de.bosch.com
Dennis Mack[1]
dennis.mack@de.bosch.com
Robert Herzog[1]
paulrobert.herzog@de.bosch.com
Juergen Gall[2,3]
gall@iai.uni-bonn.de

[1] Bosch Research
[2] University of Bonn
[3] Lamarr Institute for Machine Learning and Artificial Intelligence

Abstract

Human motion generation has made tremendous progress in recent years, with state-of-the-art approaches surpassing ground truth data in leading evaluation benchmarks. However, visual inspection of the generated motions paints a different picture. Even state-of-the-art approaches generate motions frequently containing self-intersections, i.e., body parts interpenetrating, which are strong artifacts, severely limiting the perceived motion quality. We introduce a novel loss, which explicitly penalizes self-intersections, to the training of human motion generation methods. We base our loss on a sphere proxy of human geometry, which allows us to calculate a self-intersection loss 98 % faster and uses 83 % less memory than comparable methods based on triangular meshes. The loss is agnostic to the specific approach, and we add it to the training of the recent human motion generation methods *human motion diffusion model (MDM)* and *MoMask*. Our extensive experiments show a reduction of self-intersections in generated motions of up to 49 % while improving other evaluation metrics. The code is available at https://github.com/boschresearch/humanspherereproxy.

1 Introduction

Generating realistic 3D human motions is an essential task in virtual reality applications [43], video games [40], computer animation [24], and synthetic data generation for deep learning [54]. Most recent approaches [9, 15, 49] rely on data-driven approaches using transformer-based [65] diffusion models [17, 44, 45] or variational autoencoder (VAE) [21, 53]. Considering metrics on current evaluation benchmarks [14, 35], human motion generation has

made tremendous progress over the years. However, actually looking at the generated motions is often unsatisfying. Many generated motions contain self-intersections, i.e., body parts interpenetrating, which are strong artifacts that seriously impair the perceived motion quality. This observation highlights two things: 1) Recent human motion generation approaches do not focus enough on the physical plausibility of the generated motions, and 2) current evaluation metrics fail to capture the quality of the generated motions sufficiently.

We aim to avoid self-intersections in generated human motions by explicitly penalizing self-intersections while training human motion generation methods. In the related field of 3D human pose and shape estimation, some approaches [6, 31] implement such a loss based on triangular meshes of humans. However, they focus on optimizing the poses of a human for a single motion, while recent human motion generation methods use data batches with many motions. The runtime and memory consumption of these approaches does not scale to this scenario, prohibiting their use. Instead, we propose to approximate meshes used for training with a set of spheres. This approximation has several advantages. First, only a few hundred spheres are necessary to approximate a mesh with thousands of triangles. Thus, we can significantly reduce the memory cost of the geometry representation and the number of intersection checks between geometric primitives. Second, calculating if two spheres intersect is simpler than calculating triangle intersections. Therefore, we can use this sphere proxy to efficiently compute our novel self-intersection loss, which enables us to apply it to recent human motion generation methods. Our extensive experiments show a significant reduction in generated self-intersections while improving most other metrics. Our contributions can be summarized as follows:

- we introduce a novel self-intersection loss, based on a sphere approximation of human geometry, which reduces the memory cost by 83 % and the runtime by 98 % compared to previous approaches, making the self-intersection loss applicable in the first place,
- we show that our novel self-intersection loss reduces self-intersections in human motion generation by up to 49 % while improving other evaluation metrics,
- and we introduce a novel voxel-based metric measuring the severity of self-intersections to guide human motion generation towards improving perceived motion quality.

2 Related Work

Human Motion Generation Recent human motion generation methods use learning approaches on motion capture data [5, 14, 25, 56] to either learn the complete manifold of human motions [18, 37, 54] or condition the generation process on an action class [7, 32, 63], audio [1, 28], a motion prefix [13, 58], or text [8, 10, 14, 15, 33, 39, 49, 61, 62].

MotionDiffuse [63] estimates the noise of a noisy motion to iteratively obtain a clean motion [17, 44, 45]. TEMOS [33] uses a VAE [21] by aligning the continuous latent space of a text encoder and a motion encoder, and generates motions with a motion decoder. Motion latent diffusion (MLD) [8] applies diffusion models in the latent space of a VAE. Most recent approaches focus on the controllability of motion generation [2, 8, 32, 42, 57]. In contrast, we focus on the perceived motion quality by avoiding self-intersections in generated motions.

Human Geometry Representations Most commonly, human geometries are represented using triangular meshes as in the Skinned Multi-Person Linear (SMPL) model [23, 29, 31,

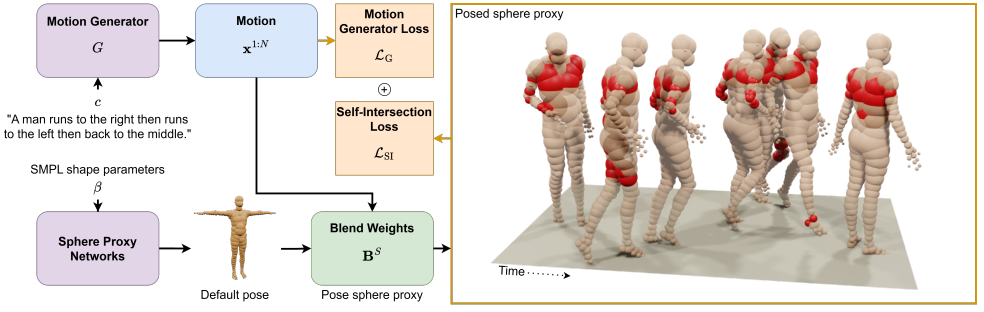

Figure 1: We make human motion generation self-intersection-aware by proposing a novel self-intersection loss. Given a condition embedding c, an arbitrary human motion generation method G generates a motion $\boldsymbol{x}^{1:N}$. Relying on SMPL shape parameters β, we obtain our sphere proxy in default pose and employ the blend weight matrix $\boldsymbol{B}^S$ to apply the generated poses to the sphere proxy. Our novel self-intersection loss $\mathcal{L}_{\text{SI}}$ is calculated on the sphere intersections of the posed sphere proxy and added to $\mathcal{L}_{\text{G}}$ of G. Red indicates intersecting spheres.

41]. Some approaches [9, 31, 52] calculate self-intersections for triangular meshes, but they have a high runtime, which can be improved at the cost of memory usage by using space partitioning methods [19, 31, 48]. Other representations include signed distance fields (SDF) [12, 30], 3D point clouds [11], or occupancy maps [26]. Finally, human meshes can be approximated by simple geometric primitives. SMPLify [6] uses a rough approximation with a set of capsules. Stoll et al. [46] use a sum of 3D Gaussians to represent human geometry. DualSDF [16] approximates arbitrary geometric shapes using spheres. We extend DualSDF by approximating human meshes with a set of spheres and attaching them to an underlying skeleton.

Physically Grounded Motion Generation Enhancing the physical plausibility of generated motions is a growing area of related research. PhysDiff [59] generates physically plausible motions by projecting intermediate noisy motions generated by motion diffusion models into a physics engine. Several approaches [9, 49] apply geometric losses based on the joints to the training of human motion generation models. HUMOS [51] uses a foot-sliding loss, ground penetration loss, and floating loss based on vertex locations. 3D human pose and shape estimation [22, 36, 47, 50] aims to recover the SMPL [23] pose and shape parameters given a single image. In this field, losses to enforce physical constraints are frequently used, like a pose prior to penalize unnatural bends of knees and elbows [6], and self-intersection losses [6, 27, 31]. We follow this line of work by explicitly penalizing self-intersections using a novel loss.

3 Methodology

We aim to generate physically plausible 3D human motions by explicitly penalizing self-intersections. Our approach is agnostic to the specific human motion generation method, and we explain the general setting in Sec. 3.1. We base our novel self-intersection loss on

a sphere approximation of the human geometry, which we call *sphere proxy* and describe it in Sec. 3.2. Fig. 1 shows an overview of our approach. The advantage of the sphere proxy is that self-intersections can be computed significantly more efficiently compared to using triangular meshes. Sec. 3.3 describes our novel *self-intersection loss* and how we integrate it into the model training.

3.1 Human Motion Generation

Human motion generation aims to generate natural and diverse human motions. Formally, a motion $\boldsymbol{x}^{1:N}$ is a discrete temporal sequence of length $N \in \mathbb{N}^+$ of individual poses $\boldsymbol{x}^n \in \mathbb{R}^{J \times D}$ with $J \in \mathbb{N}^+$ denoting the number of joints of the underlying skeleton, $D \in \mathbb{N}^+$ denoting the dimension of the pose representation, and $n \in \mathbb{N}^+$ denoting the temporal index. A pose $\boldsymbol{x}^n$ can be defined by joint locations, rotations, velocities, or a combination of them. Usually, the motion generation process is conditioned on the embedding of some real-world signal $c \in \mathbb{R}^L$, with $L \in \mathbb{N}^+$ denoting the dimension of the embedding space, like a text description or a 3D point cloud. However, unconditioned generation is also possible. The motion is generated by some generative model G and depends on the specific implementation.

3.2 Human Sphere Proxy

We follow DualSDF [16] to obtain a sphere approximation of human geometries. DualSDF uses a set of $S \in \mathbb{N}^+$ spheres $\mathbf{S} = \{(\boldsymbol{z}^i, r^i) | i = 1, ..., S\}$ with centers $\boldsymbol{z}^i \in \mathbb{R}^3$ and radii $r^i \in \mathbb{R}$ to approximate a 3D geometry X using SDFs. Given a point $\boldsymbol{p} \in \mathbb{R}^3$, the SDF specifies the distance of that point to the closest surface of the geometry. The sign encodes whether the point lies inside (negative) or outside (positive) the surface. Given a human mesh X, we sample $K \in \mathbb{N}^+$ 3D points $\boldsymbol{p}^k, k = 1, ..., K$, and corresponding SDF values $d_X^k \in \mathbb{R}$. For a set of spheres $\mathbf{S}$, the value of the SDF at point $\boldsymbol{p}$ is defined as the minimum over the SDF values of the individual spheres

$$d_{\mathbf{S}} = \min_{1 \leq i \leq S} d^i_{\text{sphere}}, \text{ with } \quad d^i_{\text{sphere}} = ||\boldsymbol{p} - \boldsymbol{z}^i||_2 - r^i. \tag{1}$$

DualSDF implements a neural network to predict the sphere parameters, following the framework of variational autodecoder (VAD) [30] by learning a Gaussian latent space. However, a learned latent space would require an optimization process to find the latent vector that corresponds to the sphere proxy of a given mesh. Instead, we use the SMPL [23] shape parameters $\beta \in \mathbb{R}^U$, $U \in \mathbb{N}^+$ denoting the number of SMPL shape parameters, as input to our neural network, because they have semantic meaning and they already approximately follow a Gaussian distribution. We use several loss terms to train the neural network. First of all, we approximate the sampled SDF values of the mesh X using the set of spheres $\mathbf{S}$

$$\mathcal{L}_{\text{SDF}} = \frac{1}{K} \sum_{k=1}^{K} \begin{cases} \max(d_S^k, 0) & d_X^k < 0, \\ |d_S^k - d_X^k| & d_X^k \geq 0. \end{cases} \tag{2}$$

For points $\boldsymbol{p}^k$ inside the 3D shape X, the loss is truncated to zero [16, 30]. Furthermore, we add an emptiness loss. We want to approximate the surface of the original mesh, so all spheres should be placed within that surface, which we achieve by checking that each sphere

contains at least one point $\boldsymbol{p}^k$ sampled from inside the mesh.

$$\mathcal{L}_{\text{emptiness}} = \frac{1}{S}\sum_{i=1}^{S}\begin{cases}\max\left(||\boldsymbol{p}^{\kappa}-\boldsymbol{z}^i||-r^i,0\right) & d_X^{\kappa}<0,\\ 0 & d_X^{\kappa}\geq 0,\end{cases} \tag{3}$$

where $\kappa = \arg\min_k(||\boldsymbol{p}^k - \boldsymbol{z}^i||)$ is the index of the closest sampled point to sphere center $\boldsymbol{z}^i$. Additionally, the spheres should be distributed equally within the boundaries of the mesh to approximate all body parts with the same level of detail, which we achieve with an intersection loss. If the distance between the centers $\boldsymbol{z}^i$ and $\boldsymbol{z}^{i'}$ of two spheres i and i' is smaller than the sum of their radii r^i and $r^{i'}$, the spheres are intersecting. The intersection distance is given by

$$b^{i,i'} = \max(r^i + r^{i'} - ||\boldsymbol{z}^i - \boldsymbol{z}^{i'}||, 0). \tag{4}$$

Minimizing the intersection distance $b^{i,i'}$ pushes the spheres apart from each other, filling the space governed by the boundaries of the mesh. Formally, the loss is given by

$$\mathcal{L}_{\text{IS}} = \frac{1}{S^2}\sum_{i=1}^{S}\sum_{i'=i+1}^{S} b^{i,i'}. \tag{5}$$

The overall loss to train the sphere proxy is defined by

$$\mathcal{L}_{\text{SP}} = \mathcal{L}_{\text{SDF}} + \lambda_{\text{emptiness}}\mathcal{L}_{\text{emptiness}} + \lambda_{\text{IS}}\mathcal{L}_{\text{IS}}, \tag{6}$$

where $\lambda_{\text{emptiness}}, \lambda_{\text{IS}} \in \mathbb{R}$ are hyperparameters.

We use SMPL meshes in default pose to train the sphere proxy and attach the spheres to the SMPL skeleton to apply a pose $\boldsymbol{x}^n$. While learning a posed sphere proxy is also possible, it would require substantially more training resources, as the training data would need to capture various poses sufficiently. In SMPL, joint locations are regressed, given the vertex locations. Instead, we train a simple neural network to predict them, given SMPL shape parameters using an L_2 loss on joint locations. To attach the spheres to the skeleton, we follow the standard linear blend skinning approach [24]. The sphere's movement is governed by a blend weight matrix $\boldsymbol{B}^S \in \mathbb{R}^{S\times J}$. Similarly, the SMPL model also has a blend weight matrix $\boldsymbol{B}^X \in \mathbb{R}^{V\times J}$, where V is the number of vertices of the mesh. To obtain $\boldsymbol{B}^S$, we calculate the $g \in \mathbb{N}^+$ nearest vertices of the given mesh to the surface of each sphere. The blend weights of a sphere are then simply the mean of the blend weights of the neighboring vertices. Finally, these blend weight matrices are calculated for all meshes in the training data, and subsequently, we take their mean to obtain one blend weight matrix $\boldsymbol{B}^S$ for the sphere proxy. The sphere centers $\boldsymbol{z}$ become dependent on a given pose $\boldsymbol{x}^n$, $\boldsymbol{z} = \boldsymbol{z}(\boldsymbol{x}^n)$. However, we omit this dependency in the following for ease of notation.

3.3 Preventing Self-Intersections

We use our sphere proxy to propose a novel self-intersection loss. At some point during the training of every human motion generation method G, a human motion $\boldsymbol{x}^{1:N}$ is generated. We apply each generated pose to the sphere proxy and calculate which spheres intersect. However, it is not ideal to calculate the intersections between all sphere pairs. Neighboring spheres will always intersect, while spheres representing the same body part will never intersect. We utilize a data-driven approach to determine the spheres that always intersect by recording all sphere intersections given the poses of a human motion dataset. Sphere pairs

intersecting in more than 90% of the poses can be seen as body model inaccuracies and are excluded from the loss calculation. We utilize the blend weight matrix $\boldsymbol{B}^S$ to determine the spheres belonging to the same body part by assigning each sphere to the joint for which the blend weight is the biggest. Sphere pairs assigned to the same joint are excluded from the loss calculation. We denote the remaining set of sphere pairs checked in the self-intersection loss as $\mathbf{W}$ and define the self-intersection loss using the intersection distance between sphere pairs

$$\mathcal{L}_{\text{SI}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{(i,i')\in\mathbf{W}}\left(b^{i,i'}\right)^2. \tag{7}$$

Note, that $b^{i,i'}$ is dependent on a given pose $\boldsymbol{x}^n$. The overall training loss for G becomes

$$\mathcal{L} = \mathcal{L}_{\text{G}} + \lambda_{\text{SI}}\mathcal{L}_{\text{SI}}, \tag{8}$$

where $\mathcal{L}_{\text{G}}$ is the training loss of G and $\lambda_{\text{SI}} \in \mathbb{R}$ is a hyperparameter.

4 Experiments

We evaluate our methods on the text-to-motion task on the datasets HumanML3D [14] and KIT-ML [35]. We use the motion representation proposed by Guo *et al.* [14] but also recover SMPL [23] joint rotations to pose our sphere proxy - details can be found in the supplementary material. KIT-ML follows a different skeletal structure than HumanML3D, but TEMOS [33] provides correspondences between the KIT-ML and SMPL joints while we interpolate missing joints. HumanML3D and KIT-ML represent motions using the skeleton of one target motion. We apply SMPLify [6] to each target motion to get one set of SMPL shape parameters for each dataset to obtain our sphere proxy.

We use the metrics *R Precision*, *Fréchet Inception Distance* (*FID*), *Multimodal distance*, *Diversity*, and *MultiModality* to evaluate our approach as commonly used in the literature [14]. In addition, we propose a novel metric, *Self-Intersect (SI)*, to measure the severity of self-intersections in the generated motions. *SI* approximates the mean self-intersection volume by generating a mesh for every generated pose, scaling each mesh to fit into a sphere with radius 1 m, subdividing the space within the sphere into voxels with edge length $v = 0.06$ cm, and determining if the voxel lies within a region of self-intersection. Values are reported in cubic centimeters. Full details on *SI* can be found in the supplementary material.

4.1 Motion Generation Methods

We integrate our novel self-intersection loss into the training of the recent human motion generation methods *human motion diffusion model (MDM)* [29] and *MoMask* [15], which we briefly explain in the following. We call our modified versions *SIA-MDM* and *SIA-MoMask*, with SIA being short for self-intersection-aware.

MDM follows the diffusion model framework by learning a reverse diffusion process. Given a noisy motion $\boldsymbol{x}_t^{1:N}$ at noise step $t \in \mathbb{N}^+$, MDM implements G with a Transformer [55] encoder-only architecture and is trained to predict the clean motion $\hat{\boldsymbol{x}}_0^{1:N}$, given the condition embedding c and the noise step t. During sampling, $\hat{\boldsymbol{x}}_0^{1:N}$ is passed through the forward diffusion process to obtain $\boldsymbol{x}_{t-1}^{1:N}$ and this procedure is iterated until $\boldsymbol{x}_0^{1:N}$ is obtained. To generate a motion, $\boldsymbol{x}_T^{1:N} \sim \mathcal{N}(0,\boldsymbol{I})$ is sampled from a Gaussian distribution and iteratively denoised, where T denotes the maximal noise step.

Dataset	Method	R Precision top 1↑	R Precision top 2↑	R Precision top 3↑	FID↓	Multimodal Dist↓	Diversity→	MultiModality↑	SI↓
HumanML3D	Real	$0.511^{\pm0.003}$	$0.703^{\pm0.003}$	$0.797^{\pm0.002}$	$0.002^{\pm0.000}$	$2.974^{\pm0.008}$	$9.503^{\pm0.065}$	-	$447^{\pm0}$
	MLD [8]	$0.481^{\pm0.003}$	$0.673^{\pm0.003}$	$0.772^{\pm0.002}$	$0.473^{\pm0.013}$	$3.196^{\pm0.010}$	$9.724^{\pm0.082}$	$2.413^{\pm0.079}$	-
	MotionDiffuse [58]	$0.491^{\pm0.001}$	$0.681^{\pm0.001}$	$0.782^{\pm0.001}$	$0.630^{\pm0.001}$	$3.113^{\pm0.001}$	$9.410^{\pm0.049}$	$1.553^{\pm0.042}$	-
	ReMoDiffuse [58]	$0.510^{\pm0.005}$	$0.698^{\pm0.006}$	$0.795^{\pm0.004}$	$0.103^{\pm0.004}$	$2.974^{\pm0.016}$	$9.018^{\pm0.075}$	$1.795^{\pm0.043}$	-
	MDM [58]	$0.418^{\pm0.005}$	$0.604^{\pm0.005}$	$0.707^{\pm0.004}$	$0.489^{\pm0.025}$	$3.631^{\pm0.023}$	$\mathbf{9.449^{\pm0.066}}$	$\mathbf{2.973^{\pm0.111}}$	$619^{\pm11}$
	MoMask [58]	$\underline{0.521}^{\pm0.002}$	$\underline{0.713}^{\pm0.002}$	$\underline{0.807}^{\pm0.002}$	$\mathbf{0.045^{\pm0.002}}$	$\underline{2.958}^{\pm0.008}$	$9.624^{\pm0.080}$	$1.241^{\pm0.040}$	$\underline{316}^{\pm02}$
	SIA-MDM (Ours)	$0.435^{\pm0.005}$	$0.628^{\pm0.006}$	$0.731^{\pm0.006}$	$0.265^{\pm0.024}$	$3.462^{\pm0.026}$	$\underline{9.568}^{\pm0.086}$	$\underline{2.893}^{\pm0.075}$	$382^{\pm09}$
	SIA-MoMask (Ours)	$\mathbf{0.525^{\pm0.003}}$	$\mathbf{0.717^{\pm0.003}}$	$\mathbf{0.813^{\pm0.002}}$	$\underline{0.068}^{\pm0.002}$	$\mathbf{2.933^{\pm0.006}}$	$9.691^{\pm0.092}$	$1.198^{\pm0.041}$	$\mathbf{290^{\pm02}}$
KIT-ML	Real	$0.424^{\pm0.005}$	$0.649^{\pm0.006}$	$0.799^{\pm0.006}$	$0.031^{\pm0.004}$	$2.788^{\pm0.012}$	$11.080^{\pm0.097}$	-	$778^{\pm0}$
	MLD [8]	$0.390^{\pm0.008}$	$0.609^{\pm0.008}$	$0.734^{\pm0.007}$	$0.404^{\pm0.027}$	$3.204^{\pm0.027}$	$10.800^{\pm0.117}$	$\underline{2.192}^{\pm0.071}$	-
	MotionDiffuse [58]	$0.417^{\pm0.004}$	$0.621^{\pm0.004}$	$0.739^{\pm0.004}$	$1.954^{\pm0.062}$	$2.958^{\pm0.005}$	$\mathbf{11.100^{\pm0.143}}$	$0.730^{\pm0.013}$	-
	ReMoDiffuse [58]	$0.427^{\pm0.014}$	$\underline{0.641}^{\pm0.004}$	$0.765^{\pm0.055}$	$\mathbf{0.155^{\pm0.006}}$	$2.814^{\pm0.012}$	$10.800^{\pm0.105}$	$1.239^{\pm0.028}$	-
	MDM [58]	$0.404^{\pm0.005}$	$0.607^{\pm0.004}$	$0.731^{\pm0.004}$	$0.513^{\pm0.046}$	$3.096^{\pm0.024}$	$10.732^{\pm0.103}$	$1.806^{\pm0.176}$	$597^{\pm07}$
	MoMask [58]	$\underline{0.433}^{\pm0.007}$	$\mathbf{0.656^{\pm0.005}}$	$\mathbf{0.781^{\pm0.005}}$	$\underline{0.204}^{\pm0.011}$	$\underline{2.779}^{\pm0.022}$	$10.780^{\pm0.080}$	$1.131^{\pm0.043}$	$930^{\pm07}$
	SIA-MDM (Ours)	$0.416^{\pm0.005}$	$0.635^{\pm0.006}$	$0.755^{\pm0.006}$	$0.321^{\pm0.021}$	$2.981^{\pm0.024}$	$\underline{10.922}^{\pm0.107}$	$\mathbf{2.234^{\pm0.080}}$	$\mathbf{441^{\pm05}}$
	SIA-MoMask (Ours)	$\mathbf{0.437^{\pm0.005}}$	$\mathbf{0.656^{\pm0.006}}$	$\underline{0.776}^{\pm0.005}$	$0.316^{\pm0.017}$	$\mathbf{2.722^{\pm0.017}}$	$10.709^{\pm0.120}$	$1.111^{\pm0.034}$	$\underline{472}^{\pm03}$

Table 1: Text-to-motion results on HumanML3D [14] and KIT-ML [35]. All experiments are repeated for 20 random seeds. ± indicates the 95% confidence interval. **Bold** indicates the best result, while underscore indicates the second best. → indicates that closer to 'Real' is better. Our self-intersection-aware methods significantly improve the respective baselines in most metrics.

MoMask follows the vector quantized VAE (VQ-VAE) [53] framework and implements G with three components. First, a residual VQ-VAE is trained, which encodes a motion $\boldsymbol{x}^{1:N}$ using a hierarchy of quantization layers of discrete motion tokens corresponding to entries of a learned codebook. Second, a masked transformer is trained to generate motion tokens of the base layer of the quantization hierarchy given the condition embedding c. Third, a residual transformer is trained to generate the motion tokens of the remaining quantization layers given c. During sampling, the motion tokens of the quantization hierarchy are progressively generated using the masked and residual transformer given c. Subsequently, the motion tokens are decoded using the decoder of the residual VQ-VAE.

4.2 Implementation Details

Sphere Proxy The sphere proxy is based on the gender-neutral SMPL-H [41] model without the hand joints and SMPL shape parameters of dimension $U = 10$ in a range between -5.0 and 5.0, following Guo et al. [14]. We randomly sample 8,000 sets of SMPL shape parameters and use the corresponding joint locations to train the joint regressor. It is trained for 300 epochs with an initial learning rate of 1e-4, which is decayed by 0.1 every 100 epochs. We randomly sample 800 sets of SMPL shape parameters to train the sphere regressor. For each corresponding mesh, we sample $K = 750,000$ points with associated SDF values, of which 250,000 points are sampled in a sphere around the mesh, and 500,000 points are sampled closely to the surface. The sphere regressor is trained for 2,800 epochs with an initial learning rate of 5e-4, which is decayed by 0.5 every 700 epochs. Each batch contains 16,384 SDF samples per mesh, of which 10% are sphere samples, and of the remaining points, 50% correspond to hands and feet, as these body parts have more details. We empirically set $\lambda_{\text{emptiness}} = 10$ and $\lambda_{\text{IS}} = 0.1$. Both models are trained using the Adam [20] optimizer with a batch size of 64. The architectures of both models are shown in the supplementary material. We use the mean of the blend weights of the $g = 8$ nearest neighbor vertices and only keep the four most significant values following SMPL [23]. Our sphere proxy contains $S = 192$ spheres unless stated otherwise. For the intersection reduction, we use the poses of the training split of the HumanML3D dataset.

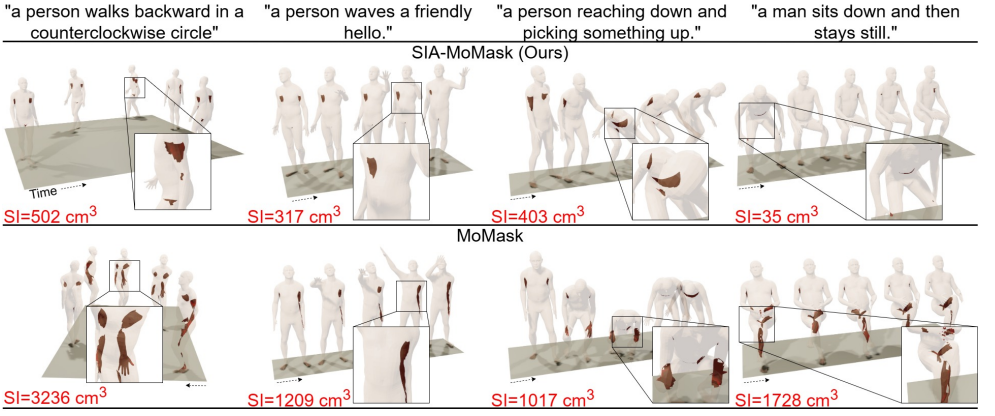

Figure 2: **Visual Comparison** between motions generated by SIA-MoMask (Ours) and MoMask [15] given text prompts from the HumanML3D [14] test set. Red patches indicate self-intersections. SI indicates the mean self-intersection volume. SIA-MoMask generates significantly fewer self-intersections while semantically following the textual description.

Motion Generation Methods We run our experiments on SIA-MDM and SIA-MoMask using the parameters given in the available code of MDM and MoMask, respectively. SIA-MDM is trained for 600,000 steps with $\lambda_{\text{IS}} = 0.01$ on HumanML3D and for 200,000 steps with $\lambda_{\text{SI}} = 0.0001$ on KIT-ML. SIA-MoMask integrates our novel loss into the residual VQ-VAE training with $\lambda_{\text{SI}} = 0.01$ for HumanML3D, and into the training of all components with $\lambda_{\text{SI}} = 0.000001$ for KIT-ML. Text conditions are embedded using CLIP [38].

4.3 Results

We compare the results obtained with SIA-MDM and SIA-MoMask to the results of MDM[1] and MoMask in Tab. 1. Both self-intersection-aware methods generate significantly fewer self-intersections than the respective baselines, emphasizing the effectiveness and generality of our novel loss. Additionally, most other metrics improve, highlighting the benefit of focusing on the physical plausibility of generated motions. Furthermore, we note the substantial amount of self-intersections in the ground truth data, the origin of which we discuss in the supplementary material. Our novel self-intersection loss facilitates the generation of motions containing few self-intersections, even if they are present in the ground truth data. However, we acknowledge that MDM and MoMask also generate fewer self-intersections than the ground truth motions on KIT-ML and HumanML3D, respectively, which we attribute to the desired diversity of motion generation, thus deviating from the ground truth. Nevertheless, focusing solely on optimizing FID, often associated with motion quality, does not result in the most realistic motions. With FID approaching the ground truth data, the amount of generated self-intersections is also expected to approach the ground truth self-intersections, highlighting the limitation of the FID metric. Therefore, evaluating FID together with SI yields a better judgment of motion quality. Fig. 2 and the supplementary material show mo-

[1]The evaluation script of the original publication of MDM contained errors. All results are obtained following bug fixes detailed at https://github.com/GuyTevet/motion-diffusion-model/issues/182

tions generated by MoMask and SIA-MoMask given text prompts of the HumanML3D test set, which confirms the improved motion quality compared to the baseline models.

4.4 Ablation Studies

We ablate design choices on the HumanML3D [14] dataset using SIA-MDM and focus on the metrics *FID*, *MultiModality*, and *SI*. All results are shown in Tab. 2.

Method	FID↓	MultiModality↑	SI↓
No intersection reduction	$0.841^{\pm 0.058}$	$3.083^{\pm 0.067}$	$715^{\pm 12}$
128 spheres	$0.411^{\pm 0.056}$	$2.711^{\pm 0.059}$	$306^{\pm 11}$
256 spheres	$0.301^{\pm 0.038}$	$2.754^{\pm 0.061}$	$331^{\pm 05}$
SIA-MDM (Ours)	$0.265^{\pm 0.024}$	$2.893^{\pm 0.075}$	$382^{\pm 09}$
MDM [49]	$0.489^{\pm 0.025}$	$2.973^{\pm 0.111}$	$619^{\pm 11}$

Table 2: Ablation studies on HumanML3D [14] using SIA-MDM. All experiments are repeated for 20 random seeds. ± indicates the 95% confidence interval.

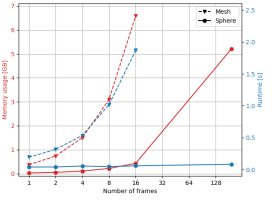

Figure 3: Comparison of the computational efficiency of the self-intersection loss between meshes and the sphere proxy.

Influence of the self-intersection method In Fig. 3, we compare the memory usage and runtime of our novel loss to the implementation of SMPLify-X [51] using meshes by applying each loss to 1, 2, 4, 8, and 16 frames of each motion. Memory usage and runtime increase dramatically with the number of frames for the mesh-based approach, prohibiting its use in human motion generation. In contrast, the runtime of the sphere proxy stays almost constant while the memory consumption is reasonable, even when using all frames.

Influence of intersection reduction We use all sphere pairs to calculate our self-intersection loss and compare it to our sphere pair reduction. Using all sphere pairs results in worse performance than using the self-intersection reduction and no self-intersection loss at all. We believe the additional self-intersections during the loss computation result in a strong but unclear gradient signal hindering model optimization. Additionally, penalizing realistic poses that contain self-intersections due to body model inaccuracies might impair model performance. This result highlights the importance of intersection reduction.

Influence of number of spheres We compare our sphere proxy to versions using 128 and 256 spheres. All versions improve MDM in *FID* and *SI*. However, 192 spheres yield the best trade-off between *FID* and *SI*; hence, we use this version in our experiments.

Evaluation of the sphere proxy We randomly sample 150 sets of SMPL shape parameters and use the corresponding meshes as a test set for our sphere proxy. We compute the SDF value of each mesh vertex for the corresponding sphere proxy and take the mean of the absolute SDF values for all vertices. The mean distance per vertex is approximately 0.6 cm, verifying the sphere proxy as a valid approximation of the surface of human meshes. The supplementary material provides a thorough analysis of the sphere proxy.

5 Conclusion

This paper investigates the problem of self-intersections in human motion generation, and we show that even state-of-the-art human motion generation approaches suffer from them, limiting the perceived motion quality. To mitigate this problem, we propose a sphere proxy of human geometry and show computational superiority in calculating self-intersections regarding runtime and memory usage compared to triangular meshes. Integrating our novel self-intersection loss into the training of MDM [49] and MoMask [15] significantly reduces self-intersections in generated motions while improving other evaluation metrics and the perceived motion quality, as we show with visual examples. Future research can use our sphere proxy to model contact between humans interacting with a scene or other humans. Finally, we acknowledge that the sphere proxy is only an approximation of human geometry, and some unrealistic self-intersections are still generated. Future research could further improve the sphere proxy, e.g., by incorporating hand and finger motions.

Acknowledgement Juergen Gall has been supported by the ERC Consolidator Grant FORHUE (101044724).

References

[1] Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *ACM TOG*, 42 (4):1–20, 2023.

[2] Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol. Teach: Temporal action compositions for 3d humans. In *3DV*, pages 414–423, Prague, Czech Republic, 2022.

[3] Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. Sinc: Spatial composition of 3d human motions for simultaneous action generation. In *ICCV*, pages 9984–9995, Paris, France, 2023.

[4] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *ECCV*, pages 640–653, Firenze, Italy, 2012.

[5] Bharat Lal Bhatnagar, Xianghui Xie, Ilya A Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *CVPR*, pages 15935–15946, New Orleans, LA, USA, 2022.

[6] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, pages 561–578, Amsterdam, The Netherlands, 2016.

[7] Pablo Cervantes, Yusuke Sekikawa, Ikuro Sato, and Koichi Shinoda. Implicit neural representations for variable length human motion generation. In *ECCV*, pages 356–372, Tel Aviv, Israel, 2022.

[8] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *CVPR*, pages 18000–18010, Vancouver, Canada, 2023.

[9] Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *CVPR*, pages 9760–9770, Vancouver, Canada, 2023.

[10] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. In *ECCV*, Milan, Italy, 2024.

[11] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, pages 605–613, Honolulu, HI, USA, 2017.

[12] Sarah F Frisken and Ronald N Perry. Designing with distance fields. *ACM SIGGRAPH 2006 Courses*, pages 60–66, 2006.

[13] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. Imos: Intent-driven full-body motion synthesis for human-object interactions. In *Comput. Graph. Forum*, volume 42, pages 1–12, 2023.

[14] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, pages 5152–5161, New Orleans, LA, USA, 2022.

[15] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *CVPR*, pages 1900–1910, Seattle, WA, USA, 2024.

[16] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge Belongie. Dualsdf: Semantic shape manipulation using a two-level representation. In *CVPR*, pages 7631–7641, Virtual, 2020.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, pages 6840–6851, Virtual, 2020.

[18] Roy Kapon, Guy Tevet, Daniel Cohen-Or, and Amit H. Bermano. Mas: Multi-view ancestral sampling for 3d motion generation using 2d diffusion. In *CVPR*, pages 1965–1974, Seattle, WA, USA, 2024.

[19] Tero Karras. Maximizing parallelism in the construction of bvhs, octrees, and k-d trees. In *EGGH-HPG*, volume 4, pages 33–37, Paris, France, 2012.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, volume 3, 2015.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[22] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, pages 5253–5263, Virtual, 2020.

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

[24] Thalmann Magnenat, Richard Laperrière, and Daniel Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *GI*, pages 26–33, Toronto, Canada, 1988.

[25] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, pages 5441–5450, Seoul, Republic of Korea, 2019.

[26] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, pages 4460–4470, Long Beach, CA, USA, 2019.

[27] Marko Mihajlovic, Shunsuke Saito, Aayush Bansal, Michael Zollhoefer, and Siyu Tang. Coap: Compositional articulated occupancy of people. In *CVPR*, pages 13201–13210, New Orleans, LA, USA, 2022.

[28] Muhammad Hamza Mughal, Rishabh Dabral, Ikhsanul Habibie, Lucia Donatelli, Marc Habermann, and Christian Theobalt. Convofusion: Multi-modal conversational diffusion for co-speech gesture synthesis. In *CVPR*, pages 1388–1398, Seattle, WA, USA, 2024.

[29] Ahmed A. A. Osman, Timo Bolkart, and Michael J. Black. STAR: Sparse trained articulated human body regressor. In *ECCV*, volume 6, pages 598–613, Glasgow, UK, 2020.

[30] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, Long Beach, CA, USA, 2019.

[31] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, pages 10975–10985, Long Beach, CA, USA, 2019.

[32] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *ICCV*, pages 10985–10995, Virtual, 2021.

[33] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *ECCV*, pages 480–497, Tel Aviv, Israel, 2022.

[34] Mathis Petrovich, Or Litany, Umar Iqbal, Michael J. Black, Gül Varol, Xue Bin Peng, and Davis Rempe. Multi-track timeline control for text-driven 3d human motion generation. In *CVPRW*, pages 1911–1921, Seattle, WA, USA, 2024.

[35] Matthias Plappert, Christian Mandery, and Tamim Asfour. The KIT motion-language dataset. *Big Data*, 4(4):236–252, 2016.

[36] Zhongwei Qiu, Qiansheng Yang, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Chang Xu, Dongmei Fu, and Jingdong Wang. Psvt: End-to-end multi-person 3d pose and shape estimation with progressive video transformers. In *CVPR*, pages 21254–21263, Vancouver, Canada, 2023.

[37] Sigal Raab, Inbal Leibovitch, Peizhuo Li, Kfir Aberman, Olga Sorkine-Hornung, and Daniel Cohen-Or. Modi: Unconditional motion synthesis from diverse data. In *CVPR*, pages 13873–13883, Vancouver, Canada, 2023.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, volume 139, pages 8748–8763, Virtual, 2021.

[39] Zeping Ren, Shaoli Huang, and Xiu Li. Realistic human motion generation with cross-diffusion models. In *ECCV*, Milan, Italy, 2024.

[40] Katja Rogers, Sukran Karaosmanoglu, Maximilian Altmeyer, Ally Suarez, and Lennart E Nacke. Much realistic, such wow! a systematic literature review of realism in digital games. In *CHI*, pages 1–21, New Orleans, LA, USA, 2022.

[41] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM TOG*, 36(6):245:1–245:17, 2017.

[42] Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a generative prior. In *ICLR*, volume 12, Vienna, Austria, 2024.

[43] Aayam Shrestha, Pan Liu, German Ros, Kai Yuan, and Alan Fern. Generating physically realistic and directable human motions from multi-modal inputs. In *ECCV*, pages 1–17, Milan, Italy, 2024.

[44] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, volume 32, pages 2256–2265, 2015.

[45] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *NeurIPS*, volume 33, pages 12438–12448, Virtual, 2020.

[46] Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *ICCV*, pages 951–958, Barcelona, Spain, 2011.

[47] Yu Sun, Qian Bao, Wu Liu, Tao Mei, and Michael J Black. Trace: 5d temporal regression of avatars with dynamic cameras in 3d environments. In *CVPR*, pages 8856–8866, Vancouver, Canada, 2023.

[48] Matthias Teschner, Stefan Kimmerle, Bruno Heidelberger, Gabriel Zachmann, Laks Raghupathi, Arnulph Fuhrmann, M-P Cani, François Faure, Nadia Magnenat-Thalmann, Wolfgang Strasser, et al. Collision detection for deformable objects. In *Comput. Graph. Forum*, volume 24, pages 61–81, 2005.

[49] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *ICLR*, volume 11, Kigali, Rwanda, 2023.

[50] Shashank Tripathi, Lea Müller, Chun-Hao P Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. 3d human pose estimation via intuitive physics. In *CVPR*, pages 4713–4725, Vancouver, Canada, 2023.

[51] Shashank Tripathi, Omid Taheri, Christoph Lassner, Michael J Black, Daniel Holden, and Carsten Stoll. Humos: Human motion model conditioned on body shape. In *ECCV*, pages 133–152, Milan, Italy, 2024.

[52] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 118:172–193, 2016.

[53] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, volume 30, page 6309–6318, Long Beach, CA, USA, 2017.

[54] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, pages 109–117, Honolulu, HI, USA, 2017.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, pages 6000–6010, 2017.

[56] Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Humanise: Language-conditioned human motion generation in 3d scenes. *NeurIPS*, 35: 14959–14971, 2022.

[57] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *ICLR*, volume 12, Vienna, Austria, 2024.

[58] Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *ICCV*, pages 14928–14940, Paris, France, 2023.

[59] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *ICCV*, pages 16010–16021, Paris, France, 2023.

[60] Amir Zadeh, Yao-Chong Lim, Paul Pu Liang, and Louis-Philippe Morency. Variational auto-decoder: A method for neural generative modeling from incomplete data. *arXiv preprint arXiv:1903.00840*, 2019.

[61] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *CVPR*, pages 14730–14740, Vancouver, Canada, 2023.

[62] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *ICCV*, pages 364–373, Paris, France, 2023.

[63] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE TPAMI*, 46(6):4115–4128, 2024.

[64] Mingyuan Zhang, Daisheng Jin, Chenyang Gu, Fangzhou Hong, Zhongang Cai, Jingfang Huang, Chongzhi Zhang, Xinying Guo, Lei Yang, Ying He, and Ziwei Liu. Large motion model for unified multi-modal motion generation. In *ECCV*, pages 397–421, Milan, Italy, 2024.