Policy-Space Diffusion for Physics-Based Character Animation

MICHELE ROCCA, Computer Science, University of Copenhagen, Copenhagen, Denmark
SUNE DARKNER, Computer Science, University of Copenhagen, Copenhagen, Denmark
KENNY ERLEBEN, Computer Science, University of Copenhagen, Copenhagen, Denmark
SHELDON ANDREWS, Software and IT Engineering, École de technologie supérieure, Montreal, Canada

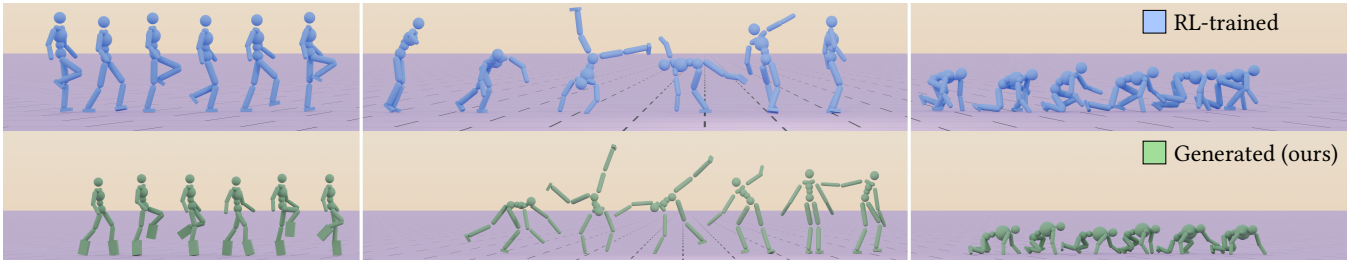

Fig. 1. Our compact, similarity-preserving policies allow us to generate the weights of the actor-network adapting to novel character morphology, unseen by the model. Using a diffusion model trained on less than 50 labeled policies, we learn and sample from the underlying data manifold.

Adapting motion to new contexts in digital entertainment often demands fast agile prototyping. State-of-the-art techniques use reinforcement learning policies for simulating the underlined motion in a physics engine. Unfortunately, policies typically fail on unseen tasks and it is too time-consuming to fine-tune the policy for every new morphological, environmental, or motion change. We propose a novel point of view on using policy networks as a representation of motion for physics-based character animation. Our policies are compact, tailored to individual motion tasks, and preserve similarity with nearby tasks. This allows us to view the space of all motions as a manifold of policies where sampling substitutes training. We obtain memory-efficient encoding of motion that leverages the characteristics of control policies such as being generative, and robust to small environmental changes. With this perspective, we can sample novel motions by directly manipulating weights and biases through a Diffusion Model. Our newly generated policies can adapt to previously unseen characters, potentially saving time in rapid prototyping scenarios. Our contributions include the introduction of Common Neighbor Policy regularization to constrain policy similarity during motion imitation training making them suitable for generative modeling; a Diffusion Model adaptation for diverse morphology; and an open policy dataset. The results show that we can learn non-linear transformations in the policy space from labeled examples, and conditionally generate new ones. In a matter of seconds, we sample a batch of policies for different conditions that show comparable motion fidelity metrics as their respective trained ones.

CCS Concepts: • **Computing methodologies** → **Animation**; *Physical simulation*; *Reinforcement learning;*

Additional Key Words and Phrases: Character animation, physics-based control, motion retargeting, control policies, deep reinforcement learning, diffusion-transformer models

ACM Reference Format:
Michele Rocca, Sune Darkner, Kenny Erleben, and Sheldon Andrews. 2025. Policy-Space Diffusion for Physics-Based Character Animation. *ACM Trans. Graph.* 44, 3, Article 25 (May 2025), 18 pages. https://doi.org/10.1145/3732285

Project page:
michelerocca.github.io/projects/policy-space_diffusion

1 Introduction

Motion synthesis for physically based characters has received increasing interest in recent years, with developed approaches successfully demonstrating skilled motion synthesis for climbing [Naderi et al. 2019], juggling [Chemin and Lee 2018; Luo et al. 2021], boxing [Won et al. 2021], locomotion [Peng et al. 2018], and many other tasks. This technology has important applications in computer graphics and animation. **Deep reinforcement learning (DRL)** is particularly interesting for interactive and real-time applications, such as video games, since control policies may be efficiently executed online, allowing virtual characters to operate in dynamic environments.

Although DRL has demonstrated impressive results for online motion synthesis, it is characterized by the long learning times required to generate a control policy. This drawback can be mitigated by employing algorithms, such as **proximal policy optimization (PPO)** [Schulman et al. 2017], that permit parallel exploration of the cross-product of the state-action space, but even imitation learning and simple locomotion tasks can require many hours of training time. This leads to time and computational resource challenges

Kenny Erleben and Sheldon Andrews equally contributed to supervision of the project.
Authors' Contact Information: Michele Rocca, Computer Science, University of Copenhagen, Copenhagen, Region Hovedstaden, Denmark; e-mail: miro@di.ku.dk; Sune Darkner, Computer Science, University of Copenhagen, Copenhagen, Region Hovedstaden, Denmark; e-mail: darkner@di.ku.dk; Kenny Erleben, Computer Science, University of Copenhagen, Copenhagen, Region Hovedstaden, Denmark; e-mail: kenny@di.ku.dk; Sheldon Andrews, Software and IT Engineering, École de technologie supérieure, Montreal, Quebec, Canada; e-mail: sheldon.andrews@etsmtl.ca.

ACM 0730-0301/2025/05-ART25
https://doi.org/10.1145/3732285

when control policies are to be generated for a large number of characters and tasks.

Recent work by Xu et al. [2023] aims to reduce training times through policy reuse when retargeting policies to characters with different morphologies or to different environments. However, their approach requires training using high-end computing resources and storing modifications to an existing policy network, in addition to the original policy. This means that expectations of computing and memory resources remain fairly high, which can be prohibitive for interactive and real-time computer graphics applications on devices with reduced resources such as mobile devices and so-called thin clients. This is particularly the case where many different control policies are to be loaded and executed simultaneously. Thus, the ability to quickly generate compact policies is particularly useful for such platforms and applications.

Diffusion models (**DMs**) are powerful deep generative models for efficiently compressing large datasets into compact representations. This efficiency facilitates the generation of new, diverse instances, with the added benefit of controllability through conditioning mechanisms. Notably, their accuracy in resolution makes them suitable for manipulating data sensitive to variations like the weights and biases of deep neural networks. Building on recent advances in using DMs to generate implicit neural fields for 3D shapes [Erkoç et al. 2023], we introduce a novel approach for policy synthesis that eliminates the need for additional training time when retargeting policies to different character morphologies, environments, or motion types. Furthermore, our synthesized policies are highly compact, making them well-suited for deployment on devices with limited memory and computational resources. We summarize the contributions of our work as follows:

- A generative learning framework that learns the conditional distribution of policy weights and allows generating novel control policies by sampling from a continuum of task and environment variations.
- Introducing a novel regularization strategy, called **common neighbor policy** (**CNP**) regularization, that preserves the similarity of policies trained from similar motions.
- A policy dataset consisting of hundreds of policies with different motions, morphology, and terrains, openly available for further studies on this topic.

2 Related Work

Related work can be divided into three sub-categories one about motion retargeting approaches, a second on DRL, and the last is about DM approaches. Our work touches on all these three categories.

2.1 Motion Retargeting

Many of the examples used to demonstrate our proposed framework focus on retargeting the control policies to new characters and environments. However, motion retargeting has long been of interest to the computer animation community. Traditional approaches rely on space-time trajectory optimization to ensure that specific position-level constraints are enforced for the motion when retargeted to different morphologies [Gleicher 1998], particularly foot-ground contact [Kovar et al. 2002b]. Other work has shown that stylistic transfer is possible between different character skeletons [Abdul-Massih et al. 2017; Feng et al. 2012]. Deep learning has recently been employed as a tool to retarget motions to different character skeletons in 2D [Aberman et al. 2019] and 3D [Aberman et al. 2020], as well as to resolve "foot skating" artifacts [Mourot et al. 2022a]. Lee et al. [2023] used a skeleton-agnostic embedding to characterize and transfer motions. Li et al. [2023] proposed a similar idea for retargeting motion from human to non-human characters. While many early works focus on kinematic retargeting approaches, physics has proven to be beneficial [Tak and Ko 2005]. Reda et al. [2023] proposed using reinforcement learning to produce plausible motions for characters and avatars when driven by a human actor. We also re-applied physics-based reinforcement learning to generate a dataset of extra input motions for our framework.

2.2 Deep Reinforcement Learning for Motion Control

Learning skills from human motion clips is a well-studied approach for generating natural motions for physics-based characters [Mourot et al. 2022b]. Specifically, DRL has emerged as a useful technique for achieving complex tasks in dynamic environments [Kwiatkowski et al. 2022]. For instance, by learning composite control policies that combine skills encoded by a small number of specific motion clips [Peng et al. 2018], or even larger collections of unstructured motion clips [Peng et al. 2022, 2021; Xu and Karamouzas 2021]. Bergamin et al. [2019] proposed to learn responsive DRL controllers based on exemplars generated by a motion-matching technique. Many works demonstrate the ability of DRL to retarget motion to different character morphologies and environments, but typically require retraining the policy for different tasks and target environments, or use a curriculum strategy [Xie et al. 2020]. Several recent works have proposed approaches to reduce the tedious training times associated with learning DRL. Ren et al. [2023] showed that the use of a differentiable physics engine could dramatically accelerate imitation learning. More recently, Xu et al. [2023] proposed to learn additional layers that augment previously trained control policies, allowing a policy to be adapted to new tasks, including changes in character morphology and terrain. The initial training time is amortized for each new task, which requires the order of minutes to train. Bohlinger et al. [2024] propose learning an abstract controller for several legged robots for the locomotion task. Close to our work, Won and Lee [2019] approach the morphology adaptation task by training a parametric controller for different characters.

Our approach differs from others by first pre-training unconditional policies and then conditioning a policy generator at a later stage, enabling more flexibility since policy learning and conditioning are done separately. Conceptually, our framework allows the reuse of policies from a database to impart new characteristics to generated policies. We demonstrate these capabilities by combining morphology adaptation with terrain adaptation, joining two independent datasets — one featuring morphological variations and the other featuring terrain variations. The resulting generated policies exhibit characteristics drawn from both datasets. Furthermore, our work introduces a regularization term to deal with the redundant nature of control policy representation. Intuitively speaking, this gives us unique representations of the policies in our policy dataset

that are used as input for a DM, and sampling from this model can instantaneously produce valid policy variations, avoiding costly retraining.

2.3 Motion Diffusion Models

Recent advances in diffusion-based generative models have had broad impacts on the field of motion synthesis. They show great promise in their ability to capture the diversity of large motion datasets and furthermore provide the ability to guide the model toward generating motion sequences with specific traits. Of particular interest are text-conditioned **motion diffusion models** (**MDM**), which allow the user to synthesize motion based on a natural language description. Tevet et al. [2023] were among the first to propose a transformer-based MDM conditioned on a CLIP based textual embedding. Zhang et al. [2024] proposed a similar architecture, but included a denoising model to support the generation of longer motion clips and more complex textual prompts. PriorMDM [Shafir et al. 2024] shows that DMs are well-suited for synthesizing long motion sequences and blending to produce prescribed trajectories. Others have demonstrated the ability of DMs to synthesize human-to-human interactions [Liang et al. 2024]. This idea has been extended to use other conditioning inputs, such as synthesizing motion from audio [Alexanderson et al. 2023; Qi et al. 2023].

Du et al. [2023] conditioned a DM on sparse VR tracking input and showed impressive results for full-body pose reconstruction. Raab et al. [2024] use a DM to learn the motion patterns of a single motion clip and generate stylized variations, useful for style transfer and crowd animation. Tevet et al. [2024] recently proposed a diffusion-based motion planner combined with a tracking policy to convert textual prompts into simulations that can interact with the environment.

The works Ze et al. [2024] and Chi et al. [2024] use a diffusion policy for generating robot manipulation actions from visual conditioning, where a DM substitutes the classic actor MLP. Similarly Truong et al. [2024] use state observations as a condition for generating actions using a diffusion policy for behavior cloning. Diffusion policy methods have recently demonstrated state-of-the-art results in robotic learning tasks. Despite their use of a DM, they are fundamentally different than our proposed learning framework. They use a diffusion process to transform states into action, we operate a diffusion process on the policy weights.

We take a novel approach by modeling the weights and biases of all policies as points on a shared multivariate differentiable manifold. This choice allows us to sample a continuum of policies, effectively representing a wide range of simulation parameters. To achieve this, we approximate the underlying manifold using a conditional DM. For categorical conditions, we establish conditional continuity by leveraging the principal components of the control policies. This design enhances our architecture's ability to generalize effectively to new samples, particularly when dealing with motion combination tasks.

3 Policy Weights as a Continuum

Our work takes a novel view of neural network-based control policies that views the network parameters as forming a continuum over task and environment variations.

Consider a motion clip $\mathbf{m} \equiv \{m_1, \ldots, m_M\}$, consisting of M successive poses. For imitation learning tasks, a neural network-based policy with weights w'

$$P^{\mathbf{m}}_{w'}(m_i) \rightarrow \mathbf{m}'$$

is trained to imitate the motion $\mathbf{m}$ from an initial pose m_i. It produces motion $\mathbf{m}'$ that is, in a generative way ($\sim$), close to $\mathbf{m}$ (i.e., $\mathbf{m}' \sim \mathbf{m}$).

Due to the over-parametrized nature of deep neural networks, the weights w' are not unique in representing the motion $\mathbf{m}$, another independent imitation learning training can lead to weights w'' of a policy

$$P^{\mathbf{m}}_{w''}(m_i) \rightarrow \mathbf{m}''$$

that still represents $\mathbf{m}$ in a generative way. Here, w' and w'' are not numerically close to each other, while $\mathbf{m}'' \sim \mathbf{m}' \sim \mathbf{m}$.

To use deep generative modeling techniques on the weight-space, we want to encourage the weights to converge to similar values if the underlying motion is the same, or $w' \approx w''$. More generally, the weights should change continuously and smoothly as the target motion clip is perturbed. Consider two motions $\mathbf{m} \equiv \{m_i\}$ and $\mathbf{n} \equiv \{n_j\}$ where the latter is a perturbation of the former (either numerically $\mathbf{m} \approx \mathbf{n}$, or in a generative way $\mathbf{m} \sim \mathbf{n}$). A desirable property is that their respective policies $P^{\mathbf{m}}_{w}$ and $P^{\mathbf{n}}_{u}$ consistently converge to similar weights, $w \approx u$. The hypothesis is that training the policies with a policy-gradient method such as PPO implies the existence of a differentiable manifold $\mathcal{P} \subseteq \mathbb{R}^D$ that describes how the D weights and biases change as the target motion deviates from the reference motion $\mathbf{m}$. Therefore the parameters $w, u \in \mathcal{P}$ are points on the manifold, and the geodesic distance

$$d\mathcal{P}(w, u) \approx 0\,.$$

We adopt a regularization strategy during the imitation training to encourage neighboring solutions and aim to approximate this manifold from data using a DM.

Our experiments utilize two distinct datasets. The first dataset comprises instances of motion capture data $\mathbf{m}$, utilized for training the policies, which we denote the Ground-Truth-set. We generate the second dataset, which consists of the policies $P^{\mathbf{m}}_{w}$ we use to train the DM. We shall refer to it as the Policy-set. The Ground-Truth-set originates from partitioned captures of the La Forge (LaFan1) dataset as described by Harvey et al. [2020]. These captures have been visualized and segmented into shorter clips to isolate individual motion types. Consequently, these shorter clips constitute our Ground-Truth-set.

3.1 Common Neighbor Policy (CNP) Regularization

Each policy is trained individually from specific motion clips within the Ground-Truth-set. The general pipeline is outlined at the top row of Figure 2. The Policy-set is derived from training the **adversarial motion prior** (**AMP**) model [Peng et al. 2021] with the PPO algorithm solely on imitation learning tasks, without incorporating any goal tasks during training. Eventually, the weights converge to one of the many local minima of the over-parametrized model. This means that for every policy learned to imitate a ground-truth motion, there exists a conspicuous number of equivalent policies with different weights that can be obtained by training

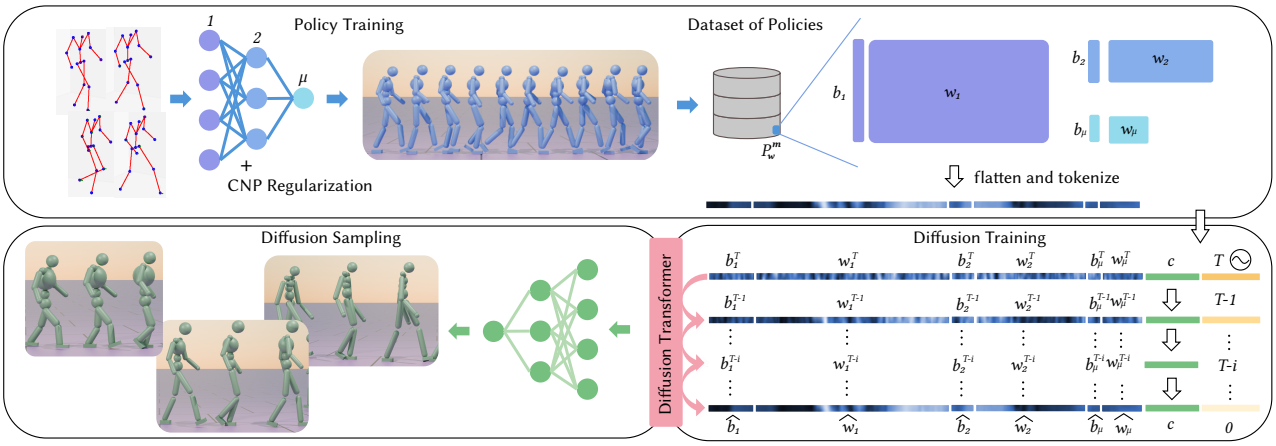

Fig. 2. An overview of the full pipeline of our method: For every motion in the kinematic motion-set we train our compact policy networks using our CNP regularization. The network outputs the mean action for each body joint given the state observation, while the variance is fixed. The architecture for generating the policies is a conditional Diffusion Transformer. It learns how to denoise standard Gaussian noise into the parameters of the control policy network. The condition is refreshed at each step. During sampling, we can present novel conditions to obtain novel policies. Here is an example of morphology adaptation.

for the same ground-truth motion. We want to restrict the solutions to the subset of solutions that are numerically close to each other.

For learning the imitation policies we follow Peng et al. [2021] and use the AMP. To encourage proximity among policies, we introduce a Gaussian prior around a CNP, which are pre-trained weights w_{CNP} from another imitation task, and it is used during the training of the whole Policy-set. This gives the augmented loss function proposed by Peng et al. [2021]:

$$\mathcal{L} \equiv \mathcal{L}_{\text{AMP}} + \lambda ||w - w_{\text{CNP}}||^2 .$$

The regularization consists of the Mean Squared Error between the current weights w of the on-training policy $P_w^{\mathbf{m}}$ and the weights of the CNP. We initialize the training of each imitation policy using the CNP weights, ensuring that the regularization loss is initially zero. As the policy is updated during training, the weights change in order to better imitate the desired motion. However, the regularization loss penalizes values of w that are different from w_{CNP}. The coefficient λ controls how much changes in the network weights are penalized, and $\lambda = 0.02$ is used for all experiments. We found that this clusters policies representing the same motion clip, while keeping policies for different motion types centered around the CNP regularization.

Figure 3 illustrates a 2D principal component reduced representation of the policies trained using this combined initialization and regularization approach. It shows how the policies trained from the same motion cluster together, and are distant from policies representing other motions.

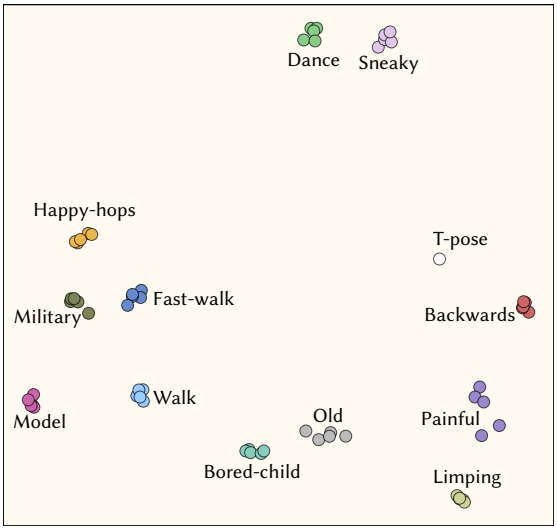

Fig. 3. Plot of the first two principal components of the control policies representing 12 motions using the T-pose policy as CNP. The figure illustrates how our regularization allows five policies, trained for imitating the same motion clip across independent training, to cluster together. Motions like Painful and Old show more loose clusters that we correlate with poor convergence. Figure 5 shows the cumulative explained variance of this PCA representation

3.2 A Compact Policy Architecture

Policies are represented by a **multi-layer perceptron** (**MLP**) with two fully connected hidden layers and ReLU activations. We use the same 223-dimensional state representation and 28-dimensional action output as AMP, where the state includes root-bone height, local body positions, rotations, velocities, and angular velocities. The output is a multivariate Gaussian distribution's mean action μ, while the standard deviation is fixed. In the original AMP architecture, the two hidden layers consisted of, respectively, 1024 and 512 neurons each, the input was standardized using its collected running mean and variance, leading to 770k parameters. To reduce the input size of the DM, we have decreased the size of the hidden layers to 128 and 32 neurons and removed the input standardization. Consequently, the weights and biases completely represent the motion, leading to 33k parameters. Even though our policies

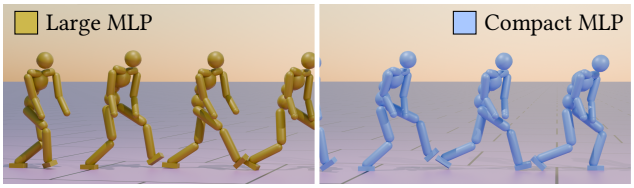

Fig. 4. The compact network we use can be trained to reach the same visual quality as a bigger network similar in size to the ones used in ASE [Peng et al. 2022]. We have reduced the number of parameters by 95% or more.

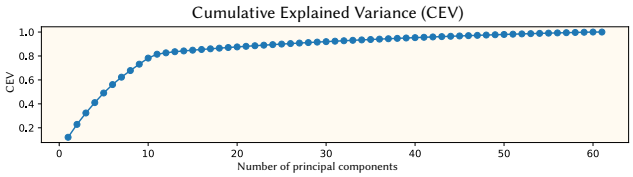

Fig. 5. Cumulative explained variance of the PCA encoding. The first 12 components account for 80% of the variation in our dataset with 5 instances of 12 motion policies in Figure 3. Therefore we consider it to be a good continuous alternative to the one-hot encoding for conditioning the DM on the motion combination task

contained fewer than 5% of the parameters used in [Peng et al. 2022], they produce motions that follow the reference motion clips, which suggests that compact networks can be used for learning individual reference motions. This is exemplified in Figure 4. Each policy file averages 2.8 MB in size.

4 Diffusion Model and Conditioning

To approximate the manifold underlying a set of policies trained on different conditions, we train a DM. We employ a modified version of the Diffusion-Transformer, building upon the framework proposed by Erkoç et al. [2023] and incorporating input-based conditioning. This conditioning method operates by concatenating the condition with the input and refreshing the corresponding part of the generated content at each diffusion step. Specifically, the weights and biases of a control policy network are concatenated with conditioning features that are specific to that policy. This integration ensures that the model remains aware of the conditioning context throughout the diffusion process, enabling accurate and context-sensitive policy generation. The input is tokenized layer by layer, treating each set of weights or biases in the network as a token, along with tokens for encoding the condition and diffusion timestep. This is exemplified in the bottom-right of Figure 2. For training, we utilize the DDPM method by Ho et al. [2020], and for sampling, we employ the more efficient DDIM method by Song et al. [2021].

Depending on the application, conditioning can take the form of either one-hot encoding for categorical factors, such as motion type, or min-max normalized scalars for continuous parameters like height, limb lengths, and terrain roughness. While for encoding the diffusion time step we use sinusoidal positional encoding, a simple normalized scalar appears to be sufficient for representing conditions such as height, morphology, and terrain. Our initial experiments on combining different motions showed that one-hot conditioning effectively directs the diffusion process to generate motion types from the training set, but struggles to generalize to combined motion types. To address this limitation, we convert categorical conditions into continuous ones using a latent representation as a condition label. Models such as **variational autoencoders** (**VAEs**) are widely used for non-linear latent encoding, but given the size of the input, they would be time-consuming to train. Our preliminary analysis of the dataset suggested that **principal component analysis** (**PCA**), although it is a linear latent representation, can separate the data clusters by motion type effectively already with the first two components as shown in Figure 3. We apply PCA to the policy network's parameters across the entire dataset and retain the first 12 components as a condition of our DM, as they capture more than 80% of the variance as illustrated in Figure 5. We consider this to be a feasible continuous replacement for the one-hot encoding. To have a standard and invertible representation of the policy weights used for training the model, we normalize them by using the minimum and maximum values across the whole training set to fit the range $[-1, 1]$. Consequently, the last layer of the decoder is adjusted to use a hyperbolic tangent activation function such that the output is in the same range as the input and can effectively represent it. We use 300 diffusion steps with a linear noise schedule in the range $[1e^{-4}, 1e^{-2}]$ as it resulted in a good balance between final-step diffusion noise, learning capability, and sampling time. By experimental tuning from the Erkoç et al. [2023] implementation, we reached a low training loss by using 3040 as the size of each token after linear projection, and 16 attention heads with 12 layers each for a total of 1537M parameters.

4.1 Quantitative Evaluation

We define policy quality based on the similarity between the simulated motion during policy replay and the motion of the ground truth clip. In this section, we present details about how motions synthesized by DM-generated policies are compared to their DRL counterparts and ground truth animations.

The quality of control policies is evaluated using the following metrics: (i) **Falling Rate (FR)**, (ii) motion similarity measured by **Dynamic Time Warping (DTW)**, and (iii) **Fréchet motion distance** (**FMD**).

Falling Rate: Running a policy in a simulator can sometimes lead to unexpected states that cause the policy to fail. Specifically, for many motions, falling during the simulation can be equivalent to an unrecoverable failure. Therefore, to quantify the reliability and success of motion imitation, the FR used by Won and Lee [2019] is defined as the percentage of instances in which the character falls during many independent simulations, e.g., percentage of zero indicates that the character never fell during the evaluation. This indicator loses meaning for motions that require the character to lie on the floor, for example crawling. For those, the FR is assumed to be zero. We evaluate the FR across 1024 simulations for every policy.

Dynamic Time Warping Similarity: For successful policies, we also want to quantify the quality of the motion imitation. Each simulation starts from a random pose from the dataset and the trajectories of the joints can slightly differ from the ground truth. For

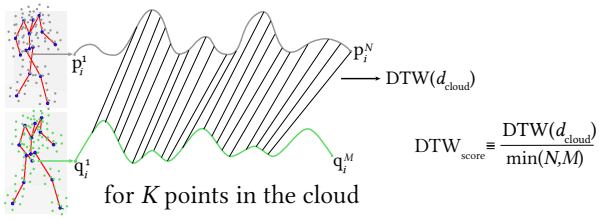

Fig. 6. The DTW score is calculated by using a custom distance on the full point cloud and we average the score by dividing for the length of the shorter sequence. The final score results in a distance between two sequences averaged over time and number of cloud points.

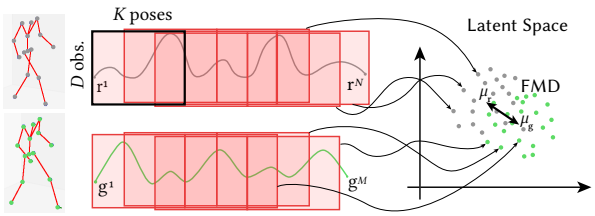

Fig. 7. The FMD is calculated by dividing the D motion observations in small patches of K poses via a moving window. Each patch is encoded into a latent space by means of a pre-trained auto-encoder, fine-tuned on the whole Ground-Truth-set. Each motion clip is represented as a distribution of points in the latent space, then the FD is computed as a measure of the similarity between the reference motion $\mathbf{r}$ and the generated motion $\mathbf{g}$.

this reason, it is important to use a metric showing robustness to time-shift and moderate amplitude deformation of the trajectories. Metrics like the Hausdorff Distance overestimate the pointwise similarity between poses in simulation and ground truth by the use of maximum and minimum operations. We instead use DTW to measure the similarity between the simulated trajectories and the retargeted, realigned ground truth. This process is illustrated in Figure 6. To avoid mixing physical units, the trajectories are represented by the temporal evolution of a 3D point cloud represented in meters. The point cloud is rigidly linked to the body joint and consists of cubes with 8 points per joint, similarly to Kovar et al. [2002a]. We use DTW to calculate the similarity as the cost of warping the simulated trajectories into the ground truth across all 3D points in the cloud, ensuring a comprehensive comparison of the full pose. The distance between points clouds is calculated as the average of the Euclidean distances between corresponding points in the two point clouds, such that

$$d_{\text{cloud}} \equiv \frac{1}{K} \sum_{i=1}^{K} \|\mathbf{p}_i - \mathbf{q}_i\|,$$

where $\mathbf{p}_i, \mathbf{q}_i \in \mathbb{R}^3$ are the ith points in the first and second cloud. To get a single score, we average across the time by dividing by the shorter sequence's length as we do not expect the warping to dominate the shifting and to avoid underestimating the metric. The resulting score is the average distance in meters of a point in the cloud to its respective ground truth after the alignment.

Fréchet Motion Distance: Some aspects of motion similarity that humans perceive through visual inspection are difficult to quantify with traditional metrics. We assume that some abstract motion features can be extracted by encoding motion sequences into a latent space, as recently done with images. **Frechet inception distance (FID)** [Heusel et al. 2017] has been used to measure the similarity between two distributions of images by comparing their latent features from a pre-trained autoncoder intermediate layer. Inspired by this approach, FMD [Hu et al. 2024; Maiorca et al. 2022] has been used to compare motion sequences by encoding them into a latent space and calculating the distance between their respective distributions, thus providing a more generative aligned measure of motion similarity.

We use FMD to assess the quality and variability of the motion $\mathbf{g} \equiv \{g^1, \ldots, g^M\}$ generated through simulation, compared to the reference motion $\mathbf{r} \equiv \{r^1, \ldots, r^N\}$. This process is illustrated in Figure 7. A ResNet34 model, pre-trained on images (IMAGENET1K_V1), is fine-tuned on the whole Ground-Truth-set to encode motion segments in a latent space. The **Fréchet distance (FD)** is calculated between the distribution of encoded simulated motion and the one of encoded reference ground truth. Assuming the two distributions being Gaussian the FMD can be calculated analytically using means $\boldsymbol{\mu}$ and covariance matrices Σ, such that

$$\text{FMD} \equiv \|\boldsymbol{\mu}_\mathbf{g} - \boldsymbol{\mu}_\mathbf{r}\|^2 + \text{trace}\left(\Sigma_\mathbf{g} + \Sigma_\mathbf{r} - 2\sqrt{\Sigma_\mathbf{g}\Sigma_\mathbf{r}}\right).$$

More in detail, every motion clip, consisting of D observations, is transformed into a set of images through a sliding window selection with window size K. Here the RGB channels are replaced with x,y, and z channels of the 3D motion. After a small test on the patch sizes, we chose the encoder with the lowest reconstruction loss and a window of $K = 34$ poses per image.

Quantitative Comparison: When evaluating the quality of the motion simulated using policies generated with the DM, we do not compare the motion distances in absolute terms. We compare them with the ones obtained for the motions of similar RL-trained policies, leading to a fairer comparison. This allows us to evaluate the quality of novel motions for which an exact ground truth cannot be calculated, or for which no policy is trained for the same conditional label. This is illustrated in Figure 8.

5 Results

In this section, we present results demonstrating that our DM can generate policies adapting to a continuum of different character morphology, generalize to other tasks such as terrain adaptation, and explore the possibility of motion combination. We present results demonstrating that our DM can generate policies that adapt to a continuum of different character morphologies and generalize to other tasks, such as terrain adaptation and motion combination. Section 5.1 provides an evaluation of morphology adaptation, tested on five distinct motions: Dance, Cartwheel, Monkey-like, Crawling, and Military, each with a separately trained DM. Section 5.2 examines terrain adaptation in conjunction with morphology adaptation, focusing on the Military motion. For this, we train the DM on two orthogonal datasets: one encompassing all morphology types on flat terrain and the other using the default character across various terrain types. Finally, Section 5.3 explores motion combination by generating policies that blend Fast-walk with neighboring motions,

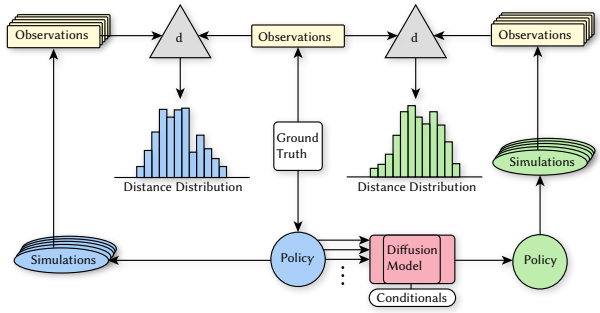

Fig. 8. Schematic of the quantitative evaluation process of the motion quality comparing RL policies and DM generated policies using the distance d. The ground truth is used to train the imitation policies using RL. We train the DM to generate new policies with the dataset of trained policies and their respective conditionals. We calculate the distance across multiple simulations for each generated policy and we compare the distance distribution with the RL-trained for the test set policy.

Table 1. List of the DMs That Have Been Trained

Name	Condition	# P	Validation
Dance Morph.	10 scalars	40	10%
Cartwheel Morph.	10 scalars	40	5 polices
Monkey Morph.	10 scalars	40	10%
Crawling Morph.	10 scalars	40	10%
Military Morph.	10 scalars	40	10%
Military Morph.+Terr.	10+2 scalars	40+12	10%
12 Motions Comb.	12 PCA comp.	72	N/A

For each model, it shows the size and type of the condition; the number of policies (# P), subset of the Policy-set, used during training; and the percentage of policies removed from the training set for validation. For the Cartwheel Morph example, we removed five policies for which the RL training was unsuccessful.

such as Walk, Happy-hops (skipping), and Military. This experiment utilizes a single DM trained on 12 motion types, as depicted in Figure 3.

All training was conducted on a computer running Ubuntu 20.04, equipped with an A6000 GPU and 48 GB of VRAM. For training control policies, we adapted the **adversarial skill embeddings** (**ASE**) [Peng et al. 2022] implementation based on NVIDIA Isaac, and for training our Diffusion-Transformer models, we used a modified implementation of GPT-2 from Erkoç et al. [2023]. A link to the code repository and the Policy-set will be provided upon acceptance.

Details of each model configuration are provided in Table 1, with training settings selected empirically to yield satisfactory performance. All results demonstrate motion synthesized by policies generated from conditions that were not encountered during training. The supplementary video contains animations of synthesized motions.

5.1 Morphology Adaptation

The morphology adaptation task demonstrates the power of our diffusion-generated policies to adapt to unseen character morphologies for the Dance motion. Each policy is trained using as a CNP the policy of the wanted motion with the default character and flat terrain. Additional examples of motions are shown in Figure 1, the appendix, and the introductory video. We compare DM-generated policies specifically for characters of the test set to the naive way of reusing RL-trained policies on the new character. Our results can be seen in Figure 10. Our diffusion approach for generating policies is better at adapting to unseen morphologies than the default RL-trained Dance policy. The default policy, when executed on a new character, results in significantly different body poses and global motion trajectory. The character eventually falls, which is generally the case when body proportions deviate excessively from the default character. Our sampled policies preserve the characteristics of the motion without further RL training.

Parametric character variations: Starting with the standard humanoid character [Peng et al. 2018] as a base, we have created morphological variations reminiscent of the exaggerated caricatures often found in entertainment media. In total, we are using 10 parameters: the height of the character between 140 and 200 centimeters, and the other nine parameters describing the morphological variations. The first five parameters determine the proportions of the character's full height, dividing it into sections for the head, torso, thighs, shins, and heel height. The remaining four parameters control the lengths of additional body segments: shoulder width, upper arms length, lower arms length, and hip width. These segments can be adjusted independently of the height proportions, offering greater customization. The change in dimension leads to a change in the physical properties of the virtual character such as mass distribution, joint torque, and collision geometry. Among the variety of possibilities this parametrization offers, we have selected 10 caricatural characters, shown in Figure 9, that represent extreme configurations and allow for interpolation of in-between parameters. A policy dataset is generated by training policies for each new morphology and for each reference motion. Specifically, policy training uses an imitation policy for the same motion from the Ground-Truth-set and trained using the default character's policy as the CNP. For ground-truth comparisons, reference motions are retargeted to the new character morphology and a morphology specific policy is learned.

A full overview of the performances is given by the box plots of the DTW, FMD, and FR in Figure 11 showing that the reliability and fidelity of our policies are comparable to RL-trained policies for the same test characters. The distribution of the distance metrics across independent simulations is narrower for our generated policies, indicating that the quality is more consistent across policy replays.

5.2 Terrain Adaptation

In addition to using a flat terrain as a playground for our characters, we introduce a heightfield with uniformly distributed depths, controlled by two parameters: a deterministic spatial scale in the x-y plane and the depth range in the z-direction. The maximum deviation in the z-direction is a linear function of the spatial scale, starting from 25 cm when the spatial scale is at its minimum (25 cm) and reaching up to 80 cm when the spatial scale is at its maximum (150 cm). The depth parameter shrinks the z-deviation distribution from 0 (flat terrain for every spatial scale) to the maximum allowed for the given spatial scale. This approach allows us to create a variety of terrains, from flat surfaces to rocky ground and tall,

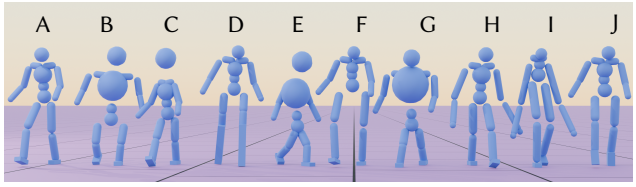

Fig. 9. The default Humanoid model (A) and the exaggerated morphological variations derived from it (B-J). For our DM on the morphology adaptation task, the training set is constituted by policies trained on these characters.

smooth hills. We train each policy for the new terrains using as CNP the policy for the default character proportions, 162 cm in height, trained on flat terrain.

Policies that are trained only on flat terrain are continuously perturbed by unexpected forces when the simulation environment presents rough ground or slopes. To address this, we train the DM on two separate datasets: one containing all morphologies on flat terrain and another featuring the default morphology across various terrains. Figure 12 illustrates examples on unseen terrains where morphology adaptation benefits from the inclusion of terrain variation, achieved by adding 12 policies for the default character on different terrains. This approach improves the FR in 63.73% of the {Terrain × Morphology} combinations compared to RL-policies trained solely on flat terrain, with an average improvement of 9.24 percentage points.

5.3 Combining Different Motion Types

We test our model on unseen motion types by generating policies through conditions in between the principal component embeddings given during training. When training policies with different motion content, we use as CNP a policy trained to reproduce a steady T-pose. This choice is assumed to make all the types of motion equally difficult to learn and stabilize the convergence time. To evaluate this task we test our method on the motion types Walk, Fast-walk, Military, and Happy-hops, which have nearby neighboring clusters in the principal component space as can be seen in Figure 3.

We generate policies by linear interpolation of the conditions while sampling the DM, and we compare them to direct linear interpolation of the policy representation. In Figure 13 we show that the directly interpolated policy representation, being the neural network weights and biases of the policy, can keep a faithful representation of the motion type only when one is very close to the original input policy representations, while the motion pattern and direction get significantly modified the further away one is from the policy representations. In comparison our generated policy better preserves the motion pattern and direction, giving a more realistic transition between the two motion types.

We can inspect the trajectory of the policies in the principal component representation during the two types of interpolation in Figure 16. Our generated policies show a curved path between the two motion types. This supports our intuition of how control policies form a large-dimension-nonlinear space and how geodesics in this space will look in a reduced linear space.

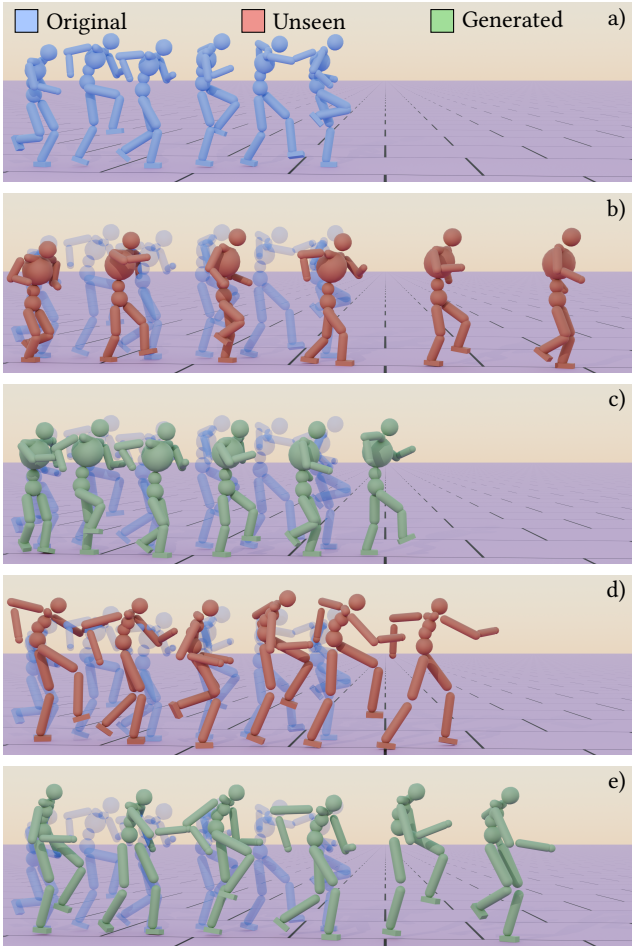

Fig. 10. The policy trained with the default 162 cm character on the Dance motion does not generalize well to different morphologies: (a) Original policy running on the default character; (b,d) Running the original policy on characters with light morphology changes. The motion loses its characteristics, such as leg rising level and dancing pattern; (c,e) Our generated policies recover the characteristics of the original motion on the specific morphology.

Combining a motion with a neighboring one often results in an asymmetrical motion that takes half the characteristics of each motion type. This phenomenon is represented in Figure 15 where Fast-walk is mixed with other styles. The effect is more noticeable in the introduction video.

6 Discussion

In this section, we analyze key findings, discuss model limitations, and propose potential improvements. We structure the discussion into distinct subsections addressing quantitative evaluation, policy synthesis challenges, failure cases, effects of the regularization, and considerations on the compatibility between combined motions.

6.1 Quantitative Evaluation of RL Policies

Numerically comparing an RL-trained policy from the validation set with its DM-generated counterpart is not trivial. Policies

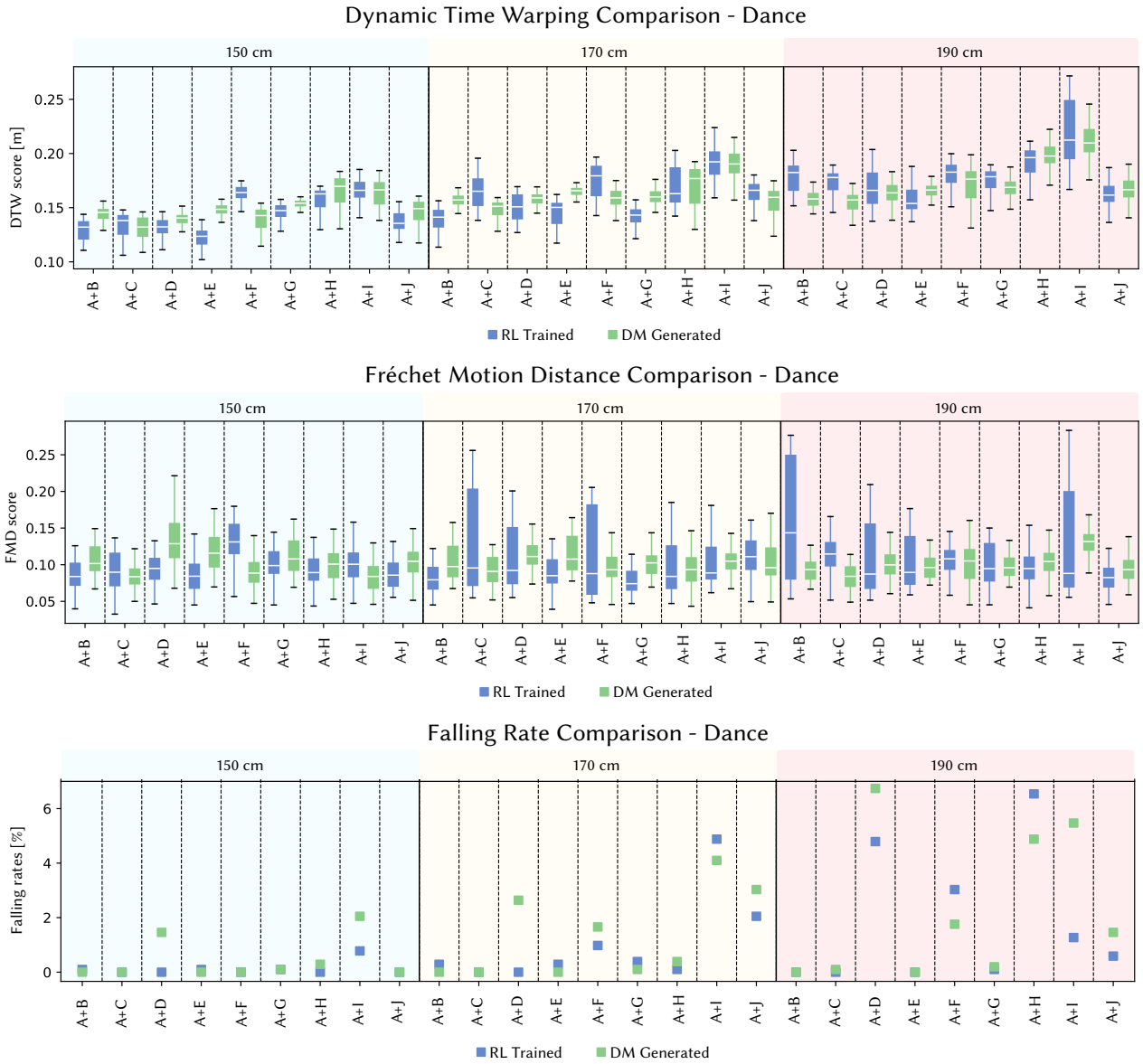

Fig. 11. Quantitative evaluation of the model on the test set. These statistics are calculated for each of the policies on 200 replay instances with different seeds. The default morphology of the humanoid character is mixed with the remaining nine morphology. The generated model achieves comparable results compared with the trained ones. The distribution of the metrics for the generated policies is narrower for the generated ones showing more stable quality across independent simulations. The FR shows comparable values, in some cases improving the reliability. Here A+X is a midpoint morphology unseen by the DM, between type A (default) and type X as in Figure 9.

corresponding to intermediate conditions are not necessarily located between the policies in the training set. In addition, multiple convergent solutions during RL training, even when regularization is applied, prevent a unique policy from emerging because several solutions may be similarly distant from the CNP. Nevertheless, we observe that the validation loss consistently decreases together with the training loss across all our experiments. This finding supports our assumption that employing CNP regularization in conjunction with a policy gradient method promotes a structured data manifold.

6.2 Challenges in Policy Synthesis

Although our experiments employed relatively small neural networks, our model synthesizes high-quality motion for almost all motion types without the need for hyperparameter tuning. However, certain motions, such as Old-walk and Painful-walk, exhibit reduced performance. As shown in Figure 3, these slow motion types form more dispersed clusters, suggesting that the policies have not converged accurately. Peng et al. [2021] introduced a gradient penalty to stabilize training and to prevent

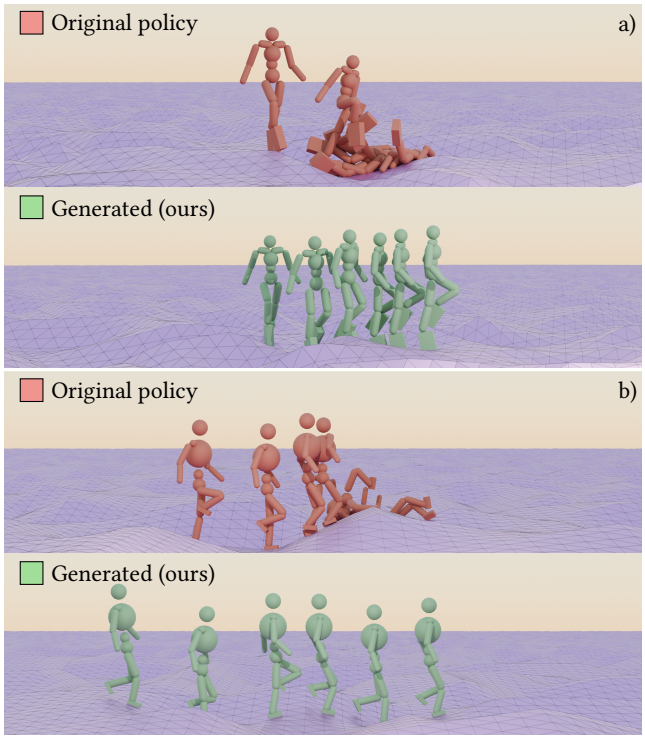

Fig. 12. Our generated policies can adapt simultaneously to morphology and different terrains, improving the FR on 64% of the cases compared with the same policy on a flat terrain. (a) On a character with taller heels the FR is recovered from 80% to 26% using our model. (b) On a character with a larger torso the FR is recovered from 100% to 20% using our model.

the generator from overshooting the data manifold. Large gradient penalties, however, appear to hinder the learning of slow motions. To mitigate this, we reduce the gradient penalty so that slower motions can be learned more effectively. Figure 17(a) illustrates the impact of poor convergence in compact model representations.

Another challenge arises from the use of AMP in training policies. Characters sometimes freeze entirely, particularly in slow-motion styles. We attribute this behavior to the Markovian nature of the algorithm and the absence of a temporal phase term. In motions such as Crawling or Monkey-like walking, policies may enter absorbing states where the optimal action is to maintain the same pose. Our generated policies occasionally exacerbate this freezing effect, suggesting that an alternative control policy model might better capture such motions. Figures 20 and 21 in the appendix highlight this issue by showing increased mean and variance in FMD scores, while DTW scores remain comparable to those of the reference policies. Other policies, as the military walk in Figure 18, do not show this freezing phenomenon.

6.3 Failure Cases and Limitations

Certain failure cases occur when the conditioning variables extend beyond the well-supported regions of the training set. For instance, as illustrated in Figure 17(b), policies fail when the thigh length is significantly shorter than the shin length. If such character types are required, the model can be improved by incorporating corresponding policies into the training set.

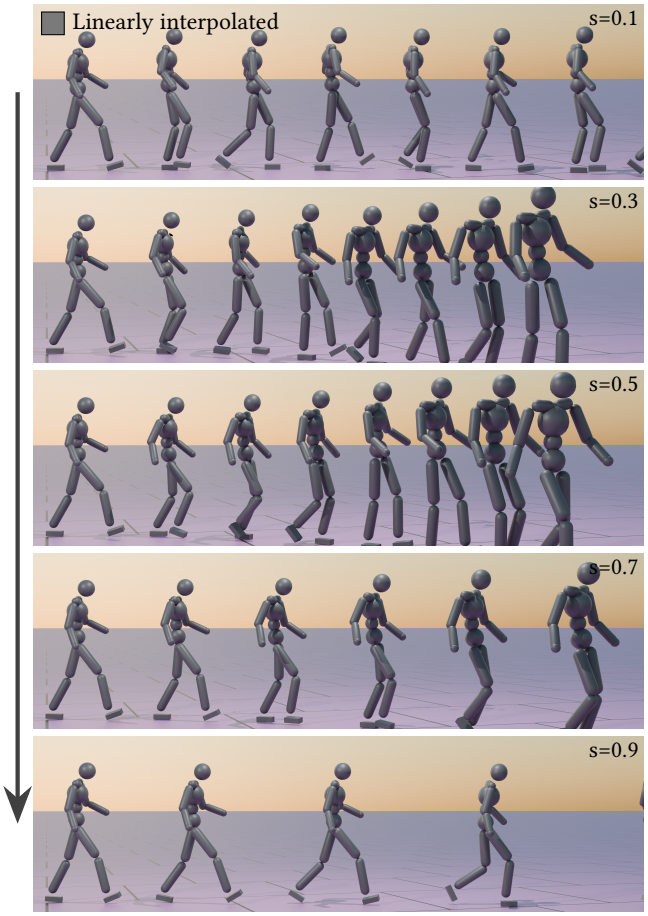

Fig. 13. Direct linear interpolation with a parameter $0 < s < 1$ of learned policies results in poor motion quality. Only when close to the end-points of the interpolation do the motions look similar to the source policies but at the mid the motions become weird looking. One can not simply use interpolation to blend different motions.

Failures also become more frequent when changes in character morphology involve increased mass, as seen in Figure 17(c). This observation suggests that the policies are less robust for larger and heavier characters under the same joint stiffness conditions. Addressing this issue requires expanding the training set to include more policies adapted to such morphologies.

Moreover, ensuring the reliability of the policy training set prior to training the DM is crucial. For example, in the Cartwheel task (Figure 19), policies trained on larger-torso characters (B, E, G) fail to recover from upside-down positions. To mitigate this, we removed the five worst-performing policies before training the DM. Consequently, test characters A+B, A+E, and A+G exhibit a 100% FR, whereas RL-trained versions succeed due to their lower mass, which makes them easier to train.

6.4 Impact of CNP Regularization

The choice of CNP is, in principle, arbitrary because any policy could serve as a regularizer. For morphology and terrain adaptation, selecting the policy corresponding to the same motion with a default

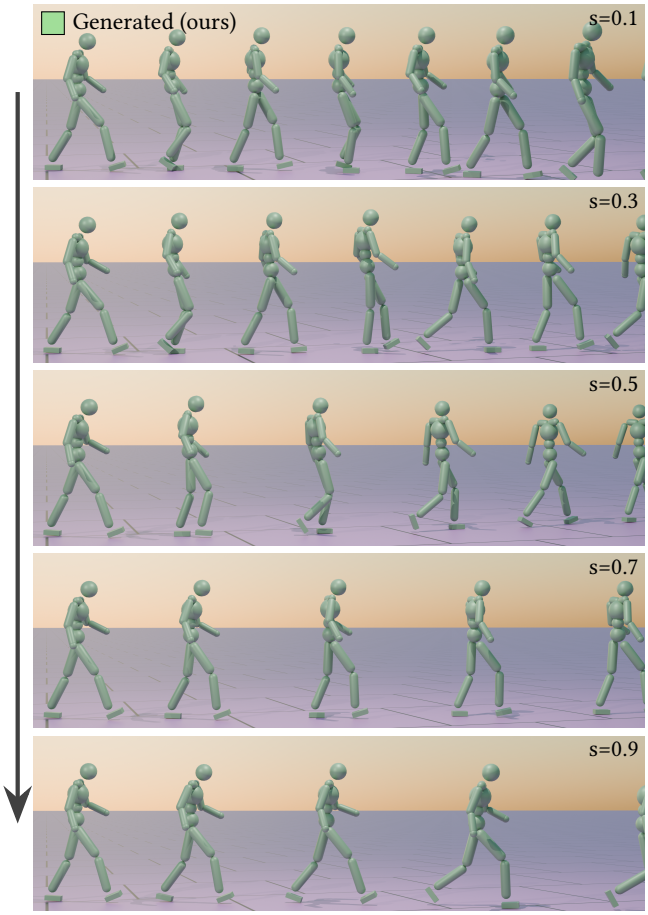

Fig. 14. We use our diffusion process trained on the 12 motion types to generate unseen motions that can be described as a gradual mixing from Slow-walk to Fast-walk motions. The top row shows the slowest walking motion and the bottom row shows the fastest walking. This demonstrates the ability to generate new motion for a character by linearly interpolating between conditions to the DM by a parameter $0 \leq s <\leq 1$.

character on flat terrain as the CNP is a natural choice since other policies can be interpreted as variations of this baseline.

For motion combination tasks, however, selecting an appropriate CNP is more complex. We opted for the T-pose (trained without regularization) because it represents a neutral, standing character and is therefore motion-agnostic. Although the "zero-policy", where all entries are zero, might appear to be a natural neutral choice because it represents inactivity in PD controllers and minimizes squared weight values, it ultimately hinders policy learning.

The selection of neighbors in CNP regularization plays a critical role in shaping the local manifold structure. For example, regularizing Fast-walk policies with Walk policies causes convergence at a different local minimum. In this scenario, even a linear interpolation in policy space yields a meaningful motion transition, with walking speed increasing progressively as expected. We attribute this behavior to the similarity between the two motions and the local linearity of gradient-based methods. Figures 22 and 23 in the appendix illustrate this effect. We speculate that carefully selecting

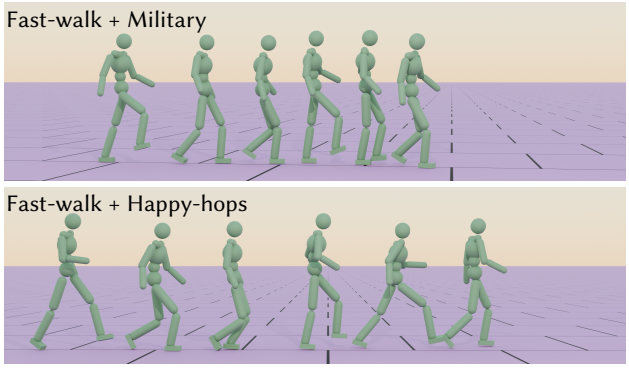

Fig. 15. Combining Fast-walk with a motion of a neighboring cluster results in an asymmetrical motion where one step is taken from Fast-walk and the next from the combined style. The introduction video offers a better representation of the phenomenon.

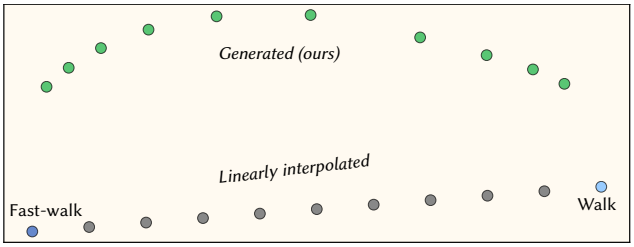

Fig. 16. Two-dimensional PCA plot of the interpolated policies. The generated policies obtained by linearly interpolating the principal component condition from Walk to Fast-walk remark the non-linearity of the transformation between the policy parameters of two similar motion types. Our generated policies follow a curved path, showing the diffusion process can model the non-linearity of the geometrical manifold of the policy space. This result is supported by Figures 13 and 14, where we show the motion quality for some of these policy points parametrized by s.

regularization policies could help shape the data manifold more effectively.

6.5 Compatibility on Motion Combination

Our observations indicate that meaningful motion combinations occur only between motions that form neighboring clusters in Figure 3. For example, combining forward and backward walking results in an incoherent policy. Future research should explore how policies encode motions and investigate model improvements to better handle diverse motion combinations.

7 Conclusion

In this work, we present a novel framework for retargeting control policies for physics-based characters to new task and character variations. Our CNP regularization technique enables the learning of a similarity-preserving policy network representation across various motion types. This approach makes the actor-network's weight space suitable for generative modeling, where different weights are assumed to represent continuous variations on a shared manifold. The compact nature of our networks suggests that motion representations can be compressed. We propose a new role for neural network-based control policies, where a compact

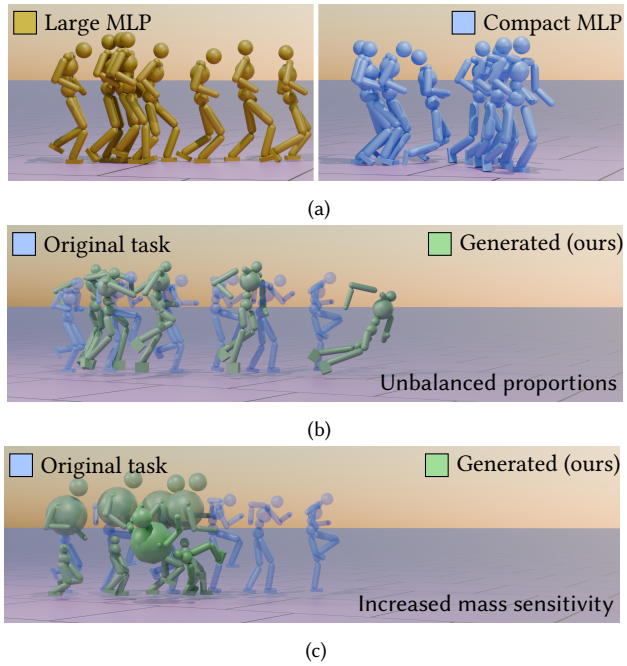

Fig. 17. Example failure cases of our approach. (a) For some specific motion types, such as Painful-walk, the compact policy performs worse than a bigger network. (b) A generated policy fails if the morphology has unbalanced proportions such as thighs way shorter than shins. (c) Generated policies become more sensitive with taller and increased mass characters. The likelihood of these failures can in our experience change depending on the choice of motion type and character morphology.

representation encodes the motion, and new variations of the motion are generated by manipulating the network's weights.

We further demonstrate that a Diffusion Transformer model can effectively manage diverse motion types and produce new policies adaptable to novel character morphologies, environmental variations, and motion combinations. In the case of the latter, a key aspect of our approach is the continuous conditioning of categorical labels, achieved through the principal component parametrization of motion types. Our results show that non-linear transformations within the policy space can be learned from labeled examples, enabling the conditional generation of new policies.

7.1 Future Work

Our findings open up several promising research directions. These include exploring conditioning on finer motion characteristics, conducting topological analyses of the newly represented motion space, and integrating physical simulation more deeply into the DM training process.

The DM attempts to learn the manifold underlying the policy weights. However, it does not have any feedback on the final quality of the motion. This opens up a future path of research to study conditional policies based on the motion itself with the need of validating the policies during training. It is not a trivial task to deal with the validation of the motion quality during the training of the DM. The input to a policy is a state observation, while the output is the joints' action. The easiest way to perform validation would be to include simulation steps during the diffusion training. This is a computationally too demanding task in our current setting, yet it may be conceptually feasible after some resource optimization. We leave this to be explored in future work.

Recent advancements in DMs and transformer architectures have showcased their ability to generate data conditioned on multi-modal parameters, facilitating complex and context-aware synthesis. We believe that achieving an efficient and interpretable encoding of control policies is essential for fully leveraging these generative models in physics-based simulation and animation.

Future research could also focus on investigating the compression limits of policy motion representation and planning conditional transitions between motions for thin-clients in the video game industry. Enhancing the concept of CNP regularization could lead to a more efficient and smooth encoding of policies on differentiable manifolds. Additionally, exploring the formalization of a policy space may improve the stability of policies in out-of-distribution states and contribute to the development of more robust controllers.

To support and accelerate research in this domain, we are releasing our comprehensive dataset of control policies, which includes hundreds of regularized, labeled examples. This resource is designed to enable further experimentation and validation of new generative approaches, fostering innovation in policy synthesis and adaptive motion generation. We anticipate that access to such data will facilitate the development of methods that extend the generative capabilities of current models, leading to more robust and versatile solutions in physics-based character animation.

References

Michael Abdul-Massih, Innfarn Yoo, and Bedrich Benes. 2017. Motion style retargeting to characters with different morphologies. *Computer Graphics Forum* 36, 6 (2017), 86–99. DOI: https://doi.org/10.1111/cgf.12860

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* 39, 4, Article 62 (Aug 2020), 14 pages. DOI: https://doi.org/10.1145/3386569.3392462

Kfir Aberman, Rundi Wu, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. 2019. Learning character-agnostic motion for motion retargeting in 2D. *ACM Transactions on Graphics* 38, 4, Article 75 (Jul 2019), 14 pages. DOI: https://doi.org/10.1145/3306346.3322999

Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. 2023. Listen, denoise, action! Audio-driven motion synthesis with diffusion models. *ACM Transactions on Graphics* 42, 4, Article 44 (Jul 2023), 20 pages. DOI: https://doi.org/10.1145/3592458

Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. 2019. DReCon: Data-driven responsive control of physics-based characters. *ACM Transactions on Graphics* 38, 6, Article 206 (Nov 2019), 11 pages. DOI: https://doi.org/10.1145/3355089.3356536

Nico Bohlinger, Grzegorz Czechmanowski, Maciej Piotr Krupka, Piotr Kicki, Krzysztof Walas, Jan Peters, and Davide Tateo. 2024. One policy to run them all: An end-to-end learning approach to multi-embodiment locomotion. In *Proceedings of the 8th Annual Conference on Robot Learning*. CoRL 2024. DOI: https://doi.org/10.48550/ARXIV.2409.06366

Jason Chemin and Jehee Lee. 2018. A physics-based juggling simulation using reinforcement learning. In *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '18)*. ACM, New York, NY, USA, Article 3, 7 pages. DOI: https://doi.org/10.1145/3274247.3274516

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*. 02783649241273668. DOI: https://doi.org/10.1177/02783649241273668

Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali Thabet, and Artsiom Sanakoyeu. 2023. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 481–490. DOI: https://doi.org/10.1109/CVPR52729.2023.00054

Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. 2023. HyperDiffusion: Generating implicit neural fields with weight-space diffusion. In *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 14254–14264. DOI: https://doi.org/10.1109/ICCV51070.2023.01315

Andrew Feng, Yazhou Huang, Yuyu Xu, and Ari Shapiro. 2012. Automating the transfer of a generic set of behaviors onto a virtual character. In *Motion in Games*. Marcelo Kallmann and Kostas Bekris (Eds.), Springer Berlin Heidelberg, Berlin, 134–145.

Michael Gleicher. 1998. Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. ACM, New York, NY, USA, 33–42. DOI: https://doi.org/10.1145/280814.280820

Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust motion in-betweening. *ACM Transactions on Graphics* 39, 4, Article 60 (Aug 2020), 12 pages. DOI: https://doi.org/10.1145/3386569.3392480

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6629–6640.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*. H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33, Curran Associates, Inc., Vancouver, Canada., 6840–6851. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

Lei Hu, Zihao Zhang, Yongjing Ye, Yiwen Xu, and Shihong Xia. 2024. Diffusion-based human motion style transfer with semantic guidance. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '24)*. Eurographics Association, Goslar, DEU, 1–12. DOI: https://doi.org/10.1111/cgf.15169

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2002a. Motion graphs. *ACM Transactions on Graphics* 21, 3 (July 2002), 473–482. DOI: https://doi.org/10.1145/566654.566605

Lucas Kovar, John Schreiner, and Michael Gleicher. 2002b. Footskate cleanup for motion capture editing. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '02)*. ACM, New York, NY, USA, 97–104. DOI: https://doi.org/10.1145/545261.545277

Ariel Kwiatkowski, Eduardo Alvarado, Vicky Kalogeiton, C. Karen Liu, Julien Pettré, Michiel van de Panne, and Marie-Paule Cani. 2022. A survey on reinforcement learning methods in character animation. *Computer Graphics Forum* 41, 2 (2022), 613–639. DOI: https://doi.org/10.1111/cgf.14504

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 (SA '23)*. ACM, New York, NY, USA, Article 45, 11 pages. DOI: https://doi.org/10.1145/3610548.3618206

Tianyu Li, Jungdam Won, Alexander Clegg, Jeonghwan Kim, Akshara Rai, and Sehoon Ha. 2023. ACE: Adversarial correspondence embedding for cross morphology motion retargeting from human to nonhuman characters. In *SIGGRAPH Asia 2023 Conference Papers (SA '23)*. ACM, New York, NY, USA, Article 46, 11 pages. DOI: https://doi.org/10.1145/3610548.3618255

Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. 2024. InterGen: Diffusion-based multi-human motion generation under complex interactions. *International Journal of Computer Vision* 132, 9 (September 2024), 3463–3483. DOI: https://doi.org/10.1007/s11263-024-02042-6

Yunhao Luo, Kaixiang Xie, Sheldon Andrews, and Paul Kry. 2021. Catching and throwing control of a physically simulated hand. In *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '21)*. ACM, New York, NY, USA, Article 15, 7 pages. DOI: https://doi.org/10.1145/3487983.3488300

Antoine Maiorca, Youngwoo Yoon, and Thierry Dutoit. 2022. Evaluating the quality of a synthesized motion with the fréchet motion distance. In *ACM SIGGRAPH 2022 Posters (SIGGRAPH '22)*. ACM, New York, NY, USA, Article 9, 2 pages. DOI: https://doi.org/10.1145/3532719.3543228

Lucas Mourot, Ludovic Hoyet, François Le Clerc, and Pierre Hellier. 2022a. UnderPressure: Deep learning for foot contact detection, ground reaction force estimation and footskate cleanup. *Computer Graphics Forum* 41, 8 (Dec. 2022), 195–206. DOI: https://doi.org/10.1111/cgf.14635

Lucas Mourot, Ludovic Hoyet, François Le Clerc, François Schnitzler, and Pierre Hellier. 2022b. A survey on deep learning for skeleton-based human animation. *Computer Graphics Forum* 41, 1 (2022), 122–157. DOI: https://doi.org/10.1111/cgf.14426

Kourosh Naderi, Amin Babadi, Shaghayegh Roohi, and Perttu Hämäläinen. 2019. A reinforcement learning approach to synthesizing climbing movements. In *Proceedings of the 2019 IEEE Conference on Games (CoG)*. ACM, London, UK, 1–7. DOI: https://doi.org/10.1109/CIG.2019.8848127

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. DeepMimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics* 37, 4 (July 2018), 1–14. DOI: https://doi.org/10.1145/3197517.3201311

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. 2022. ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions on Graphics* 41, 4, Article 94 (Jul 2022), 17 pages. DOI: https://doi.org/10.1145/3528223.3530110

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics* 40, 4, Article 144 (Jul 2021), 20 pages. DOI: https://doi.org/10.1145/3450626.3459670

Qiaosong Qi, Le Zhuo, Aixi Zhang, Yue Liao, Fei Fang, Si Liu, and Shuicheng Yan. 2023. DiffDance: Cascaded human motion diffusion model for dance generation. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23)*. ACM, New York, NY, USA, 1374–1382. DOI: https://doi.org/10.1145/3581783.3612307

Sigal Raab, Inbal Leibovitch, Guy Tevet, Moab Arar, Amit Haim Bermano, and Daniel Cohen-Or. 2024. Single motion diffusion. In *Proceedings of the 12th International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=DrhZneqz4n

Daniele Reda, Jungdam Won, Yuting Ye, Michiel van de Panne, and Alexander Winkler. 2023. Physics-based motion retargeting from sparse inputs. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6, 3, Article 33 (Aug 2023), 19 pages. DOI: https://doi.org/10.1145/3606928

Jiawei Ren, Cunjun Yu, Siwei Chen, Xiao Ma, Liang Pan, and Ziwei Liu. 2023. DiffMimic: Efficient motion mimicking with differentiable physics. In *Proceedings of the 11th International Conference on Learning Representations*. OpenReview.nt, Kigali Rwanda, 19. Retrieved from https://openreview.net/forum?id=06mk-epSwZ

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR* abs/1707.06347 (2017). Retrieved from http://arxiv.org/abs/1707.06347

Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. 2024. Human motion diffusion as a generative prior. In *Proceedings of the 12th International Conference on Learning Representations*. OpenReview.net, Vienna Austria, 17. Retrieved from https://openreview.net/forum?id=dTpbEdN9kr

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations*. OpenReview.net, Vienna, Austria, 20. Retrieved from https://openreview.net/forum?id=St1giarCHLP

Seyoon Tak and Hyeong-Seok Ko. 2005. A physically-based motion retargeting filter. *ACM Transactions on Graphics* 24, 1 (Jan 2005), 98–117. DOI: https://doi.org/10.1145/1037957.1037963

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. Retrieved from https://arxiv.org/abs/2410.03441

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-Or, and Amit Haim Bermano. 2023. Human motion diffusion model. In *Proceedings of the 11th International Conference on Learning Representations*. OpenReview.net, Kigali Rwanda, 16. Retrieved from https://openreview.net/forum?id=SJ1kSyO2jwu

Takara Everest Truong, Michael Piseno, Zhaoming Xie, and Karen Liu. 2024. PDP: Physics-based character animation via diffusion policy. In *SIGGRAPH Asia 2024 Conference Papers (SA'24)*, Association for Computing Machinery, Tokyo, Japan. DOI: https://doi.org/10.1145/3680528.3687683

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2021. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics* 40, 4, Article 146 (Jul 2021), 11 pages. DOI: https://doi.org/10.1145/3450626.3459761

Jungdam Won and Jehee Lee. 2019. Learning body shape variation in physics-based characters. *ACM Transactions on Graphics* 38, 6, Article 207 (Nov. 2019), 12 pages. DOI: https://doi.org/10.1145/3355089.3356499

Zhaoming Xie, Hung Yu Ling, Nam Hee Kim, and Michiel van de Panne. 2020. ALLSTEPS: Curriculum-driven learning of stepping stone skills. *Computer Graphics Forum* 39, 8 (2020), 213–224. DOI: https://doi.org/10.1111/cgf.14115

Pei Xu and Ioannis Karamouzas. 2021. A GAN-like approach for physics-based imitation learning and interactive character control. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 4, 3, Article 44 (Sep 2021), 22 pages. DOI: https://doi.org/10.1145/3480148

Pei Xu, Kaixiang Xie, Sheldon Andrews, Paul G. Kry, Michael Neff, Morgan Mcguire, Ioannis Karamouzas, and Victor Zordan. 2023. AdaptNet: Policy adaptation for physics-based character control. *ACM Transactions on Graphics* 42, 6, Article 177 (Dec 2023), 17 pages. DOI: https://doi.org/10.1145/3618375

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 2024. 3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations. In *Proceedings of Robotics: Science and Systems (RSS)*.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2024. MotionDiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 46, 06 (Jun 2024), 4115–4128. DOI: https://doi.org/10.1109/TPAMI.2024.3355414

Appendix

In this appendix, we show further results for a more complete overview of the characteristics of our method.

Morphology Adaptation

Fig. 18. Quantitative evaluation for the Military motion. Here A+X is a midpoint morphology between type A (default) and type X as in Figure 9.

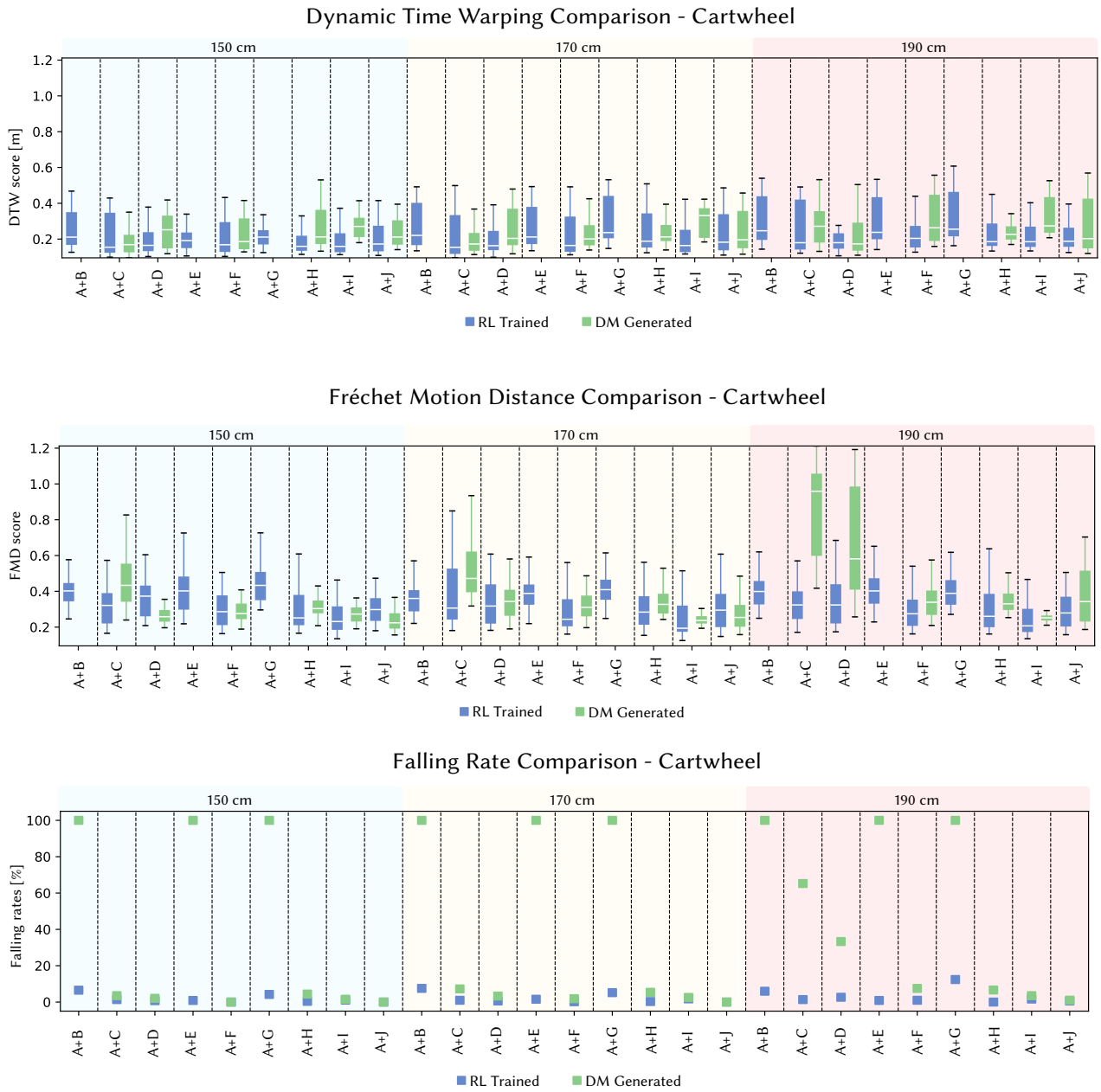

Fig. 19. Quantitative evaluation for the Cartwheel motion. Here A+X is a midpoint morphology between type A (default) and type X as in Figure 9.

Fig. 20. Quantitative evaluation for the Monkey motion. This type of motion is subject to the freezing phenomenon due to the Markov Chain nature of the imitation framework. The policies can favor absorbing states rather than moving actions for slow motions like this one. This explains the high FMD scores for some of the morphologies. Here A+X is a midpoint morphology between type A (default) and type X as in Figure 9.

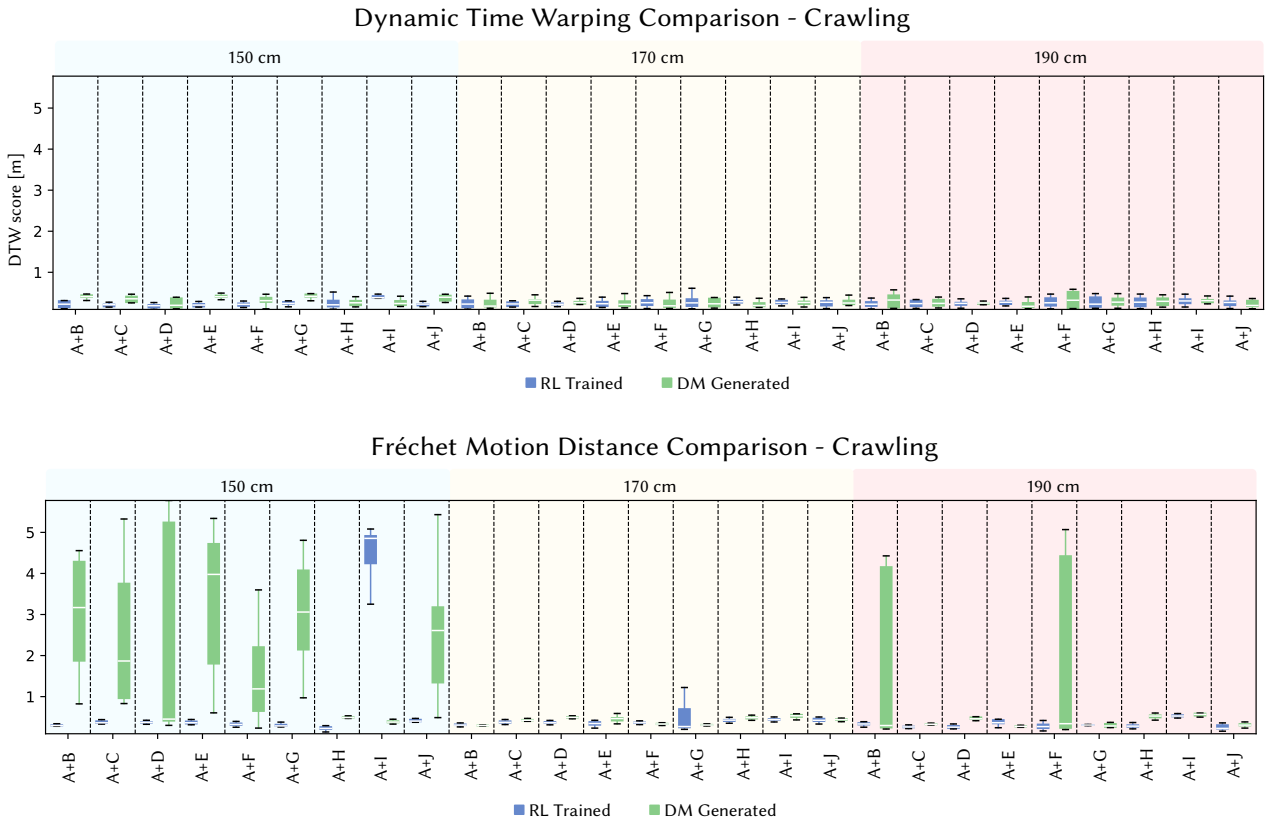

Fig. 21. Quantitative evaluation for the Crawling motion. This type of motion is subject to the freezing phenomenon due to the Markov Chain nature of the imitation framework. The policies can favor absorbing states rather than moving actions for slow motions like this one. This explains the high FMD scores for some of the morphologies. The FR is omitted because the character lies on the floor. Here A+X is a midpoint morphology between type A (default) and type X as in Figure 9.

Motion Combination

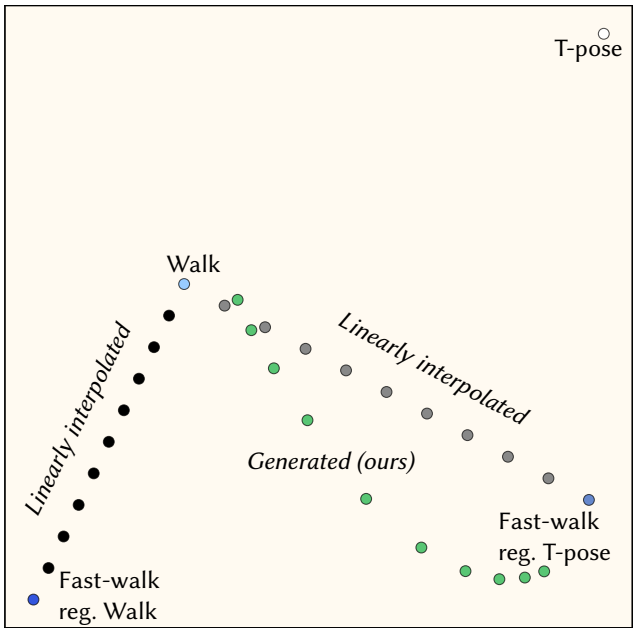

Fig. 22. Two-dimensional PCA plot of the interpolated policies. When a Fast-walk is fine-tuned by regularizing from a Walk instead of the T pose, it converges to another policy. In this specific case, Walk and Fast-walk have very similar policies, and linear interpolation in the policy space is sufficient for good motion interpolation. This is better illustrated in Figure 23.

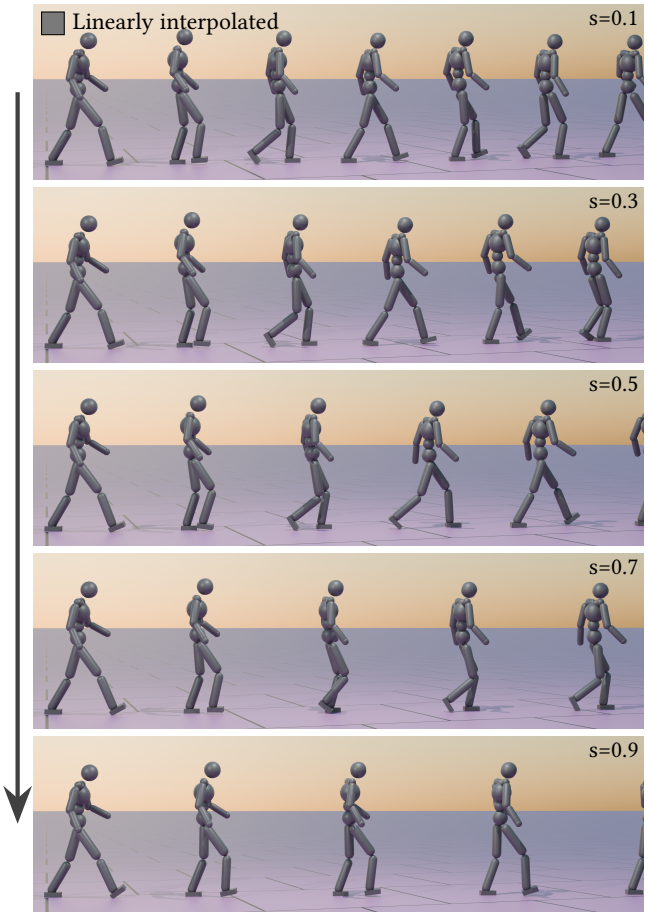

Fig. 23. When a Fast-walk is fine-tuned using as regularization a slower Walk, the policies are so close to each other that linear interpolation on the policy space is sufficient to produce the expected motion.

Received 11 November 2024; revised 13 February 2025; accepted 25 March 2025