

Cross-Morphology Motion Transfer with Semantic Style Alignment

Alexios Mylordos^{ID}, Jose Luis Ponton^{ID}, Nuria Pelechano^{ID}, and Andreas Aristidou^{ID}, *Senior Member, IEEE*

Abstract—Transferring animations between characters with diverse skeletal structures is challenging. Traditional retargeting pipelines rely on fixed correspondences, canonical skeletons, or human-centric datasets, which can lead to artifacts when applied across heterogeneous morphologies. We introduce a framework for cross-morphology motion transfer with semantic style alignment that uses morphology-agnostic control signals (e.g., velocity, angular velocity, relative height) to align behaviors across species. Our method supports all-to-all retargeting: motions from any source can be mapped to any trained target while preserving target-specific style. For each target morphology, we train a Vector Quantized VAE and an autoregressive sequence model to construct a compact, morphology-specific codebook that captures stylistic priors. This modular design scales to new morphologies without retraining existing models and allows optional user control (e.g., phase, velocity scaling) for fine-grained alignment. Experiments across bipeds and quadrupeds demonstrate accurate, plausible, and style-faithful motion transfer, establishing a scalable approach to retargeting across arbitrary skeletal topologies.

Index Terms—Cross-morphology, motion transfer, retargeting, semantic alignment, stylistic alignment

I. INTRODUCTION

ANIMATING characters with diverse skeletal structures and stylistic priors remains a central challenge in computer graphics. Traditional pipelines for motion retargeting often depend on carefully crafted skeleton correspondences or reference templates, which makes them fragile in the presence of heterogeneous morphologies such as bipeds, quadrupeds, or fantastical creatures. Differences in joint connectivity and constraints, body proportions, and stylistic priors frequently lead to artifacts, e.g., anthropomorphized animal motion or degraded fidelity when mapping back to the source domain. Compounding these challenges, datasets that encompass truly diverse skeletal topologies are scarce, limiting the effectiveness of purely data-driven approaches. An ideal system would (i) enable bidirectional translation of motion across arbitrary skeletons, (ii) preserve the target character’s stylistic priors (a dog should still *move like a dog*), and (iii) scale seamlessly as new target characters’ morphologies are introduced. These requirements motivate our central objective: to transfer the semantic content of a source motion across substantially

different morphologies without requiring explicit skeletal correspondences, while preserving the characteristic motion prior of each trained target morphology.

Recent learning-based advances have substantially pushed the state of motion generation and retargeting. Neural models have demonstrated impressive results in motion synthesis, editing, and style transfer [1]–[3]. However, most existing methods remain tied to fixed skeletal topologies [4], [5], bone-proportion variations [6], or skeletal homeomorphism [7], limiting their applicability to cross-morphology retargeting. Attempts at multi-skeleton scalability often rely on topology-specific modules [8], [9] or entirely separate models per character [10], [11], undermining extensibility. While approaches based on intrinsic motion representations [12]–[14] or unpaired translation frameworks [15], [16] offer greater flexibility, they typically operate in one-to-one or one-to-all paradigms, often anchored by a canonical reference skeleton. More recently, Gat et al. [17] introduced a skeleton-conditional diffusion model that can animate characters of arbitrary topology; however, this formulation is conditioned on a single skeleton at a time and does not perform explicit pairwise mappings across skeletons.

In this work, we propose a framework for *cross-morphology stylistic motion transfer*, capable of synthesizing plausible motions for a target morphology given control signals originating from any other morphology. Unlike traditional joint-matching or phase-matching approaches, our method dispenses with fixed reference topologies and instead uses morphology-agnostic control signals (e.g., velocity, angular velocity, relative height). These signals capture semantically consistent cues across species and enable our model to operate in an *all-to-all* fashion at inference: motions from any source morphology can be mapped to any previously trained target morphology, as illustrated in Figure 1, aligning behaviors such as locomotion, sitting, or lying down, even in the absence of direct joint correspondences. We evaluate this setting on a wide range of morphologies, including human, dog, ostrich, monkey, scorpion, T-Rex, and dragon skeletons, demonstrating all-to-all cross-morphology transfer with semantic style preservation.

Our design is modular and skeleton-aware: for each target morphology, we train a dedicated instance composed of a Vector Quantized Variational Autoencoder (VQVAE) [18] and an autoregressive codebook predictor. Each instance learns its own morphology-specific codebook independently, which captures stylistic priors from motion data. At inference, control signals may originate from *any* morphology, but synthesis is limited to the trained target. Extending the framework to new target morphologies requires training a new instance, but

A. Mylordos and A. Aristidou are with the Department of Computer Science, University of Cyprus, Nicosia, 1678, Cyprus, and the CYENS Centre of Excellence, Nicosia, 1016, Cyprus.

E-mail: alexmylordos@gmail.com; a.aristidou@ieee.org.

J. L. Ponton and N. Pelechano are with the Universitat Politècnica de Catalunya, 08034 Barcelona, Spain.

E-mail: jose.luis.ponton@upc.edu; npelechano@cs.upc.edu.

Manuscript received September 1, 2026; revised September 1, 2026.

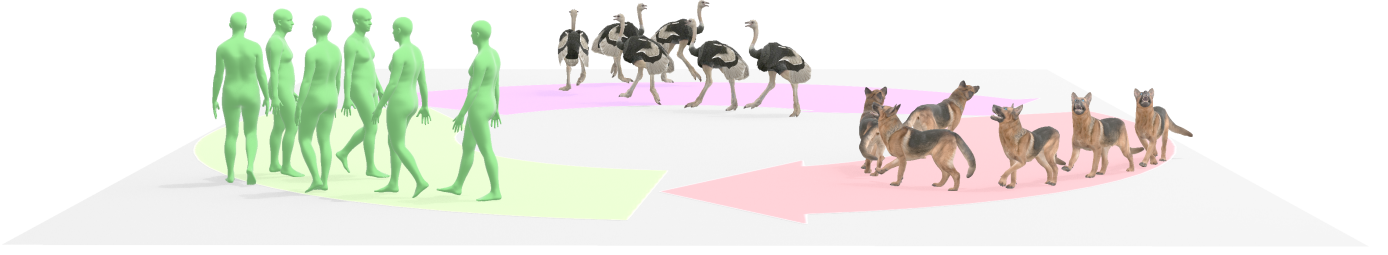


Fig. 1: **All-to-all cross-morphology motion transfer.** Our method transfers motion among humans, dogs, and ostriches without relying on a reference morphology, while preserving the motion characteristics of each target.

all previously trained models remain valid without retraining. Thus, the scalability of our framework does not arise from zero-shot generation on unseen target skeletons, but from its modularity: adding a new target requires training only its corresponding model, without modifying or retraining any existing source–target mappings. This yields a practical form of *all-to-all* transfer, since any input morphology can be mapped to any trained output. Unlike a unified *all-to-all* model, separate instances do not share a common latent embedding, but our experiments demonstrate that morphology-agnostic control signals provide sufficient semantic alignment for high-quality transfer. The key novelty is therefore the separation between a shared, morphology-agnostic behavioral interface and independent morphology-specific motion priors: semantic alignment is performed through the former, while target-specific motion style is preserved by the latter.

Training proceeds in two stages: (i) a VQVAE is optimized with MSE and vector quantization losses to construct a morphology-specific codebook; (ii) an autoregressive predictor is trained with cross-entropy loss to model valid sequences of codebook indices. At inference, predicted indices retrieve vectors from the codebook, which are decoded into motion trajectories. This design requires no cyclic consistency or explicit phase supervision, though optional control signals, such as phase or velocity scaling, can be introduced to refine alignment (e.g., ensuring that a running input maps to running output rather than walking).

Importantly, our framework preserves stylistic fidelity by disentangling morphology-agnostic signals from morphology-specific reconstructions. In addition, users can adjust alignment via simple signal scaling, affording flexible high-level control. This design respects non-human priors and avoids collapsing behaviors into a human-centric latent space, while still supporting fine-grained constraints when desired.

In summary, our contributions include:

- A **morphology-specific motion synthesis framework** for *all-to-all* cross-morphology retargeting, which generates target motions from control signals of any source morphology without requiring joint- or phase-level correspondences. Semantic alignment is achieved through **morphology-agnostic signals** (e.g., velocity, angular velocity, relative height), with *user-controllable cues* (e.g., phase, velocity scaling) for intuitive style and pace adjustment.
- A **two-stage training procedure** combining VQVAE-

based reconstruction with autoregressive sequence modeling. This design yields compact, morphology-specific codebooks that capture stylistic priors, enabling scalable extension to new characters without retraining existing ones.

- A comprehensive **evaluation across heterogeneous morphologies** (bipeds $\leftrightarrow$ quadrupeds), demonstrating kinematic accuracy, physical plausibility, and faithful preservation of target-style dynamics.

II. RELATED WORK

While motion synthesis and retargeting have advanced rapidly, most methods assume fixed skeletal topologies, canonical references, or human-centered datasets. Such constraints hinder their ability to generalize across heterogeneous morphologies (e.g., bipeds vs. quadrupeds) and to preserve target-specific style. In contrast, our work enables *all-to-all*, *cross-morphology retargeting* without reliance on canonical skeletons, while supporting scalable extension to new target characters without retraining the entire model.

Human→human motion retargeting: Traditional motion retargeting explicitly addressed adapting motion across characters with different proportions or skeletal structure using joint correspondences, kinematic constraints, or importance-based mappings [19], [20]. Learning-based retargeting methods later emerged to improve robustness and automation. For instance, Neural Kinematic Networks (NKN) [15] introduced unsupervised retargeting via forward kinematics constraints, while Skeleton-Aware Networks [7] embedded motion into a shared latent space to enable retargeting across *homeomorphic* skeletons. SAME [21] proposed skeleton-agnostic motion embeddings for retargeting within a unified representation, while pose-aware attention models [22] further improved robustness to body-part variation within the human domain. While effective for human-to-human retargeting, these methods assume compatible skeletal structure or shared topology and do not generalize to morphologically diverse characters.

Cross-morphology retargeting and embodiment: Before deep learning, systems for highly varied morphologies relied on IK with morphology-dependent goal generation (e.g., Spore [23] or constrained optimization [24]. Recent VR embodiment approaches have mapped human motion to quadrupeds using learned models, emphasizing first-person control fidelity: NeuroDog [25] and Dog Code [26] achieved realistic quadruped motion but are typically one-directional

(human→animal) and rely on a human reference. Other approaches attempted to scale across multiple morphologies using topology-specific modules [9], [27]. Motion2Motion [28] introduced a recent learning-based baseline for cross-morphology motion transfer using patchwise motion matching with sparse bone correspondences. In contrast, our method enables scalable, bidirectional, all-to-all transfer across morphologies without explicit skeletal correspondences, or mapping motions to a fixed human reference, while preserving target-specific style.

Discrete motion representations & phase alignment:

Recent work has explored discrete and phase-aware motion representations primarily for motion generation, control, or semantic alignment rather than explicit retargeting. MoGlow [29] modeled human motion using continuous latent flows, while MotionDreamer [30] and CoMo [31] discretized motion into codebook representations for controllable synthesis. Phase-centric approaches such as WalkTheDog [8] aligned locomotion semantics across humans and dogs via shared phase manifolds. These approaches enable indirect alignment typically assuming canonical reference or phase latent spaces, but do not perform direct skeleton-to-skeleton retargeting across heterogeneous morphologies. Related phase-aware embeddings [4], [6] improved robustness for human locomotion but remain limited to homeomorphic skeletons.

Non-human motion generation and control: Recent generative models condition motion synthesis on skeletal structure to support arbitrary topologies. AnyTop [17] conditioned diffusion models on skeletal graphs and joint descriptors to synthesize motion for unseen skeletons. Motion Anything [32] targeted multimodal “any-to-motion” generation with multimodal conditioning and attention-based mask modeling. Other works adapted text-to-motion pipelines to non-human characters [2], [33], [34], focusing on motion generation from scratch. Single-sequence generative methods such as GANimator [10] and SinMDM [11] further demonstrated that diverse non-human motions can be synthesized from limited motion data, but they do not perform motion transfer between morphologically different source and target characters. Beyond generative approaches, motion-control methods have also been developed specifically for non-human characters. Mode-Adaptive Neural Networks (MANN) [35], for example, introduced phase-conditioned gating for quadruped control learned from unstructured data. While these generation and control approaches address different animation tasks, neither directly targets cross-morphology motion retargeting with target-style preservation, which is the focus of our work.

Physics-based retargeting and character control: Reinforcement learning and physics priors have been applied to retarget sparse human motion to non-human morphologies, emphasizing balance, contact, and physical realism via plausible control policies [36]. ACE [37] introduced adversarial correspondence embeddings to learn cross-morphology mappings between human and non-human characters within a physics-based control framework, enabling robust human-to-animal retargeting without hand-designed joint correspondences. In contrast, related physics-based character-control methods such as AMP [1] learned adversarial motion priors

for producing stylized and physically plausible behaviors, rather than explicitly addressing cross-morphology retargeting or correspondence learning. While such methods emphasize balance, contact, and physical realism, they often sacrifice kinematic and stylistic fidelity. Our work is complementary: we focus on kinematic retargeting with morphology-specific style preservation, while physics-based methods can serve as downstream controllers or constraints.

III. METHODOLOGY

The core challenge of cross-morphology motion transfer is twofold: (i) achieving semantic or content alignment (e.g., ensuring a dog runs when a human runs) without relying on shared skeletal structures, and (ii) ensuring morphological adaptation or style preservation, where the output motion respects the physical constraints and style of the target creature.

To address this, we propose a framework that decouples motion content from skeletal topology. Unlike prior methods such as Walk-the-Dog [8], which aligns motions via a shared learned phase manifold and require retraining for new pairs, our approach utilizes a modular *any-to-one* design that scales to an *all-to-all* system. We define a set of universal, morphology-agnostic control signals (e.g., velocity, relative height) that act as the common language between morphologies.

Our method learns a generative model of motion through a two-stage process (Figure 2):

- 1) **Morphological Adaptation (Motion Generator).** We train a morphology-specific VQ-VAE to learn a discrete *vocabulary* of valid poses for the target character. This effectively compresses the continuous pose space into a compact codebook of stylistic priors.
- 2) **Semantic Alignment (Autoregressive Predictor).** We train a sequence model to learn the *grammar* of the motion, predicting how these discrete pose tokens are used over time conditioned on the universal control signals. This model effectively learns the temporal dynamics of the motion by generating sequences of these discrete codes over time, conditioned on the control signals.

This separation allows us to train a distinct model for each target morphology once. At inference, control signals extracted from *any* source morphology can be fed into the target’s predictor, achieving transfer without explicit source-target pairing or retraining. Each target morphology therefore learns an independent VQ-VAE and morphology-specific codebooks; semantic alignment across morphologies is achieved exclusively through the shared morphology-agnostic control signals rather than through a shared latent representation or shared codebooks.

A. Dataset and Preprocessing

We employ three primary datasets in our study: the dog motion database introduced by Zhang et al. [35], the human locomotion database from Starke et al. [38], and the zoo dataset used in Gat et al. [17]. While the dog and human datasets provide extensive coverage of their respective motion domains, the zoo dataset is comparatively limited in scope

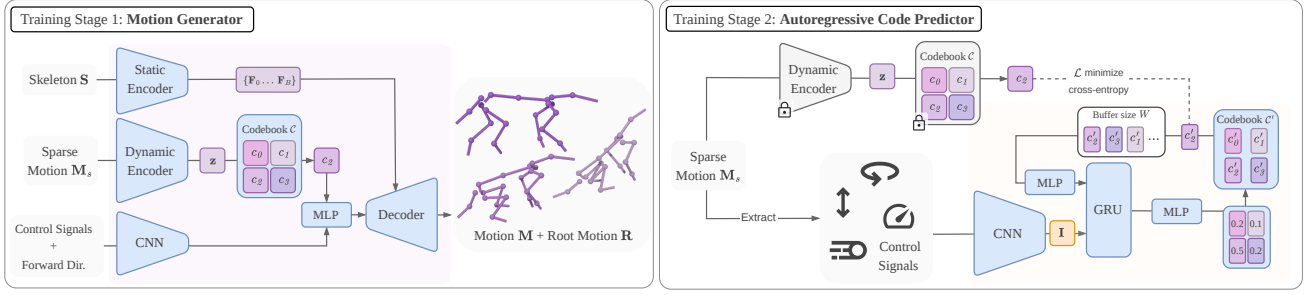


Fig. 2: **Overview of the two-stage training pipeline.** (Left) **Stage 1: Motion Generator.** We train a skeleton-aware VQ-VAE to reconstruct full root-relative motion M and root trajectory R from sparse inputs M_s and skeletal offsets S . This stage learns a discrete, morphology-specific codebook $\mathcal{C}$ that captures the valid pose manifold. (Right) **Stage 2: Autoregressive Code Predictor.** The Motion Generator is *frozen* (indicated by lock icons), and it processes the input motion to provide ground-truth code indices from codebook $\mathcal{C}$. The predictor learns to forecast these indices autoregressively using a secondary, lightweight codebook $\mathcal{C}'$, conditioned on extracted control signals. The interaction between stages is explicitly enforced by minimizing the cross-entropy loss (dashed line).

and size. To mitigate this imbalance, we apply data augmentation techniques, including spatial mirroring and applying random rotations to the root orientation of input motions to improve robustness. In certain cases, we also use synthetic motion generation via generative models [10]; the proportion of augmented data did not exceed 50% of the total samples for any species. Consequently, our combined database offers a diverse collection of human locomotion and a comprehensive set of motions for other animal species. The final durations of our dataset are: 82 minutes of dog motion, 78 minutes of human motion, 10 minutes of ostrich motion, and 90 seconds each for T-Rex, scorpion, monkey, and dragon motions. Note that for these latter species, the dataset only contains a limited set of fundamental locomotion behaviors, including walking, running, turning, and idle motions.

Most available zoo datasets, which primarily consist of cyclic keyframed motions, lack translational components and thus loop in place, making them difficult to use directly in our implementation. Since a core element of our control signals is the velocity vector (to be discussed later in this section), critical for both interpreting the performed action and transferring it to the target character, we artificially introduce global translation into the motion capture sequences of the ostrich, T-Rex, scorpion, monkey, and dragon using a foot-contact-based approach. Specifically, we detect foot-contact events, frames in which a foot remains stationary relative to the ground, and use them to anchor the feet in world space. The root joint is then advanced forward to ensure the feet remain planted, thereby generating forward translation. We define synthetic paths for our creatures to follow, but this procedure often introduces artifacts such as misalignments and foot sliding, especially during turns and side-walking. To address these issues, we apply inverse kinematics (IK), which reduces residual misalignments and minimizes foot sliding. Note that earlier methods [17], [28] do not incorporate root translations in their zoo database and reconstructions.

B. Motion Representation

Our method processes motion sequences of length T for a skeleton with J joints. We decompose the motion into three components: the static skeleton structure, the root-relative pose, and the root’s global motion.

Static Skeleton Structure: A tensor $S \in \mathbb{R}^{J \times 3}$ containing the 3D offset vectors that define each joint’s local position in the bind pose relative to its parent. This creates a hierarchical representation of the skeleton’s proportions.

Root-relative Motion: The motion tensor $M \in \mathbb{R}^{T \times J \times 9}$, which contains the per-frame local positions (3D) and rotations (6D) of all joints, expressed within the root’s coordinate system. This component captures the body’s articulation, independent of its global trajectory. The sparse input to our generator, $M_s \in \mathbb{R}^{T \times J_s \times 9}$, is a subset of M corresponding to J_s joints, where the subset is determined automatically by the skeletal hierarchy after the $B = 3$ pooling operations described below. The root and leaf joints are preserved during pooling and are therefore always included, while additional intermediate joints may remain for more complex skeletal hierarchies. Consequently, J_s is morphology-dependent and does not require manually defined correspondences or a fixed set of semantic end-effectors. Additionally, a lightweight root proxy containing the root velocity magnitude, pelvis height, and yaw angular velocity is included as input. Unlike the full root trajectory, these signals provide local motion cues without introducing global orientation information, allowing the latent representation to remain largely invariant to the absolute root orientation while improving consistency between root motion and body pose.

Root Motion: The root’s trajectory $R \in \mathbb{R}^{T \times 9}$, which stores the root joint’s 3D displacement and 6D rotation relative to the previous frame.

For all rotational data in M and R , we use the 6D representation [39] consisting of the first two columns of the rotation matrix. This representation avoids the discontinuity issues associated with quaternions and has been shown to be more effective for training neural networks on rotational data.

C. Stage 1: Motion Generator

The Motion Generator is a skeleton-aware Vector Quantized Variational Autoencoder (VQ-VAE). Its primary objective is to learn a *morphology-specific* codebook that captures the valid pose manifold of the target creature, effectively solving the problem of *morphological adaptation*. To construct this morphology-specific representation, we adapt the sparse-to-dense framework of SparsePoser [40], designed to reconstruct the full root-relative pose $\mathbf{M}$ from its sparse counterpart $\mathbf{M}_s$ (typically the end-effectors). This reconstruction task acts as a powerful regularizer, compelling the encoder to learn a semantically meaningful latent representation rather than a trivial one-to-one mapping. By forcing the model to infer the complete pose from sparse information, it must capture the underlying kinematic constraints and correlations of the skeleton. This leads to a more structured and robust codebook of plausible poses.

To capture the hierarchical nature of the skeleton, we process it as a graph across B resolution levels (with $B = 3$ in our experiments). We employ skeletal pooling to iteratively collapse consecutive joints, progressively coarsening the graph while preserving the root and leaf joints. After the three pooling operations, the joints remaining at the coarsest resolution define the morphology-specific sparse joint set used in $\mathbf{M}_s$; depending on the complexity of the hierarchy, this set may also contain intermediate joints. This multi-scale approach allows the network to capture kinematic features at varying granularities while downsampling the temporal dimension.

The generator comprises three modules, each utilizing B blocks:

- 1) **Static Encoder (*se*):** To handle skeletal variations (e.g., different limb lengths within a species), we explicitly model the bone structure. The encoder takes the static offsets $\mathbf{S}$ and processes them through B blocks of skeletal pooling operations. Crucially, at each level $l \in \{0, \dots, B\}$, we extract a static feature map $\mathbf{F}_l$. These features are injected into the decoder via concatenation, allowing the learned codebook to remain agnostic to specific bone lengths while the decoder adapts the motion to the specific skeleton geometry $\mathbf{S}$.
- 2) **Dynamic Encoder (*de*):** This module maps the continuous sparse motion $\mathbf{M}_s$ to a discrete latent sequence. It employs temporal skeletal 1D-convolutions where the weight matrix is masked to only allow influence between topologically neighboring joints. We use a temporal stride of 2 at each of the B blocks. This design choice explicitly imposes temporal compression, resulting in a latent representation $\mathbf{Z} \in \mathbb{R}^{\frac{T}{2^B} \times L}$ that is temporally downsampled by a factor of 2^B (i.e., $8\times$ smaller for $B = 3$). This downsampling increases the receptive field, allowing each code to represent a short motion primitive rather than a single static pose.
- 3) **Decoder (*d*):** The decoder reconstructs the full motion $\mathbf{M}$ from the discrete latents and the static features. It mirrors the encoder's hierarchy, but using skeletal unpooling and linear temporal upsampling to recover the full motion. At each level l , the static features $\mathbf{F}_l$ are concatenated with

the dynamic features, ensuring the reconstructed motion is physically consistent with the input skeleton $\mathbf{S}$.

We introduce three key modifications to the original architecture to enable robust cross-morphology transfer. First, we operate primarily in a root-relative space. The global root transformation is omitted from $\mathbf{M}$, allowing the latent representation to focus on the body pose rather than the absolute trajectory. As described in Section III-B, the encoder additionally receives a lightweight root proxy that provides coarse motion dynamics.

Second, we modify the architecture by replacing the continuous bottleneck with a Vector Quantization (VQ) layer. In contrast to continuous latent models like MoGlow [29], which can produce artifacts by interpolating between distinct behaviors, our approach discretizes motion into a learnable codebook $\mathcal{C} \in \mathbb{R}^{N \times L}$, where N is the number of codes, and L is the dimensionality of each latent embedding. Each latent vector $\mathbf{z}_i$ is mapped to its nearest codebook entry $\mathbf{c}_k$ (where $k = \arg \min_j \|\mathbf{z}_i - \mathbf{c}_j\|_2^2$). This discretization serves two critical purposes: (1) it restricts the generated motion to a set of valid motion sequences, ensuring the output always synthesizes a plausible pose for the target creature; and (2) due to the encoder's temporal stride, each code $\mathbf{c}_k$ encapsulates a temporal sequence (a sequence of 2^B frames) rather than a static pose. This abstraction promotes temporal coherence and allows the subsequent autoregressive model to predict high-level motion dynamics rather than per-frame jittery results.

Finally, we condition the decoder on the global heading information to recover the complete motion. To recover the full root trajectory while maintaining consistency with the body pose, the heading sequence is first temporally downsampled to match the latent resolution and concatenated with the sampled codebook vectors. A lightweight projection MLP network maps the resulting feature vector to the dimensionality expected by the first decoder layer, after which the decoder reconstructs the complete motion, including both the body pose motion $\mathbf{M}$ and the global root trajectory $\mathbf{R}$. Conditioning the decoder in this manner allows root motion and body pose to be predicted jointly, improving their temporal consistency.

D. Stage 2: Autoregressive Code Predictor

While Stage 1 learns the codebook of valid poses, Stage 2 learns how to use the codebook of motion sequences to achieve semantic alignment with the control signals. We employ an autoregressive recurrent model to predict per-timestep categorical distributions over the learned codebook indices.

Decoupled Codebooks. The predictor operates on a secondary, lower-dimensional codebook $\mathcal{C}' \in \mathbb{R}^{N' \times L'}$ (where $L' < L$). Crucially, $\mathcal{C}'$ shares the same categorical index space $\{1, \dots, N\}$ as the primary codebook $\mathcal{C}$. This ensures a strict 1-to-1 correspondence during inference: an index k predicted by the autoregressive model using the lightweight codebook $\mathcal{C}'$ is used to retrieve the corresponding high-fidelity pose embedding from the primary codebook $\mathcal{C}$ for final reconstruction. We denote by $\mathbf{c}'_k$ the k -th entry of the secondary codebook $\mathcal{C}'$, and by $\mathbf{e}'_t$ the time-indexed embedding used by the autoregressive predictor at timestep t .

Using a separate codebook $\mathcal{C}'$, with reduced dimensionality ($L' = 128$), allows the autoregressive model to focus on the temporal dynamics and ordinal relationships between codes, rather than the fine-grained kinematic details embedded in $\mathcal{C}$ ($L = 512$), which leads to faster convergence and better adherence to control signals.

Although each target morphology learns an independent morphology-specific codebook, the predicted code indices are driven by the same morphology-agnostic control signals. These shared signals provide a common behavioral interface across morphologies, allowing each target model to select motion codes that are semantically consistent with the source behavior, such as locomotion, turning, sitting, or lying down, without requiring a shared latent embedding or aligned codebook across morphologies.

Universal Control Signals: The model is conditioned on a set of morphology-agnostic control signals designed to capture the high-level intent of the motion. This control vector $\mathbf{I}$ includes the root's **velocity** and **acceleration** magnitudes, its **yaw angular velocity** and **angular acceleration**, and the normalized **heights of the root and head** joints relative to a canonical standing pose. To ensure semantic consistency across morphologies, the angular velocity is derived from the root's rotation about the vertical axis for bipeds, and from the orientation of the root-to-chest vector (projected onto the ground plane) for quadrupeds. Note that our control sequence vector is kept deliberately simple to facilitate more robust matching. All signals are smoothed using a Gaussian filter.

Since these control signals are extracted directly from physically plausible source motions, mutually conflicting inputs are uncommon in practice. Nevertheless, if inconsistent control signals arise, e.g., a low body-height signal combined with a high running velocity, the predictor may generate semantically ambiguous motions. Such cases were not observed in our experiments; however, if needed, additional morphology-specific control signals can be incorporated to better constrain the target behavior.

Optional Phase Alignment: For tasks requiring precise phase alignment, the speed and acceleration signals can be replaced with binary foot contact labels. This substitution is motivated by the observation that similar kinematic profiles (e.g., comparable speeds and accelerations) across differing morphologies do not necessarily imply consistent gait phase alignment, and may therefore be disregarded by the model or even lead to failure. For each morphology, we define a fixed ordered list of foot joints, and the corresponding target model contains one foot-contact input channel per target foot. During training, individual foot-contact channels are randomly masked, allowing the predictor to operate when only partial contact information is available at inference time. When transferring between morphologies with different numbers of feet, we use the first $N = \min(N_{\text{source}}, N_{\text{target}})$ contact channels according to these predefined orderings. If the source has fewer feet than the target, the remaining target contact channels are masked; if the source has more feet, the additional source channels are discarded. Thus, this optional phase-control mechanism uses a simple deterministic channel assignment rather than requiring explicit semantic foot-to-foot

correspondences.

Sequence Modeling: The synthesis process begins by encoding the control signals with a CNN to match the latent code's temporal resolution, yielding $\mathbf{I} \in \mathbb{R}^{\frac{T}{2^B} \times H}$, where H denotes the encoded control sequence dimensionality. At each latent timestep t , a Gated Recurrent Unit (GRU) models the temporal dynamics of the motion. We chose a GRU for its proven effectiveness in capturing long-term dependencies in sequential data. Its gating mechanism allows the model to generate coherent, long-term motion by selectively remembering or forgetting information about past poses. The input to the GRU is a single large vector, $\mathbf{x}_t$, formed by concatenating two sliding windows: 1) a window of the W most recent encoded control signals $\{i_{t-W+1}, \dots, i_t\}$, and 2) a window of the W most recent past pose embeddings $\{e'_{t-W}, \dots, e'_{t-1}\}$. These pose embeddings e' are the L' -dimensional vectors looked up directly from the secondary codebook $\mathcal{C}'$. This allows the model to predict the next action based on a rich history of recent controls and poses. A Multilayer Perceptron (MLP) then converts the GRU's output into logits, representing a probability distribution over the codebook entries in $\mathcal{C}'$ for the next pose.

Training Details: To train this predictive process, we employ the Gumbel-Softmax trick, which provides a differentiable approximation for sampling discrete codes from the model's predicted logits, enabling end-to-end training. For stability, we use a mixed-input strategy: the embedding passed to the next timestep is a mix of the model's own Gumbel-Softmax sample and the ground-truth code. This form of scheduled teacher-forcing prevents the compounding of errors during training and improves convergence by keeping the model grounded in the correct motion history.

E. Training

Training proceeds in two distinct stages. First, we train the Motion Generator to learn a robust and discrete pose prior. Second, we freeze the generator's weights and train the Autoregressive Code Predictor to model the temporal dynamics of the pose indices. In the following, we detail the network configuration used for both stages and the corresponding optimization objectives.

1) **Network Architectures: Motion Generator.** The dynamic encoder (*de*) uses three stages of skeleton-aware convolutions with a temporal stride of 2 and LeakyReLU activations, resulting in an 8 times temporal downsampling of the sparse motion $\mathbf{M}_s$. The primary codebook $\mathcal{C}$ contains $N = 512$ entries, each with a dimensionality of $L = 512$. The decoder (*d*) mirrors this architecture, using skeleton-aware convolutions with temporal upsampling at each stage. Before decoding, the heading information is temporally downsampled and concatenated with the quantized latent representation. A lightweight MLP projects the resulting feature vector to the dimensionality expected by the decoder, which then reconstructs the complete motion sequence including both body pose motion and root trajectory.

Autoregressive Code Predictor. The control signals are first downsampled to match the Motion Generator temporal

rate using a CNN, producing the encoded control sequence $\mathbf{I} \in \mathbb{R}^{\frac{T}{2^B} \times H}$. The predictor itself is a 2-layer GRU with a hidden dimension of 512 that processes a sliding window of the $W = 4$ most recent VQ embeddings concatenated with the corresponding control embeddings. The secondary codebook $\mathcal{C}'$ has $N = 512$ entries (sharing the same indices as $\mathcal{C}$) but a smaller dimensionality of $L' = 128$. The GRU output is passed to an MLP that produces logits over the $N = 512$ codebook entries.

2) *Training Objectives: Stage 1: Motion Generator.* The generator is trained to optimize a combined reconstruction and vector quantization objective:

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{recon}} + \beta \mathcal{L}_{\text{commit}}$$

The reconstruction loss $\mathcal{L}_{\text{recon}}$ is the mean squared error for both the root-relative pose and the global root motion:

$$\mathcal{L}_{\text{recon}} = \|\hat{\mathbf{M}} - \mathbf{M}\|_2^2 + \|\hat{\mathbf{R}} - \mathbf{R}\|_2^2$$

The commitment loss $\mathcal{L}_{\text{commit}}$ encourages the encoder's output $\mathbf{Z}$ to stay close to the chosen codebook vectors $\mathbf{c}$:

$$\mathcal{L}_{\text{commit}} = \|\mathbf{Z} - \text{sg}[\mathbf{c}]\|_2^2$$

where sg is the stop-gradient operator. We do not use a separate codebook loss as commonly used in VQ-VAE architectures; instead, the codebook $\mathcal{C}$ is updated via an Exponential Moving Average (EMA). We also employ codebook resets to reinitialize unused codes to random encoder latents from the current batch.

Stage 2: Autoregressive Code Predictor. With the generator frozen, the predictor is trained to minimize the cross-entropy (CE) loss between its predicted logits and the ground-truth indices y_t from the generator's codebook $\mathcal{C}$:

$$\mathcal{L}_{\text{autoreg}} = \sum_{t=1}^{\frac{T}{2^B}} \text{CE}(\mathbf{f}_{\theta}(\mathbf{h}_t), y_t)$$

Here, $\mathbf{f}_{\theta}$ is the output MLP, and the GRU hidden state $\mathbf{h}_t$ is computed autoregressively:

$$\mathbf{h}_t = \text{GRU}(\mathbf{x}_t, \mathbf{h}_{t-1})$$

where the input vector $\mathbf{x}_t$ is a concatenation of two sliding windows:

$$\mathbf{x}_t = [\mathbf{i}_{t-W+1:t}; \mathbf{e}'_{t-W:t-1}]$$

$\mathbf{i}_{t-W+1:t}$ represents the window of the W most recent encoded control signals (from $\mathbf{I}$), and $\mathbf{e}'_{t-W:t-1}$ is the window of the W past time-indexed pose embeddings from $\mathcal{C}'$. Note that $\mathbf{e}'_{t-W:t-1}$ is projected to the embedding dimension using an MLP.

The embedding $\mathbf{e}'_t$ (which will be used in the window for $\mathbf{x}_{t+1}$) is computed using a mixed strategy for teacher-forcing:

$$\mathbf{e}'_t = \mathbf{v}_t \cdot \mathbf{W}_{\mathcal{C}'}$$

where $\mathbf{W}_{\mathcal{C}'}$ is the weight matrix of the codebook $\mathcal{C}'$, and $\mathbf{v}_t$ is a one-hot vector selected as:

$$\mathbf{v}_t = \begin{cases} \mathbf{v}_{\text{gt}} & \text{with probability 0.8} \\ \mathbf{v}_{\text{gs}} & \text{otherwise} \end{cases}$$

Here, $\mathbf{v}_{\text{gt}}$ is the one-hot vector of the ground-truth index y_t , and $\mathbf{v}_{\text{gs}}$ is the one-hot vector from the Gumbel-Softmax sample of the model's own logits. This multiplication $\mathbf{v}_t \cdot \mathbf{W}_{\mathcal{C}'}$ is an efficient way of selecting the correct embedding vector from the codebook $\mathcal{C}'$.

F. Inference Setup

As illustrated in Figure 3, during inference, we provide a sequence of control signals from a source motion (which can be of any morphology). These signals are fed to the target character's trained Autoregressive Code Predictor. The model generates motion autoregressively, one timestep at a time.

At each timestep t :

- 1) The GRU's input vector $\mathbf{x}_t$ is assembled by concatenating the window of recent control signals $\{\mathbf{i}_{t-W+1}, \dots, \mathbf{i}_t\}$ with the window of past generated pose embeddings $\{\mathbf{e}'_{t-W}, \dots, \mathbf{e}'_{t-1}\}$.
- 2) The GRU processes $\mathbf{x}_t$ and (via the MLP) predicts logits for the current pose.
- 3) An argmax operation is performed on these logits to select the single, most likely code index, $\hat{y}_t$.

This index $\hat{y}_t$ is then used to retrieve two separate vectors:

- (1) The pose embedding $\mathbf{c}_{\hat{y}_t}$ is retrieved from the primary codebook $\mathcal{C}$ and passed to the Motion Generator's Decoder.
- (2) The context embedding $\mathbf{c}'_{\hat{y}_t}$ is retrieved from the secondary codebook $\mathcal{C}'$ and stored in the buffer to be used as input for the next step (i.e., it becomes $\mathbf{e}'_t$ in the window for $\mathbf{x}_{t+1}$). Thus, during inference, $\mathbf{e}'_t = \mathbf{c}'_{\hat{y}_t}$.

Finally, the Decoder uses the pose embedding $\mathbf{c}_{\hat{y}_t}$ and the current control signal $\mathbf{i}_t$ to reconstruct the final root-relative pose $\mathbf{M}$ and root motion $\mathbf{R}$. This process allows motions from any source morphology to drive the synthesis of stylistically consistent motion on any trained target morphology. Importantly, the sparse representation $\mathbf{M}_s$ is used only during the morphology-specific training of the Motion Generator and does not define correspondences between source and target joints at inference. Consequently, target joints or limbs without a counterpart in the source morphology, such as wings, tails, or additional legs, are reconstructed by the target-specific decoder according to the learned target motion prior and do not require explicit source-joint mappings.

IV. RESULTS AND EVALUATION

We evaluate our framework across a wide range of heterogeneous morphologies, focusing on three key aspects: (i) semantic fidelity in cross-domain motion transfer, (ii) kinematic and physical plausibility, and (iii) controllability through user-adjustable cues. We compare our method against three baselines: an adapted version of the skeleton-aware approach of Aberman et al. [7], Walk-the-Dog, a framework specifically designed for human-to-dog motion transfer [8], and Motion2Motion (M2M), a recent method for cross-topology motion transfer based on sparse skeletal correspondences [28]. We provide both quantitative and qualitative evaluations through numerical comparisons, visualizations of trajectories, foot contacts, and style transfer across morphologies.

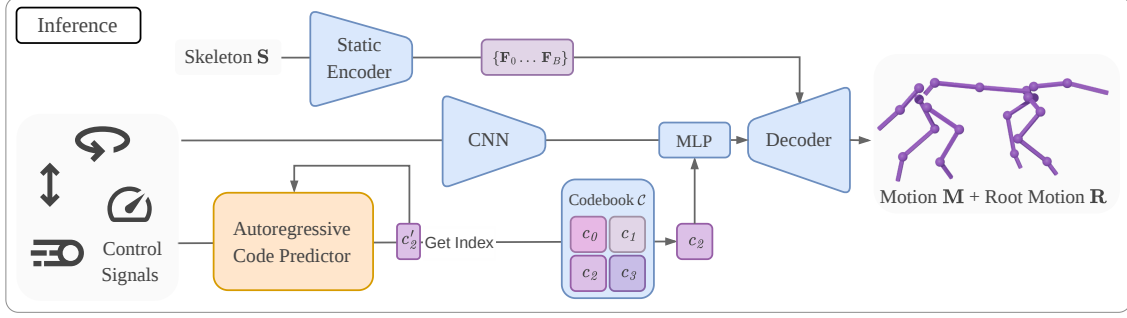


Fig. 3: **Inference Pipeline.** The **Autoregressive Code Predictor** generates a sequence of discrete indices based on input control signals (which can originate from any source morphology). These predicted indices serve as lookup keys to retrieve the corresponding high-fidelity embeddings from the primary codebook $\mathcal{C}$. The Motion Generator’s decoder then reconstructs the full root-relative pose $\mathbf{M}$ and global root motion $\mathbf{R}$ using these retrieved codes and the target’s static skeletal features, ensuring the synthesized motion adheres strictly to the target’s learned morphological priors.

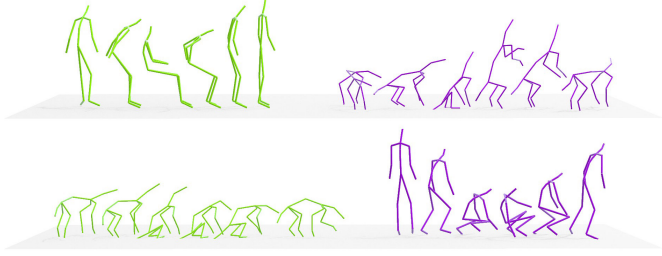


Fig. 4: **Cross-Morphology Motion Transfer with Semantic Alignment.** The top example illustrates motion transfer from a human sitting to a dog, while the bottom example shows the inverse transfer from a dog sitting to a human. Green skeletons denote the source (input) motion, and purple skeletons denote the transferred (output) motion.

A. Implementation details

Our system is implemented in PyTorch using the AdamW optimizer, and was trained on motion datasets spanning humans, quadrupeds, and birds. The two-stage pipeline first learns compact morphology-specific codebooks with VQ-VAE, followed by autoregressive sequence modeling for long-horizon synthesis. Training is performed on an RTX 2080 Ti GPU with a batch size of 128 and a learning rate of 10^{-3} . The Motion Generator is trained on 128-frame-long sequences from the dataset with a stride of 16. During the training of the Autoregressive Code Predictor, this window length is set to 256 frames. Training each target-specific model requires approximately 10 hours across both training stages. During inference, the isolated neural path synthesizes motion at approximately 8000 FPS. The full pipeline including input pre-processing and control signal extraction runs at approximately 3000 FPS, enabling efficient generation of long sequences. Visualizations are made in Blender.

B. Quantitative and Qualitative Evaluation

1) *Cross-Morphology Transfer:* Our method achieves semantic alignment without requiring explicit joint correspon-

dences, using morphology-agnostic signals (e.g., velocity, angular velocity, and relative height). Figure 4 shows a cross-morphology transfer of a sitting motion between a human and a dog, demonstrating our method’s capability for bi-directional semantic alignment across domains. Figures 1 and 6 showcase additional examples of **all-to-all, cross-domain motion transfer** among humans, dogs, and ostriches, demonstrating consistent semantic and stylistic alignment.

We further demonstrate the **scalability** of our framework by extending it to a wider range of morphologies. As shown in Figure 5, an input dog motion is successfully transferred to characters with substantially different body structures, skeletal proportions, and numbers of control points, including a human, ostrich, T-Rex, dragon, scorpion, and monkey. Despite these significant anatomical differences, the transferred motions preserve the original action semantics while naturally adapting to the locomotion characteristics of each target character. This demonstrates that additional target morphologies can be incorporated by training their corresponding target-specific models without retraining the previously learned character models.

Importantly, the transferred motion also preserves target-specific characteristics. Since each target morphology learns its own codebook and motion prior, the source control signals determine the high-level behavior without imposing the source morphology’s movement style on the target. Therefore, dogs behave like dogs, ostriches retain species-specific traits (e.g., wing-raising during running), humans maintain human-like motion, etc. Figure 7 demonstrates this concept, showing how a human running and jumping motion can be transferred to a dog in a semantically similar way.

Beyond these examples, our method can also transfer a source motion to a target species even when no semantically similar motion exists in the target’s training corpus. For instance, Figure 8 shows a human push-up motion transferred to the dog domain, despite push-ups not appearing in the dog training data. In this case, the normalized head helps associate the low body configuration of the human push-up with the dog’s lying posture, while distinguishing it from sitting. Please

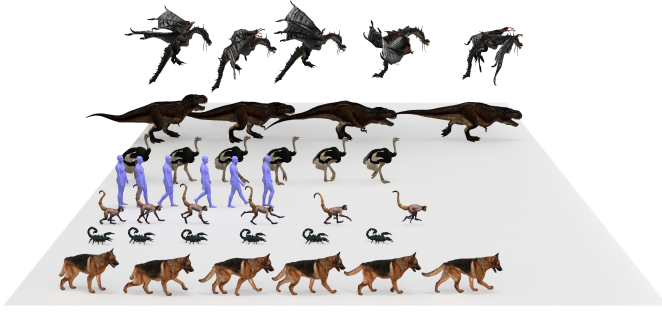


Fig. 5: **Scalability across morphologies.** A dog motion is transferred to characters with diverse body structures, including a scorpion, monkey, human, ostrich, T-Rex, and dragon. Despite the anatomical differences, our method preserves the action semantics while naturally adapting the motion to each target morphology.

refer to the supplemental video for animated versions of all these examples.

2) *Comparison and Baselines:* We compare our method against the three baselines [7], [8], [28]. For a fair comparison, all baselines are retrained on our dataset. Since these methods make different assumptions regarding skeletal compatibility, we evaluate each baseline in the setting supported by its formulation. The skeleton-aware baseline is restricted to a qualitative human→dog comparison and is not applied to pairs with substantially different structures, such as human→ostrich. We implement this comparison using a sparse-to-dense reconstruction network following SparsePoser [40], based on the skeleton-aware operators of Aberman et al. [7]. The dog model is trained from sparse inputs corresponding to the root, head, tail, front paws, and hind paws. For human→dog transfer, the human root and head are mapped to the corresponding dog inputs, the human feet to the hind paws, and the human hands to the front paws. The tail remains part of the dog output structure, but no semantically corresponding human joint is available for this input; the complete dog motion is reconstructed using the learned dog motion prior and the dog skeletal offsets. The Walk-the-Dog framework does not provide access to exported motion files, restricting our evaluation of this method to qualitative comparisons. M2M supports transfer across different skeletal topologies through sparse skeletal correspondences and is evaluated both qualitatively and quantitatively in the comparisons below.

For the skeletal-aware baseline, in the human→dog setting, the generated motions clearly fail to support cross-morphology retargeting, resulting in abnormal behavior and loss of semantic fidelity. Regarding Walk-the-Dog, the dog’s turning motions often appear unnatural, and the transferred motion is not consistently semantically aligned with the source. This occurs because the alignment in their latent space is purely structural; when the structural properties of the two datasets differ, the retargeting process fails. Regarding M2M, our experiments indicate that the method performs well when the source and target datasets contain semantically similar motions, particularly short motion sequences with close correspondence. Since the method relies on sparse joint correspondences and a

database search to retrieve candidate motions, it can fail when the target database is large and semantically heterogeneous, often producing animations that are semantically inconsistent with the source motion. An example of this behavior is shown in the supplementary video, where the M2M-generated dog stops while the source human continues walking, or continues moving while the source human stops, or remains standing while the source human sits down.

In contrast, our method successfully transfers motion across morphologies while preserving the semantic style of the target creature. Figure 9 presents representative qualitative comparisons between our approach, M2M, and the skeleton-aware method, highlighting the robustness of our method under challenging cross-morphology motion transfer scenarios. Animated visual comparisons are provided in the supplementary video.

3) *Motion Fidelity, Correspondence, and Control:* **Motion Fidelity:** The quality of the transferred motions generated by our method is high. To assess this quantitatively, we report the *Fréchet Inception Distance (FID)* between retargeted motions and real target skeleton motions, along with *jitter* and *foot-skating* metrics. The FID metric evaluates both the realism of the generated motions and their alignment with the target motion distribution. Following Chen et al. [28], we compute FID using the kinematic feature representation proposed in [41]. Jitter quantifies the temporal smoothness of the generated motion and is computed as the mean joint jerk. Jerk is defined as the third derivative of joint position with respect to time, $\mathbf{j}(t) = \frac{d^3 \mathbf{x}(t)}{dt^3}$, and lower jitter values indicate smoother trajectories across all joints. Foot skating measures violations of ground-contact constraints and is computed as the average horizontal displacement per frame when a foot is simultaneously close to the ground ($h < \tau_h$) and moving above a minimal velocity threshold ($\|\mathbf{v}_{xy}\| > \tau_v$). Lower values correspond to better adherence to stationary-foot constraints during the contact phase. Table I summarizes the FID, jitter, and foot-skating scores for our method, M2M, and the original motion sequences. Our method consistently outperforms M2M in terms of FID, indicating that the generated motions more closely match the distribution of the target motions. Furthermore, our method produces smoother animations, as reflected by lower jitter values, often even lower than those of the original sequences due to the temporal smoothing effect of the model. Finally, our method maintains low foot-skating errors, with skating values that remain comparable to the competing methods.

TABLE I: Motion fidelity comparison using jitter (10^2m/s^3), foot skating (cm/frame), and FID for our method, M2M, and the original (Og) motions. FID is computed from 8-second clips sampled from 7-minute animations for the dog, and from 1-minute animations for the human and ostrich, respectively.

	Dog			Human			Ostrich		
	Ours	M2M	Og	Ours	M2M	Og	Ours	M2M	Og
Jitter ($\downarrow$)	1.72	2.77	3.00	0.54	1.29	0.66	2.29	5.18	3.18
Skating ($\downarrow$)	1.78	1.11	0.07	0.19	0.87	0.0024	0.86	0.66	0.57
FID ($\downarrow$)	16.37	36.46	–	53.17	70.35	–	102.32	130.36	–

Spatial Correspondence: We evaluate spatial accuracy by projecting motion trajectories onto the ground plane. For chal-

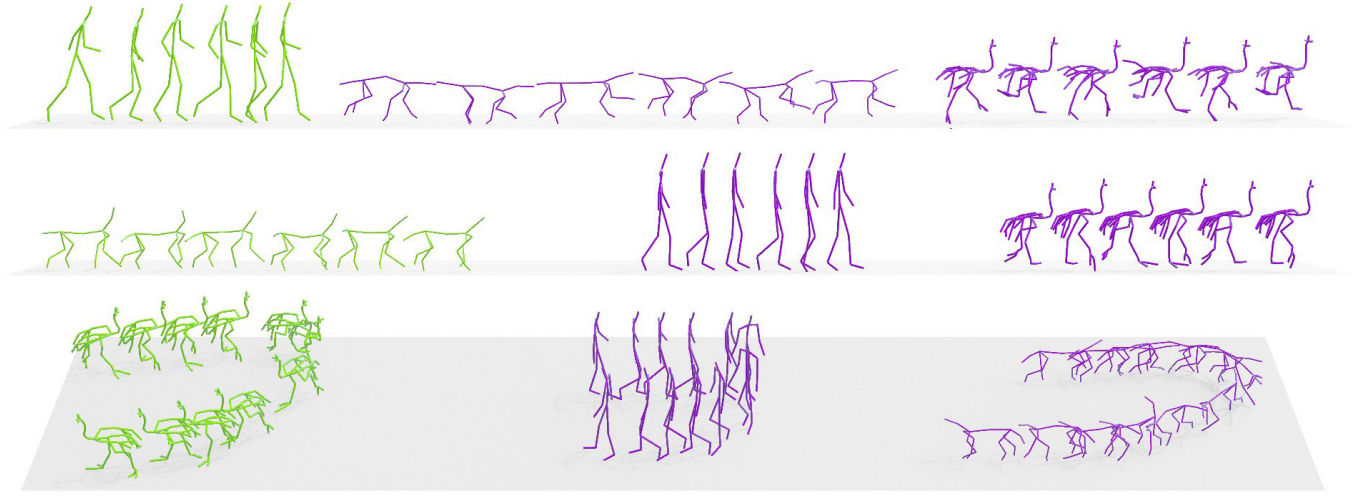


Fig. 6: **All-to-all motion transfer.** Our method transfers motions bidirectionally across species. Top: a human running motion (source) is transferred to a dog and an ostrich (outputs). Middle: a dog walking motion (source) is transferred to a human and an ostrich. Bottom: an ostrich walking-with-turn motion (source) is transferred to a human and a dog.

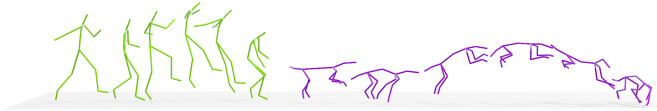


Fig. 7: Our method transfers a human running and jumping motion into a semantically similar dog motion.

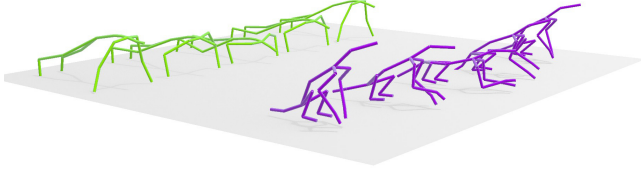


Fig. 8: Our method transfers human push-up motion (shown in green) into the dog space (purple), even though dog push-ups are not part of the training corpus. It can be seen that the dog’s motion closely mimics the style of the human.

lenging transfers such as *human* $\rightarrow$ *ostrich* and *human* $\rightarrow$ *dog*, our method produces trajectories that remain consistent with the target locomotion style while preserving the intended path. Naturally, the scale differs due to variations in species size, gait length, and the transition from bipedal to quadrupedal motion. Nevertheless, there is no drift from the intended paths, and our method successfully maintains spatial coherence as well as the correct body height (see Figure 10).

Temporal Correspondence: We investigate temporal alignment across morphologies with different gait cycles (e.g., biped $\leftrightarrow$ quadruped). Here, temporal alignment refers to phase matching, i.e., synchronizing characters’ footsteps to keep their foot movements in sync. By incorporating foot-contact signals, our method achieves phase-consistent alignment, enabling synchronized turns, jumps, and transitions across species (see Figure 11). The numerical results in Table II confirm a significant reduction in phase-shift error

and improved timing alignment compared to the condition in which foot-contact signals were disabled. This improvement is evident in both the contact disagreement ratio and the angular phase error (measured in radians) for the periodic gait signals during motion transfer across the different domains. The angular phase difference between the source and target motions is computed as

$$\Delta\phi(t) = \text{atan2}(\sin(\phi_t(t) - \phi_s(t)), \cos(\phi_t(t) - \phi_s(t))),$$

where $\phi_t(t)$ and $\phi_s(t)$ denote continuous locomotion phase values inferred from the detected binary foot-contact sequences of the target and source motions at time t , respectively. For each foot, a gait cycle is defined from one contact-on event to the next, with stance mapped to $[0, \pi)$ and swing to $[\pi, 2\pi)$. The phase difference $\Delta\phi(t)$ is computed as a wrapped angular difference constrained to the interval $[-\pi, \pi]$.

TABLE II: Phase alignment and timing consistency in motion transfer across morphologies, comparing conditions with and without foot-contact information.

Metric	With FC signals	Without FC signals
Angular phase (rad) ($\downarrow$)	0.345 ± 0.297	1.621 ± 0.885
Contact disagreement (% of frames) ($\downarrow$)	6.737	50.587

FC = Foot Contacts.

Contact disagreement counts only mismatches lasting ≥ 100 ms, or > 2 frames.

Speed Variation Control: We evaluate pace controllability by applying multiplicative scaling factors to the velocity-related control signals. The results demonstrate that our autoregressive model remains stable across different speeds, producing plausible locomotion that proportionally adapts in style while preserving the target’s gait. Visual results (Figure 12) illustrate a *human* $\rightarrow$ *dog* transfer with speed variations of $0.5\times$, $1.0\times$, and $4\times$, confirming that faster/slower motions maintain realism.

C. Ablation Studies

We conducted an extensive ablation study to evaluate the contribution of each component of our architecture. To vali-

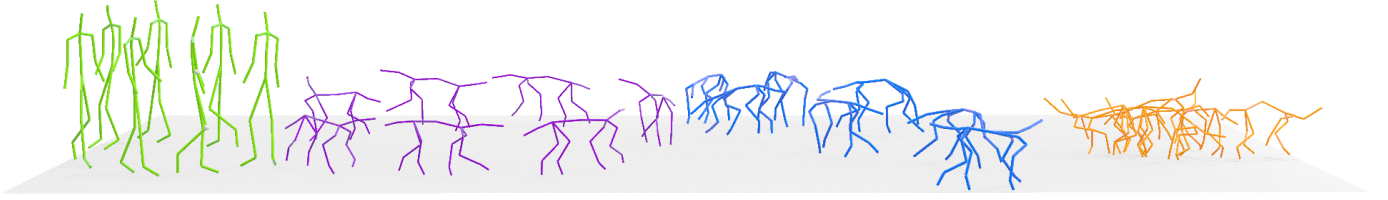


Fig. 9: **Comparison with state-of-the-art methods.** Visual comparison of our approach (purple) with M2M (blue) and the skeletal-aware baseline (orange). Our method produces higher-quality motions with better semantic preservation, while competing methods introduce motion artifacts or generate semantically incorrect behaviors.



Fig. 10: **Spatial Correspondence.** Our method generates trajectories that faithfully preserve the intended path while aligning with the target motion style. Human (in green) serves as the source motion, and the ostrich (purple) and dog (red) represent the transferred motions with stylistic adaptation. Although the trajectory scales differ due to variations in body size, gait length, and the transition from bipedal to quadrupedal locomotion, the results remain spatially coherent, showing no drift from the target paths.

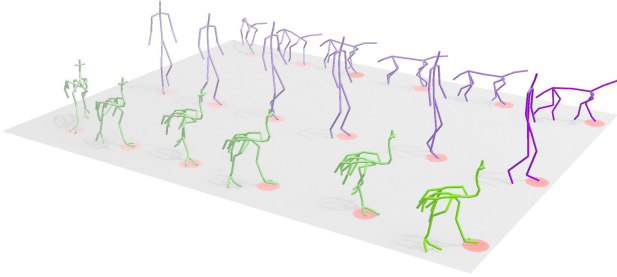


Fig. 11: **Phase alignment and timing consistency across morphologies.** Using an ostrich motion as input (source), our method transfers the motion to human and dog outputs while maintaining temporal correspondence, demonstrating consistent phase alignment and timing coherence across domains.

date the importance of our design choices, we systematically remove or modify individual elements of the framework and compare the resulting motion quality, semantic consistency, and stability.

Network Architecture: To assess the design of the Motion Generator, we compare three architectural variants: (1) our skeleton-aware network, (2) a non-skeleton-aware CNN

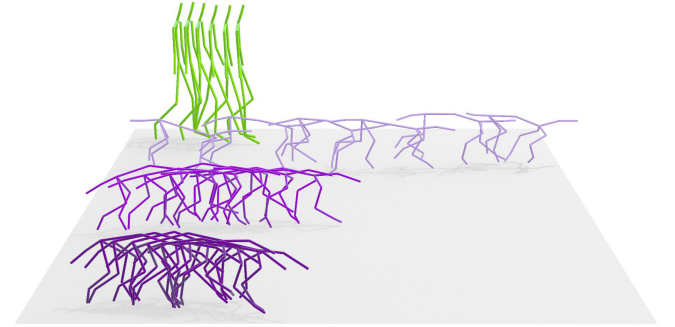


Fig. 12: **Speed variation control.** Using the same human motion (source, shown in green), our method transfers it to the dog (output, shown in purple with varying saturation) at different speed factors ($4\times$, $1\times$, $0.5\times$), demonstrating precise control over motion speed and style across domains. Effect of speed scaling: at $4\times$ speed the motion corresponds to running, at $1\times$ to normal walking, and at $0.5\times$ to slow walking.

baseline, and (3) the skeleton-aware model with the global root orientation included in the input. For the non-skeleton-aware baseline, we remove all skeletal convolution, pooling, and unpooling operations from the Motion Generator. Each skeleton-aware convolutional layer is replaced by a standard convolutional layer whose stride and number of input/output channels are selected to preserve the same output dimensionality at each stage of the encoder. Correspondingly, the decoder uses standard non-skeletal upsampling in place of skeletal unpooling. All remaining components of the VQ-VAE, as well as the training procedure, are kept unchanged. As shown in Table III, the skeleton-aware configuration achieves the lowest reconstruction error across both mean per-joint position error (MPJPE) and mean end-effector position error (MEEPE). Replacing it with a standard CNN increases reconstruction error, highlighting the benefit of topology-aware processing. Adding the global orientation to the encoder substantially worsens performance, confirming that separating root motion yields cleaner codebooks, improves VQ-VAE convergence, and results in more compact and transferable latent representations.

Separate Codebook: In addition, we evaluate the benefit of using a separate low-dimensional codebook rather than reusing the primary high-dimensional codebook. Using the primary high-dimensional codebook during the training of the Autoregressive Code Predictor leads to slower convergence,

TABLE III: Mean per-joint and end-effector position error across three network configurations.

Model Variant	MPJPE ($\downarrow$)	MEEPE ($\downarrow$)
SAN + root proxy	0.026	0.037
SAN + global root orientation	0.073	0.119
Non-skeleton-aware CNN	0.040	0.062

SAN = Skeleton-Aware Network

degraded motion quality, including less stable trajectories and weaker style preservation. The low-dimensional codebook provides a more compact and semantically meaningful latent space, specifically learned for the task of modeling the temporal and ordinal dynamics between the different codebook vectors, improving both consistency and generalization.

GRU vs. MLP Temporal Modeling: We also compare GRU with MLP for temporal modeling. The GRU more effectively captures long-term temporal dependencies, yielding smoother and more coherent reconstructions of the transferred motion. In contrast, the MLP struggles to maintain temporal continuity, resulting in noticeable drift and jitter.

Teacher Forcing: We further analyze the impact of teacher forcing during training. Removing teacher forcing leads to slower convergence and greater instability during inference, with errors compounding autoregressively. With full teacher forcing, the model suffers from exposure bias and generalizes poorly to unseen data. In contrast, a mixed-input strategy allows the model to learn a more stable temporal prior, yielding significantly more reliable long-horizon predictions and stronger generalization.

Control Signals: We evaluate the contribution of each control signal used by our predictor. Removing these signals consistently reduces motion quality and semantic alignment.

- *Trajectory-related signals (acceleration, velocity, pelvis height, yaw acceleration/rate):* Removing these signals causes sharp performance drops. Style mismatches become frequent; for example, incorrect pelvis height disrupts sitting behavior, missing yaw cues leads to unnatural turns, and removing velocity or acceleration signals reduces overall motion fidelity.
- *Foot-contact signals:* Without explicit foot-contact supervision, motions exhibit phase ambiguity, incorrect gait timing, and substantial foot sliding. Including foot-contact labels improves cycle synchronization, reduces foot-skating artifacts, and ensures more realistic ground interactions.
- *Additional signals:* We also tested other signals, including normalized head height. While potentially useful for certain species or specific movements (e.g., push-ups), it did not provide measurable benefits for locomotion in the human–dog–ostrich domain. For broader species families, richer or hierarchical signals (e.g., tail dynamics or body orientation relative to trajectory) may become important.

Autoregressive Modeling: Removing autoregressive modeling severely impacts temporal coherence. Motions become jittery and less stable, especially during faster gaits or complex transitions. Autoregression ensures local temporal continuity and higher-fidelity long-term predictions.

Continuous vs. Discrete Latent Representations: Finally, we compare continuous and discrete latent spaces. Continuous representations often produce overly smoothed motions that may lie outside the feasible motion manifold, whereas discrete (VQ) representations generate more natural and physically plausible motions. When the VQ-VAE reconstruction stage is removed, the resulting latent-spaces become less compact and lose semantic structure, degrading style preservation across species.

Overall, these ablations confirm that both the two-stage training procedure and the morphology-agnostic control design are essential for achieving scalable, realistic, and semantically consistent cross-morphology motion transfer. An animated visual evaluation of the ablation study can be found in the supplementary video.

V. DISCUSSION AND CONCLUSIONS

We introduced a framework for cross-morphology motion transfer with semantic stylistic alignment, enabling all-to-all retargeting without joint correspondences or reference skeletons. By combining morphology-agnostic control signals with morphology-specific codebooks, our method produces high-quality motion across heterogeneous topologies while preserving stylistic fidelity. The modular design scales to new morphologies without retraining existing models, making it practical for animation pipelines involving diverse character sets.

Our experiments highlight the strengths and limitations of morphology-agnostic motion transfer. By relying on semantically consistent control signals rather than explicit joint mappings, the framework generalizes across heterogeneous topologies such as humans, dogs, and ostriches, while avoiding artifacts like distorted limb motion or anthropomorphized behaviors. The modular design also supports scalability: new morphologies can be added without retraining existing models, enabling libraries of stylistically faithful motion priors. Unlike unified all-to-all models that enforce a shared latent space, our morphology-specific codebooks capture priors independently, reducing human-centric bias and better respecting non-human locomotion. However, this separation comes at the cost of explicit pairwise consistency, making alignment dependent on the expressiveness of the control signals.

Limitations: Despite its flexibility, our framework requires training a separate VQ-VAE and autoregressive predictor for each target morphology, rather than a single universal model. As a result, synthesis is limited to previously seen target morphologies, which may constrain scalability when deploying a large and diverse set of characters. Performance also depends on the quality of the control signals: when source and target motions are semantically far apart (e.g., mapping human dancing or martial arts to animals), alignment may degrade. Failure cases arise when the autoregressive network favors frequently occurring codes over rare ones, causing infrequent motions in the target dataset to be under-represented during sampling. Source motion semantics may not be preserved when the combination of control signals lies outside the training distribution. Our evaluation primarily

focused on locomotion and coarse postural changes, leaving complex behaviors such as tool use, acrobatics, or multi-character interactions as open challenges. Furthermore, since most zoo-style cyclic keyframed datasets lack translational motion and loop in place, we artificially introduce translation. However, this procedure introduces misalignments and foot sliding, which degrade the quality of the reconstructed motion. Finally, while morphology-agnostic signals such as velocity and height enable robust motion transfer, they may overlook subtler stylistic cues (e.g., tail dynamics or wing articulation). For example, our method accounts for trajectory and velocity but not the body's orientation relative to the motion path, therefore it does not currently support side-walking behaviors. Incorporating additional control signals or hierarchical representations may help overcome these limitations.

Future Work: Promising directions include scaling to larger multi-species datasets and extending the framework to interactive retargeting and real-time generation (e.g., token-level modifications during encoding or decoding [40]). An important future direction is zero-shot transfer to unseen target morphologies, where semantic motion knowledge learned across multiple characters could generalize to a new skeleton without requiring morphology-specific motion training. Recent advances in zero-shot generalization for unseen robot embodiments and large-scale human motion generation [42], [43] suggest that embodiment-agnostic representations and large-scale multi-domain training may provide useful directions toward this goal. Incorporating physics-based priors could improve contact fidelity and energy efficiency, while coupling with generative diffusion models may further enhance diversity and controllability.

ACKNOWLEDGMENTS

We would like to thank Theodoros Kyriakou for his assistance with the visualizations of our results. This project has received funding from the European Commission's Horizon Europe program under grant agreement 101178362, the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 739578, the Government of the Republic of Cyprus, and the MCIN/AEI/10.13039/501100011033/FEDER "A way to make Europe", UE (PID2021-122136OB-C21).

REFERENCES

- [1] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: adversarial motion priors for stylized physics-based character control," *ACM Trans. Graph.*, vol. 40, no. 4, Jul. 2021.
- [2] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano, "Human motion diffusion model," in *The Eleventh International Conference on Learning Representations*, ser. ICLR'23, 2023.
- [3] J. L. Ponton, A. Winkler, L. Kavan, Y. Ye, and P. Kadlec, "Stymo: Fast and controllable few-shot motion style transfer," *ACM Trans. Graph.*, 2026.
- [4] D.-K. Jang, S. Park, and S.-H. Lee, "Motion puzzle: Arbitrary motion style transfer by body part," *ACM Trans. Graph.*, vol. 41, no. 3, Jun. 2022.
- [5] M. Shi, K. Aberman, A. Aristidou, T. Komura, D. Lischinski, D. Cohen-Or, and B. Chen, "MotionNet: 3d human motion reconstruction from monocular video with skeleton consistency," *ACM Trans. Graph.*, vol. 40, no. 1, Sep. 2020.
- [6] S. Tripathi, O. Taheri, C. Lassner, M. J. Black, D. Holden, and C. Stoll, "HUMOS: Human motion model conditioned on body shape," in *Proceedings 18th European Conference on Computer Vision Part XVI*, ser. ECCV'24. Berlin, Heidelberg: Springer-Verlag, Oct. 2024, p. 133–152.
- [7] K. Aberman, P. Li, D. Lischinski, O. Sorkine-Hornung, D. Cohen-Or, and B. Chen, "Skeleton-aware networks for deep motion retargeting," *ACM Trans. Graph.*, vol. 39, no. 4, Aug. 2020.
- [8] P. Li, S. Starke, Y. Ye, and O. Sorkine-Hornung, "Walkthedog: Cross-morphology motion alignment via phase manifolds," in *ACM SIGGRAPH 2024 Conference Papers*, ser. SIGGRAPH '24. NY, USA: ACM, 2024.
- [9] Q. Zhao, P. Li, W. Yifan, S.-H. Olga, and G. Wetzstein, "Pose-to-motion: Cross-domain motion retargeting with pose prior," *Computer Graphics Forum*, vol. 43, no. 8, p. e15170, 2024.
- [10] P. Li, K. Aberman, Z. Zhang, R. Hanocka, and O. Sorkine-Hornung, "GANimator: neural motion synthesis from a single sequence," *ACM Trans. Graph.*, vol. 41, no. 4, Jul. 2022.
- [11] S. Raab, I. Leibovitch, G. Tevet, M. Arar, A. H. Bermano, and D. Cohen-Or, "Single motion diffusion," in *The Twelfth International Conference on Learning Representations*, ser. ICLR '24, 2024.
- [12] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *ACM Trans. Graph.*, vol. 36, no. 4, Jul. 2017.
- [13] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "DeepMimic: example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018.
- [14] I. Jang, S. Choi, S. Hong, C. Kim, and J. Noh, "Geometry-aware retargeting for two-skinned characters interaction," *ACM Trans. Graph.*, vol. 43, no. 6, Nov. 2024.
- [15] R. Villegas, J. Yang, D. Ceylan, and H. Lee, "Neural kinematic networks for unsupervised motion retargeting," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, ser. CVPR'18, 2018, pp. 8639–8648.
- [16] K. Aberman, Y. Weng, D. Lischinski, D. Cohen-Or, and B. Chen, "Unpaired motion style transfer from video to animation," *ACM Trans. Graph.*, vol. 39, no. 4, Aug. 2020.
- [17] I. Gat, S. Raab, G. Tevet, Y. Reshef, A. H. Bermano, and D. Cohen-Or, "Anytop: Character animation diffusion with any topology," in *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers*, ser. SIGGRAPH Conference Papers '25. NY, USA: ACM, 2025.
- [18] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6309–6318.
- [19] M. Gleicher, "Retargeting motion to new characters," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '98. NY, USA: ACM, 1998, p. 33–42.
- [20] H. J. Shin, J. Lee, S. Y. Shin, and M. Gleicher, "Computer puppetry: An importance-based approach," *ACM Trans. Graph.*, vol. 20, no. 2, p. 67–94, Apr. 2001.
- [21] S. Lee, T. Kang, J. Park, J. Lee, and J. Won, "SAME: Skeleton-agnostic motion embedding for character animation," in *SIGGRAPH Asia 2023 Conference Papers*, ser. SA '23. NY, USA: ACM, 2023.
- [22] L. Hu, Z. Zhang, C. Zhong, B. Jiang, and S. Xia, "Pose-aware attention network for flexible motion retargeting by body part," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 8, pp. 4792–4808, 2024.
- [23] C. Hecker, B. Raabe, R. W. Enslow, J. DeWeese, J. Maynard, and K. van Prooijen, "Real-time motion retargeting to highly varied user-created morphologies," in *ACM SIGGRAPH 2008 Papers*, ser. SIGGRAPH '08. NY, USA: ACM, 2008.
- [24] M. Abdul-Massih, I. Yoo, and B. Benes, "Motion style retargeting to characters with different morphologies," *Comput. Graph. Forum*, vol. 36, no. 6, p. 86–99, Sep. 2017.
- [25] D. Egan, D. Cosker, and R. McDonnell, "Neurodog: Quadruped embodiment using neural networks," *Proc. ACM Comput. Graph. Interact. Tech.*, vol. 6, no. 3, Aug. 2023.
- [26] D. Egan, A. Jovane, J. Szkaradek, G. Fletcher, D. Cosker, and R. McDonnell, "Dog code: Human to quadruped embodiment using shared codebooks," in *Proceedings of the 17th ACM SIGGRAPH Conference on Motion, Interaction, and Games*, ser. MIG '24. NY, USA: ACM, 2024.
- [27] G. Martinelli, N. Garau, N. Bisagno, and N. Conci, "MoMa: Skinned motion retargeting using masked pose modeling," *Computer Vision and Image Understanding*, vol. 249, p. 104141, 2024.

- [28] L.-H. Chen, Y. Zhang, Z. Yin, Z. Dou, X. Chen, J. Wang, T. Komura, and L. Zhang, "Motion2motion: Cross-topology motion transfer with sparse correspondence," in *SIGGRAPH Asia 2025 Conference Papers*, ser. SA '25. NY, USA: ACM, 2025.
- [29] G. E. Henter, S. Alexanderson, and J. Beskow, "MoGlow: probabilistic and controllable motion synthesis using normalising flows," *ACM Trans. Graph.*, vol. 39, no. 6, Nov. 2020.
- [30] Y. Wang, C. Guo, Y. Mu, M. G. Javed, X. Zuo, J. Lu, H. Jiang, and L. Cheng, "Motiondreamer: One-to-many motion synthesis with localized generative masked transformer," in *The 13th International Conference on Learning Representations*, ser. ICLR'25, 2025.
- [31] Y. Huang, W. Wan, Y. Yang, C. Callison-Burch, M. Yatskar, and L. Liu, "CoMo: Controllable motion generation through language guided pose code editing," in *Proceedings of the 18th European Conference on Computer Vision, Part XXIX*, ser. ECCV'24. Berlin, Heidelberg: Springer-Verlag, 2024, p. 180–196.
- [32] Z. Zhang, Y. Wang, W. Mao, D. Li, R. Zhao, B. Wu, Z. Song, B. Zhuang, I. Reid, and R. Hartley, "Motion anything: Any to motion generation," 2025.
- [33] W. Lee, J. Jeong, T. Moon, H.-J. Kim, J. Kim, G. Kim, and B.-U. Lee, "How to move your dragon: Text-to-motion synthesis for large-vocabulary objects," 2025.
- [34] Z. Yang, M. Zhou, M. Shan, B. Wen, Z. Xuan, M. Hill, J. Bai, G.-J. Qi, and Y. Wang, "OmniMotionGPT: Animal motion generation with limited data," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, ser. CVPR'24, 2024, pp. 1249–1259.
- [35] H. Zhang, S. Starke, T. Komura, and J. Saito, "Mode-adaptive neural networks for quadruped motion control," *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018.
- [36] D. Reda, J. Won, Y. Ye, M. van de Panne, and A. Winkler, "Physics-based motion retargeting from sparse inputs," *Proc. ACM Comput. Graph. Interact. Tech.*, vol. 6, no. 3, Aug. 2023.
- [37] T. Li, J. Won, A. Clegg, J. Kim, A. Rai, and S. Ha, "ACE: Adversarial correspondence embedding for cross morphology motion retargeting from human to nonhuman characters," in *SIGGRAPH Asia 2023 Conference Papers*, ser. SA '23. NY, USA: ACM, 2023.
- [38] S. Starke, H. Zhang, T. Komura, and J. Saito, "Neural state machine for character-scene interactions," *ACM Trans. Graph.*, vol. 38, no. 6, Nov. 2019.
- [39] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, ser. CVPR'19, 2019, pp. 5738–5746.
- [40] J. L. Ponton, H. Yun, A. Aristidou, C. Andujar, and N. Pelechano, "Sparseposer: Real-time full-body motion reconstruction from sparse data," *ACM Trans. Graph.*, vol. 43, no. 1, Oct. 2023.
- [41] L. Siyao, W. Yu, T. Gu, C. Lin, Q. Wang, C. Qian, C. C. Loy, and Z. Liu, "Bailando: 3d dance generation by actor-critic gpt with choreographic memory," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, ser. CVPR'22, 2022, pp. 11 040–11 049.
- [42] Y. Ge, Y. Yan, M. Han, C. Li, Z. Ding, J. Li, F. Yang, T. Yang, T. Shen, C. Ouyang *et al.*, "Crosszero: Zero-shot of unseen embodiment manipulation via object-centric hierarchical diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 4446–4454.
- [43] K. Fan, S. Lu, M. Dai, R. Yu, L. Xiao, Z. Dou, J. Dong, L. Ma, and J. Wang, "Go to zero: Towards zero-shot motion generation with million-scale data," in *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2025, pp. 13 336–13 348.



Alexios Mylordos is an MSc student in Artificial Intelligence at the University of Cyprus and a research associate at the CYENS Centre of Excellence. He holds a BSc in Computer Science (with honors) from the University of Cyprus. His research interests include computer graphics, character animation, generative models, and machine learning. Contact him at alexmylordos@gmail.com.



Jose Luis Ponton is a Postdoctoral Researcher at the Max Planck Institute for Informatics, Germany. Jose Luis obtained his PhD from the Universitat Politècnica de Catalunya, Spain, where he also received his M.Sc. in Computer Graphics and B.S. in Computer Science. His research interests include character animation, robotics, generative AI, virtual reality, machine learning, and computer graphics. Contact him at jpononm@mpi-inf.mpg.de.



of virtual humans and crowds, and human-avatar interaction in Virtual Reality. Contact her at npelechano@cs.upc.edu.



Andreas Aristidou is an Associate Professor in the Department of Computer Science at the University of Cyprus and Senior Research Fellow at CYENS Centre of Excellence. He was previously a European Trust fellow at the University of Cambridge, where he earned his PhD. Andreas holds a BSc from the National and Kapodistrian University of Athens and an MSc (with honors) from King's College London. His research focuses on computer graphics, particularly character animation, where he uses techniques like machine learning and generative models to create virtual humans. His work also extends to digital heritage, VR/AR/MR environments, and the application of Conformal Geometric Algebra in computer graphics. Contact him at a.aristidou@ieee.org.