

OmniMotionGPT: Animal Motion Generation with Limited Data

Zhangsihao Yang^{1*}, Mingyuan Zhou², Mnegyi Shan³, Bingbing Wen³, Ziwei Xuan²
 Mitch Hill², Junjie Bai², Guo-Jun Qi^{2,4}, Yalin Wang¹

¹Arizona State University, USA

²OPPO Seattle Research Center, USA

³University of Washington, USA

⁴Westlake University, China

Abstract

Our paper aims to generate diverse and realistic animal motion sequences from textual descriptions, without a large-scale animal text-motion dataset. While the task of text-driven human motion synthesis is already extensively studied and benchmarked, it remains challenging to transfer this success to other skeleton structures with limited data. In this work, we design a model architecture that imitates Generative Pretraining Transformer (GPT), utilizing prior knowledge learned from human data to the animal domain. We jointly train motion autoencoders for both animal and human motions and at the same time optimize through the similarity scores among human motion encoding, animal motion encoding, and text CLIP embedding. Presenting the first solution to this problem, we are able to generate animal motions with high diversity and fidelity, quantitatively and qualitatively outperforming the results of training human motion generation baselines on animal data. Additionally, we introduce AnimalML3D, the first text-animal motion dataset with 1240 animation sequences spanning 36 different animal identities. We hope this dataset would mediate the data scarcity problem in text-driven animal motion generation, providing a new playground for the research community.

1. Introduction

Computational modeling of 3D motions is an important topic with a wide range of applications, including robotics, virtual/mixed/augmented reality, gaming, and visual media. Traditional methods for obtaining computational models of motions rely on human artists who use their observations of the real world to animate 3D assets [27], or extensive motion capture process [34]. This process requires great effort and skill from artists or an expensive and time-consuming capture procedure. Recent advances in generative modeling have led to breakthrough success for synthesizing realistic human motions using natural language textual descriptions

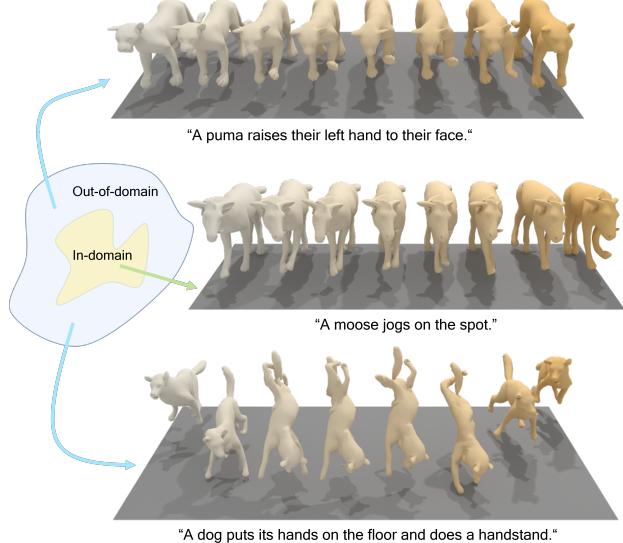


Figure 1. Visualization of in-domain and out-of-domain motion generation from textual descriptions. Our model generates animal motion ranging from conventional movements to complex, out-of-domain behaviors. The in-domain motion semantic latent space, highlighted by the yellow region, encapsulates common animal movements described in textual data. The out-of-domain latent space, delineated by the blue region, includes complex motions that are less frequently associated with animal behaviors, such as performing a handstand. The blue and green arrows denote our motion generation process from out-of-domain and in-domain prompts.

[13, 14, 20, 36, 42, 43]. Text-driven motion generation has the potential to greatly increase the efficiency and accessibility of motion animation. Despite the success of motion generation in the domain of human motions, significant obstacles remain which prevent similar techniques from being used to generate other kinds of motions.

In this work, we showcase a method to tackle the difficult problem of animal motion generation from text descriptions. Text-driven animal motion generation is much less studied than human motion generation mainly due to dataset availability issues. Animal motion data in the research commu-

*Work done while interning at OPPO

nity is very limited and not available at a comparable scale as human motion datasets [13, 29, 37]. Specifically, there is no paired text-motion dataset for animal motion sequences at all, akin to HumanML3D [13] in the human motion domain. This fundamental data scarcity problem motivates us to leverage information from human motions to supplement significantly smaller animal motion datasets.

To incorporate human motion data when training an animal motion model, we must address several key problems. Animals have different motion representations than humans, notably in terms of the number of joints and joint definitions [64]. This makes it hard to directly transfer the knowledge from human motion models to animal ones. Moreover, human motion generators do not care too much about the skeleton information beyond joints [13, 43], while for animals, the skeleton offsets for different species could be different even if they share the same skeleton topology [64]. Furthermore, animals perform much less diverse motion patterns than human beings in reality, even though animals are capable of mimicking most motion patterns of human beings. It is straightforward to collect a motion of hand clapping for human, but requires more effort for animals either in reality, which requires animal training, or in virtual, which requires the artists’ manual calibration of the animal arm movements.

To address the aforementioned challenges, we propose an architecture to transfer the knowledge from the human motion domain to enrich the generation of both in-distribution and out-of-distribution animal motions. We first design a transformer-based [46] motion encoder that projects different skeletal motions to a primal joint’s latent space which enables the translation between two different motion domains. By registering the motion both on a common textural space, we are able to connect human motion modality, language space, and animal motion modality, with CLIP [39] similarity loss. We design three loss functions, latent consistency, CLIP similarity, and end-effector loss, to regularize the transformation of the latent feature from human motion to animal motion generation model. We additionally create the first animal language-motion dataset AnimalML3D for training and evaluation of our method. We generate skeleton motions and annotate textural descriptions for the existing DeformingThings4D [27] dataset that only contains animal motion mesh sequences.

Our contribution can be summarized as follows:

- We present OmniMotionGPT, a new framework that trains on sparse animal motion data and generates diverse motions from complex texts by transferring learned human motion knowledge.
- We propose a new method to train motion autoencoders for both animal and human motion by aligning their semantic representation. Extensive experiments demonstrate that our method significantly outperforms existing methods both qualitatively and quantitatively.

- We introduce AnimalML3D, the first dataset pairing text descriptions with 3D animal motions, which consists of 3720 human-written textual descriptions accompanying 1240 motions of 36 different animal identities. We hope our new dataset can provide a solid new playground for researchers interested in the animal text-motion task.

2. Related work

Animal Representations. Several models have been developed to represent animal motion, including LASSIE [57], SMAL [64], and LASR [54–56]. SMAL and its enhanced variant SMALR [65], with more expressive features, extend of the widely-used human motion representation SMPL [30], catering to the motion representations of five animal categories. LASR is introduced following SMAL to accommodate a broader range of animal species. LASSIE, along with its subsequent iteration Hi-LASSIE [58], employs a neural field around detected bones in images, but they are used more often in image or video reconstruction instead of motion generation. Our approach utilizes SMAL as the core representation due to its explicit skeletal structure and the semantic meaning provided for each joint. Additionally, the compatibility of SMAL with the standard human motion representation SMPL [30], facilitates the knowledge transfer from human to animal motion distribution, which is crucial to our research.

Human Motion Synthesis. Human motion synthesis aims to generate diverse and natural 3D human motion. One major line of research focuses on motion generation based on existing motion frames. For example, predicting future motion from given frames [4, 7, 10, 17, 33, 51], motion in-betweening [9, 15, 16, 41], and motion generation from a simple sequence [26]. Traditionally this has been modeled as a one-to-one relationship until recent generative models handle the stochastic nature of motion space and greatly increase the result diversity. Another line of work incorporates multi-modal inputs as conditioning signals, including action label [12, 35, 49], music and audio [19, 24, 44], scene geometry [48, 50], object interaction [23], and text [13, 14, 20, 42, 61]. Despite the amount of research effort in human motion generation, it remains an open problem whether such approaches could be migrated to other skeleton structures like animals, mainly due to the lack of datasets with comparable scales.

Text-driven Human Motion Generation. With the development of pre-trained language models, text-driven human motion synthesis becomes one of the most important conditional motion generation tasks. The goal is to synthesize realistic, diverse 3D human motion sequences that align semantically with given textual descriptions. MotionCLIP [42] uses auto-encoder structures to learn a joint embedding of language and pose and thus generate animations. TEMOS [36] and T2M [13] leverage a VAE structure to map text

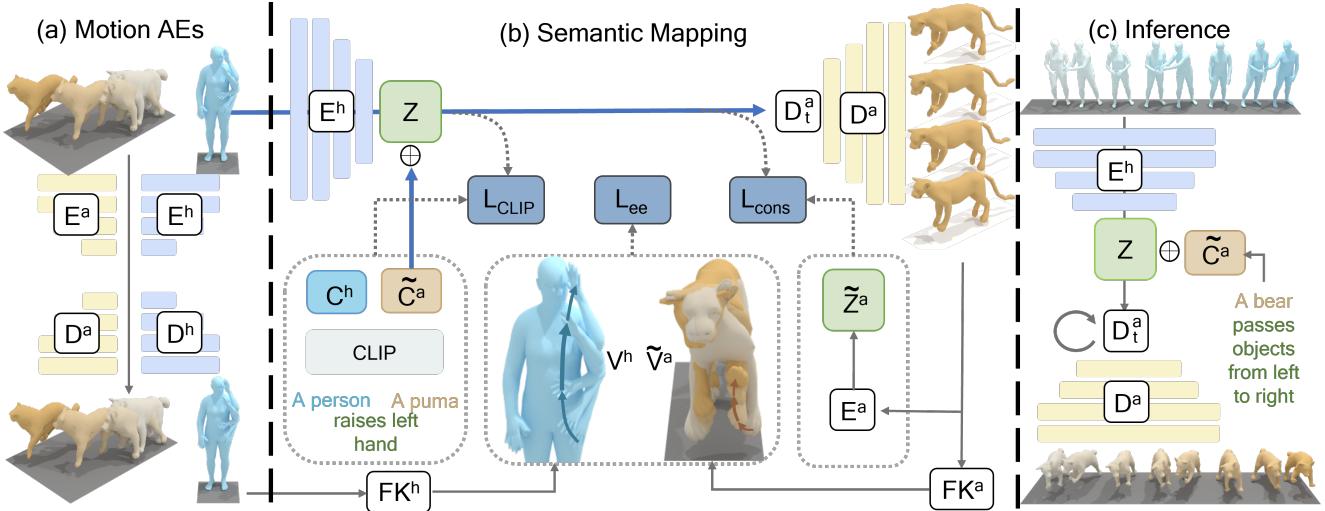


Figure 2. **The architecture of our training and inference stages.** We train part (a) and part (b) at the same time. In (a), we train two motion autoencoders simultaneously, each within their domain, leveraging primal joints to maintain dimensional coherence in the latent space. Details on the structure and loss functions can be found in Section 3.1. In (b), human motion is fed into the human motion encoder E^h to produce a semantic-aware, subject-invariant latent code Z . The CLIP feature of the subject-translated sentence and Z are concatenated together and passed into the animal text decoder D_t^a and motion decoder D^a . We introduce three losses to regularize the generated animal motions. CLIP similarity loss L_{CLIP} extracts subject-invariant latent features. Latent consistency loss L_{cons} pushes the generated animal motion to be closer to the subject-invariant motion feature Z . End-effectors loss L_{ee} injects human motion velocity information into animals. During inference in (c), we generate animal motions based on human motion sequences sampled from generative models. Details on the architecture, loss functions, and inference process are elaborated in Section 3.2.

into a normal distribution in the latent space. Later work TM2T [14], MotionGPT [20], and T2MGPT [60] learn to encode the motion sequences as discrete, quantized text/motion tokens in a fixed size codebook, and generate through an auto-regressive process. A parallel line of work utilizes diffusion model [18] with text embedding as a condition. MDM [43] and MotionDiffuse[61] apply diffusion model to text-motion dataset through a transformer structure. Re-MoDiffuse [62] further integrates a retrieval mechanism to refine the denoising process. MLD [8] achieves better results and is two orders of magnitude faster than previous diffusion models by using the latent diffusion model. PhysDiff [59] further incorporates physical simulation to enforce realistic human motion rules. Nevertheless, the nature of diffusion models and VAEs requires a huge amount of data during training, and thus won't directly apply to animal motions.

Motion Retargeting. Many works in motion retargeting focus on transferring motion data between entities with topologically equivalent skeletons, particularly in human [2, 11, 25] and animal contexts [31]. Some other works retarget human motion data to non-humanoid characters; these methods typically require humans to mimic animal motions [40] or necessitate the creation of a paired dataset for motion transfer [1, 53]. Skeleton-free retargeting [22, 28, 47] is another emerging approach to retargeting 3D objects. Our task differs from traditional retargeting as we directly gener-

ate motions from text descriptions.

Motion and Pose Datasets. HumanML3D [13] is built upon HumanAct12[12] and AMASS[32], containing a broad range of human actions such as daily activities. Similarly, KIT language-motion dataset [37] contains 3911 motions and 6278 natural language annotations. Motion-X [29] is another large-scale 3D expressive whole-body motion dataset paired with textual annotations. On the other hand, for animals, we have Animal3D [52] which estimates static poses from animal images but doesn't contain dynamic motion sequences. DeformingThings4D [27] is perhaps the only animal motion dataset, but it's built for depth and optical flow estimation and therefore doesn't come with textual annotations and has a limited amount of motion sequences. To the best of our knowledge, there are no public animal text-motion datasets before us.

3. Method

Our goal is to generate high-quality animal motions that are consistent with text descriptions. The overall training framework consists of two parts optimized simultaneously: motion autoencoder training for animals and humans, and joint training for knowledge transfer, as illustrated in Figure 2. Section 3.1 explains the separate training procedure of human motion and animal motion autoencoders. Section 3.2 describes the joint training mechanism that aligns human

and animal motion spaces, along with integrating the text semantic latent space. It also illustrates how this mechanism decodes human motion embedding to generate animal motion in the inference stage.

3.1. Integrating Joint and Text Awareness in Motion Autoencoders

Motion Representation. In object motion representation, the kinematics can be abstracted through a skeletal model. This skeletal structure is conceptualized as a tree graph, with joints as nodes and armatures as edges as defined in [2]. The number of joints J is consistently one greater than the number of armatures A . We represent skeletal motion using a static component $\mathcal{S} \in \mathbb{R}^{(J-1) \times S}$, with S as static features' dimensionality, usually set as a 3D vector ($S = 3$). Beyond this static representation, our dynamic component comprises three parts: global rotation $\mathcal{R} \in \mathbb{R}^{T \times Q}$, global translation $\mathcal{T} \in \mathbb{R}^{T \times 3}$, and joint rotations $\mathcal{Q} \in \mathbb{R}^{T \times (J-1) \times Q}$ relative to their parents, excluding the root joint. We select $Q = 6$, following [63], to represent the rotations of each joint and global root. After augmenting the global translation to a Q -dimensional vector by padding zeroes to it, the dynamic component can be represented as $\mathcal{D} \in \mathbb{R}^{T \times (J+1) \times Q}$ by concatenating \mathcal{R} , \mathcal{T} , and \mathcal{Q} . \mathcal{D} is a sequence of poses $\mathcal{P}_t \in \mathbb{R}^{(J+1) \times Q}$ at frame t . Primal joints are the joints that have a degree not equal to 2 in the skeletal graph. Intersecting primal joints is the intersection of primal joints between skeleton graphs.

Joint-aware Motion Autoencoder. Figure 3 shows an overview of our autoencoder model. Our model begins with a transformer encoder extracting joint-level features from each pose. The input is the concatenation of poses $\mathcal{P}_t \in \mathbb{R}^{(J+1) \times Q}$ and the corresponding, zero-padded static offsets $\mathcal{S}' \in \mathbb{R}^{(J+1) \times S}$. The shared joint transformer encoder generates a feature $\mathcal{F}_j \in \mathbb{R}^{(J+1) \times f_j}$ for each pose. Similarly, another joint level transformer encoder is used to extract feature $\mathcal{F}_o \in \mathbb{R}^{(J+1) \times f_j}$ from \mathcal{S} . Subsequently, a second transformer encoder extracts temporal features $\mathcal{F}_t \in \mathbb{R}^{T \times F_t}$, where $F_t = (J+1) \times f_t$, with concatenated input of \mathcal{F}_j and \mathcal{F}_o . Following this, a 1D pooling layer reduces the temporal dimension. And a primal joint pooling layer selectively extracts features from intersecting primal joints (uniformly across different skeleton graphs) to form the latent feature $\mathcal{Z} = E(\mathcal{D}, \mathcal{S}) \in \mathbb{R}^{(T/l) \times J_p \times f_z}$, where l represents the temporal downsampling rate and J_p is the number of primal joints. This is followed by a temporal unpooling layer, which replicates \mathcal{Z} by a factor of l , and a joint unpooling layer that introduces zero-padding at non-primal joint locations. Further refinement is executed via two transformer encoders, operating on temporal and joint dimensions similar to the initial encoding phase. The output, $\mathcal{F}_o = D(\mathcal{Z}, \mathcal{S})$, is formatted to match the dimensionality of the input dynamic \mathcal{D} .

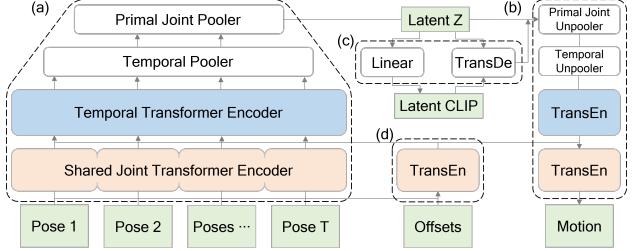


Figure 3. Overview of the proposed motion autoencoder. (a) shows the initial processing with the Temporal Transformer and Shared Joint Transformer Encoders. (b) illustrates the Primal Joint Unpooling and Temporal Unpooling sections. (c) represents the MLP and TransDe components leading to the Latent Z space. (d) indicates the final processing involving Latent CLIP and generation of motion and offsets.

Text-aware Motion Autoencoder. To incorporate textual information into our autoencoder architecture, we develop a cross-modal encoding and decoding scheme. This involves encoding the latent vector \mathcal{Z} into the CLIP feature domain $\mathcal{Z}_{CLIP} = E_t(\mathcal{Z})$, where E_t is a latent encoder. Then we have a latent decoder D_t to decode back to joint-aware latent space $\mathcal{Z}_t = D_t(\mathcal{C}, \mathcal{Z})$. The decoder, a causal attention [38] based transformer, accepts both CLIP features and the latent vector \mathcal{Z} as inputs. Its output, subsequently channeled into the joint-aware decoder to get \mathcal{F}_{text} , enables synchronous training of both autoencoder networks. This dual functionality facilitates the conversion of motion into CLIP representations and vice versa, extending the capabilities of the autoencoder to include sequential motion decoding from textual descriptions.

Training Objectives. There are three losses used to train the motion autoencoder: the reconstruction loss from the joint-aware autoencoder $\mathcal{L}_{jrec} = \|\mathcal{P} - \mathcal{F}_o\|_2$; the CLIP similarity loss $\mathcal{L}_{CLIP} = 1 - \cos(\mathcal{Z}_{CLIP}, \hat{\mathcal{Z}}_{CLIP})$; and the CLIP forward reconstruction loss $\mathcal{L}_{trec} = \|\mathcal{P} - \mathcal{F}_{text}\|_2$. The total loss to train motion autoencoder is

$$\mathcal{L}_{ae} = \mathcal{L}_{jrec} + \lambda_1 \mathcal{L}_{CLIP} + \lambda_2 \mathcal{L}_{trec} \quad (1)$$

where $\lambda_1 = 1.0$ and $\lambda_2 = 1.0$ in our experiment.

3.2. Semantic Mappings between Motion Autoencoders

Architecture. Our objective is to generate new animal motions by leveraging human motion data, which encompasses a wide range of types and semantic interpretations. We train two autoencoders: a human-focused model on abundant human motion data and an animal-focused model on the animal's limited dataset enriched with latent features extracted from the human motion model.

In Figure 2, static \mathcal{S}^h and dynamic \mathcal{D}^h components of human motions are encoded into a latent motion feature

space \mathcal{Z}^h through the human encoder E^h . For simplicity, we use h to represent *human* and a to represent *animal*. We then replace the subject of the sentence describing the human motion with the name of the targeted animal. The CLIP embedding of the original sentence is \mathcal{C}^h and the edited sentence is $\tilde{\mathcal{C}}^a$. These features are subsequently passed into the animal motion decoders, D^a and D_t^a , which incorporate the static components \mathcal{S}^a of animal motions to generate the synthetic output $\tilde{\mathcal{F}}_o = D^a(D_t^a(\tilde{\mathcal{C}}^a, \mathcal{Z}^h), \mathcal{S}^a)$. we simplify this process as $\tilde{\mathcal{F}}_o = \tilde{D}^a(\tilde{\mathcal{C}}^a, \mathcal{Z}^h, \mathcal{S}^a)$.

Training Objectives. To supervise training of the aforementioned architecture, we design three loss functions, as illustrated in Figure 2.

CLIP Similarity Loss. Our objective is to extract a subject-invariant latent feature \mathcal{Z}^h from human motion data, encapsulating the action independent of the subject. For instance, the extracted latent feature \mathcal{Z}^h of ‘a person is running’ should encapsulate the notion of ‘running’ exclusively, abstracting away from ‘a person’. We integrate this subject-invariant feature into our network by employing two distinct CLIP cosine similarity losses. The first loss function minimizes the distance between the CLIP feature \mathcal{C}^h of the human motion sentence and \mathcal{Z}^h , as introduced in Section 3.1. The second loss function minimizes the distance between the modified CLIP feature $\tilde{\mathcal{C}}^a$, obtained by substituting the subject in the sentence with an animal name, and \mathcal{Z}^h , represented as

$$\mathcal{L}_{CLIP} = 1 - \cos(E_t^a(\mathcal{Z}^h), \tilde{\mathcal{C}}^a). \quad (2)$$

This dual loss strategy promotes subject-invariance in the latent feature \mathcal{Z}^h .

Latent Consistency Loss. To ensure the integrity of the latent feature transformation within our framework, we define the Latent Consistency Loss, \mathcal{L}_{cons} . This loss quantifies the discrepancy between the human latent feature Z_h and its reconstructed counterpart obtained after processing through the animal motion decoder and encoder, $E^a(\tilde{D}^a(\tilde{\mathcal{C}}^a, \mathcal{Z}^h, \mathcal{S}^a), \mathcal{S}^a)$. It is expressed as the L2 norm of their difference:

$$\mathcal{L}_{cons} = \|Z^h - E^a(\tilde{D}^a(\tilde{\mathcal{C}}^a, \mathcal{Z}^h, \mathcal{S}^a), \mathcal{S}^a)\|_2. \quad (3)$$

End-Effectors Loss. Our End-Effectors Loss ensures that the dynamic translation of motion from humans to animals maintains kinematic integrity by comparing the velocities at the skeletal structure’s extremities, known as end-effectors. These points, defined as terminal nodes on the skeleton graph, are crucial for generating realistic motion. Velocities for these points are computed using forward kinematics, FK_{ee} (see Appendix for methodology). The velocity for human motion end-effectors is calculated as $\mathcal{V}^h = FK_{ee}(\mathcal{D}^h, \mathcal{S}^h)$, and for synthetic animal motion as $\tilde{\mathcal{V}}^a = FK_{ee}(\tilde{\mathcal{F}}_o, \mathcal{S}^a)$.

The loss is defined by the L2 norm of the velocity difference:

$$\mathcal{L}_{ee} = \|\mathcal{V}^h - \tilde{\mathcal{V}}^a\|_2 \quad (4)$$

guiding the network to generate animal motions that reflect the dynamic properties of human movements.

The total loss function for cross-domain motion adaptation is represented as:

$$\mathcal{L}_{cross} = \lambda_3 \mathcal{L}_{cons} + \lambda_4 \mathcal{L}_{CLIP} + \lambda_5 \mathcal{L}_{ee} \quad (5)$$

where $\lambda_3 = 0.1$, $\lambda_4 = 1.0$, and $\lambda_5 = 100$. The training objective for the entire framework is thus represented by:

$$\mathcal{L}_{total} = \mathcal{L}_{ae}^h + \mathcal{L}_{ae}^a + \mathcal{L}_{cross}^a. \quad (6)$$

Inference. During the inference phase, our framework starts by converting a textual description into the corresponding CLIP feature $\tilde{\mathcal{C}}$. In parallel, a human motion—either from an existing motion generation method or from a ground truth motion—is encoded through E^h to produce the latent human motion feature \mathcal{Z}^h . These features are inputs to the animal textual decoder D_t^a , which samples a new latent feature $\tilde{\mathcal{Z}}$. Then the feature is fed into the animal motion decoder D^a , generating the intended animal motion.

4. AnimalML3D Dataset

To address the data scarcity problem, we introduce AnimalML3D, the first animal language-motion dataset which has 922 training pairs and 318 test pairs. It extends DeformingThings4D [27] which consists of 1972 animation sequences spanning 31 different animals or humanoid categories with dense 4D annotation. We select motion sequences that correspond to the SMAL categories [64], and precisely extract skeletal data from the selected motions. This curation process resulted in a robust set of 1,240 animation sequences, which are then divided into a training set of 922 sequences (23 identities) and a test set of 318 sequences (13 identities).

We introduce two significant enhancements to DeformingThing4D. First, we created three descriptive captions by a group of well-trained human annotators for each motion, generating a comprehensive dataset that consists of 3,720 sentences, with a minimum sentence length criterion of five words. Second, we generated skeletal motion data derived from the original animations.

We first fit a SMAL template to the first frame of the mesh, employing the approach detailed in [5]. While this initial step establishes an approximate starting alignment, it necessitates further refinement for a precise fit to the target mesh. To achieve a more precise overlay with the target mesh, we utilized Wrap4D, a commercial software specifically designed for processing 4D sequences. We determined

Methods	R-Precision \uparrow			FID-OOD	MM-Dist \downarrow	Diversity \uparrow	MModality \uparrow
	Top-1	Top-2	Top-3				
T2M-GPT [60]	0.089 \pm .007	0.153 \pm .007	0.214 \pm .007	2.792 \pm .033	0.775 \pm .004	44.761 \pm 2.693	22.958 \pm 0.731
MotionGPT [20]	0.148 \pm .008	0.226 \pm .008	0.285 \pm .008	2.211 \pm .034	0.741 \pm .004	44.334 \pm 2.733	13.967 \pm 1.098
MDM [43]	0.336 \pm .010	0.523 \pm .012	0.649 \pm .014	1.167 \pm .027	0.501 \pm .003	52.137 \pm 2.690	22.108 \pm 2.338
MotionDiffuse [61]	0.407 \pm .017	0.614 \pm .015	0.733 \pm .015	1.019 \pm .014	0.464 \pm .004	38.821 \pm 1.790	31.350 \pm 0.646
OMGPT (Ours)	0.850 \pm .009	0.935 \pm .007	0.964 \pm .006	1.453 \pm .021	0.355 \pm .003	43.804 \pm 1.701	34.492 \pm 0.874

Table 1. **Comparison with the state-of-the-art methods on out-of-distribution text descriptions.** We evaluate all methods using metrics from [13]. FID-OOD is used to gauge out-of-distribution performance, differentiating it from typical in-distribution assessments. We report each metric’s average and standard deviation, based on 20 evaluations. The best and second-best results are highlighted in cyan and blue.

Methods	R-Precision \uparrow			FID \downarrow	MM-Dist \downarrow	Diversity \uparrow	MModality \uparrow
	Top-1	Top-2	Top-3				
Real motion	0.558 \pm .049	0.734 \pm .040	0.839 \pm .032	0.105 \pm .005	0.357 \pm .006	22.795 \pm 1.843	-
T2M-GPT [60]	0.080 \pm .024	0.168 \pm .023	0.248 \pm .042	1.084 \pm .042	0.636 \pm .013	33.403 \pm 1.902	20.078 \pm 1.096
MotionGPT [20]	0.142 \pm .016	0.233 \pm .032	0.307 \pm .042	0.748 \pm .050	0.558 \pm .010	29.265 \pm 2.453	10.311 \pm 1.537
MDM [43]	0.379 \pm .051	0.554 \pm .058	0.646 \pm .048	0.505 \pm .038	0.487 \pm .008	27.826 \pm 1.643	13.593 \pm 1.038
MotionDiffuse [61]	0.505 \pm .037	0.695 \pm .045	0.805 \pm .041	0.401 \pm .024	0.421 \pm .007	25.194 \pm 1.510	7.081 \pm 0.357
OMGPT (Ours)	0.539 \pm .064	0.721 \pm .063	0.830 \pm .043	0.223 \pm .036	0.348 \pm .007	37.487 \pm 1.575	17.487 \pm 0.792

Table 2. **Comparison with the state-of-the-art methods on our AnimalML3D test set.** Methods are evaluated using metrics from [13], with top results in cyan (best) and blue (second-best). We report each metric’s average and standard deviation, based on 20 evaluations.

corresponding keypoints, ranging from 10 to 30, on the fitted SMAL template and the target mesh geometry. Having established this keypoint correspondence in the inaugural frame, Wrap4D is then employed to systematically morph the SMAL template across the entire sequence, ensuring that the adapted mesh conformed to the keypoint definitions and maintained the topological consistency of the SMAL model throughout the frames. Subsequently, the joint positions were computed using the joint regression matrix as outlined in [30]. Comprehensive details of dataset curation and visual illustrations of the mesh quantities and procedural results are included in the Appendix.

5. Experiments

Baselines and Evaluation Settings. We compare our model performance with various motion generation models, including T2MGPT [60], MotionGPT [20], MDM [43] and MotionDiffuse [61]. T2MGPT and MotionGPT employ a two-stage pipeline with VQVAE [45] and GPT [6], whereas MDM and MotionDiffuse utilize a single-stage diffusion model. All models are trained on the proposed AnimalML3D dataset.

We evaluate the results on two tasks. In in-distribution (ID) setting, we generate with prompts from the AnimalML3D dataset. In out-of-distribution (OOD) setting, we generate with prompts from the HumanML3D dataset by replacing the subject phrase with an animal name.

We use the same set of evaluation metrics as in [13]. *R-*

precision measures retrieval accuracy by comparing the input text to the generated motions. *Frechet Inception Distance (FID)* measures the distance between generated motion distribution and testing motion distribution for ID experiments. As there is no ground truth animal motion for OOD experiments, we compare the distance between generated OOD motions and whole ground truth Animal3D dataset to compute the FID-OOD metric. *Multimodal Distance (MM-Dist)* gauges the distance between the generated motion and the corresponding sentences in the latent space, using the outputs from the human latent encoder E_t^h and CLIP features. *Diversity* evaluates the differences between independently sampled motions. *Multimodality (MModality)* assesses the variance within multiple motions generated from a single text description.

Implementation Details. We use a two-layer transformer with a dimension of 16 for the joint encoder/decoder, and a two-layer transformer with a dimension of 256 for the temporal encoder/decoder. The latent encoder E_t head is a linear layer with an input size of $49 \times 7 \times 16$ and the caption decoder is a four-layer transformer decoder with a dimension of 256. We train with the total loss described in Section 3 in an end to end manner for 30000 steps. We use an Adam optimizer with learning rate $lr = 10^{-4}$, betas $\beta = (0.9, 0.999)$, batch size $B = 256$, exponential moving constant $\lambda = 0.99$.

We configure the SMPL and SMAL representations with

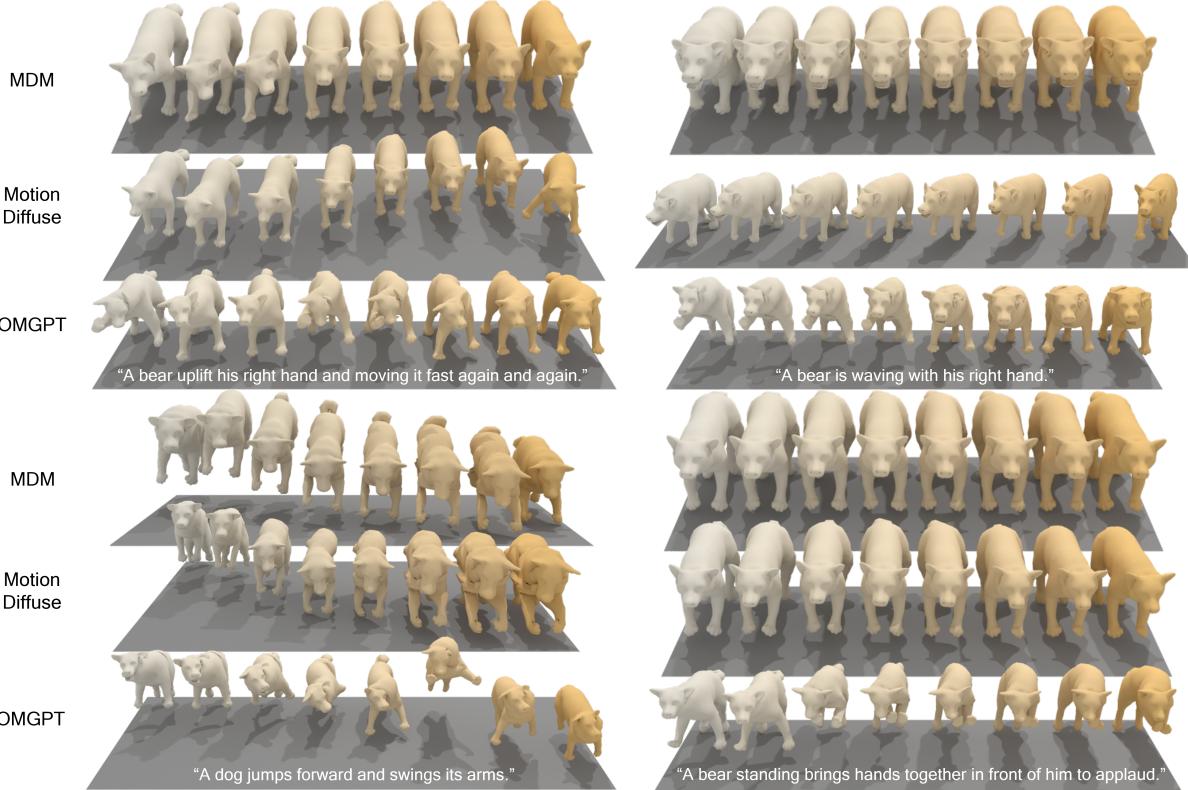


Figure 4. Visual comparison between our method, OMGPT, and other baselines. Motions are generated according to the captions shown in the figure, evenly arranged in rows from left to right, showcasing a progression from beginning to end. Our method demonstrates enhanced versatility and adherence to captions, outperforming baselines MDM [43] and MotionDiffuse [61]. We assess performance using text descriptions adapted from HumanML3D [13], modified to replace human subjects with various animals. Notably, our method effectively processes OOD caption inputs, demonstrating significant improvements in alignment to these captions. Meanwhile, baselines are less adept at responding to such text descriptions.

22 and 35 joints respectively. For the HumanML3D dataset, following the data processing step in [13], we only keep motion sequences between 20 and 196 frames. For the AnimalML3D dataset, given its smaller size, we only keep motion sequences between 10 and 196 frames. For details on the convergence of each loss component, readers are referred to the Appendix.

Quantitative Motion Generation Comparison. Table 1 and Table 2 show our quantitative animal motion generation results, on both ID and OOD prompts. ID prompts are taken from our AnimalML3D test set, while out-of-distribution prompts are annotations from the HumanML3D test set with the subject replaced by an animal name. We compare our results with four recent human motion generation baselines. Two of them are based on VQVAE and GPT [20, 60] and the other two are based on diffusion models [43, 61].

The GPT-based models [20, 60] exhibit low R-precision scores for both ID and OOD experiments, attributed to the sparse training dataset of only 922 motion sequences and a relatively large codebook size of 512. This significant dis-

crepancy leads to two key issues. First, the VQVAE tends to overfit, reducing its ability to generalize in ID motion generation. Secondly, the large codebook size complicates the training of the transformer decoder, making it prone to generating repetitive motion patterns or noisy motions due to data sparsity (more details in the Appendix). This observation underscores that, although FID-OOD and Diversity scores are high, the extensive codebook size frequently results in unrealistic or repeated motions. In contrast, our method, without relying on a fixed-size codebook, effectively handles small datasets with limited motion diversity.

Our OMGPT model outperforms the diffusion-based models MDM [43] and MotionDiffuse [61] in all metrics, both ID and OOD. While these models produce slightly higher R-precision and lower diversity scores compared to the GPT-based baselines, indicating better robustness to small datasets and text-motion alignment, they fall short in generating diverse motions from OOD prompts.

Additionally, note that OMGPT’s superiority on OOD prompts is more prominent than ID prompts. This is because

Exp	Configuration Difference	R-Precision Top-1 \uparrow	MM-Dist \downarrow	Diversity \uparrow
A	MLP Mapping	$0.351 \pm .009$	$0.476 \pm .002$	31.406 ± 1.137
B	E_t : MLP	$0.404 \pm .013$	$0.466 \pm .003$	38.412 ± 1.900
C	$\lambda_5 \mathcal{L}_{ee} = 0$	$0.477 \pm .017$	$0.468 \pm .003$	51.441 ± 2.107
D	$\lambda_3 \mathcal{L}_{cons} = 0$	$0.508 \pm .019$	$0.452 \pm .003$	43.946 ± 2.075
E	-	$0.850 \pm .009$	$0.355 \pm .003$	43.804 ± 1.701

Table 3. **Ablation Study on the configurations of our framework.**

In our ablation study, we evaluate various configurations of our framework in comparison to the fully integrated model, with a focus on architectural choices and loss weights. The impact of these elements is assessed using metrics including OOD R-Precision, MM-Dist, and Diversity. Each metric is evaluated 20 times to compute the average and standard deviation. These results demonstrate the essential role of each design component in our model in achieving optimal performance.

of its ability to incorporate human motion knowledge into the training process, and thus adaptable to a wider range of potential prompts. It is infeasible to jointly train with human data in all four baseline methods due to the motion representation difference in nature.

Qualitative Motion Generation Analysis. Figure 4 presents our generated motion sequences in comparison with baseline approaches MDM [43] and MotionDiffuse [61]. With abundant knowledge transferred from human motion datasets, our model is able to generate results with better fidelity, alignment with the textual inputs, and diversity in complex motion descriptions.

Our OMGPT model outperforms baseline methods in three aspects. First, OMGPT demonstrates the ability to generate OOD motions that are out of the existing animal data distribution but in the human motion distribution. The bottom right and top right examples show a bear clapping and waving hands which could be faithfully and reasonably generated by incorporating human motion knowledge with our framework but rarely happens in reality. Second, OMGPT is able to comprehend a broader range of motion patterns not appearing in the animal dataset, like ‘fast’ and ‘again and again’ in the top left example. Third, OMGPT is capable of capturing complicated and composite motion descriptions, despite being built on an animal motion dataset with limited motion diversity and relatively simple prompts. The bottom left example illustrates OMGPT generating a sequence of motions (‘jumping’ and then ‘swinging arms’) whereas the baseline methods are not able to handle.

Ablation Study. To validate the effectiveness of our designed semantic mapping configuration, we present ablation studies in Table 3. We alter the structure and loss weights of our final model to analyze their impact on motion generation quality and visual representation, as shown in Figure 5.

Architecture (Exp A & B). Exp A shows that adding

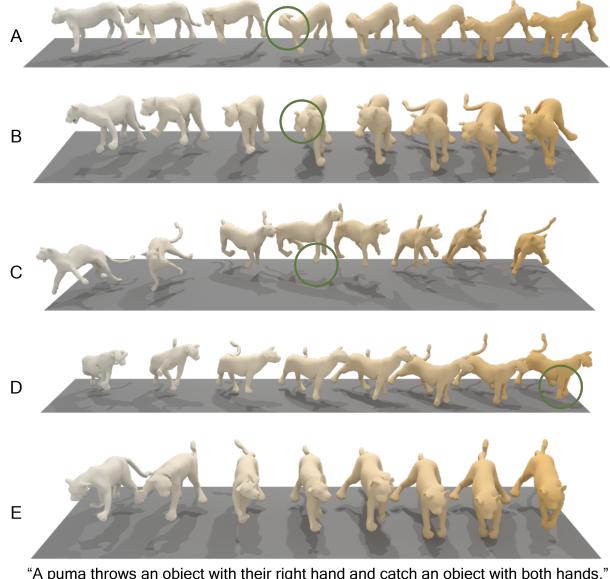


Figure 5. **Visualization of generated motion under different configurations.** The letters A–E correspond to the Exp identities in Table 3. Motions are generated according to the caption shown in the figure. The green circles highlight the unrealistic parts in the motions by making changes to the configurations of the designed framework. Motions are evenly arranged in rows from left to right, showcasing a temporal progression from beginning to end.

an MLP mapping between the human latent space and the generated animal latent space results in less dynamic motion, as illustrated in row A of Figure 5. Exp B shows that altering the semantic head from a linear layer to MLP allows more flexibility in the latent space. However, this indirectly affects the motion latent, leading to reduced movement in some joints, as observed in row B of Figure 5.

Loss Weight (Exp C & D). Exp C sets the weight for L_{ee} to 0 and achieves higher motion diversity but at the expense of realism. Without the end effector loss, the generated motion appears unnaturally elevated above the ground, as shown in row C of Figure 5. Exp D demonstrates that omitting consistency loss leads to incomplete motion sequences. This is evident in the ‘catch an object’ sequence, where the final part is missing in the generated motion, as depicted in row D of Figure 5.

6. Conclusion

In this work, we propose the first text-driven animal motion generation algorithm. We design a one-stage jointly-training architecture that first trains motion autoencoder for both animal and human domains and simultaneously trains a knowledge mapping mechanism to generate animal motion with human motion encodings. We demonstrate diverse and realistic animal motion generation results and present metrics

quantitatively surpassing all baseline methods. Moreover, we contribute the first animal text-motion dataset AnimalML3D, creating a new playground to encourage future investigation in the field of animal motion generation.

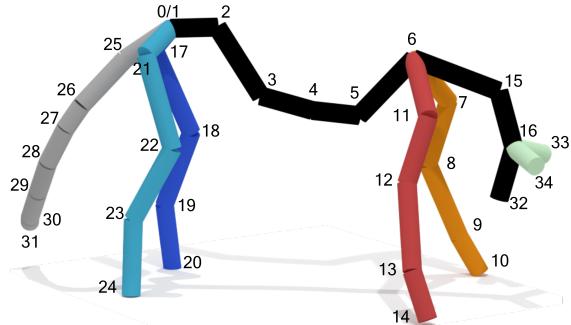
References

- [1] Michel Abdul-Massih, Innfarn Yoo, and Bedrich Benes. Motion style retargeting to characters with different morphologies. In *Computer Graphics Forum*, pages 86–99, 2017. [3](#)
- [2] Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)*, 39(4):62–1, 2020. [3, 4](#)
- [3] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. [12](#)
- [4] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan, 2017. [2](#)
- [5] Benjamin Biggs, Thomas Roddick, Andrew Fitzgibbon, and Roberto Cipolla. Creatures great and smal: Recovering the shape and motion of animals from video. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 3–19, 2019. [5, 12](#)
- [6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901, 2020. [6](#)
- [7] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1591–1599, 2017. [2](#)
- [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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023. [3](#)
- [9] Yinglin Duan, Tianyang Shi, Zhengxia Zou, Yenan Lin, Zhe-hui Qian, Bohan Zhang, and Yi Yuan. Single-shot motion completion with transformer, 2021. [2](#)
- [10] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015. [2](#)
- [11] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, page 33–42, New York, NY, USA, 1998. Association for Computing Machinery. [3](#)
- [12] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. [2, 3](#)
- [13] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. [1, 2, 3, 6, 7, 14](#)
- [14] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *Proceedings of the European Conference on Computer Vision*, 2022. [1, 2, 3](#)
- [15] Félix G. Harvey and Christopher Pal. Recurrent transition networks for character locomotion. In *ACM SIGGRAPH Asia 2018 Technical Briefs*, New York, NY, USA, 2018. Association for Computing Machinery. [2](#)
- [16] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4), 2020. [2](#)
- [17] A. Hernandez, J. Gall, and F. Moreno. Human motion prediction via spatio-temporal inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7133–7142, 2019. [2](#)
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020. [3](#)
- [19] Ruozhi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Dixin Jiang. Dance revolution: Long-term dance generation with music via curriculum learning, 2023. [2](#)
- [20] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiogpt: Human motion as a foreign language. *arXiv preprint arXiv:2306.14795*, 2023. [1, 2, 3, 6, 7, 16](#)
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [12](#)
- [22] Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 452–461, 2020. [3](#)
- [23] Nilesh Kulkarni, Davis Rempe, Kyle Genova, Abhijit Kundu, Justin Johnson, David Fouhey, and Leonidas Guibas. Nifty: Neural object interaction fields for guided human motion synthesis. *arXiv preprint arXiv:2307.07511*, 2023. [2](#)
- [24] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *Advances in Neural Information Processing Systems*, 2019. [2](#)
- [25] Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 39–48, 1999. [3](#)
- [26] Peizhuo Li, Kfir Aberman, Zihan Zhang, Rana Hanocka, and Olga Sorkine-Hornung. G animator: Neural motion synthesis from a single sequence. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022. [2](#)

- [27] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12706–12716, 2021. 1, 2, 3, 5
- [28] Zhouyingcheng Liao, Jimei Yang, Jun Saito, Gerard Pons-Moll, and Yang Zhou. Skeleton-free pose transfer for stylized 3d characters. In *Proceedings of the European Conference on Computer Vision*, pages 640–656, 2022. 3
- [29] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset, 2023. 2, 3
- [30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seamless Graphics Papers: Pushing the Boundaries*, pages 851–866. 2023. 2, 6, 12
- [31] Shubh Maheshwari, Rahul Narain, and Ramya Hebbalaguppe. Transfer4d: A framework for frugal motion capture and deformation transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12836–12846, 2023. 3
- [32] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5442–5451, 2019. 3
- [33] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 2
- [34] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001. 1
- [35] Mathis Petrovich, Michael J. Black, and Gü̈l Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. 2
- [36] Mathis Petrovich, Michael J. Black, and Gü̈l Varol. TEMOS: Generating diverse human motions from textual descriptions. In *Proceedings of the European Conference on Computer Vision*, 2022. 1, 2
- [37] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big Data*, 4(4):236–252, 2016. 2, 3
- [38] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 4
- [39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2
- [40] Yeongho Seol, Carol O’Sullivan, and Jehee Lee. Creature features: Online motion puppetry for non-human characters. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 213–221, 2013. 3
- [41] Xiangjun Tang, He Wang, Bo Hu, Xu Gong, Ruifan Yi, Qilong Kou, and Xiaogang Jin. Real-time controllable motion transition for characters. *ACM Transactions on Graphics*, 41(4):1–10, 2022. 2
- [42] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *Proceedings of the 17th European Conference on Computer Vision*, pages 358–374, 2022. 1, 2
- [43] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *Proceedings of the 11th International Conference on Learning Representations*, 2023. 1, 2, 3, 6, 7, 8
- [44] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 448–458, 2023. 2
- [45] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 6
- [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 2
- [47] Haoyu Wang, Shaoli Huang, Fang Zhao, Chun Yuan, and Ying Shan. Hmc: Hierarchical mesh coarsening for skeleton-free motion retargeting. *arXiv preprint arXiv:2303.10941*, 2023. 3
- [48] Jingbo Wang, Sijie Yan, Bo Dai, and Dahua Lin. Scene-aware generative network for human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12206–12215, 2021. 2
- [49] Zhenyi Wang, Ping Yu, Yang Zhao, Ruiyi Zhang, Yufan Zhou, Junsong Yuan, and Changyou Chen. Learning diverse stochastic human-action generators by learning smooth latent transitions. In *AAAI Conference on Artificial Intelligence*, 2019. 2
- [50] Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Humanise: Language-conditioned human motion generation in 3d scenes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2
- [51] Mao Wei, Liu Miaomiao, and Salzemann Mathieu. History repeats itself: Human motion prediction via motion attention. In *Proceedings of the European Conference on Computer Vision*, 2020. 2
- [52] Jiacong Xu, Yi Zhang, Jiawei Peng, Wufei Ma, Artur Jesslen, Pengliang Ji, Qixin Hu, Jiehua Zhang, Qihao Liu, Jiahao Wang, et al. Animal3d: A comprehensive dataset of 3d animal pose and shape. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9099–9109, 2023. 3
- [53] Katsu Yamane, Yuka Ariki, and Jessica Hodgins. Animating non-humanoid characters with human motion data. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 169–178, 2010. 3
- [54] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. Lasr: Learning articulated shape reconstruction from a monocular video. In *Proceedings of*

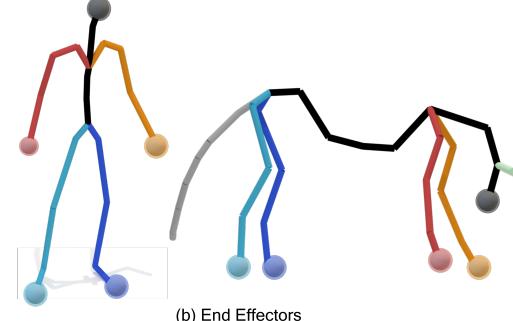
the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15980–15989, 2021. 2

- [55] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan. Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. In *NeurIPS*, 2021.
- [56] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2863–2873, 2022. 2
- [57] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Lassie: Learning articulated shapes from sparse image ensemble via 3d part discovery. *Advances in Neural Information Processing Systems*, 35:15296–15308, 2022. 2
- [58] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Hi-lassie: High-fidelity articulated shape and skeleton discovery from sparse image ensemble. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4853–4862, 2023. 2
- [59] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. PhysDiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 3
- [60] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3, 6, 7, 16
- [61] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 2, 3, 6, 7, 8
- [62] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. *arXiv preprint arXiv:2304.01116*, 2023. 3
- [63] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 4
- [64] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6365–6373, 2017. 2, 5, 12, 13
- [65] Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black. Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3955–3963, 2018. 2

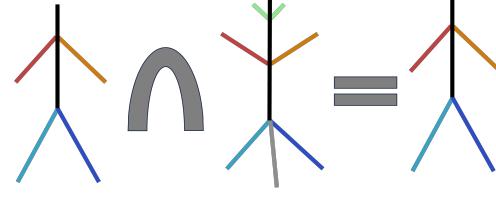


	Joint Name	Joint Index	Joint Name	Joint Index
'root':	0,	10,	'LFoot':	10,
'pelvis0':	1,	'RLeg1':	11,	
'spine':	2,	'RLeg2':	12,	
'spine0':	3,	'RLeg3':	13,	
'spine1':	4,	'RFoot':	14,	
'spine2':	5,	'Neck':	15,	
'spine3':	6,	'Head':	16,	
'LLeg1':	7,	'LLeg2':	8,	
'LLeg2':	8,	'LLeg3':	9,	
'LLeg3':	9,	'LLegBack1':	17,	
		'LLegBack2':	18,	
		'LLegBack3':	19,	
		'LFootBack':	20,	
		'RLegBack1':	21,	
		'RLegBack2':	22,	
		'RLegBack3':	23,	
		'RFootBack':	24,	
		'Tail1':	25,	
		'Tail2':	26,	
		'Tail3':	27,	
		'Tail4':	28,	
		'Tail5':	29,	
		'Tail6':	30,	
		'Tail7':	31,	
		'Mouth':	32,	
		'Ear':	33,	
		'Rear':	34,	

(a) SMAL Joint Names and Locations



(b) End Effectors



(c) Intersecting Primal Joints

Figure 6. **Illustration of joints-related information.** In part (a), we present the skeleton of the SMAL model [64], including the names and indices of the joints. Part (b) displays the locations of the end-effectors in both the SMPL and SMAL models, represented by spheres of the same color for corresponding joints. In part (c), we depict the process of intersecting the primal skeleton graphs of SMPL and SMAL, illustrating the resulting intersecting primal skeleton between the two models.

A. Configurations of Joints

In part (a) of Figure 6, we outline the skeletal structure of the Skinned Multi-Animal Linear (SMAL) model [64]. The SMAL skeleton is comprised of 35 joints, notably with the “root” and “pelvis0” joints situated at the same location. A key distinction between the SMAL model and the Skinned Multi-Person Linear (SMPL) model [30] lies in the addition of a tail in SMAL, an element absent in the SMPL model.

We define essential concepts such as “end effectors”, “primal joints”, and “intersecting primal joints” in Section 3. These concepts are visually elaborated upon in Figure 6. For instance, in part (b) of the figure, we illustrate the end-effector joints for both SMAL and SMPL models, each marked with distinct color spheres to denote the five end-effector joints in both models.

Part (c) of Figure 6 showcases the intersection of the primal skeletons of SMPL and SMAL. This intersection is subject to potential ambiguity. For example, the left leg branch in the SMPL graph could correspond to multiple components in SMAL, such as the left back leg, the tail, or even the right leg branch. Our approach aligns these intersections based on their semantic meanings, ensuring a meaningful and contextually appropriate mapping. The intersecting primal joints are clearly indicated in the figure, providing a nuanced understanding of the skeletal overlaps between the two models.

B. Details of Data Processing

In Figure 7, we illustrate the three-stage data processing workflow for our AnimalML3D dataset, using a representative example. The initial stage involves fitting a SMAL model [64] to the animal’s identity in the first frame, typically in a resting pose as depicted in the lower section of (a) in Figure 7. Our approach is developed upon the framework established by [5], with a notable modification replacing the losses with Chamfer Distance [3]. We build upon the framework presented by [5], incorporating a significant adaptation: we employ the Chamfer Distance as our loss function, as described by [3], instead of the original loss terms used in [5]. The model optimization targets four parameters: scale (S), global translation (T), and the SMAL model parameters β and θ , which are refined using the Chamfer Distance [3] between points sampled from the computed mesh of the SMAL model and the target mesh, with 3000 points sampled per iteration. Optimization is executed in two phases using the Adam optimizer [21] with a learning rate of 0.005: initially, S and T are optimized over 50 epochs, followed by a comprehensive optimization of S , T , β , and θ for an additional 400 epochs to obtain the final mesh.

In the second stage, we utilize the software Wrap4D for mesh registration, aligning the roughly fitted mesh from the previous stage to the meshes of each frame. The blueprint code for this process is depicted in part (b) of Figure 7. Within the software environment, we establish corresponding points between the fitted mesh and the target mesh. For

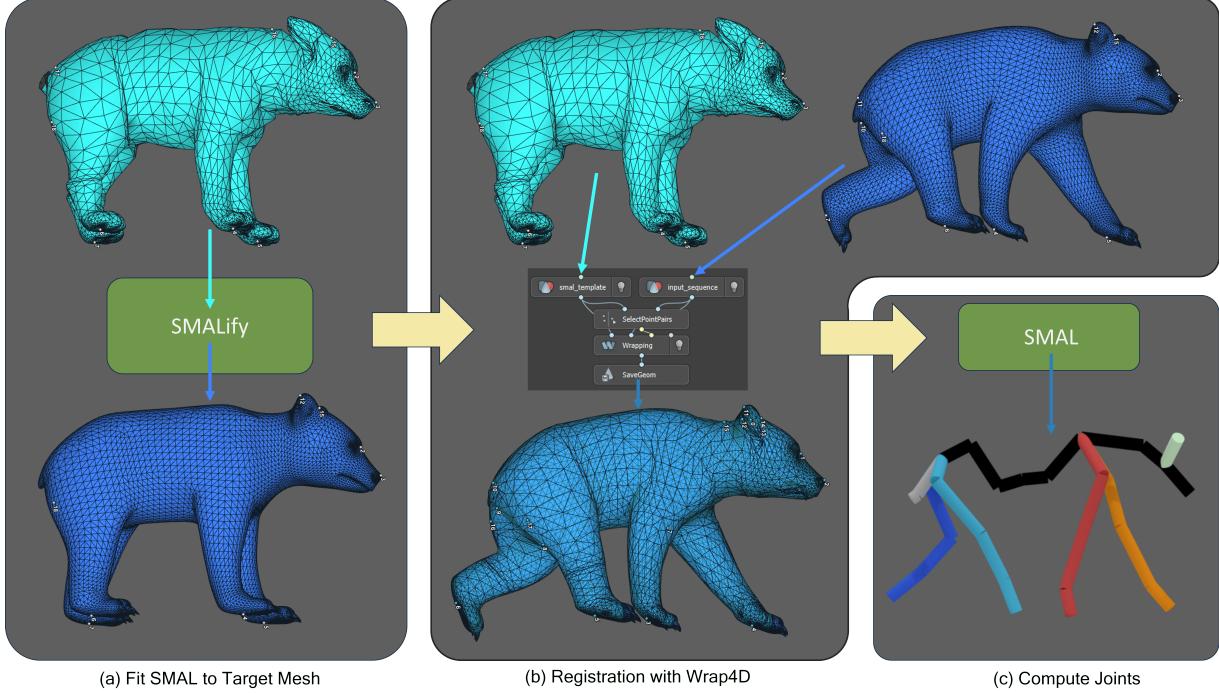


Figure 7. Data processing pipeline for our AnimalML3D dataset. Our data processing pipeline is delineated into three stages: (a) fitting the SMAL model [64] to the target mesh, (b) registering the fitted mesh to a sequence of motions, and (c) computing joint positions from the registered mesh. In stage (a), we illustrate the target mesh (at the bottom) and the resulting fitted mesh (at the top). For stage (b), inputs to Wrap4D include the fitted mesh alongside the target mesh sequence (top right), with the output being the registered mesh maintaining SMAL topology (bottom), where white dots signify the corresponding points utilized for registration. In stage (c), we calculate the joint positions from the registered mesh; the figure highlights a short tail representation, typical of bear species where the tail is not prominently visible.

every unique identity in the dataset, we generate a distinct correspondence map, culminating in a total of 36 correspondence mappings required to process the entire dataset.

In the third stage, which is elaborated upon in Section 4 of the main paper, we apply the joint regression matrix to the vertices of the SMAL model that preserve the topology. This application yields the positional data for the joints.

C. Loss Details and Convergence

In addition to the losses defined in Sections 3.1 and 3.2, we introduce another loss function that employs global translation \mathcal{T} to regularize generated motion. This loss is applied to both motions generated from the joint autoencoder and the text autoencoder, with a weight of 1.0. Empirically, we observed that incorporating global translation results in smoother motion generation, significantly reducing the shaking effect.

Figure 8 illustrates the convergences of all the losses. Notably, the semantic loss \mathcal{L}_{CLIP} does not converge close to 0. There are two primary reasons for this. First, achieving complete alignment between the motion and CLIP features is challenging. The motion encompasses attributes like velocity and facing direction, which are not fully captured in

the CLIP features. Additionally, the CLIP features encode semantic nuances, such as differentiating between “run” for first and second-person pronouns and “runs” for third-person pronouns. These disparities hinder a full alignment between motion and CLIP features. Second, our use of cosine similarity as a metric reveals that when similarity falls below 0.75, the resulting r-precision is approximately 63%, a respectable rate in motion recall. This outcome underscores the nuanced relationship between motion and CLIP features, suggesting that perfect alignment may not be necessary for effective motion synthesis.

D. More Our Results

In Figure 9, we present additional motions generated by our OMGPT model. These results further validate our model’s capability to generate both ID and OOD. For instance, walking backward is categorized as ID, while stomping with the left foot is considered OOD. A notable challenge is the generation of motions involving complex body interactions, such as stretching one arm with the assistance of the other. This aspect represents a critical area for future development, particularly in translating human motion interactions to animal models. Supplementary material, including a video that

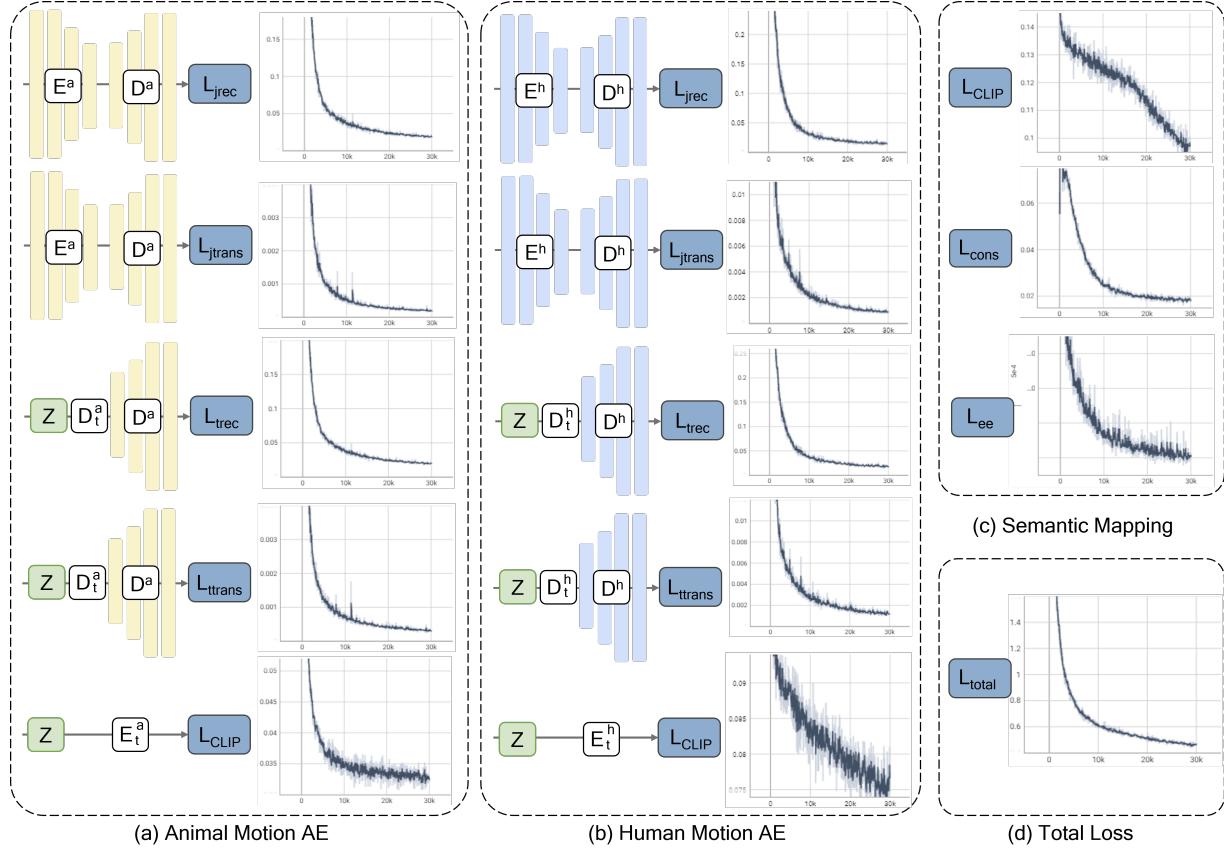


Figure 8. **Visualization of computation of loss functions and their convergence.** Parts (a) and (b) illustrate the loss functions defined in Section 3.1. Part (c) showcases the specific loss function introduced in Section 3.2. Finally, part (d) depicts the overall convergence of the total loss, represented as a weighted sum of all individual loss functions.

showcases these motions in a continuous format, is available. This video, named after the figures in this paper, provides a comprehensive view of the generated motions.

E. Baseline Implementations

For all baseline comparisons, we trained the models using our dataset, converting motions into a 36 by 6 dimensional format (details in Section 3.1). These baseline models, originally designed for human motion generation, do not typically account for offsets, which are crucial in animal motion generation. Therefore, we incorporate offsets into the dynamic features as an additional input and output target. During inference, we directly use animal offsets for a fair comparison with our method. We adhere to the default settings provided in the baseline methodologies for both training and evaluation, ensuring consistency across all comparisons.

F. More Baseline Results

In Figure 10, we present results from T2M-GPT and MotionGPT. The analysis reveals that both models struggle with

generating accurate motions: MotionGPT often produces motionless outputs in response to OOD inputs, whereas T2M-GPT tends to generate erratic and noisy motions under similar OOD conditions. This discrepancy highlights the challenge of aligning motion generation with the corresponding textual descriptions, especially when handling OOD instructions.

G. Metric Computation Details

We elaborate on several evaluation metrics, previously utilized in [13]. The metrics involve three types of features: ground-truth motion features (f_{gt}), generated motion features (f_{pred}), and text features (f_{text}). These features are extracted using the animal encoder, denoted as E^a , following the training of the network.

FID (Fréchet Inception Distance). This metric assesses the overall quality of generated motions. The FID is calcu-

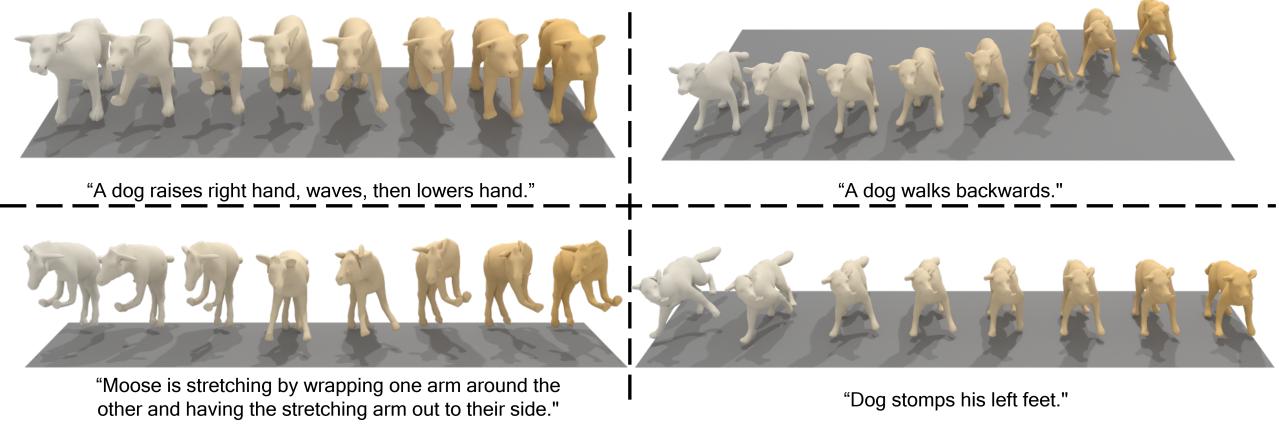


Figure 9. **More results of generated motions from our model.** Our model demonstrates robust performance in generating both ID and OOD motions. Except for walking backward, all evaluated motions are OOD, underscoring the model’s effectiveness in handling a variety of challenging scenarios.

lated using the equation:

$$\text{FID} = \|\mu_{gt} - \mu_{pred}\|^2 - \text{Tr}(\Sigma_{gt} + \Sigma_{pred} - 2(\Sigma_{gt}\Sigma_{pred})^{\frac{1}{2}}) \quad (7)$$

where μ_{gt} and μ_{pred} are mean of f_{gt} and f_{pred} . Σ is the covariance matrix and Tr denotes the trace of a matrix. we calculate FID based on 1024 randomly generated motions.

MM-Dist. This metric calculates the feature-level distance between text embeddings and generated motion features. For N randomly generated samples, MM-Dist is the average Euclidean distance between each text feature and its corresponding generated motion feature, defined as:

$$\text{MM-Dist} = \frac{1}{N} \sum_{i=1}^N \|f_{pred,i} - f_{text,i}\| \quad (8)$$

where $f_{pred,i}$ and $f_{text,i}$ are the features of the i -th text-motion pair. We set N to 1024 in our experiments.

Diversity. Diversity quantifies the variance among all motion sequences in the dataset. We calculate this by randomly selecting S_{dis} pairs of motion features ($f_{pred,i}$ and $f'_{pred,i}$) and then computing:

$$\text{Diversity} = \frac{1}{S_{dis}} \sum_{i=1}^{S_{dis}} \|f_{pred,i} - f'_{pred,i}\| \quad (9)$$

S_{dis} is set to 1024 for OOD and 64 for ID.

MModality. this metric evaluates the diversity of human motions generated from the same text description. For each text description, we generate 100 motions and select two subsets containing 10 motions each. The features of the j -th

pair for the i -th text description are denoted as $(f_{pred,i,j}, f'_{pred,i,j})$. MModality is then defined as:

$$\text{MModality} = \frac{1}{10N} \sum_{i=1}^N \sum_{j=1}^{10} \|f_{pred,i,j} - f'_{pred,i,j}\| \quad (10)$$

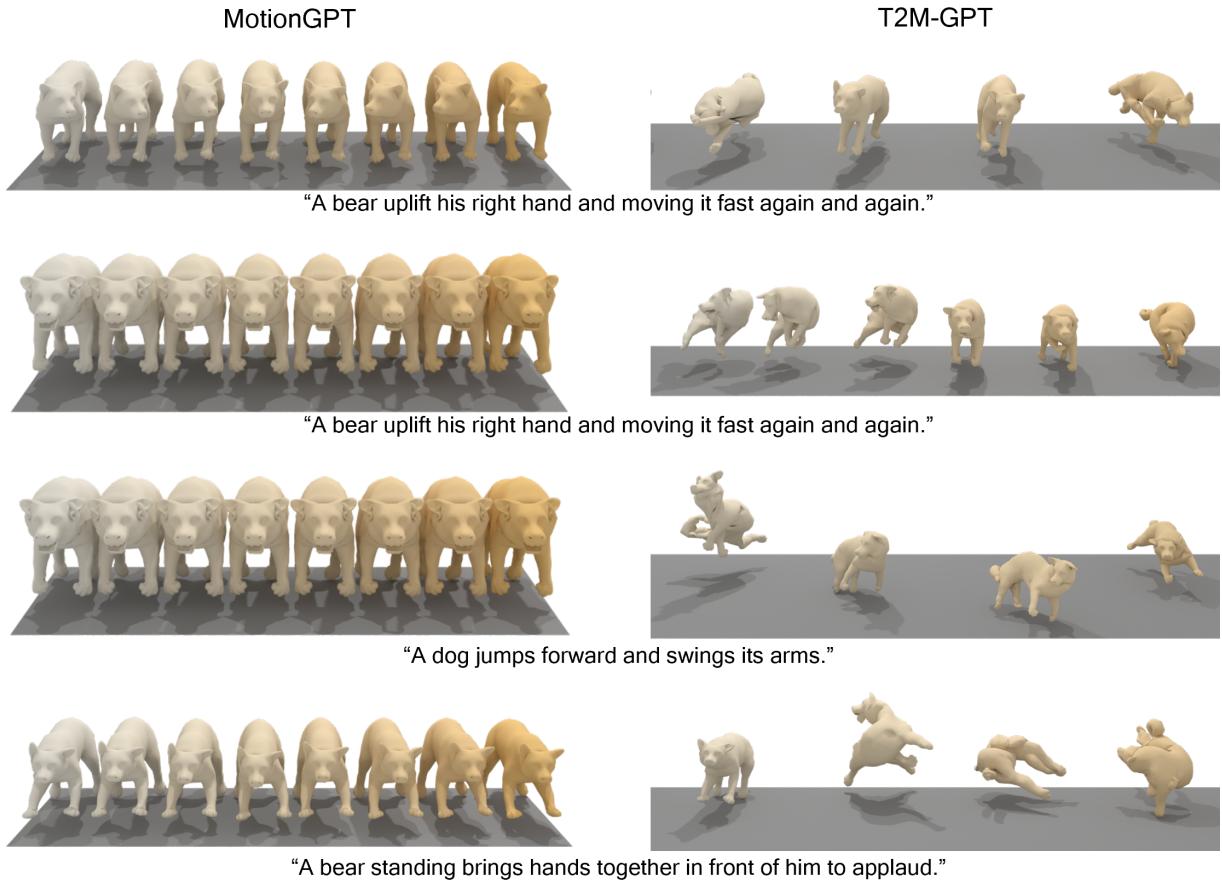


Figure 10. **Generated motions from T2M-GPT and MotionGPT.** Figure illustrates motions generated by T2M-GPT [60] and MotionGPT [20], corresponding to comparisons in Figure 4. These results demonstrate comparatively lower quality, as evidenced by reduced metrics in R-Precision and MM-Dist.