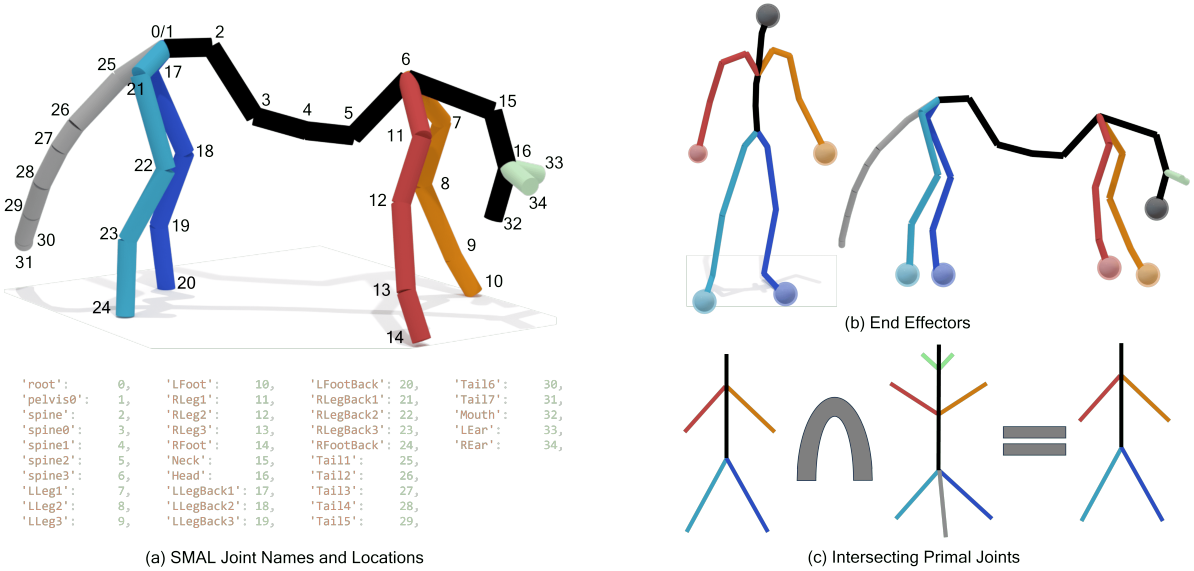

Figure 6. **Illustration of joints-related information.** In part (a), we present the skeleton of the SMAL model [72], including the names and indices of the joints. Part (b) displays the locations of the end-effectors in both the SMPL and SMAL models, represented by spheres of the same color for corresponding 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 [72]. 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 [32] 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 [72] 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 [23] 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

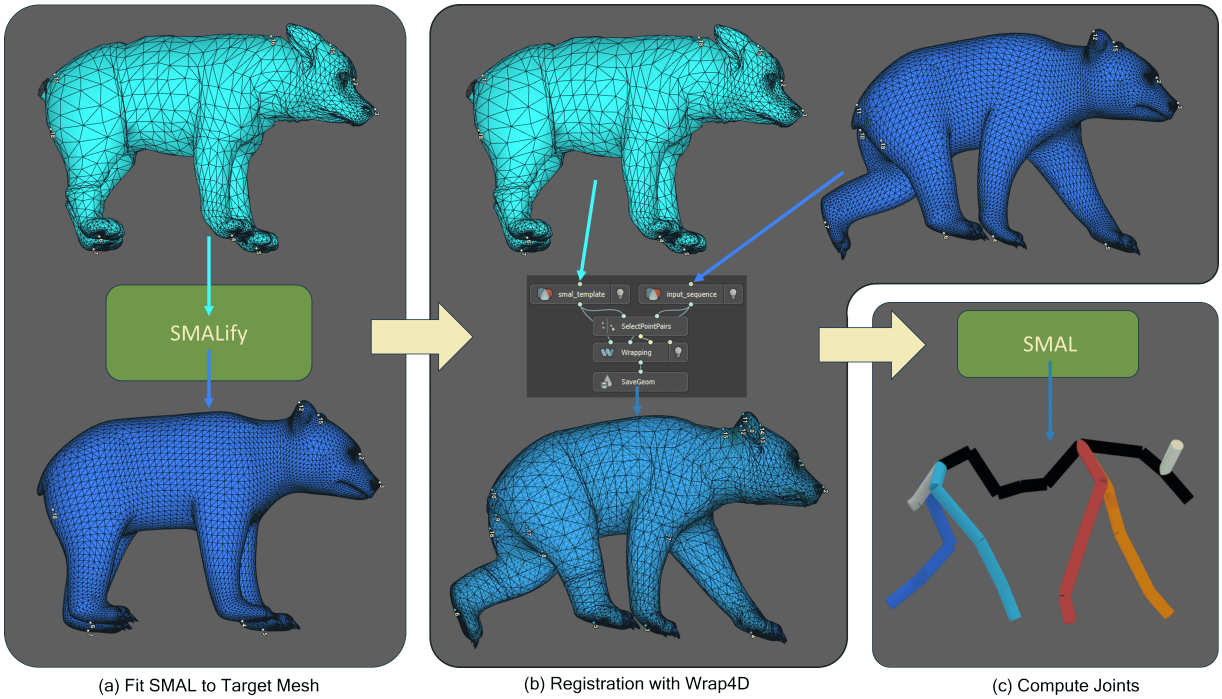

Figure 7. **Data processing pipeline for our AnimalML3D dataset.** Our data processing pipeline is delineated into three stages: (a) fitting the SMAL model [72] to the target mesh, (b) registering the fitted mesh to a sequence of motions, and (c) computing joint positions from the registered mesh. In 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

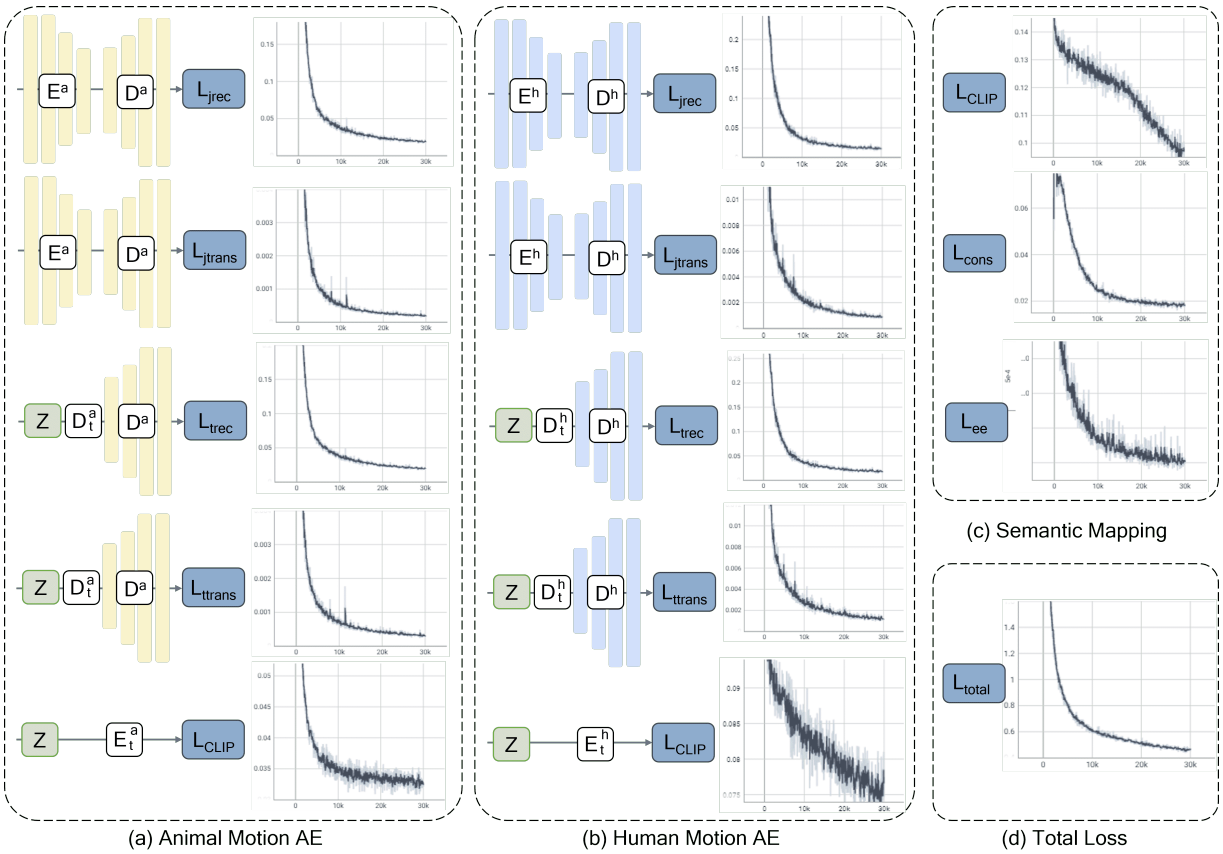

Figure 8. **Visualization of computation of loss functions and their convergence.** Parts (a) and (b) illustrate the loss functions defined in Section 3.1. Part (c) showcases the specific loss function introduced in Section 3.2. Finally, part (d) depicts the overall convergence of the total loss, represented 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 [15]. 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-

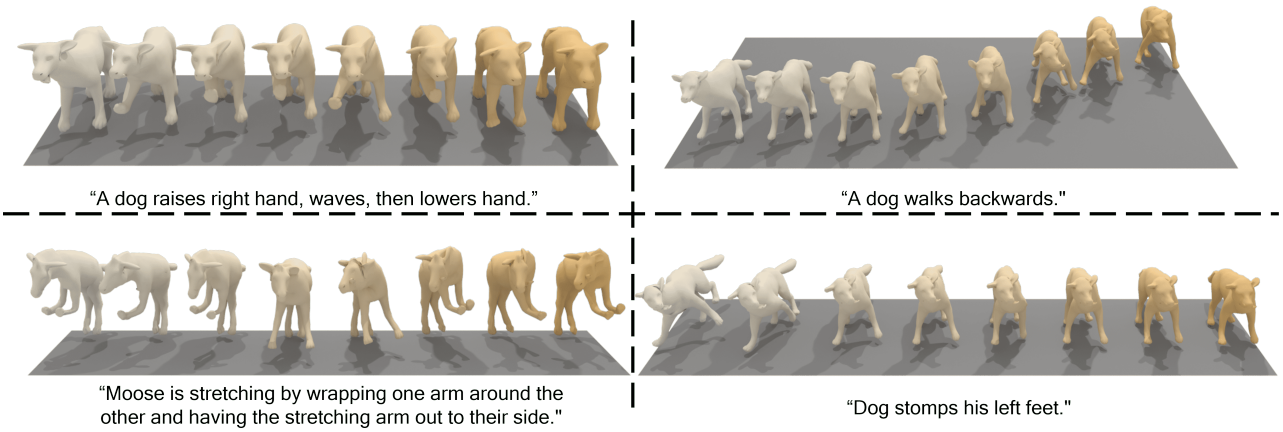

Figure 9. **More results of generated motions from our model.** Our model demonstrates robust performance in generating both ID and OOD motions. Except for walking backward, all evaluated motions are OOD, underscoring the model's effectiveness in handling a variety of challenging scenarios.

lated using the equation:

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

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:

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

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:

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

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:

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

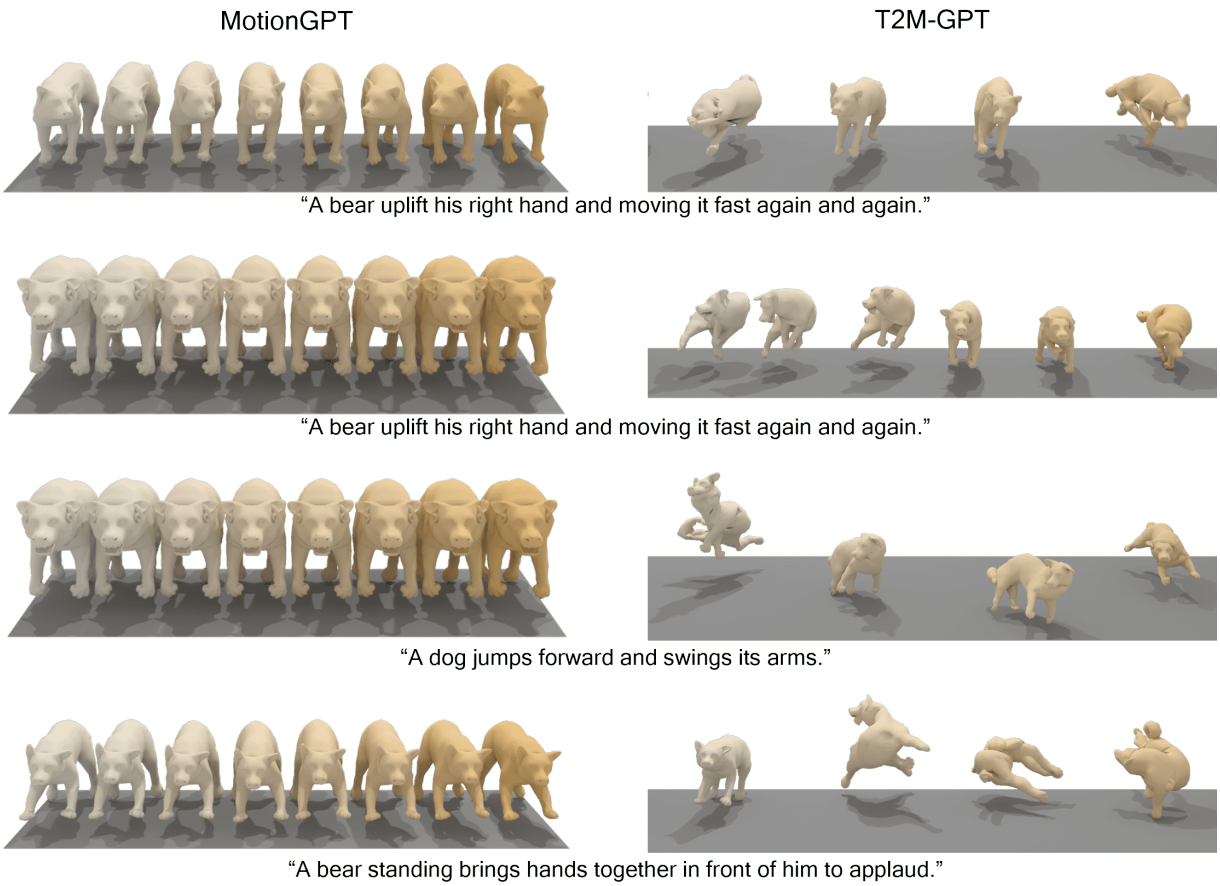

Figure 10. **Generated motions from T2M-GPT and MotionGPT.** Figure illustrates motions generated by T2M-GPT [68] and MotionGPT [22], corresponding to comparisons in Figure 4. These results demonstrate comparatively lower quality, as evidenced by reduced metrics in R-Precision and MM-Dist.