AnyTop: Character Animation Diffusion with Any Topology

Anonymous Author(s)

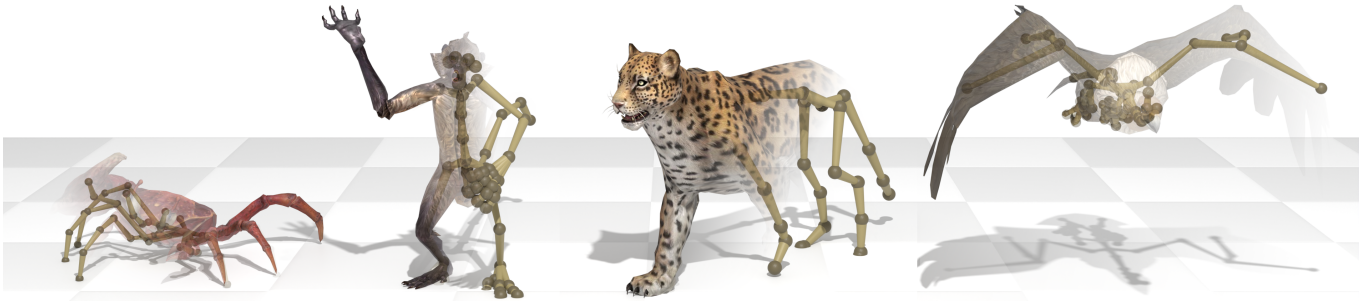

Fig. 1. Anytop generates motions for diverse characters with distinct motion dynamics, using only their skeletal structure as an input.

Generating motion for arbitrary skeletons is a longstanding challenge in computer graphics, remaining largely unexplored due to the scarcity of diverse datasets and the irregular nature of the data. In this work, we introduce AnyTop, a diffusion model that generates motions for diverse characters with distinct motion dynamics, using only their skeletal structure as input. Our work features a transformer-based denoising network, tailored for arbitrary skeleton learning, integrating topology information into the traditional attention mechanism. Additionally, by incorporating textual joint descriptions into the latent feature representation, AnyTop learns semantic correspondences between joints across diverse skeletons. Our evaluation demonstrates that AnyTop generalizes well, even with as few as three training examples per topology, and can produce motions for unseen skeletons as well. Furthermore, our model's latent space is highly informative, enabling downstream tasks such as joint correspondence and temporal segmentation. Our code and dataset will be shared. Please refer to the supplementary video.

CCS Concepts: • **Computing methodologies** → **Motion processing**; *Computer graphics*; *Computer vision*; *Machine learning approaches*.

Additional Key Words and Phrases: Animation, Motion synthesis, Deep Features, Computer Graphics.

1 INTRODUCTION

Character animation is a fundamental task in computer animation, playing a crucial role in industries such as film, gaming, and virtual reality. Animating 3D characters is a complex and time-consuming task that requires manual high-skill effort. Typically, animation pipelines involve a *unique* skeleton for each character, defining its motion span, over which the animation is carefully crafted.

In recent years, neural network-based approaches have simplified the animation process, showing impressive results in tasks such as motion generation and editing [Dabral et al. 2023; Holden et al. 2016; Tevet et al. 2023; Zhang et al. 2024a]. However, most existing methods cannot handle different skeletons and focus on a single topology [Kapon et al. 2023; Shafir et al. 2024], target skeletons that differ only in bone proportions [Tripathi et al. 2025; Yang et al. 2023], or rely on skeletal homeomorphism [Aberman et al. 2020].

While effective within their scopes, these methods overlook the broader opportunities presented by diverse character animation, which require handling a wide variety of skeletal topologies. Conversely, methods designed to handle multiple skeletons often lack scalability, relying on topology-specific adjustments such as additive functional blocks for each skeleton [Li et al. 2024] or entirely distinct instances of the model [Li et al. 2022; Raab et al. 2024].

There are two main reasons keeping arbitrary skeleton animation generation largely under-explored. First, the irregular nature of the data, with skeletons varying in the number of joints and their connectivity, challenges standard methods for processing and analysis. Second, the lack of datasets encompassing diverse topologies presents significant challenges for data-driven approaches.

In this work, we introduce AnyTop, a diffusion framework designed to generate motions for arbitrary skeletal structures, as illustrated in Fig. 1. AnyTop is carefully designed to handle any skeleton in a general manner with no need for topology-specific adjustments.

AnyTop is based on a transformer encoder, specifically adapted for graph learning. While many works embed an entire pose in one tensor [Han et al. 2024; Xie et al. 2023], we embed each joint independently at each frame [Aberman et al. 2020; Agrawal et al. 2024], enabling capturing both joint interactions within the skeleton and universal joint behaviors across diverse skeletal structures. AnyTop applies attention along both the temporal and skeletal axes. Notably, the skeletal attention is between *all* joints. This is in contrast to previous art, and is made possible thanks to our topological conditioning scheme; we integrate graph characteristics [Park et al. 2022; Ying et al. 2021a], such as joint parent-child relations, into the attention maps. Consequently, each joint has access to information from all skeletal parts while also being able to prioritize topologically closer joints. Furthermore, to bridge the gap between similarly behaved parts in different skeletons, AnyTop incorporates textual descriptions of joints into the latent feature representation.

AnyTop is trained on Truebones Zoo dataset [Truebones Motions Animation Studios 2022], which includes motion captures of diverse skeletal structures. We contribute a processed version, aligned with the popular HumanML3D [Guo et al. 2022a] representation, which will be made publicly available. Using quantitative and qualitative evaluations, we show that AnyTop outperforms current art.

Our model demonstrates three forms of generalization in its generations: *In-skeleton Generalization* allows for new motion variants that preserve the character's original motion motifs; *Cross-skeleton*

generalization facilitates generating motions that adapt motifs from several characters; and *Unseen-skeleton generalization* enables motion generation for skeletons not encountered during training. Beyond its generative capabilities, AnyTop's highly informative Diffusion Features (DIFT) [Tang et al. 2023] enable various downstream applications, including unsupervised correlation, temporal segmentation, and motion editing.

The approach presented here, and its ability to share information across characters, opens doors for more flexible generation, better equipped to learn and operate on more complex characters and scenarios, that better fit the real-world needs of 3D content creators.

2 RELATED WORK

Skeletal variability in generative motion models . We refer to four types of skeletal variability (Tab. 1). The naming draws from terminology in the graph domain, hence we interchangeably use the terms joint and vertex, as well as edge and bone. A *single* skeleton type refers to identical skeletons — that is, skeletons with the same vertices, connectivity, and edge lengths. *Isomorphic* skeletons correspond to isomorphic graphs, sharing vertices and edges but potentially differing in edge proportions. *Homeomorphic* skeletons may vary in structure, yet correspond to homeomorphic graphs, *i.e.*, use topologies obtained from the same primal graph by subdivision of edges. Specifically, homeomorphic skeletons share the same number of kinematic chains and end-effectors. Finally, *non-homeomorphic* skeletons vary in their structure and have no common primal graph.

Most motion generative methods focus on a single skeletal structure [Karunratanakul et al. 2023; Petrovich et al. 2021; Raab et al. 2023]. Others train on isomorphic skeletons [Villegas et al. 2021; Zhang et al. 2023b], including works that use the SMPL [Loper et al. 2015] body model [Jang et al. 2024; Petrovich et al. 2022; Tripathi et al. 2025] and SMAL [Zuffi et al. 2017] body model or its derivatives [Rueegg et al. 2023; Yang et al. 2023]. A smaller portion of generative works support homeomorphic skeletons [Cao and Yang 2024; Lee et al. 2023; Ponton et al. 2024; Studer et al. 2024; Zhang et al. 2024d]. Among these works, some [Aberman et al. 2020] require a designated encoder and decoder per skeleton, and some [Zhang et al. 2024b] offer a unified framework for all skeletons.

Only a handful of works can handle non-homeomorphic skeletons. Martinelli et al. [2024] performs motion retargeting by learning a shared manifold for all skeletons, and decoding it to motions using learned skeleton-specific tokens. The learned tokens capture the skeletal information of characters in the dataset, limiting the model's ability to generalize to skeletons unseen during training. Its results are shown exclusively on bipeds, leaving the applicability to other character families (*e.g.*, quadrupeds, millipedes) unexplored. WalkTheDog [Li et al. 2024] uses a latent space that encodes motion phases and accommodates non-generative motion matching.

A different class of generative models bypasses the handling of the skeletal structure by generating motion directly from point clouds [Mo et al. 2025], shape-handles [Zhang et al. 2023a] or meshes [Muralikrishnan et al. 2024; Song et al. 2023; Ye et al. 2024; Zhang et al. 2024c]. These works demonstrate great flexibility in target character structure, but overlook the advantage of skeletons, which are more compact and semantically meaningful, easier to manipulate via rig-based animation, and compatible with physics engines [Tevet et al. 2024] and inverse kinematics systems. Some works [Wang et al. 2024] perform automatic rigging after the generation, but automatic rigging often necessitates manual adjustments.

Finally, methods that support arbitrary skeletons [Li et al. 2022; Raab et al. 2024] involve a separate training process for each skeleton, exhibiting scaling issues and lacking Cross-skeleton generalization.

AnyTop addresses training on non-homeomorphic skeletons and is the only skeletal-based approach capable of generating natural, smooth motions on a diverse range of characters, including bipeds (e.g., raptor, bird), quadrupeds (e.g., dog, bear), multi-legged arthropods (e.g., spider, centipede), and limbless creatures (e.g., snakes). To the best of our knowledge, our work is the only one capable of accepting an input topology, including unseen ones, and generating motions based on that topology.

Table 1. **Skeletal Variability.** Character skeletons can vary in edge length, kinematic chain complexity, or overall topology. Each level of variation introduces greater challenges for motion synthesis. AnyTop can generate motions for dozens of non-homeomorphic skeletons using a single model.

Skeleton Variability type	Edge lengths variations	Kinematic chains variations	Primal skeleton variations
Single	✗	✗	✗
Isomorphic	✓	✗	✗
Homeomorphic	✓	✓	✗
Non-homeomorphic	✓	✓	✓

Transformer-based Graph Learning. Early versions of deep networks on graphs relied on convolutional architectures [Kipf and Welling 2016]. The emergence of transformers has sparked a new avenue of research, integrating graphs and transformers. GAT [Veličković et al. 2018] replace the graph-convolution operation with a self-attention module, where attention is restricted to neighboring nodes. Rong et al. [2020] iteratively stack self-attention layers alongside graph convolutional ones to account for long-range interactions between nodes. Unlike transformers in the language and imaging domains, and due to the irregular structure of graphs, these earlier works do not use positional encoding.

Subsequent works [Dwivedi and Bresson 2021; Kreuzer et al. 2021] linearize the graphs into an array of nodes and add absolute positional encoding to each node. However, linearization is unnatural to the graph structure, requiring a reconsideration of the approach.

Encoding relative positional information has been explored to maintain positional precision while adhering to the graph's structure. Works using it [Park et al. 2022; Shaw et al. 2018; Ying et al. 2021b] integrate relative positional encoding into the attention map based on relative measures, such as shortest path distance between nodes or edge type.

The aforementioned approaches are *discriminative*, applied to tasks such as regression and segmentation. AnyTop leverages the relative positional encoding approach for *generative* tasks and tailors it to the *motion* domain. In particular, our work redefines edge types to capture joint relations within skeletal structures and considers a temporal axis, which is not present in the graph domain.

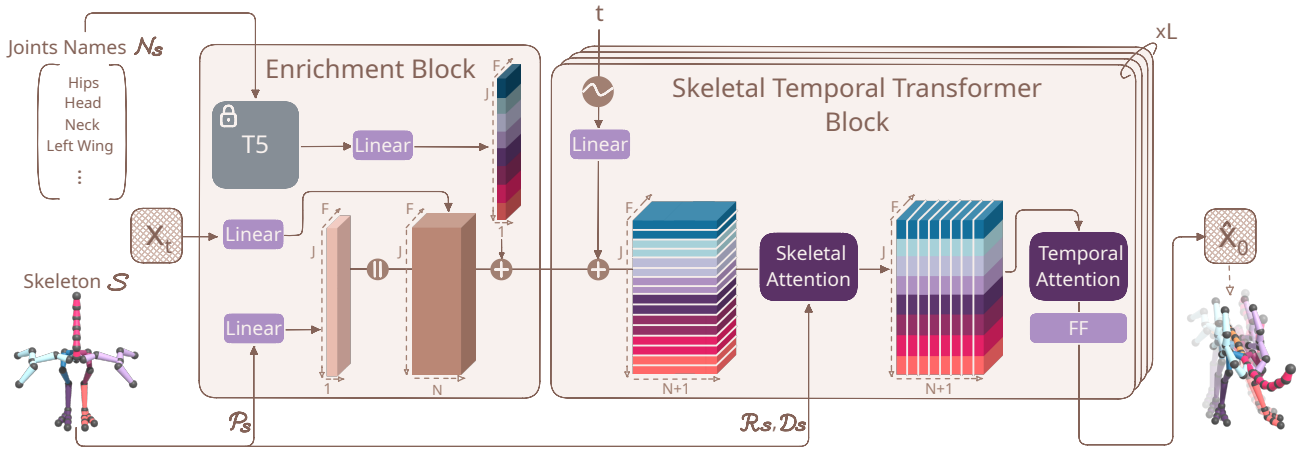

Fig. 2. **Overview.** The input to AnyTop is a noised motion X_t and the skeleton $\mathcal{S} = \{\mathcal{P}_S, \mathcal{R}_S, \mathcal{D}_S, \mathcal{N}_S\}$, where $\mathcal{P}_S$ is the rest-pose, $\mathcal{R}_S$ is the joints relations, $\mathcal{D}_S$ is the topological distances between each pair of joints and $\mathcal{N}_S$ is the joints names. The *Enrichment Block* incorporates the skeletal features into the noised motion by concatenating the embedded $\mathcal{P}_S$ to the sequence and adding a T5-embedded name to each joint. The enriched motion is then passed through a stack of L *Skeletal Temporal Transformer* layers. We apply skeletal attention along the joint axis, capturing interactions between all joints, and incorporate the topology information $\mathcal{R}_S$ and $\mathcal{D}_S$ to attention maps. Next, we apply temporal attention along the frames axis. Finally, the output is projected back to the motion features dimension, facilitating the reconstruction of the motion sequence.

3 METHOD

AnyTop is a diffusion model synthesizing motions for multiple different characters with arbitrary skeletons. Given a skeletal structure for input, it generates a natural motion sequence with high fidelity to ground-truth characters. AnyTop is based on a transformer encoder, specifically adapted for graph learning, as depicted in Fig. 2.

3.1 Preliminaries

Motion Representation. We represent motion as a 3D tensor $X \in \mathbb{R}^{N \times J \times D}$, where N and J are the maximum number of frames and joints across all motions in the dataset, and D is the number of motion features per joint. As motions vary in duration and skeletal structure, we pad the original number of frames and joints of each motion to match the maximum values N and J, respectively. We adopt a redundant representation, where each joint j (except the root) consists of its root-relative position $p_j \in \mathbb{R}^3$, 6D joint rotation $r_j \in \mathbb{R}^6$ [Zhou et al. 2018], linear velocity $v_j \in \mathbb{R}^3$, and foot contact label $fc_j \in \{0, 1\}$. Altogether a joint is represented by $\{p_j, r_j, v_j, fc_j\} \in \mathbb{R}^{13}$, hence $D = 13$. For the root joint, features include its rotational velocity, linear velocity and height, which are concatenated and zero-padded to match the size D. Our representation is inspired by Guo et al. [2022a]; however, our approach maintains features at the joint level by representing each joint as a separate tensor, resulting in J tokens per frame. In contrast, Guo et al. concatenate features from all joints into one tensor, resulting in a single token per frame.

Skeletal structure Representation. In the context of 3D motion, *topology* is a directed, acyclic, and connected graph (DAG). Adding geometric information to this graph makes it a *skeleton*. We use the terms "topology" and "skeleton" interchangeably throughout this work, clarifying any distinction when necessary. A *rest-pose* is the character's natural pose, represented by $(\mathcal{G}, O)$, where $\mathcal{G}$ is a DAG defining the topological hierarchy and $O \in \mathbb{R}^{J \times 3}$ is a set of 3D offsets, specifying each joint's parent-relative position. In our work, we represent a skeleton by $\mathcal{S} = \{\mathcal{P}_S, \mathcal{R}_S, \mathcal{D}_S, \mathcal{N}_S\}$. The first term, $\mathcal{P}_S \in \mathbb{R}^{J \times D}$, is the rest-pose, converted to the format of individual poses in the motion sequence. The second term, $\mathcal{R}_S \in \mathbb{N}_0^{J \times J}$, is the joints relations, where $\mathcal{R}_S[i, j]$ holds the relation type between i and j. We allow six types of relations, which are *child*, *parent*, *sibling*, *no-relation*, *self* and *end-effector*. *Self* and *end-effector* are valid only in case $i = j$, and *end-effector* specifies if the joint is a leaf in $\mathcal{G}_S$. The third term, $\mathcal{D}_S \in \mathbb{N}_0^{J \times J}$, represents the graph distances, where $\mathcal{D}_S[i, j]$ holds the topological distance between i and j in $\mathcal{G}_S$, up to a maximal distance d_{max}. The topological conditions, $\mathcal{R}_S$ and $\mathcal{D}_S$, are illustrated in Fig. 3. Finally, $\mathcal{N}_S$ is the joints' textual descriptions, which are typically included in 3D asset formats (*e.g.*, bvh, fbx).

3.2 Architecture

AnyTop is a generative Denoising Diffusion Probabilistic Model (DDPM) [Ho et al. 2020]. At each denoising step $t \in [1, T]$ it gets a noisy motion X_t and a skeleton $\mathcal{S} = \{\mathcal{P}_S, \mathcal{R}_S, \mathcal{D}_S, \mathcal{N}_S\}$ as input, and predicts the clean motion $\hat{X}_0$ [Tevet et al. 2023] rather than the noise ϵ_t.

AnyTop consists of two primary components, illustrated in Fig. 2. The first is an *Enrichment Block*, which integrates skeleton-specific information into the noised motion. The second is a *Skeletal Temporal Transformer Block*, which employs attention across both skeletal and temporal axes while embedding topological information into the skeletal attention maps.

Enrichment block. This block incorporates semantic information from the rest-pose $\mathcal{P}_S$ and the joint descriptions $\mathcal{N}_S$, into the noised sample X_t. It projects $\mathcal{P}_S$ to feature lentgh F and concatenates it

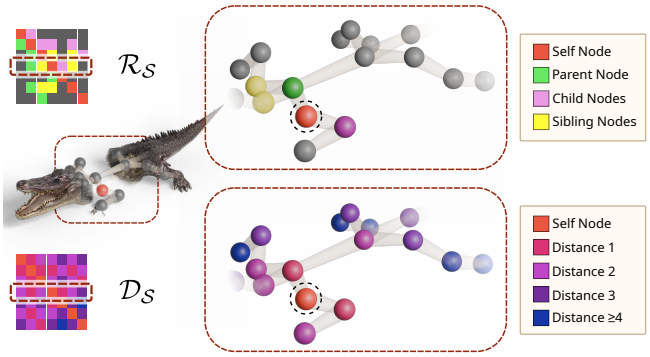

Fig. 3. **Topological Conditions.** Joint relations $\mathcal{R}_\mathcal{S}$ (top) and graph distances $\mathcal{D}_\mathcal{S}$ (bottom), visualized for a specific joint marked in red. Different colors indicate different values in the row corresponding to the visualized joint in the $\mathcal{R}_\mathcal{S}$, $\mathcal{D}_\mathcal{S}$ matrices.

with X_t along the temporal axis, effectively making it frame 0. The joint descriptions $\mathcal{N}_\mathcal{S}$ are encoded by a T5 model, projected to length F, and added to their corresponding joint features across all frames. Finally, the block outputs enhanced data of shape $\mathbb{R}^{(N+1)\times J\times F}$.

Skeletal Temporal Transformer block. The inputs to this block are the embedded tokens of X_t emitted from the *Enrichment Block*, the diffusion step t, and the precomputed values $\mathcal{D}_\mathcal{S}, \mathcal{R}_\mathcal{S}$. The block comprises a stack of L identical *Skeletal Temporal Transformer* (STT) encoder layers, each consisting of three parts. The first component is a *Skeletal Attention*, which performs spatial self-attention across joints within the same frame. Unlike concurrent approaches that limit attention or convolution to adjacent joints within the skeletal hierarchy, our method enables each joint to attend to all others, capturing long-range relations. To regain the local joint knowledge, we incorporate topology information $\mathcal{R}_\mathcal{S}$ and $\mathcal{D}_\mathcal{S}$ into the attention maps. This allows each joint to access information from all skeletal parts while also prioritizing topologically closer joints.

The second component is a *Temporal Attention*, which applies self-attention along the temporal axis for each joint independently, observing its motion over time. To enhance efficiency and mitigate overfitting, the temporal attention is applied within a temporal window of length W. The third component is a feed-forward block. Finally, the output is projected to the original motion dimension, enabling motion reconstruction.

Incorporating Graph-based Features via Skeletal Attention. We extend transformers for graph-based learning by incorporating both graph topology and node interaction information through our *Skeletal Attention* mechanism. Inspired by *discriminative* works in the *graphs* domain [Ying et al. 2021b], AnyTop introduces a novel method for *generative* tasks, specifically tailored to the *motion* domain. We integrate graph properties directly into attention maps, enabling the structural characteristics of the graph to influence the learning process. Our work uses two types of node affinity, the topological distance, $\mathcal{D}_\mathcal{S}$, and relations, $\mathcal{R}_\mathcal{S}$, as detailed in Sec. 3.1. We incorporate the graph information into the attention maps [Park et al. 2022], by learning distinct query and key embeddings for distances, denoted by $E_q^\mathcal{D}$, $E_k^\mathcal{D} \in \mathbb{R}^{d_{max}\times F}$, and embeddings for relation, denoted by $E_q^\mathcal{R}$, $E_k^\mathcal{R} \in \mathbb{R}^{6\times F}$, where $E_q^{(\cdot)}$ and $E_k^{(\cdot)}$ denote embeddings that relate to queries and keys, respectively, and F is the latent feature size. These embeddings are used to form two new attention maps, $a^\mathcal{D}$ and $a^\mathcal{R}$ defined for a given pair of joints $i, j \in [J]$:

$$a_{ij}^\mathcal{D} = q_i \cdot E_q^\mathcal{D}[\mathcal{D}_{ij}] + k_j \cdot E_k^\mathcal{D}[\mathcal{D}_{ij}], \tag{1}$$

$$a_{ij}^\mathcal{R} = q_i \cdot E_q^\mathcal{R}[\mathcal{R}_{ij}] + k_j \cdot E_k^\mathcal{R}[\mathcal{R}_{ij}], \tag{2}$$

where q_i, k_j denote the i'th joint query and j'th joint key, respectively, and $[\cdot]$ denotes an index in the embedding matrix. Finally, we incorporate graph information by adding the two attention maps to the standard attention map and scaling their sum:

$$a_{ij} = \frac{q_i \cdot k_j + a_{ij}^\mathcal{D} + a_{ij}^\mathcal{R}}{\sqrt{F}}. \tag{3}$$

The final attention score is computed by applying the standard row-wise softmax to a_{ij}.

3.3 Training

Data Sampling and Augmentations. We train AnyTop using minibatches sampled with a *Balancing Sampler* to address the imbalanced nature of the data (described in Sec. 6.1) and mitigate the dominance of specific skeletons. To further enhance generalization, we apply skeletal augmentations to the data samples, including randomly removing 10% to 30% of the joints and adding new joints at the midpoint of existing edges. Further details on our data augmentation are provided in the sup. mat.

Training Objectives. Given a motion X_0 of skeleton $\mathcal{S}$, its noised counterpart X_t, with diffusion step $t \sim [1, T]$, our model predicts the clean motion, $\hat{X}_0 = AnyTop(X_t, t, \mathcal{S})$. Our main objective is defined by the *simple* formulation [Ho et al. 2020], namely,

$$\mathcal{L}_{simple} = E_{t\sim[1,T]} \left\| \hat{X}_0 - X_0 \right\|_2^2. \tag{4}$$

The Mean Squared Error (MSE) over rotations does not directly correlate to their distance in the rotation space, hence we apply a geodesic loss [Huang et al. 2017; Tripathi et al. 2025] over the learned rotations. Let $r, \hat{r} \in \mathbb{R}^{N\times J\times 6}$ denote the 6D rotations of X_0 and $\hat{X}_0$ respectively. The geodesic loss is defined as follows:

$$\mathcal{L}_{rot} = \sum_{n=1}^{N}\sum_{j=1}^{J} \arccos \frac{Tr(GS(r_{n,j})(GS(\hat{r}_{n,j})^T) - 1}{2}, \tag{5}$$

where GS is the Gram-Schmidt process, used to convert 6D rotations to rotation matrices [Zhou et al. 2019], and Tr is the matrix Trace operation. Overall, the final training objective is

$$\mathcal{L} = \mathcal{L}_{simple} + \lambda_{rot}\mathcal{L}_{rot}. \tag{6}$$

4 ANALYSIS

4.1 Latent Space Analysis

In this section, we examine AnyTop's latent space and show that it features a unified manifold for joints across all skeletons. We use DIFT [Tang et al. 2023], a framework designed for detecting correspondence in the latent space of models undergoing diffusion. DIFT features are intermediate activations from layer l_{corr}, extracted

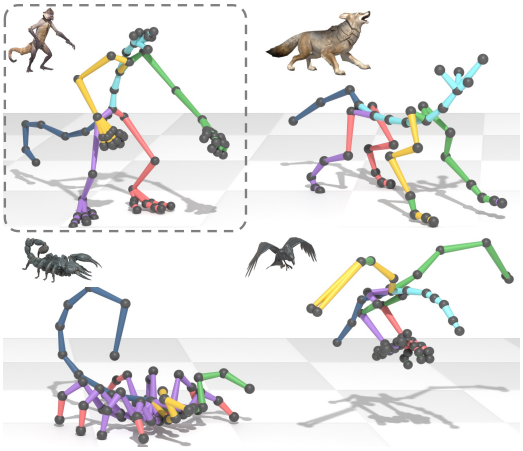

Fig. 4. **Spatial Correspondence.** Monkey (top left) depicts the reference skeleton, while the fox, scorpion and bird depict different target skeletons. Targets skeletons joints are color-coded to match their corresponding joints in the reference. For better visualization, we color the bones to match their adjacent joints. Note the correspondence in limbs, spine, and tail.

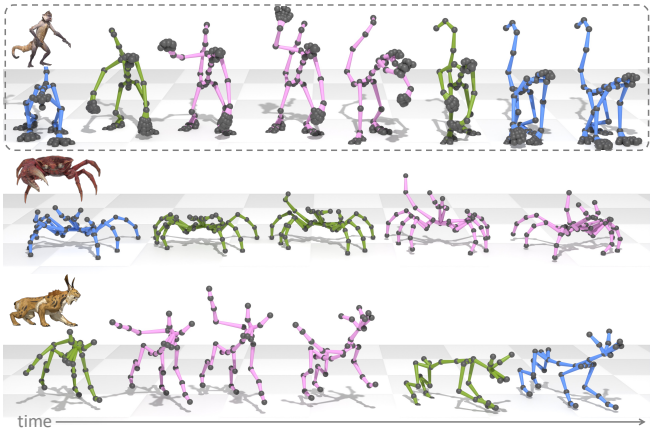

Fig. 5. **Temporal Correspondence.** Monkey (top row) features the reference motion, while the Crab and Lynx represent two target motions. The frames of the targets are color-coded to align with their corresponding reference frames. Note the correspondence: aggressive motion segments are pink, idle frames blue, and transitional frames green.

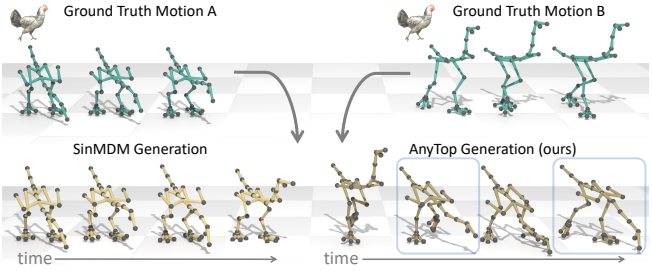

Fig. 6. **In-skeleton Generalization.** The top row depicts two ground truth chicken motions: pecking (left) and walking (right). The bottom row presents synthesized motions of an adapted SinMDM (left) and AnyTop (right). The emphasized frames in AnyTop demonstrate spatial composition of walking and pecking, introducing novel poses not present in the ground truth. SinMDM embeds entire poses, hence cannot spatially-compose joints.

during a single denoising pass on a sample that has been noised directly to diffusion step t_{corr}. These features serve as effective semantic descriptors for predicting correspondence. Note that the values we choose for l_{corr} and t_{corr} align with those used in the original DIFT work. Let X^{ref} denote a reference motion, and let X^{tgt} denote a motion in which we search for corresponding parts. Let S^{ref}, S^{tgt} denote their skeletons, respectively.

Our spatial and temporal correspondence results are illustrated in Figs. 4 and 5 respectively, and in the supplementary video.

Spatial Correspondence. We show that manifold features of semantically similar skeletal joints across different characters are close to each other. Our objective is to find the most similar *joint* in $\mathcal{S}^{ref}$ for each joint in $\mathcal{S}^{tgt}$. To achieve this, we extract DIFT features for both motions X^{ref}, X^{tgt} at diffusion step $t_{corr} = 2$ and layer $l_{corr} = 0$, average them along the temporal axis, and obtain a single feature vector per joint. Using cosine similarity, we detect the closest counterpart for each joint in S_{tgt}.

Temporal Correspondence. We show that AnyTop can recognize pose-level similarities and identify analogous actions across different skeletons. This time, our objective is to find the most similar *frame* in X^{ref} for each frame in X^{tgt}. To accomplish this goal, we extract DIFT features at diffusion step $t_{corr} = 3$ and layer $l_{corr} = 1$, and average them along the skeletal axis, resulting in a single feature vector per frame. We use cosine similarity to detect the closest counterpart for each frame in X_{tgt}.

4.2 Generalization Forms

We identify three forms of generalization in our generated motions.

In-skeleton Generalization. dubbed in-gen, refers to generalization within a specific skeleton, featured as both *temporal composition* – combining motion segments from dataset instances, and *spatial composition* – introducing novel poses by combining skeletal parts of ground truth poses. Notably, *spatial composition* is enabled by our per-joint encoding, which provides the flexibility required for such diversity. In Fig. 6 and in our supp. video, we showcase AnyTop's in-gen and highlight how other methods, which embed the entire pose, fail to achieve a comparable variety.

Cross-skeleton generalization. dubbed *cross-gen*, captures shared motion motifs across skeletons. This type of generalization is particularly useful for adapting one animal to exhibit the behavior of others, as shown in Fig. 7 and in our video. When motions must strictly align with typical behaviors, the training dataset can be restricted accordingly.

Unseen-skeleton generalization. extends to skeletons not encountered during training, and illustrated in Fig. 8 and the video.

5 APPLICATIONS

AnyTop enables various downstream tasks; we demonstrate two.

Temporal Segmentation. Temporal segmentation is the task of partitioning a temporal sequence into disjoint groups, where frames

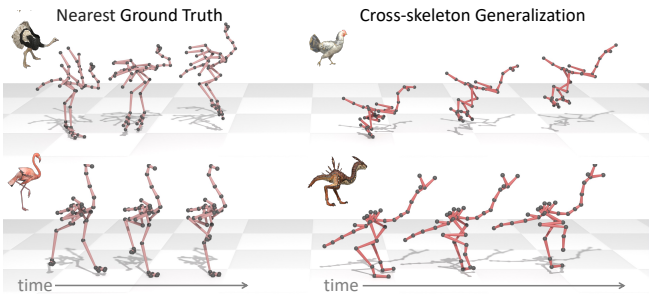

Fig. 7. **Cross-skeleton generalization.** Right: a generated motion featuring an action not in the performing skeleton's ground truth. Left: notably, the nearest ground truth originates from a different character.

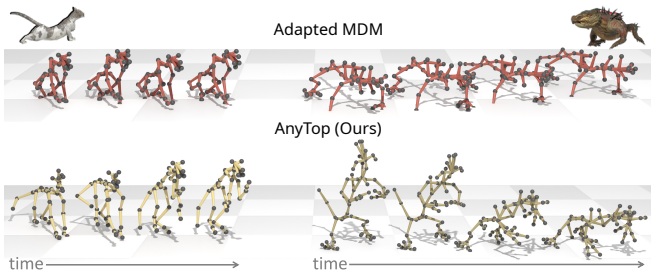

Fig. 8. **Unseen-skeleton generalization** Zero-shot inference of the cat (left) and komodo dragon (right) using AnyTop (top) and adapted MDM baseline (bottom). AnyTop's generated motions maintain natural appearance while MDM's generated motions are static and jittery.

sharing similar characteristics are grouped. For a clean sample X_0, either generated or given, and skeleton $\mathcal{S}$, we extract DIFT features at diffusion step t_{seg} =3 and layer l_{seg} =1. The features are averaged along the joint dimension to produce a single feature vector per frame. We apply PCA for dimensionality reduction and then use K-means to cluster the frames into k=3 categories. Our results are visualized in Fig. 9 and in the supp. video. This application reinforces Sec. 4, showing that AnyTop's latent features are effective frame descriptors. However, in Sec. 4, frames are grouped by similarity to X^{ref}, while here they are grouped by similarity to each other.

Editing. We demonstrate our method's versatility through two motion editing applications: *in-betweening* for temporal manipulation and *body-part editing* for spatial modifications, both leveraging the same underlying approach. For *in-betweening*, the prefix and suffix of the motion are fixed, allowing the model to generate the middle. For *body-part editing*, we fix some of the joints and let the model generate the rest. Given a fixed subset (temporal or spatial) of the motion sequence tokens, we override the denoised $\hat{x}_0$ at each sampling iteration with the fixed motion part. This approach ensures fidelity to the fixed input while synthesizing the missing elements of the motion. Our results, in Fig. 10 and the supp. video, show a smooth and natural transition between the given and the synthesized parts, and demonstrate that our model successfully generalizes techniques previously limited to human skeletons [Tevet et al. 2023] to accommodate diverse skeletal structures.

6 EXPERIMENTS

6.1 Dataset and Preprocessing

The Truebones Zoo [Truebones Motions Animation Studios 2022] dataset comprises motion captures featuring 70 diverse skeletons, including mammals, birds, insects, dinosaurs, fish, and snakes. The number of motions per skeleton ranges from 3 to 40, adding up to 1219 motions and 147,178 frames in total. The dataset includes variations in orientation, root definition, and scale. Additionally, the skeletons vary in joint order, naming conventions, and connectivity standards. To address these variations, we have performed comprehensive preprocessing of the data, including aligning all motions to the same orientation and average bone length, centering the first frame at the origin, and ensuring it is located on the ground. This process is described in details in the sup. mat. and the processed data will be made available.

Skeletal Subsets. In addition to experimenting with the full dataset, we categorize the skeletons into four groups based on their motion dynamics and train AnyTop on these subsets, alongside a model trained on the entire dataset. The four skeletal categories are *Quadrupeds*, *Bipeds*, *Flying*, and *Insects*. These subsets allow us to constrain *cross-gen* to characters with similar behavior. Our visualizations illustrate generations from models trained on the entire dataset or sub-datasets, depending on the context (*e.g.*, Fig. 8).

6.2 Implementation details

We use $T = 100$ diffusion steps, $L = 4$ STT layers, and latent dimension $F = 128$. We train the model using a single NVIDIA RTX A6000 GPU for 24 hours. Inference runs on an NVIDIA GeForce RTX 2080 Ti GPU. More implementation details can be found in the sup. mat.

6.3 Evaluation

Benchmark. To evaluate AnyTop, we introduce a benchmark comprising 30 skeletons randomly selected from those with cumulative frame counts ranging between 600 and 1200. The benchmark includes 43% Quadrupeds, 17% Bipeds, 23% Flying, and 17% Insects, reflecting the relative proportions of these categories in the dataset.

Metrics. We report four metrics that measure different aspects of the generated motions, following Li et al. [2022]; Raab et al. [2024]. The metrics are calculated separately for each skeleton, and the mean and standard deviation across all tested skeletons are reported in the form $mean^{\pm std}$. For each skeleton, we evaluate a number of samples proportional to its sample count in the dataset. Let M,G denote the group of ground truth (GT) and generated motions of the assessed skeleton, respectively. The metrics that we use are (a) *coverage*, which is the rate of temporal windows in M, that are reproduced in G, (b) *local diversity*, which is the average distance between windows in G and their nearest neighbors in M, and (c) *inter diversity*, the diversity between synthesized motions. We define *intra diversity* to be the diversity between sub-windows internal to a motion and define (d) *intra diversity diff*, which is the difference between the intra diversity of G and that of M. Metrics (a) and (d) evaluate fidelity to the GT, while metrics (b) and (c) assess diversity. An ideal score features both high fidelity and high diversity. High

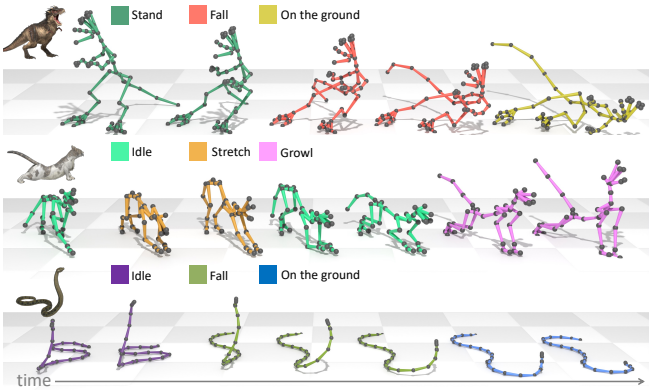

Fig. 9. **Temporal Segmentation.** Temporal clustering on a tyrannosaurus, a cat, and an anaconda snake, using K-means on PCA-reduced DIFT features.

fidelity with low diversity suggests overfitting, while low fidelity with high diversity indicates divergence and noise.

6.4 Baselines

To the best of our knowledge, no current works address such a diverse range of skeletal structures within a single model. Hence, we compare AnyTop to adaptations of two baselines. The first is *MDM* [Tevet et al. 2023], originally designed for a single humanoid skeleton. MDM uses per-frame embedding, so to match its representation format, we concatenate all joint features for each character, and pad them to a length of $J \times D$. For fairness, we also concatenate the vectorized rest-pose embedding $\mathcal{P}_{\mathcal{S}}$ along the temporal axis as frame 0. Since MDM accepts textual conditions, we use the skeleton's name (e.g., Cat, Dragon) as the input text. Additionally, since MDM's original configuration was designed for a dataset 14 times larger than ours [Guo et al. 2022b], we reduced its latent dimension size to mitigate overfit.

The second baseline is *SinMDM* [Raab et al. 2024], designed to be trained on a *single* motion sequence. We modify it to enable training on multiple sequences of the same character, resulting in a separate model for each skeleton.

6.5 Quantitative Results

Table 2 shows a quantitative comparison of AnyTop and the baselines. AnyTop outperforms MDM in all categories and SinMDM in all but coverage, which is expected since SinMDM is trained separately for each skeleton. Note the significant gap in diversity metrics, where the table shows AnyTop generalizes well, while the others struggle to do so. We also report the models' parameter count,

Table 2. **Comparison with baselines.** Our model clearly outperforms the baselines. **Bold** and underline denote best and second best, respectively. * indicates the work was adapted to align with the terms of our experiment.

Model	Coverage ↑	Local Div. ↑	Inter Div. ↑	Intra Div. Diff. ↓	#Param. (M) ↓
MDM* [2023]	$71.3^{\pm 31}$	$\underline{0.168}^{\pm 0.12}$	$0.139^{\pm 0.13}$	$0.177^{\pm 0.08}$	$\underline{5.96}$
SinMDM* [2024]	$\mathbf{89.3}^{\pm 15}$	$0.080^{\pm 0.13}$	$\underline{0.280}^{\pm 0.13}$	$\underline{0.144}^{\pm 0.09}$	176.1 (5.87 × 30)
AnyTop (Ours)	$\underline{80.5}^{\pm 20}$	$\mathbf{0.252}^{\pm 0.14}$	$\mathbf{0.312}^{\pm 0.17}$	$\mathbf{0.118}^{\pm 0.07}$	**2.28**

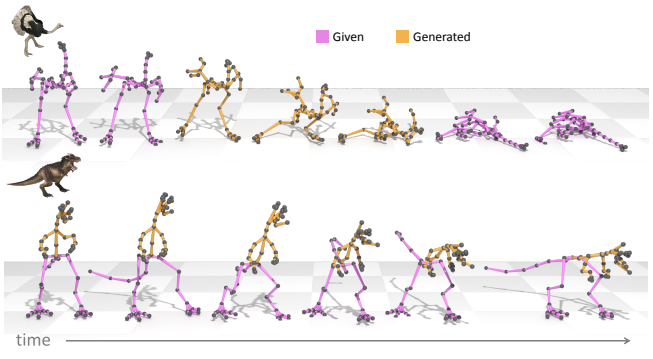

Fig. 10. **Editing.** Top: In-betweening. Given motion prefix and suffix, AnyTop can generate the middle frames. Bottom: Body part editing. Given the motion of the lower body, AnyTop can generate its complement for the upper body. Both editing strategies produce smooth and natural transition between the given and the synthesized parts.

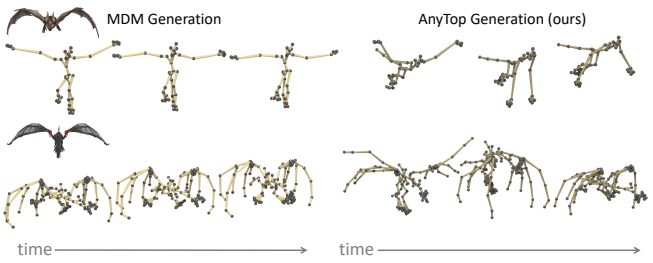

Fig. 11. **Comparison with MDM Baseline.** AnyTop (right) generates natural motions, while MDM (left) produces static, jittery motions.

showing ours uses fewer parameters, enabling lower computation and faster inference. In the sup. mat., we provide a comparison with the baselines on the data subsets, demonstrating our model's superiority on these as well.

6.6 Qualitative Results

Our supp. video reflects the quality of our results. It presents generated motions for various skeletons and comparisons to baselines.

In Fig. 11, we show that AnyTop produces natural and lively motions while MDM produces static, jittery motions. Moreover, MDM's results in our video show jittery transitions and unnatural poses. Figure 6 and our supp. video show that AnyTop can generate novel poses by effectively combining joints from different ground truth poses. In contrast, SinMDM is limited to temporal in-skeleton generalization and cannot handle spatial composition, due to its reliance on per-frame features. Moreover, since SinMDM trains a separate model per skeleton, it cannot feature cross-skeleton or unseen-skeleton generalization. As accurate foot contact is one of the major factors of motion quality, we follow Li et al. [2022]; Raab et al. [2023] and use an IK post-process to ensure proper contact.

Unseen skeleton. We present two unseen skeleton motions. One is a komodo dragon, generated by the *Bipeds* model. The second is a Cat, generated by a model trained on *Quadrupeds*, excluding the cat.

Figure 8 and our supp. video demonstrate AnyTop generalizes well to unseen skeletons, while adapted MDM under the same settings generates static and jittery motions.

6.7 Ablation

In Tab. 3, we explore three key components of AnyTop's architecture. First, the results confirm that without access to topological information, the model struggles to prioritize joints based on their hierarchical relations. Omitting the incorporation of $\mathcal{D}$ and $\mathcal{R}$ leads to degradation in all metrics. Next, excluding the rest pose $\mathcal{P}_{\mathcal{S}}$ produces inferior results, reinforcing the idea that $\mathcal{P}_{\mathcal{S}}$ encodes vital information about joint offsets and bone lengths. Lastly, we examine cross-skeletal prior sharing via the addition of joint name embeddings. While *cross-gen* improves motion diversity, it introduces a tradeoff, as generated motions may exhibit motifs absent in the skeleton's ground truth, reducing coverage. Results show that removing joint name embeddings increases coverage but severely sacrifices diversity and cross-skeleton generalization.

7 CONCLUSION, LIMITATIONS AND FUTURE WORK

We have presented AnyTop, a generative model that synthesizes diverse characters with distinct motion dynamics using a skeletal structure as input. It uses a transformer-based denoising network, integrating graph information at key points in the pipeline. Our evaluation shows a highly informative latent space and notable generalization, even for characters with few or no training samples.

One limitation of our method stems from imperfections in the input data. Despite our cleaning procedure, certain data artifacts remain unresolved. Another limitation is that our data augmentation process is computationally expensive with $O(J^2)$ complexity.

In the future, we plan to use AnyTop for skeletal retargeting, multi-character interaction, editing, and various control modalities such as text-based and music-driven animation. Another potential direction is editing animations by simply modifying joint labels in the text descriptions. Finally, future work could further explore DIFT features in the motion domain.

Table 3. **Ablation.** Removing architectural choices leads to a degradation in AnyTop's performance.

Model	Coverage ↑	Local Div. ↑	Inter Div. ↑	Intra Div. Diff. ↓
w/o graph property embedding ($\mathcal{D}, \mathcal{R}$)	$76.8^{\pm 23}$	$0.249^{\pm 0.14}$	$\underline{0.303}^{\pm 0.17}$	$0.127^{\pm 0.11}$
w/o rest-pose token	$\underline{77.2}^{\pm 25}$	$\underline{0.250}^{\pm 0.14}$	$0.292^{\pm 0.18}$	$0.130^{\pm 0.09}$
w/o joint name embedding	$\mathbf{82.3}^{\pm 17}$	$0.218^{\pm 0.12}$	$0.276^{\pm 0.15}$	$\mathbf{0.113}^{\pm 0.06}$
AnyTop (Ours)	$\underline{80.5}^{\pm 20}$	$\mathbf{0.252}^{\pm 0.14}$	$\mathbf{0.312}^{\pm 0.17}$	$\underline{0.118}^{\pm 0.07}$

REFERENCES

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 62–1.

Dhruv Agrawal, Jakob Buhmann, Dominik Borer, Robert W Sumner, and Martin Guay. 2024. SKEL-Betweener: a Neural Motion Rig for Interactive Motion Authoring. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–11.

Yu Cao and MingHui Yang. 2024. CAR: Collision-Avoidance Retargeting for Varied Skeletal Architectures. In *SIGGRAPH Asia 2024 Technical Communications*. ACM, New York, NY, USA, 1–5.

Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. 2023. MoFusion: A Framework for Denoising-Diffusion-based Motion Synthesis. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Washington, DC, USA.

Vijay Prakash Dwivedi and Xavier Bresson. 2021. A Generalization of Transformer Networks to Graphs. In *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*. AAAI Press, Washington, DC, USA.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. 2022a. Generating Diverse and Natural 3D Human Motions From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Washington, DC, USA, 5152–5161.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. 2022b. Generating Diverse and Natural 3D Human Motions From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5152–5161.

Bo Han, Hao Peng, Minjing Dong, Yi Ren, Yixuan Shen, and Chang Xu. 2024. AMD: Autoregressive Motion Diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. AAAI Press, Washington, DC, USA, 2022–2030.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

Daniel Holden, Jun Saito, and Taku Komura. 2016. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. 2017. Deep Learning on Lie Groups for Skeleton-Based Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Washington, DC, USA.

Inseo Jang, Soojin Choi, Seokhyeon Hong, Chaelin Kim, and Junyong Noh. 2024. Geometry-Aware Retargeting for Two-Skinned Characters Interaction. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–17.

Roy Kapon, Guy Tevet, Daniel Cohen-Or, and Amit H. Bermano. 2023. MAS: Multi-view Ancestral Sampling for 3D motion generation using 2D diffusion. arXiv:2310.14729 [cs.CV]

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. 2023. Guided Motion Diffusion for Controllable Human Motion Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE Computer Society, Washington, DC, USA, 2151–2162.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. 2021. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems* 34 (2021), 21618–21629.

Sunmin Lee, Taeho Kang, Jungnam Park, Jehee Lee, and Jungdam Won. 2023. Same: Skeleton-agnostic motion embedding for character animation. In *SIGGRAPH Asia 2023 Conference Papers*. ACM, New York, NY, USA, 1–11.

Peizhuo Li, Kfir Aberman, Zihan Zhang, Rana Hanocka, and Olga Sorkine-Hornung. 2022. GANimator: Neural Motion Synthesis from a Single Sequence. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 138.

Peizhuo Li, Sebastian Starke, Yuting Ye, and Olga Sorkine-Hornung. 2024. WalkTheDog: Cross-Morphology Motion Alignment via Phase Manifolds. In *ACM SIGGRAPH 2024 Conference Papers*. ACM, New York, NY, USA, 1–10.

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

Giulia Martinelli, Nicola Garau, Niccoló Bisagno, and Nicola Conci. 2024. MoMa: Skinned motion retargeting using masked pose modeling. *Computer Vision and Image Understanding* 249 (2024), 104141.

Clinton Mo, Kun Hu, Chengjiang Long, Dong Yuan, and Zhiyong Wang. 2025. Motion Keyframe Interpolation for Any Human Skeleton via Temporally Consistent Point Cloud Sampling and Reconstruction. In *European Conference on Computer Vision*. Springer, Springer International Publishing, Berlin/Heidelberg, Germany, 159–175.

Sanjeev Muralikrishnan, Niladri Dutt, Siddhartha Chaudhuri, Noam Aigerman, Vladimir Kim, Matthew Fisher, and Niloy J. Mitra. 2024. Temporal Residual Jacobians for Rig-Free Motion Transfer. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LVIII* (Milan, Italy). Springer-Verlag, Berlin, Heidelberg, 93–109. https://doi.org/10.1007/978-3-031-73636-0_6

Wonpyo Park, Woong-Gi Chang, Donggeon Lee, Juntae Kim, and Seungwon Hwang. 2022. GRPE: Relative Positional Encoding for Graph Transformer. In *ICLR2022 Machine Learning for Drug Discovery*. OpenReview.net, OpenReview.net.

Mathis Petrovich, Michael J. Black, and Gül Varol. 2021. Action-Conditioned 3D Human Motion Synthesis with Transformer VAE. In *International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Washington, DC, USA, 10985–10995.

Mathis Petrovich, Michael J. Black, and Gül Varol. 2022. TEMOS: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision (ECCV)*. Springer International Publishing, Berlin/Heidelberg, Germany.

Jose Luis Ponton, Eduard Pujol, Andreas Aristidou, Carlos Andujar, and Nuria Pelechano. 2024. Dragposer: Motion reconstruction from variable sparse tracking signals via latent space optimization.

Sigal Raab, Inbal Leibovitch, Peizhuo Li, Kfir Aberman, Olga Sorkine-Hornung, and Daniel Cohen-Or. 2023. MoDi: Unconditional Motion Synthesis from Diverse Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Washington, DC, USA, 13873–13883.

Sigal Raab, Inbal Leibovitch, Guy Tevet, Moab Arar, Amit H Bermano, and Daniel Cohen-Or. 2024. Single Motion Diffusion. In *The Twelfth International Conference on Learning Representations (ICLR)*. OpenReview.net, OpenReview.net. https://openreview.net/pdf?id=DrhZneqz4n

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. 2020. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems* 33 (2020), 12559–12571.

Nadine Rueegg, Silvia Zuffi, Konrad Schindler, and Michael J Black. 2023. Barc: Breed-augmented regression using classification for 3d dog reconstruction from images. *International Journal of Computer Vision* 131, 8 (2023), 1964–1979.

Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. 2024. Human Motion Diffusion as a Generative Prior. In *The Twelfth International Conference on Learning Representations*. OpenReview.net, OpenReview.net.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *Proceedings of NAACL-HLT*. Association for Computational Linguistics, Pennsylvania, USA, 464–468.

Chaoyue Song, Jiacheng Wei, Ruibo Li, Fayao Liu, and Guosheng Lin. 2023. Unsupervised 3d pose transfer with cross consistency and dual reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 8 (2023), 10488–10499.

Justin Studer, Dhruv Agrawal, Dominik Borer, Seyedmorteza Sadat, Robert W Sumner, Martin Guay, and Jakob Buhmann. 2024. Factorized Motion Diffusion for Precise and Character-Agnostic Motion Inbetweening. In *Proceedings of the 17th ACM SIGGRAPH Conference on Motion, Interaction, and Games*. ACM, New York, NY, USA, 1–10.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. 2023. Emergent Correspondence from Image Diffusion.

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit H Bermano, and Michiel van de Panne. 2024. CLoSD: Closing the Loop between Simulation and Diffusion for multi-task character control.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. 2023. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations (ICLR)*. OpenReview.net, OpenReview.net. https://openreview.net/forum?id=SJ1kSyO2jwu

Shashank Tripathi, Omid Taheri, Christoph Lassner, Michael J. Black, Daniel Holden, and Carsten Stoll. 2025. HUMOS: Human Motion Model Conditioned on Body Shape. In *European Conference on Computer Vision*. Springer, Springer International Publishing, Berlin/Heidelberg, Germany, 133–152.

Truebones Motions Animation Studios. 2022. Truebones. https://truebones.gumroad.com/ Accessed: 2022-1-15.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*. OpenReview.net, OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

Ruben Villegas, Duygu Ceylan, Aaron Hertzmann, Jimei Yang, and Jun Saito. 2021. Contact-Aware Retargeting of Skinned Motion. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Washington, DC, USA, 9700–9709. https://api.semanticscholar.org/CorpusID:237513547

Haoyu Wang, Shaoli Huang, Fang Zhao, and Chun Yuan. 2024. MMR: Multi-scale Motion Retargeting between Skeleton-agnostic Characters. In *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, IEEE, Washington, DC, USA, 1–8.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. 2023. OmniControl: Control Any Joint at Any Time for Human Motion Generation. In *The Twelfth International Conference on Learning Representations*. OpenReview.net, OpenReview.net.

Zhangsihao Yang, Mingyuan Zhou, Mengyi Shan, Bingbing Wen, Ziwei Xuan, Mitch Hill, Junjie Bai, Guo-Jun Qi, and Yalin Wang. 2023. OmniMotionGPT: Animal Motion Generation with Limited Data. arXiv:2311.18303 [cs.CV]

Zijie Ye, Jia-Wei Liu, Jia Jia, Shikun Sun, and Mike Zheng Shou. 2024. Skinned Motion Retargeting with Dense Geometric Interaction Perception.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021a. Do Transformers Really Perform Badly for Graph Representation?. In *Thirty-Fifth Conference on Neural Information Processing Systems*. Curran Associates Inc., NY, USA. https://openreview.net/forum?id=OeWooOxFwDa

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021b. Do transformers really perform badly for graph representation? *Advances in neural information processing systems* 34 (2021), 28877–28888.

Haodong Zhang, Zhike Chen, Haocheng Xu, Lei Hao, Xiaofei Wu, Songcen Xu, Rong Xiong, and Yue Wang. 2024b. Unified Cross-Structural Motion Retargeting for Humanoid Characters. *IEEE Transactions on Visualization and Computer Graphics* 1 (2024).

Jiaxu Zhang, Shaoli Huang, Zhigang Tu, Xin Chen, Xiaohang Zhan, Gang Yu, and Ying Shan. 2023a. TapMo: Shape-aware Motion Generation of Skeleton-free Characters.

Jiaxu Zhang, Junwu Weng, Di Kang, Fang Zhao, Shaoli Huang, Xuefei Zhe, Linchao Bao, Ying Shan, Jue Wang, and Zhigang Tu. 2023b. Skinned motion retargeting with residual perception of motion semantics & geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Washington, DC, USA, 13864–13872.

Jia-Qi Zhang, Miao Wang, Fu-Cheng Zhang, and Fang-Lue Zhang. 2024d. Skinned Motion Retargeting with Preservation of Body Part Relationships. *IEEE Transactions on Visualization and Computer Graphics* 1 (2024).

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2024a. MotionDiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1 (2024).

Xinyi Zhang, Naiqi Li, and Angela Dai. 2024c. DNF: Unconditional 4D Generation with Dictionary-based Neural Fields.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2018. On the Continuity of Rotation Representations in Neural Networks. *CoRR* abs/1812.07035 (2018). arXiv:1812.07035 http://arxiv.org/abs/1812.07035

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Washington, DC, USA, 5745–5753.

Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 2017. 3D menagerie: Modeling the 3D shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE Computer Society, Washington, DC, USA, 6365–6373.

APPENDIX

This Appendix provides additional details to complement the information presented in the main paper. While the main paper is self-contained, the details provided here offer further insights and clarifications.

In appendix A, we provide implementation details of AnyTop, and in Appendix B, we elaborate on our preprocessing and augmentation pipelines. Finally, in appendix C and present additional quantitative results beyond those in the main paper.

A IMPLEMENTATION DETAILS

The maximum topological distance we allow in $\mathcal{D}$ is $d_{max} = 6$, and our *Temporal Attention* is applied on temporal windows of length $W = 31$. For our model inputs, we allow maximum number of joints $J = 143$. *During training*, we use cropped sequences of $N = 40$ frames. To enable our model handle higher frame positions and generate longer sequences, we incorporate positional encoding relative to the cropping index. For training, we used batch size of 16 when training on the entire dataset, and a batch size of 8 to train in the data subsets.

B DATA

Truebones Zoo dataset. In addition to the data misalignment issues discussed in the main paper, the dataset also contains vulnerabilities such as excessive dummy joints, qualitative artifacts like foot sliding and floating, and 20% of the frames involve skeletons connected to the origin via an additional bone, resulting in artefacts such as walking or running in place. We address some of these issues as part of our data processing pipeline, which is detailed in the main paper and further extended in the following paragraph.

Data Preprocessing. In this section, we provide further details on the preprocessing steps mentioned in the main paper, as well as describe additional refinements applied to the dataset. As part of the alignment process, we ensure that all skeletons are properly grounded. This is achieved by using the textual descriptions of the joints to identify the foot joints of each skeleton. Based on their height in the rest pose, we determine the ground height for each skeleton and subtract it from the corresponding root height in the motion data. For skeletons that do not interact with the ground, such as flying birds or swimming fish, the ground height is determined by the position of the lowest joint in the rest pose. Another important preprocessing step is ensuring that the rest poses of all skeletons are natural. This is essential for two key reasons. First, many animals feature a similar span of rotation angles in organs that have similar functionality, *e.g.*, the forearm. We would like this span of rotations to constitute a manifold representing multiple animals. Once all rotation angles are defined relative to a character's rest-pose, we have a common representation basis, hence the desired manifold can be obtained.

Second, the rest pose is encoded as a single frame within the motion sequence. To maintain consistency with the other frames, which represent natural poses, the rest pose must also exhibit a natural configuration. To accomplish this, we transform all motion rotations so that they are relative to a natural rest pose, which can either be provided as an additional motion capture (mocap) file or selected from the skeleton's idle motion.

In addition to the alignment procedure, we also extract relevant information from the skeletons and motion data. First, we use foot joints labels to generate foot-contact indicators for each frame, which are concatenated with the motion features. Next, we compute the mean and standard deviation for each skeleton's frames and use these statistics to normalize the motions before feeding them into the model during training.

Input Preprocessing. The input to our model is a skeleton $\mathcal{S} = \mathcal{P}_\mathcal{S}, \mathcal{R}_\mathcal{S}, \mathcal{D}_\mathcal{S}, \mathcal{N}_\mathcal{S}$, derived from the raw rest pose of the character, represented as $(\mathcal{G}_\mathcal{S}, O_\mathcal{S})$, along with the corresponding joint names. Both $(\mathcal{G}_\mathcal{S}, O_\mathcal{S})$ and the joint names can be obtained from standard motion capture formats (*e.g.*, bvh, fbx).

The skeletal features $\mathcal{S} = \{\mathcal{P}_\mathcal{S}, \mathcal{R}_\mathcal{S}, \mathcal{D}_\mathcal{S}, \mathcal{N}_\mathcal{S}\}$ are computed as follows: First, to compute $\mathcal{P}_\mathcal{S}$, we apply forward kinematics with zero rotations on $(\mathcal{G}_\mathcal{S}, O_\mathcal{S})$ obtaining the global joint positions in the rest pose. These positions are then converted to root-relative coordinates. To align the rest pose with the format of individual frames in a motion sequence, we append to each root-relative position a 6D representation of zero rotation, zero velocity, and foot contact indicators. The topological conditions, $\mathcal{R}_\mathcal{S}$ and $\mathcal{D}_\mathcal{S}$, are derived through a traversal of the skeletal hierarchy $\mathcal{G}_\mathcal{S}$. Finally, the joint names $\mathcal{N}_\mathcal{S}$ are extracted from the motion capture data and undergo text-preprocessing, which includes the removal of digits, symbols, irrelevant words, and redundant prefixes. Additionally, side indicators such as 'L/R' are replaced with 'Left/Right', non-English joint names are translated, and similar actions are standardized.

Data augmentation. *Skeletal Augmentation* exposes our model to a wider variety of skeletons, as described in the Method section of the main paper. Next, we further elaborate about this process. The first augmentation we apply is *joint removal*, which randomly removes up to 30% of the joints from the skeleton, where feet joints are never removed to maintain physical correctness. For efficiency consideration, we exclude joints with more than a single child from the removal procedure. The second augmentation is *joint addition*, which introduces a new joint at the midpoint of a randomly selected edge. After removing or adding joints to the skeleton, we update $\mathcal{R}_\mathcal{S}$, $\mathcal{D}_\mathcal{S}$ and $\mathcal{N}_\mathcal{S}$ accordingly. Note that updating $\mathcal{D}_\mathcal{S}$ is computationally expensive with a complexity of $0(J^2)$, as it requires recomputing the path between each pair of joints in the DAG.

C COMPARISON WITH BASELINES ON SUBSET MODELS

We provide a quantitative evaluation of AnyTop trained on the data subsets defined in the Experiments section of the main paper. To maintain fairness in comparison, we train the adapted MDM baseline separately for each subset. Since SinMDM is independently trained for each skeleton, no additional adjustments are needed. Each model is evaluated using the corresponding skeletons from our benchmark that match the relevant data subset. The results, shown in Tab. 4, indicate that AnyTop achieves the optimal coverage-diversity tradeoff compared to all other baselines presented.

Table 4. **Comparison on Data Subsets.** Quantitative results of AnyTop trained on different data subsets, compared to the baselines trained under equivalent settings. * indicates the work has been adjusted to our experimental terms and † indicates that a specific skeleton (Scorpion) has been removed from the SinMDM evaluation set, as SinMDM fails to converge on this skeleton. This exclusion ensures that its impact does not skew the overall score.

Subset	Model	Coverage ↑	Local Div. ↑	Inter Div. ↑	Intra Div. Diff. ↓
Quadrupeds	MDM*	$83.3^{\pm 23}$	$\underline{0.103}^{\pm 0.14}$	$0.112^{\pm 0.07}$	$0.160^{\pm 0.03}$
	SinMDM*	$\mathbf{94.0}^{\pm 06}$	$0.050^{\pm 0.04}$	$\underline{0.230}^{\pm 0.12}$	$\underline{0.151}^{\pm 0.08}$
	AnyTop	$\underline{89.2}^{\pm 09}$	$\mathbf{0.215}^{\pm 0.08}$	$\mathbf{0.291}^{\pm 0.17}$	$\mathbf{0.114}^{\pm 0.06}$
Bipeds	MDM*	$87.9^{\pm 13}$	$0.034^{\pm 0.01}$	$0.081^{\pm 0.03}$	$\underline{0.108}^{\pm 0.05}$
	SinMDM*	$\mathbf{95.0}^{\pm 05}$	$\underline{0.040}^{\pm 0.02}$	$\underline{0.251}^{\pm 0.12}$	$\mathbf{0.090}^{\pm 0.03}$
	AnyTop	$\underline{93.5}^{\pm 05}$	$\mathbf{0.191}^{\pm 0.09}$	$\mathbf{0.288}^{\pm 0.19}$	$0.120^{\pm 0.06}$
Flying	MDM*	$63.7^{\pm 31}$	$\underline{0.219}^{\pm 0.25}$	$0.193^{\pm 0.18}$	$\underline{0.154}^{\pm 0.08}$
	SinMDM*	$\mathbf{78.9}^{\pm 18}$	$0.071^{\pm 0.04}$	$\underline{0.320}^{\pm 0.13}$	$\mathbf{0.095}^{\pm 0.03}$
	AnyTop	$\underline{72.6}^{\pm 18}$	$\mathbf{0.289}^{\pm 0.13}$	$\mathbf{0.410}^{\pm 0.19}$	$0.166^{\pm 0.07}$
Insects	MDM*	$88.4^{\pm 07}$	$0.063^{\pm 0.03}$	$0.185^{\pm 0.10}$	$\mathbf{0.117}^{\pm 0.05}$
	SinMDM *	$77.8^{\pm 04}$	$\mathbf{0.235}^{\pm 0.29}$	$\mathbf{0.419}^{\pm 0.08}$	$0.152^{\pm 0.05}$
	SinMDM *†	$\mathbf{92.9}^{\pm 03}$	$0.061^{\pm 0.015}$	$0.348^{\pm 0.10}$	$0.136^{\pm 0.06}$
	AnyTop	$\underline{90.6}^{\pm 09}$	$\underline{0.189}^{\pm 0.07}$	$\underline{0.317}^{\pm 0.117}$	$\underline{0.127}^{\pm 0.05}$