

SMP: Reusable Score-Matching Motion Priors for Physics-Based Character Control

YUXUAN MU*, Simon Fraser University, Canada
 ZIYU ZHANG*, Simon Fraser University, Canada
 YI SHI*, Simon Fraser University, Canada
 MINAMI MATSUMOTO, Sony Playstation, Japan
 KOTARO IMAMURA, Sony Playstation, Japan
 GUY TEVET, Stanford University, USA
 CHUAN GUO, Meta, USA
 MICHAEL TAYLOR, Sony Playstation, USA
 CHANG SHU, National Research Council Canada, Canada
 PENGCHENG XI, National Research Council Canada, Canada
 XUE BIN PENG, Simon Fraser University, Canada and NVIDIA, Canada



Fig. 1. Our framework constructs reusable and modular motion priors. A general motion prior can be trained on a large dataset spanning 100 styles, and then be repurposed into 100 style-specific priors without requiring further data access or model updates. These style priors serve as stationary reward models and can be reused to train control policies for diverse tasks with stylistic yet natural behaviors.

Data-driven motion priors that can guide agents toward producing naturalistic behaviors play a pivotal role in creating life-like virtual characters. Adversarial imitation learning has been a highly effective method for learning motion priors from reference motion data. However, adversarial priors, with few exceptions, need to be retrained for each new controller, thereby limiting their reusability and necessitating the retention of the reference motion data when training on downstream tasks. In this work, we present Score-Matching Motion Priors (SMP), which leverages pre-trained motion diffusion models and score distillation sampling (SDS) to create reusable task-agnostic motion priors. SMPs can be pre-trained on a motion dataset, independent of any control policy or task. Once trained, SMPs can be kept frozen and reused as general-purpose reward functions to train policies to

produce naturalistic behaviors for downstream tasks. We show that a general motion prior trained on large-scale datasets can be repurposed into a variety of style-specific priors. Furthermore SMP can compose different styles to synthesize new styles not present in the original dataset. Our method produces high-quality motion comparable to state-of-the-art adversarial imitation learning methods through reusable and modular motion priors. We demonstrate the effectiveness of SMP across a diverse suite of control tasks with physically simulated humanoid characters.

Additional Key Words and Phrases: character animation, score distillation sampling, diffusion model, reinforcement learning

1 Introduction

Creating virtual characters that move with natural and life-like behaviors is fundamental to immersive digital experiences in animation, films, games, and virtual reality (VR) applications. While motion capture and procedural animation techniques can produce high-quality movements, the challenge lies in developing controller that enable physically simulated characters to exhibit similarly natural behaviors dynamically, in response to diverse tasks and environmental contexts. Traditional physics-based controllers trained without reference motion data often result in visually unnatural

*Joint First Authors.

Authors' Contact Information: Yuxuan Mu, yma101@sfu.ca, Simon Fraser University, Canada; Ziyu Zhang, zza333@sfu.ca, Simon Fraser University, Canada; Yi Shi, ysa273@sfu.ca, Simon Fraser University, Canada; Minami Matsumoto, Minami.X.Matsumoto@sony.com, Sony Playstation, Japan; Kotaro Imamura, Kotaro.Imamura@sony.com, Sony Playstation, Japan; Guy Tevet, guy.tvt@gmail.com, Stanford University, USA; Chuan Guo, guochuan5513@gmail.com, Meta, USA; Michael Taylor, Mike.Taylor@sony.com, Sony Playstation, USA; Chang Shu, Chang.Shu@nrc-cnrc.gc.ca, National Research Council Canada, Canada; Pengcheng Xi, Pengcheng.Xi@nrc-cnrc.gc.ca, National Research Council Canada, Canada; Xue Bin Peng, xbpeng@sfu.ca, Simon Fraser University, Canada and NVIDIA, Canada.

movements, lacking the subtle qualities of life-like motion [Coros et al. 2010; Hodgins et al. 1995; Yin et al. 2007]. Tracking-based methods improve realism by following reference motion clips, but they typically require the controller to rigidly mimic target motions frame-by-frame [Liu and Hodgins 2017; Peng et al. 2018; Won et al. 2017], limiting its flexibility to deviate from the reference and adapt behaviors to perform new tasks. Alternatively, distribution-matching methods, such as adversarial imitation learning [Ho and Ermon 2016; Peng et al. 2021], provide a more versatile approach for imitating motion data. Adversarial methods can learn flexible motion priors from motion dataset, which can act as task-agnostic measures of motion naturalness, allowing policies to produce behaviors that resemble natural motions across different tasks. However, adversarial methods typically require a prior (i.e. discriminator) to be trained jointly with a given policy, which limits the reusability and modularity of the learned priors. For each new policy, the prior must be continuously trained with data from the policy and data from the original reference dataset. As a result, the original dataset must be retained for perpetuity.

We contend that an ideal motion prior should be *modular* and *reusable*:

- *Modular*: The prior should function as an independent motion-quality objective that can guide policy training without requiring access to the original reference dataset.
- *Reusable*: Once constructed, the same prior should be applicable across diverse tasks and multiple policies without retraining.

To achieve these criteria, we propose Score-Matching Motion Priors (SMP): a method for constructing reusable, modular motion priors that can be utilized to train control policies to produce naturalistic behaviors across diverse tasks. Given an unstructured motion dataset, we first train a motion diffusion model to capture the underlying data distribution independent of any task or control policy. Once trained, the diffusion model is kept frozen and repurposed as a prior via score distillation sampling (SDS) [Poole et al. 2022], providing a robust similarity measure between simulated motions and the behaviors in the reference dataset. By reusing the learned SMP as a general style reward, our system enables a prior to be used to train new policies to perform a diverse suite of tasks with natural life-like behaviors.

The central contribution of this work is a method for constructing *modular* and *reusable* motion priors for physics-based character animation by combining reinforcement learning with score distillation. Our framework produces high-quality motions comparable to state-of-the-art adversarial imitation learning methods, while substantially improving modularity and reusability. SMP enables our system to completely discard the reference dataset during policy training. We show that a score-matching motion prior can be constructed from a task-agnostic diffusion model pretrained on large-scale motion datasets. This general-purpose motion prior can then be repurposed into style-specific priors through prompting and guidance. Furthermore, we show that priors for different styles can be composed to create new behavioral styles that are not present in the original dataset.

2 Related Work

Developing embodied agents that can act and react with life-like motions is essential for creating immersive experiences in games and virtual reality applications. This capability is also of vital importance in robotics, where naturalistic behaviors can improve safety, energy efficiency [Escontrela et al. 2022], and the learning of broadly applicable skills from human data [Grauman et al. 2022]. Given the expressive and nuanced nature of human motion, data-driven approaches have emerged as an effective paradigm for producing realistic, life-like behaviors by learning from reference motion data. *Kinematics-based methods* typically train models, such as autoregressive models [Holden et al. 2017, 2016; Zhang et al. 2018], variational autoencoders (VAEs) [Ling et al. 2020; Rempe et al. 2021; Starke et al. 2024], or diffusion models [Shi et al. 2024; Tevet et al. 2023; Zhang et al. 2022], on large motion datasets to synthesize plausible character animations. However, most of these methods do not explicitly enforce physical constraints, often leading to physically implausible behaviors, especially for new scenarios and tasks.

Physics-Based Methods. In contrast, *physics-based methods* aim to create control policies that generate motion within simulated environments, governed by dynamic equations and realistic physical laws [Raibert and Hodgins 1991; Wampler et al. 2014]. These controllers are often constructed via trajectory optimization techniques or reinforcement learning (RL) [Peng et al. 2017; Wang et al. 2009]. Previous data-free approaches based on heuristics, such as SIMBICON [Yin et al. 2007], can achieve mechanically functional behaviors, but often produce unnatural motions. Manually designing heuristics that capture the expressiveness of real human movement remains a significant challenge [Coros et al. 2010; Faloutsos et al. 2001; Hodgins et al. 1995; Witkin and Kass 1988]. Instead of relying on hand-crafted heuristics, data-driven approaches based on model predictive control (MPC) or trajectory optimization have shown promise in emulating naturalistic human motions [Lee et al. 2010; Liu et al. 2010; Muico et al. 2009; Sharon and van de Panne 2005]. More recently, reinforcement learning-based motion tracking methods have enabled controllers to effectively imitate a wide range of reference motion clips, resulting in more life-like and agile behaviors [Liu and Hodgins 2017; Peng et al. 2018; Won et al. 2017]. However, tracking-based methods limit a controller to closely following a given motion clip, which can impede the controller’s ability to generalize to new tasks, particularly when task objectives are not closely aligned with the reference motions. To address this limitation, many systems incorporate task-specific motion planners [Bergamin et al. 2019; Liu et al. 2012; Park et al. 2019], or high-level policies [Yao et al. 2022, 2024; Zhu et al. 2023], which select appropriate reference motions or latent-space skills for the controller to perform in order to complete a given task.

Distribution matching offers an alternative to motion tracking by training controllers to imitate the broader behavioral distribution of a motion dataset rather than tracking reference motions frame-by-frame. Generative Adversarial Imitation Learning (GAIL) approximates this objective using a learned discriminator to distinguish agent’s motions from dataset motions [Ho and Ermon 2016]. Adversarial Motion Priors (AMP) extend this idea to model flexible style objectives, enabling controllers to produce life-like behaviors

for novel tasks that are not observed in the original dataset [Peng et al. 2021]. However, adversarial priors must be trained jointly with a specific policy for each new task, often requiring retraining and persistent access to the dataset. In contrast, our method achieves comparable performance to state-of-the-art adversarial imitation learning approaches by introducing a reusable and modular score-matching motion prior, which can even be shaped into stylistic priors beyond those contained in the original dataset.

Diffusion Models for Control. Diffusion models’ state-of-the-art performance across a wide range of domain has spurred growing interest in leveraging diffusion models for control. Given their ability to generate high-fidelity motion, a straightforward application is to use task-oriented diffusion models as motion planners [Ren et al. 2023; Serifi et al. 2024; Tevet et al. 2024; Xu et al. 2025]. These methods leverage auto-regressive motion diffusion models to predict target future trajectories, which are then executed by a low-level tracking controller. In addition to their use as motion planners, diffuse models have also been used to directly model controllers, enabling controllers to produce flexible multi-modal behaviors for tasks such as manipulation [Chi et al. 2023; Janner et al. 2022], and character control [Huang et al. 2024a, 2025; Truong et al. 2024; Wu et al. 2025b]. Unlike these prior methods that focus on incorporating diffusion models as components of a controller, our work aims to repurpose *pretrained* diffusion models as task-agnostic behavioral priors within an optimization objective. Our method utilizes diffusion models as a reward function that can be integrated into general reinforcement learning frameworks to guide policy learning via score distillation sampling.

Score Distillation Sampling. Pretrained diffusion models are not only capable of generating samples via a denoising process, but can also serve as optimization objectives through score distillation sampling (SDS) [Poole et al. 2022]. SDS has enabled pretrained image and video diffusion models to be adapted for a wide range of downstream generative modeling tasks [Jiang et al. 2024; Wang et al. 2023; Yin et al. 2024]. Inspired by the success of SDS in the vision domain, recent works have also explored incorporating diffusion-based priors into reinforcement learning frameworks for control. A series of methods replace the discriminator in a GAIL-style setup with a diffusion model [Huang et al. 2024b; Lai et al. 2024; Pang et al. 2025; Wang et al. 2024], leveraging its expressive capabilities to differentiate between real and fake trajectories. However, these methods still rely on adversarial training and typically require continual updates to the diffusion model during policy optimization. Beyond adversarial frameworks, other systems have attempted to apply SDS-like techniques directly to policy learning. For example, Luo et al. [2024] guide policy training using pretrained image or video diffusion models conditioned on text prompts. While these methods can produce behaviors that align with textual instructions, the resulting motions often appear unnatural. The work that is most reminiscent to ours is SMILING [Wu et al. 2025a], which train controllers using a score-matching objective similar to variational score distillation (VSD) [Wang et al. 2023]. However, unlike SMILING, which requires training task-specific diffusion models, our method trains a task-agnostic motion prior from unstructured motion data. This prior can then be reused to train diverse control policies by

combining it with separate task objectives in a goal-conditioned reinforcement learning framework. We further introduce several key design decisions that enable high-quality naturalistic motions across a diverse repertoire of tasks.

3 Background

3.1 Reinforcement Learning

Our physics-based controllers are trained through a reinforcement learning (RL) framework, where an agent interacts with an environment according to a policy π in order to optimize a given objective $J(\pi)$ [Sutton et al. 1998]. At each time step t , the agent observes the current state s_t , then samples and executes an action according to the policy $a_t \sim \pi(a_t | s_t)$. The environment then transitions to a new state according to the dynamics $s_{t+1} \sim p(s_{t+1} | s_t, a_t)$, and the agent receives a scalar reward $r_t = r(s_t, a_t, s_{t+1})$. The objective is to learn a policy that maximizes the expected discounted return:

$$J(\pi) = \mathbb{E}_{\tau \sim p(\tau | \pi)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right], \quad (1)$$

where $\tau = \{s_0, a_0, r_0, s_1, \dots, s_{T-1}, a_{T-1}, r_{T-1}, s_T\}$ is a trajectory produced by the policy π , and $\gamma \in [0, 1]$ is a discount factor. The reward function serves as a flexible interface for specifying task objectives. However, crafting effective rewards that consistently induce naturalistic behaviors can be challenging and often requires extensive manual tuning.

3.2 Diffusion Models and Score Distillation Sampling

In score-based generative modeling methods [Song and Ermon 2019; Song et al. 2021], a neural network $f : \mathbb{R}^D \rightarrow \mathbb{R}^D$ is trained to estimate the *score* of a point x under a given distribution $p(x)$, which is defined as the gradient of the log-density $\nabla_x \log p(x)$. However, the predicted scores may be unreliable, as the available training data typically do not cover the entire space \mathbb{R}^D . In regions with low data density, score estimates tend to be inaccurate. To address this issue, Vincent [2011] and Song et al. [2021] propose adding noise to the data. When the noise level is sufficiently large, the perturbed data distribution covers the entire space, enabling the training of more robust and scalable score estimators. This principle forms the foundation of our approach. Score-based generative modeling with multi-level noise perturbation allows the construction of reusable motion priors from relatively limited data that lies in a low-dimensional manifold.

Diffusion Models. Diffusion models approximate a data distribution by progressively corrupting and subsequently denoising samples through a sequence of transformations [Ho et al. 2020]. Given a clean sample from the data distribution, the forward process iteratively adds Gaussian noise over N steps to form a Markov chain $\{x^i\}_{i=0}^N$, defined as:

$$q(x^i | x^{i-1}) = \mathcal{N}(x^i; \sqrt{1 - \beta_i} x^{i-1}, \beta_i I), \quad (2)$$

where i indicates the noise level, $\{\beta_i\}_{i=1}^N$ is a predefined noise schedule. Since additive *i.i.d.* Gaussian noise is used in the diffusion process, x^i can be conveniently sampled from x^0 directly via:

$$x^i = \sqrt{\alpha_i} x^0 + \sqrt{1 - \alpha_i} \epsilon, \quad (3)$$

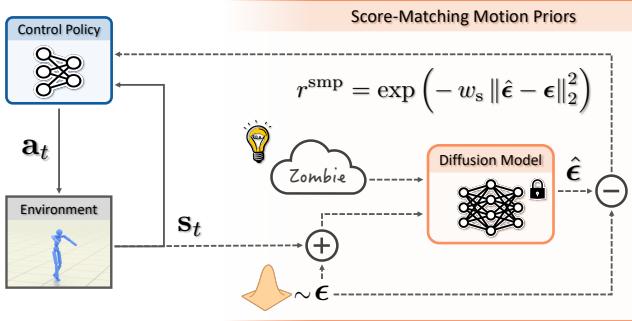


Fig. 2. Schematic overview of the system. The dashed arrows indicate components used only during policy training. A pretrained motion diffusion model serves as a reusable reward model for motion naturalness via score distillation sampling. The model can be style-conditioned, enabling the policy to learn specific skills or styles without retraining or continuous access to original motion data.

without performing the full iterative diffusion process. Here, $\bar{\alpha}_i = \prod_{j=1}^i (1 - \beta_j)$ and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. This forward diffusion process resembles the multi-level noise perturbation used in score-based generative models [Song et al. 2021]. To sample from the learned distribution $p(\mathbf{x})$, a denoising network f is typically trained using the *simple* DDPM objective [Ho et al. 2020]:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{i, \mathbf{x}^0, \epsilon} \left[\|\epsilon - f(\mathbf{x}^i)\|_2^2 \right], \quad (4)$$

where \mathbf{x}^i is obtained using Equation (3).

Score Distillation Sampling (SDS). Once a diffusion model f is trained, samples can be generated by applying the reverse diffusion process or through score distillation sampling (SDS) [Poole et al. 2022]. SDS minimizes the KL divergence between the distribution of diffused samples derived from the forward diffusion process $q(\mathbf{x}^i | \mathbf{x}^0)$ and the reference data distribution learned by the pretrained diffusion model $f(\mathbf{x}^i)$. The gradient of the KL divergence can be estimated using the difference between the score from the forward diffusion process and the prediction from the diffusion model:

$$\nabla \mathcal{L}_{\text{SDS}} = \mathbb{E}_{i, \epsilon} \left[w(i) (f(\mathbf{x}^i) - \epsilon) \nabla \mathbf{x} \right], \quad (5)$$

where \mathbf{x} is the sample being optimized, and $w(i)$ are coefficients determined by the diffusion noise schedule. Previous work has shown that the weighting function $w(i)$ has limited impact and can be substituted with uniform weighting without negatively affecting performance [Guo et al. 2023]. The SDS gradient can also be derived from the diffusion loss in Equation (4), omitting the Jacobian with respect to the score estimator f . The SDS loss can be simplified as

$$\mathcal{L}_{\text{SDS}} = \|\hat{\epsilon} - \epsilon\|_2^2, \quad (6)$$

where $\hat{\epsilon}$ denotes the noise predicted by the denoising network f . The optimum of the SDS objective corresponds to samples that minimize the diffusion loss, thereby resembling the characteristics of the dataset on which the diffusion model was originally trained.

4 Overview

In this work, we introduce the *Score-Matching Motion Prior (SMP)*, a *reusable, modular* imitation objective derived from a pretrained diffusion model via score distillation sampling (SDS). The pretrained diffusion model is used to estimate the gradient of the log-likelihood (i.e., the *score*) of the reference distribution, which evaluates the similarity between an agent’s motions and motions in a reference dataset. The robustness of score-based generative modeling enables a pre-trained diffusion model to provide reliable guidance even for motions that are different from those in the original motion dataset. SMP can be combined with task-specific objectives to train control policies that can accomplish diverse tasks using natural life-like behaviors.

Figure 2 illustrates an overview of our framework. Unlike prior approaches that use diffusion models as *planners* [Ren et al. 2023; Serifi et al. 2024; Tevet et al. 2024], SMP repurposes a pretrained diffusion model as a general *reward model* for evaluating motion naturalness to guide training of a control policy. The task-agnostic diffusion model f is trained solely on reference motion data, independently of any control policy. Once trained, it is frozen and utilized as a reward function via score distillation sampling (SDS). When training a policy, the SDS objective encourages the policy to minimize the discrepancy between the noise ϵ added to the simulated motion, and the noise $\hat{\epsilon}$ estimated by the pretrained diffusion model. The SDS error is minimized when the agent’s motions closely aligns with the reference distribution. Similar to other distribution-matching objectives, SMP does not require the simulated character to exactly replicate specific reference motions. Instead, it encourages behaviors that capture the general characteristics of the reference data, enabling smooth transitions and adaptation to tasks that may require skills not explicitly present in the dataset. Furthermore, SMP can be trained on large and diverse datasets by employing a conditional diffusion model. This enables control policies to acquire different stylistic behaviors by conditioning the pretrained diffusion model on style labels, without the need to retain the original motion dataset.

5 Score-Matching Motion Priors

A score-matching motion prior is modeled as a diffusion model, which is trained to predict the *score* of noisy input motions. The predicted score can be interpreted as a correction applied to a noisy sample to denoise back to a clean sample from the training data distribution. This enables SMP to construct motion imitation objectives directly from the pretrained motion diffusion model.

Given a motion dataset, we first train a motion diffusion model to generate motion clips consisting of H consecutive frames $\mathbf{x} := (\mathbf{s}_{t-H+2}, \dots, \mathbf{s}_{t+1})$. During policy training, each simulated motion clip produced by the agent is diffused with Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ to a diffusion timestep i through the forward diffusion process in Equation (3), resulting in a noisy sample \mathbf{x}^i . The pretrained diffusion model $\hat{\epsilon} = f(\mathbf{x}^i)$ then predicts the score $\hat{\epsilon}$ at \mathbf{x}^i , which provides a denoising direction back toward the reference motion distribution. The correction that should be applied to align the agent’s motion with the reference distribution is therefore given by the noise residual ($\epsilon - \hat{\epsilon}$), as illustrated in Figure 3. The SMP reward

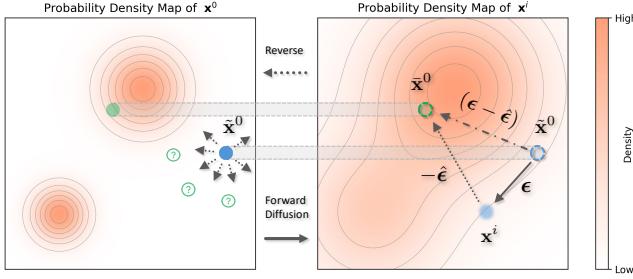


Fig. 3. A qualitative illustration of the probability density maps of \mathbf{x}^0 and \mathbf{x}^i . When the agent’s motion $\tilde{\mathbf{x}}$ deviates significantly from the reference distribution, the gradient of $p(\mathbf{x}^0)$ cannot be reliably approximated in low-density regions. The forward diffusion process in Equation (3) maps $\tilde{\mathbf{x}}$ to a diffused sample \mathbf{x}^i , where the density $p(\mathbf{x}^i)$ is higher and the score estimate is more reliable. This estimated score can then be used to obtain a pseudo target $\bar{\mathbf{x}}^0$ from the reference distribution via a reverse process. The residual between the predicted noise $\hat{\epsilon}$ and the added noise ϵ provides the correction that aligns the agent’s motion $\tilde{\mathbf{x}}$ with the reference distribution.

used for motion imitation is then defined as

$$r^{\text{smp}} = \exp(-w_s \|\hat{\epsilon} - \epsilon\|_2^2), \quad (7)$$

which is maximized when the simulated motion aligns with the reference distribution. Following standard RL reward design [Peng et al. 2018], we apply an exponential transformation to the SDS loss to normalize the reward between [0, 1]. While this deviates from the original SDS formulation, we find that this normalization produces empirical improvements when training RL policies.

Previous SDS-based methods have primarily shown promise in the 3D generation, but often produce low-quality, “blurry” results [Liang et al. 2024]. Earlier attempts to use SDS as an objective for reinforcement learning have also struggled to match the performance of adversarial methods for training control policies [Luo et al. 2024; Wu et al. 2025a]. In the following sections, we introduce several key design choices that improve stability for policy training using SMP, which enable policies to learn high-quality naturalistic human behaviors.

5.1 Ensemble Score-Matching

The SDS objective can be highly sensitive to the choice of diffusion timestep i , as different noise scales provide disparate forms of guidance [Lin et al. 2023]. At higher noise levels, diffused samples are heavily corrupted and dominated by Gaussian noise, which matches the training settings of the diffusion model and leads to more reliable score predictions due to less susceptibility to out-of-distribution samples. Therefore, evaluating the objective at higher diffusion timesteps i can be more instructive when the agent’s motions deviate substantially from the reference data distribution. However, excessive noise also removes much of the information present in the agent behaviors, which may prevent the objective from correcting more minute errors in the agent’s motions. In such cases, the guidance signal may encourage the policy to produce generic or averaged behaviors, as the model no longer retains sufficient information to steer toward more realistic and detailed motions.

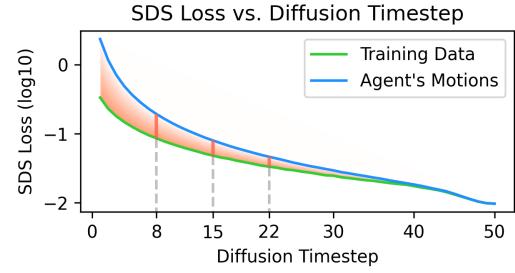


Fig. 4. An example of the SDS loss across diffusion noise levels, averaged over 1024 samples. The y -axis shows \log_{10} values for better visualization. Larger diffusion timesteps correspond to higher noise levels applied to the sample. The orange region highlights the discrepancy between agent’s motions and training data. SDS loss varies across different diffusion noise levels. At high noise levels, the SDS loss is always minimized, providing limited information about differences between the agent’s motion and the reference distribution, whereas lower noise levels generally yield larger losses. When the diffusion model’s prediction is sufficiently reliable, i.e., on less out-of-distribution (OOD) data, SDS loss computed at lower noise levels can better capture such discrepancies. In our experiments, we compute SDS loss at diffusion timesteps [22, 15, 8].

Therefore, when the difference between the agent’s motions and motions in the dataset is relatively small, diffusing the sample to lower noise levels can better preserve information of the agent’s motions and apply finer-grained corrections.

In prior SDS-based methods in vision domains [Guo et al. 2023; Poole et al. 2022], it is common to sample the diffusion noise level uniformly $i \sim \mathcal{U}(1, N)$. However, in reinforcement learning, this stochasticity can introduce undesirable variance into the reward signal, leading to less reliable value and advantage estimation. For example, samples diffused with high noise levels are nearly pure Gaussian noise, allowing the diffusion model to predict the noise easily, resulting in small loss values. Whereas samples with lower noise levels generally produce larger denoising errors, as illustrated in Figure 4. This noise-dependent variation in the SDS loss prevents values from randomly sampled timesteps from providing a consistent indicator of the similarity between the agent’s motion and the reference data distribution.

To mitigate this variance, rather than randomly sampling a single timestep per update, we compute a more consistently behaved SDS objective by ensembling multiple SDS evaluations over a fixed set of diffusion timesteps $i \in \mathbb{K}$. The resulting SMP reward is:

$$r^{\text{smp}} = \exp\left(-\frac{w_s}{|\mathbb{K}|} \sum_{i \in \mathbb{K}} \|\hat{\epsilon}_i - \epsilon_i\|_2^2\right), \quad (8)$$

where $\hat{\epsilon}_i = f(\sqrt{\alpha_i} \tilde{\mathbf{x}}^0 + \sqrt{1 - \alpha_i} \epsilon_i)$, $\tilde{\mathbf{x}}$ denotes the simulated character’s motion, and \mathbb{K} is a predefined set of diffusion timesteps. In all experiments, we use $\mathbb{K} = \{0.44N, 0.30N, 0.16N\}$, where N is the total number of diffusion steps.

In addition, we apply adaptive normalization using the running mean μ_i of the SDS error at each diffusion timestep i to mitigate the varying loss scales at different noise levels. This adaptive normalization also reduces SMP’s sensitivity to variations across different

ALGORITHM 1: Policy Training with SMP

```

1: Input (optional): style label  $c$ 
2:  $f \leftarrow$  load pretrained diffusion model
3:  $\pi \leftarrow$  initialize policy
4:  $V \leftarrow$  initialize value function
5:  $\mathcal{B} \leftarrow \emptyset$  initialize reply buffer

6: while not done do
7:   for trajectory  $j = 1, \dots, m$  do
8:      $\tau^j \leftarrow \{(s_t, \tilde{x}_t, a_t, r_t^g)_{t=0}^{T-1}, s_T^g, \tilde{x}_T\}$  collect trajectory with  $\pi$ 
9:     for  $t = 0, \dots, T - 1$  do
10:      for diffusion timestep  $i \in \mathbb{K}$  do
11:         $\epsilon_i \sim \mathcal{N}(0, I)$ 
12:         $\hat{\epsilon}_i \leftarrow f(\sqrt{\alpha_i} \tilde{x}_{t+1} + \sqrt{1 - \alpha_i} \epsilon_i, c)$ 
13:      end for
14:       $r_t^{\text{smp}} \leftarrow$  compute prior reward via eq. (8) using  $\{\epsilon, \hat{\epsilon}\}_i^{\mathbb{K}}$ 
15:       $r_t \leftarrow w_{\text{prior}} r_t^{\text{smp}} + w^g r_t^g$ 
16:      record  $r_t$  in  $\tau^j$ 
17:    end for
18:    store  $\tau^j$  in  $\mathcal{B}$ 
19:  end for

20:  update  $V$  and  $\pi$  using data from trajectories  $\{\tau^j\}_{j=1}^m$ 
21: end while

```

pre-trained diffusion models and behavioral styles, thereby reducing the need for manual parameter tuning for the SMP reward function.

5.2 Generative State Initialization

Reference state initialization (RSI), where the simulated character is initialized to states randomly sampled from a reference motion dataset, has been shown to be a vital technique for improving exploration in motion imitation methods [Peng et al. 2018]. However, RSI requires access to the motion dataset to sample initial states from during policy training. To remove this reliance on the original dataset, SMP can instead leverage the generative prior to *generate* initial states when training the policy. When using SMP to train a policy, initial states are generated by sampling from the diffusion model used as the pretrained motion prior. SMP therefore serves dual purposes in our framework, as both a reward function and an initial state distribution when training new policies. This *generative state initialization* (GSI) method alleviates the need to retain the original motion dataset once the SMP has been trained by leveraging the generative capabilities of diffusion models to produce diverse, high-quality initial states.

6 Model Representation

Our framework utilizes a pretrained diffusion model as a reusable, modular motion prior for imitation learning. The task-agnostic motion diffusion model serves as a reward function that evaluates the naturalness of the agent’s motion within a reinforcement learning framework. This allows for the training of control policies that accomplish diverse tasks while exhibiting naturalistic behaviors that resemble those in the original motion dataset. In this section, we detail key design decisions of the learning framework.

6.1 Motion Representation

Designing an appropriate motion representation is critical both for training the diffusion model to capture the dataset distribution, and for repurposing it as a motion prior for policy training. The representation should contain sufficient information to reconstruct motion while remaining easy to extract from both kinematic motion clips and simulator state observations. Following the design of prior works in motion diffusion and AMP [Peng et al. 2021; Tevet et al. 2023; Zhang et al. 2022], our motion features include:

- Root linear and angular velocities, represented in the character’s local coordinate frame at the last timestep in a motion segment.
- Local joint rotations.
- 3D positions of end-effectors (e.g., hands and feet), represented in the character’s local coordinate frame.

The character’s local coordinate frame is defined with the origin at the root (i.e., pelvis), the x -axis aligned with the root link’s facing direction, and the y -axis aligned with the global up vector. Joint rotations are represented using a 6D representation for spherical joints [Zhou et al. 2019].

6.2 Diffusion Model Implementation

The motion diffusion model is implemented as a transformer encoder, using adaptive normalization to inject noise-level conditions (and style conditions, when available). Unless otherwise specified, we use a window of $n = 10$ frames. We find that a carefully designed two-layer transformer encoder with only 3M parameters is sufficient to capture the distribution of large-scale motion datasets, such as 100STYLE [Mason et al. 2022] with over 20 hours of stylized motions. The number of diffusion timesteps is set to $N = 50$, and the model is trained to predict ϵ . Following standard practice for training diffusion models, we apply exponential moving average (EMA) on the model parameters during training [Dhariwal and Nichol 2021; Karras et al. 2024]. The model is trained for 400k–800k iterations, depending on the dataset, with convergence typically achieved within five hours on a single RTX 4090 GPU.

6.3 Policy Implementation

Following prior work, the control policy π is modeled using a multi-layer perceptron (MLP) that maps an input state s_t and goal g to a Gaussian distribution over the action space $a_t \in \mathcal{A}$ [Peng et al. 2018]. The value function $V(s_t, g)$ is modeled by a separate MLP network with a similar architecture to the policy.

States and Actions. The state s_t is represented using features similar to those in Peng et al. [2021], such as body link positions, link rotations encoded in a 6D representation, and linear and angular velocities of each link. All features are calculated in the character’s local coordinate frame. Since our policies are not trained to track specific reference motions, no phase variable or target reference state information is provided in the state. The action a_t specifies target joint positions for PD controllers at each joint. For spherical joints, each target rotation is parameterized using a 3D exponential map $q \in \mathbb{R}^3$ [Grassia 1998].

Training. The policy is trained using proximal policy optimization (PPO) [Schulman et al. 2017]. At each timestep t , the agent queries the motion prior (see Algorithm 1) to obtain a prior reward r_t^{smp} , and may also receive a task reward r_t^g from the environment. These are combined linearly to form the composite reward at that timestep

$$r_t = w^{\text{prior}} r_t^{\text{smp}} + w^g r_t^g, \quad (9)$$

following Peng et al. [2021]. After collecting a batch of trajectories, mini-batches are sampled from the buffer to update both the policy and the value function. The policy is updated with PPO using advantages estimated via GAE(λ) [Schulman et al. 2015], while the value function is trained with targets computed via TD(λ) [Sutton et al. 1998]. The pretrained diffusion model is kept fixed during the policy training process, and does not need to be updated.

7 Tasks

We evaluate the effectiveness of SMP across six motion control tasks, showcasing its ability to train policies that can perform various control tasks using diverse motion styles. Below, we summarize each task and the corresponding goal observation g , which provide the agent with task-relevant information from the environment. More comprehensive details and task reward functions are provided in Appendix.

Target Speed. This task requires the character to move at a target speed. The goal for the policy is specified as $g_t = v_t^*$. The target speed v_t^* is randomly sampled from $[1.2, 6.8]$ m/s. To focus on speed tracking and gait transitions induced by different speeds, the target movement direction is kept fixed.

Steering. The task requires the character to face a specified 2D heading direction h_t^* while simultaneously traveling at a target speed v_t^* along a target horizontal direction d_t^* . The steering policy receives the goal information as $g_t = (d_t^*, v_t^*, h_t^*)$.

Target Location. In this task, the character is instructed to reach a 2D target location specified on the floor plane. The agent perceives the goal via $g_t = p_t^*$, where the target location p_t^* is represented in the character’s local coordinate frame.

Dodgeball. In this task, a ball is launched toward the character from a random position up to 10m away, with a launch speed sampled from $[20, 25]$ m/s. This gives the agent less than 0.5s to react and dodge. If the character is hit, the episode terminates early and the agent receives zero reward for all remaining timesteps as a penalty. The agent receives ball information through the goal vector $g_t = (p_t^{\text{ball}}, \dot{p}_t^{\text{ball}})$.

Object Carry. In addition to training priors for human motion, SMP can also be used to train priors that jointly model the interactions between humans and objects. First, we train an SMP on a dataset of human-object carrying motions. This prior is then used to train policies for an object carrying task, where the character is required to carry a box from the ground to a randomly placed target location. The goal $g_t = p_t^{\text{box}*}$ records the target box position. The state s_t is augmented with additional features that describe the state of the box, including the position p_t^{box} and the orientation q_t^{box} . All box information is represented in the character’s local coordinate.

Setup. In this task, the policy trained to recover from arbitrary fallen states and maintain a minimum root height of $0.8m$.

8 Results

To evaluate the effectiveness of SMP, we apply our framework to train policies for on a suite of challenging motion control tasks. In Section 8.1, we demonstrate the *modularity* of SMP by using a pre-trained prior to train new control policies without requiring access to the original motion dataset. By training a single style-conditioned diffusion model on the 20-hour 100STYLE dataset, the prior can then be used to train policies to perform tasks in a variety of different styles, where each style-specific prior reward is obtained simply by conditioning the pretrained model on the desired style label. New styles can also be synthesized by composing the pretrained diffusion model when conditioned on different styles. In Section 8.2, we show that a single motion prior can be *reused* to train policies across diverse tasks, such as steering, target location, and dodgeball. In addition to controlling the behavioral style of the character, in Section 8.3 we show that SMP can also model *human-object interaction priors*. By training the diffusion model on human-object interaction data, SMP learns to jointly model both object and character motions, allowing agents to use human-like behavior to perform object interaction tasks. SMP also enables agents to learn robust and natural recovery behaviors, as demonstrated in Section 8.4. Similar to other distribution-matching objectives, the policies trained with SMP can synthesize new skills that are not explicitly present in the reference dataset. In Section 8.5, when trained on a 3-second walk-jog-run dataset, SMP automatically leads to the emergence of different locomotion gaits that enable a character to closely follow a wide spectrum of target speeds, as well as natural transitions between the various gaits. Finally, to benchmark SMP’s effectiveness for motion imitation, we evaluate SMP on single-clip imitation tasks (Section 8.6). SMP is able to closely reproduce a diverse set of dynamic and acrobatic skills, achieving comparable motion quality to state-of-the-art adversarial imitation learning methods.

Baselines: In the following experiments we compare the performance of SMP with AMP [Peng et al. 2021], a widely-used adversarial imitation learning method. Following Peng et al. [2021], the reward weights for AMP are set to $w^{\text{prior}} = w^g = 0.5$. In addition to the standard AMP method, we also compare with a variant of AMP that uses a *frozen discriminator* to examine whether the learned adversarial motion prior can be effectively *reused* without further training. First, a control policy and discriminator are trained jointly using AMP, then the trained discriminator is reused to train new control policies on the same task without further finetuning using data from the new policy. Furthermore, we also include a simple tabula-rasa learning baseline (w/o Prior), where the policy is trained solely to maximize the task reward ($w^g = 1$), without any motion priors.

8.1 One Motion Prior for 100 + N Styles

SMP serves as a modular motion style reward model that guides policy training without requiring access to the original motion dataset. This enables the application of various adaptation techniques to our diffusion-based reward model to further shape the motion prior. We

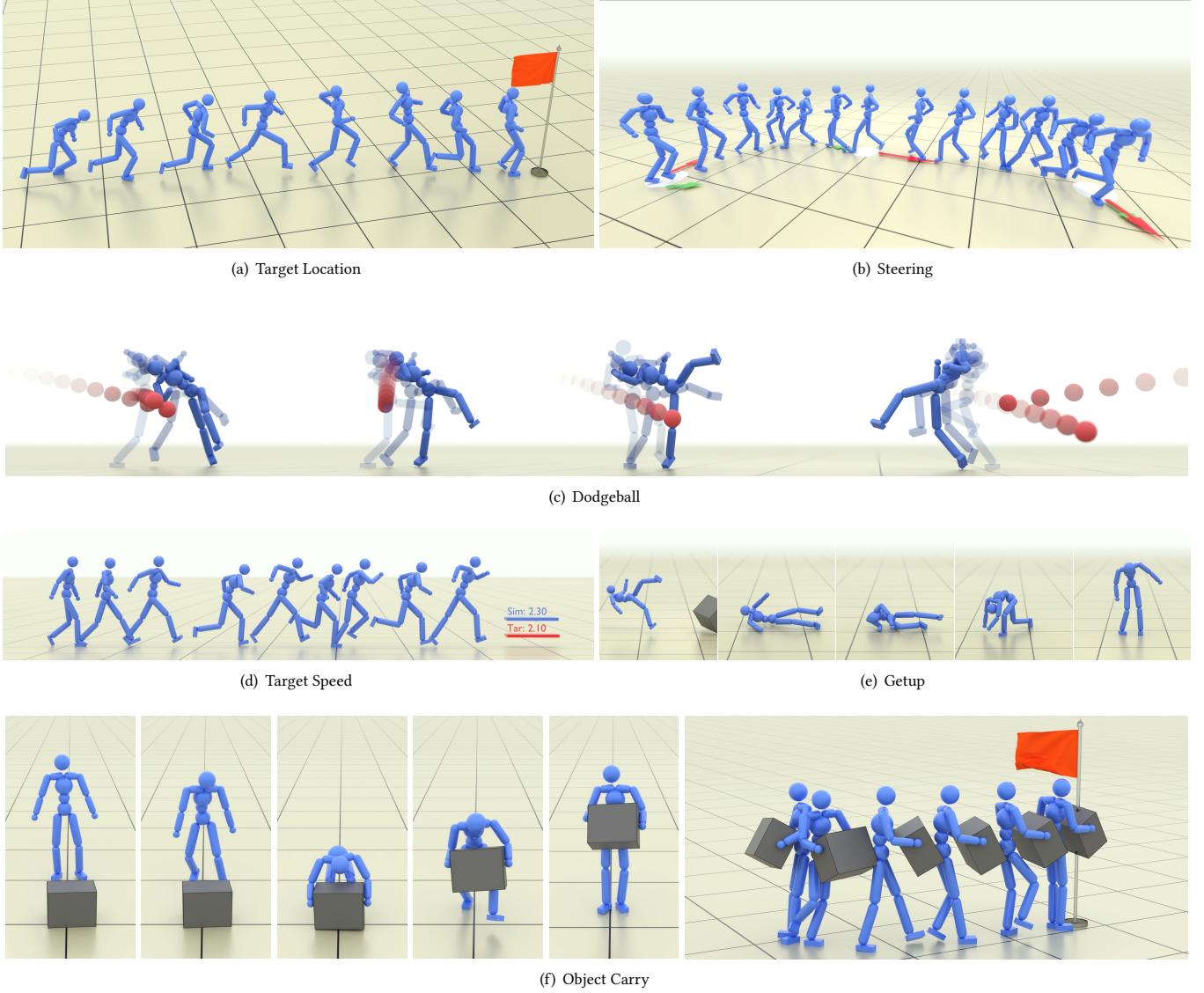


Fig. 5. Score-matching motion priors can be trained on datasets of varying sizes, independently of any task or control policy. Once trained, an SMP provides a motion imitation objective r^{smp} , which can be composed with task rewards r^g to train multiple policies that complete a diverse array of tasks while exhibiting natural, life-like behaviors.

train a general 100style-conditioned motion diffusion model $f(\mathbf{x}^i, c)$ using the entire 20-hour 100STYLE dataset, where c is a style label. Classifier-free guidance (CFG) can then be applied to reshape this general prior into style-specific motion priors:

$$f_{\text{zombie}} = f(\mathbf{x}^i, \emptyset) + w_{\text{cfg}} (f(\mathbf{x}^i, c_{\text{zombie}}) - f(\mathbf{x}^i, \emptyset)),$$

where the style-conditioned prediction $f(\mathbf{x}^i, c_{\text{zombie}})$ is applied to the unconditional prediction $f(\mathbf{x}^i, \emptyset)$ according to the guidance weight w_{cfg} . These adapted priors can be used to train policies for a given task using different behavioral styles. The performance statistics for policies trained with various styles are reported in

Table 1. All policies are trained using the same underlying SMP model with different style labels. We find that simply setting the CFG scale to 1.0 is generally sufficient to specialize the 100-style prior into distinct style-specific priors, enabling agents to perform tasks with diverse stylistic behaviors, as shown in Figure 1. While AMP can achieve comparable performance, it requires training different style-specific discriminators using style-specific datasets. In contrast, using fixed pretrained discriminators (AMP-Frozen) is ineffective in producing the desired stylistics behaviors. With AMP-Frozen, we observe that the discriminator’s accuracy drops over the course of policy training, which indicates that the policy is exploiting the

Table 1. Performance of policies trained with different styles on the target location task. Task returns are normalized to [0, 1]. Style accuracy is evaluated using a style classifier trained on the 100STYLE dataset. AMP models are trained using style-specific motion datasets, whereas SMP is pretrained once on the full 100STYLE dataset and adapted to each style using CFG during policy training. The modularity of SMP enables effective training of style-specific policies without access to the original dataset. AMP with a frozen discriminator is ineffective for producing policies of the desired styles.

Dataset	Style	Task Return			Style Accuracy		
		AMP	AMP Frozen	SMP (Ours)	AMP	AMP Frozen	SMP (Ours)
100STYLE	AeroPlane	0.867 \pm 0.001	0.871 \pm 0.008	0.882 \pm 0.010	0.981 \pm 0.004	0.000 \pm 0.000	0.995 \pm 0.003
	Chicken	0.876 \pm 0.005	0.874 \pm 0.003	0.877 \pm 0.008	0.973 \pm 0.019	0.306 \pm 0.530	0.989 \pm 0.019
	CrossOver	0.866 \pm 0.001	0.470 \pm 0.363	0.879 \pm 0.002	0.994 \pm 0.005	0.320 \pm 0.554	0.994 \pm 0.005
	Dinosaur	0.886 \pm 0.001	0.856 \pm 0.020	0.882 \pm 0.015	0.998 \pm 0.004	0.004 \pm 0.006	0.999 \pm 0.001
	FlickLegs	0.881 \pm 0.001	0.580 \pm 0.269	0.868 \pm 0.012	0.983 \pm 0.004	0.009 \pm 0.011	0.979 \pm 0.024
	HandsBetweenLegs	0.870 \pm 0.003	0.888 \pm 0.007	0.850 \pm 0.004	0.690 \pm 0.198	0.000 \pm 0.000	0.978 \pm 0.007
	HighKnees	0.882 \pm 0.003	0.872 \pm 0.014	0.897 \pm 0.003	0.988 \pm 0.009	0.012 \pm 0.017	0.992 \pm 0.005
	Neutral	0.873 \pm 0.007	0.755 \pm 0.103	0.891 \pm 0.016	0.991 \pm 0.005	0.426 \pm 0.478	0.999 \pm 0.001
	Skip	0.875 \pm 0.003	0.512 \pm 0.321	0.891 \pm 0.008	0.985 \pm 0.003	0.425 \pm 0.479	0.646 \pm 0.350
	Spin (Clockwise)	0.871 \pm 0.001	0.874 \pm 0.012	0.858 \pm 0.004	0.990 \pm 0.003	0.006 \pm 0.010	0.978 \pm 0.015
	Superman	0.872 \pm 0.004	0.870 \pm 0.013	0.893 \pm 0.005	0.998 \pm 0.004	0.304 \pm 0.492	0.999 \pm 0.002
	Zombie	0.872 \pm 0.003	0.823 \pm 0.039	0.875 \pm 0.014	0.973 \pm 0.005	0.649 \pm 0.563	0.996 \pm 0.006
Average		0.874	0.771	0.879	0.962	0.205	0.962

reward model. This exploitation often leads to unnatural behaviors. Qualitative comparisons of the various methods are available in the supplementary video.

Style Composition. SMP supports crafting novel motion priors by manipulating the pretrained diffusion model’s outputs, enabling the creation of new styles that are not present in the original without requiring additional training of the prior. As shown in Figure 6, two different styles can be composed by blending their style-conditioned predictions from the pretrained diffusion model to produce a new “AeroPlane + HighKnees” prior. The composite prior is constructed via:

$$f_{\text{comp}} = M_{\text{upper}} \odot f(\mathbf{x}^i, c_{\text{aeroplane}}) + M_{\text{lower}} \odot f(\mathbf{x}^i, c_{\text{highknees}}),$$

where M_{upper} and M_{lower} are binary masks applied to the upper- and lower-body features, respectively. The resulting prior f_{comp} can be used directly as the SMP reward r^{smp} . Moreover, our proposed generative state initialization (GSI) can also adopt the composite prior to generate initialization states of the new style, enabling effective exploration when training policies for the new style. Using SMP and GSI together with f_{comp} , our framework is able to train an agent that follows task commands while spreading its arms and lifting its knees high, all without relying on any reference motion data of the specific style.

8.2 One Motion Prior for Multiple Tasks

One pretrained score-matching motion prior can be reused to train multiple policies for different tasks, such as steering, target location and dodgeball, as show in Table 2. The score-matching motion prior is trained on a subset of the LaFAN1 dataset [Harvey et al. 2020], containing unstructured running behaviors. The resulting policies achieve high task returns while producing natural gaits. As shown in Figures 5(a) and 5(b), the character can dynamically select and transition between suitable gaits from the prior, to maintain

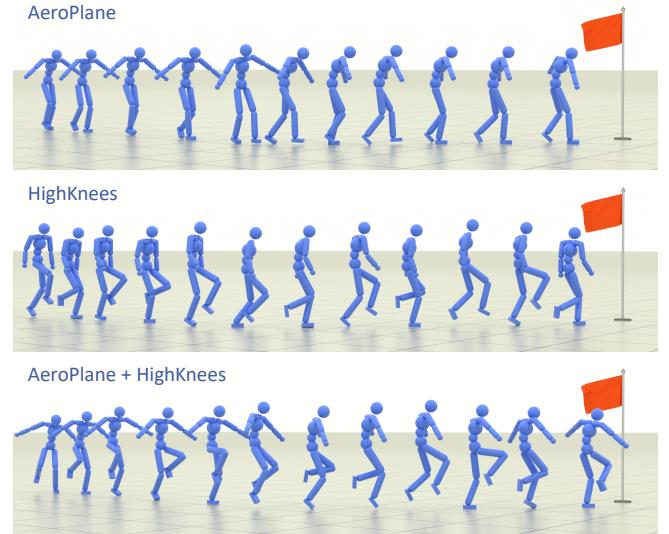


Fig. 6. A pretrained 100-style motion prior can also be adapted to synthesize motion priors of new styles. For example, we create a novel “AeroPlane + HighKnees” prior by blending two existing styles in the ϵ -space. This crafted prior, which is used for both generative state initialization and the motion prior objective r^{smp} , enables the agent to perform the target location task with a new, agile style, all without requiring any reference data.

both task performance and motion naturalness. The *Steering* policy automatically executes a backward jog when the facing direction opposes the target velocity and transitions to a forward jog once they align. It also learns lateral gaits, such as side-stepping and cross-stepping, without the need for an explicit motion planner. The policy trained with SMP also discovers human-like strategies in the *Target Location* task. The character automatically slows down as it

Table 2. Performance of combining different motion priors with task objectives, as well as optimizing task objectives alone. Task returns are normalized to [0, 1]. A single score-matching motion prior can be effectively reused to train policies across different tasks. Even with a small dataset of only three reference snippets, SMP still enables the agent to perform the target-speed task effectively and with natural motion.

Dataset	Task	Task Return			
		w/o Prior	AMP	AMP Frozen	SMP (Ours)
LaFAN1	Steering	0.901 \pm 0.006	0.634 \pm 0.019	0.243 \pm 0.008	0.914 \pm 0.006
	Target Location	0.615 \pm 0.268	0.737 \pm 0.014	0.101 \pm 0.012	0.793 \pm 0.006
	Dodgeball	0.277 \pm 0.046	0.233 \pm 0.003	0.204 \pm 0.004	0.733 \pm 0.035
Walk-Jog-Run	Target Speed	0.905 \pm 0.013	0.904 \pm 0.002	0.158 \pm 0.021	0.918 \pm 0.002

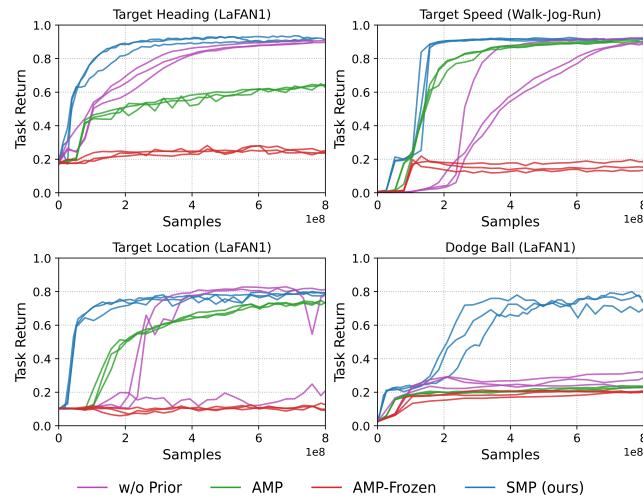


Fig. 7. Comparison of normalized task returns across motion control tasks. SMPs demonstrate better sample efficiency, potentially due to the more consistently informative guidance provided by the stationary SMP reward function.

approaches the target. When the target is far away, the character transitions to a fast running behavior. These nuanced behaviors arise purely from the stationary score-matching motion prior combined with a simple target-distance task objective. No motion planner is required to explicitly select which gait the character should perform. SMP provides the policy with the flexibility to adapt behaviors in the dataset in order to create new skills for different tasks. In the challenging *Dodgeball* task, agents trained with the locomotion prior spontaneously develop agile jumping and dodging skills, as shown in Figure 5(c). These behaviors were not present in the original dataset, but they closely resemble real human strategies in dodgeball. In contrast, AMP fails on this task, potentially due to the instability of the adversarial objective when the required dodging skill deviates significantly from the reference dataset, which contains only locomotion motions. Policies trained using AMP with a frozen discriminator further fail to produce natural behaviors or achieve strong task performance across all tasks, indicating that

Table 3. Performance of SMP on object carry and getup tasks. SMP can be extended to model human-object interaction priors and train effective interaction policies. It also guides the policy to learn robust getup skills from arbitrary fallen states, an essential recovery capability for everyday tasks.

Task	Task Return	Success Rate
Object Carry	0.909 \pm 0.022	0.997 \pm 0.004
Getup	0.897 \pm 0.029	0.998 \pm 0.003

adversarial priors cannot be effectively reused, even for identical tasks. In comparison, SMP allows a single pretrained motion prior to be reused across different policies and tasks. Moreover, as shown by the task return curves in Figure 7, a fixed score-matching motion prior also offers higher sample efficiency, consistently providing stable and informative guidance throughout reinforcement learning.

8.3 Human-Object Interaction Priors

SMP can be applied not only to locomotion patterns but also to model *interaction priors* that jointly capture both character and object motions. Given a human-object interaction (HOI) dataset, we train an unconditional HOI diffusion model $f(\mathbf{x}_{\text{char}}^i, \mathbf{x}_{\text{obj}}^i)$, which simultaneously learns the dynamics of the character motion \mathbf{x}_{char} and the object motion \mathbf{x}_{obj} . The object motion is represented by its rotation and position, both expressed in the character’s local coordinate frame. The prior reward r_t^{prior} is computed following the same procedure described in Algorithm 1 to evaluate the naturalness of character-object interaction dynamics. We demonstrate that the score-matching interaction prior, when combined with task rewards, effectively guides policies to accomplish complex, multi-stage tasks, where the agent can walk toward a box, pick it up, and carry it to an arbitrary target location, with natural, coordinated, and physically faithful motions.

8.4 Learning to Get Up

Getting up from arbitrary fallen states is an essential skill for both humans and humanoid agents. However, this task is particularly challenging, as many existing methods rely on auxiliary rewards to suppress overly dynamic or erratic motions. Existing approaches, such as AMP, which adaptively update the motion prior during training, can effectively learn natural get-up behaviors. As shown in Table 3, our method demonstrates that even with a stationary motion prior, SMP can successfully train a robust get-up policy capable of recovering from random fallen states. In Figure 5(e), the character is knocked far away yet is still able to roll over, push itself up with its hands, and successfully get back to standing. More qualitative results are best viewed in the supplementary video.

8.5 Skill Emergence under Data Scarcity

Reinforcement learning with distribution-matching objectives naturally allows agents to generalize and develop skill that not present in the dataset. Different from the typical application of SDS with image-based priors on large dataset, SMP can learn from as little as three seconds of motions, while still allowing the agent to adapt

Table 4. Position tracking error for individual skills. SMP successfully imitates a variety of skills, with imitation accuracy comparable to AMP while not requiring access to reference motion data during policy training. AMP-Frozen, which attempts to eliminate AMP’s data dependency by substituting a pre-trained discriminator, results in severely degraded behavior.

Skill	DM	Position Tracking Error [m]			
		AMP	AMP Frozen	SMILING	SMP (Ours)
Walk	0.010 ± 0.001	0.028 ± 0.004	0.044 ± 0.010	0.042 ± 0.007	0.030 ± 0.004
Run	0.013 ± 0.000	0.088 ± 0.010	0.129 ± 0.039	0.115 ± 0.040	0.067 ± 0.001
Spinkick	0.073 ± 0.061	0.049 ± 0.001	0.324 ± 0.040	0.088 ± 0.005	0.059 ± 0.006
Cartwheel	0.243 ± 0.157	0.043 ± 0.002	0.419 ± 0.013	0.104 ± 0.005	0.043 ± 0.005
Backflip	0.073 ± 0.001	0.058 ± 0.002	0.272 ± 0.034	0.144 ± 0.017	0.069 ± 0.008
Crawl	0.006 ± 0.000	0.011 ± 0.000	0.285 ± 0.008	0.061 ± 0.057	0.011 ± 0.001
Average	0.070	0.046	0.246	0.092	0.046

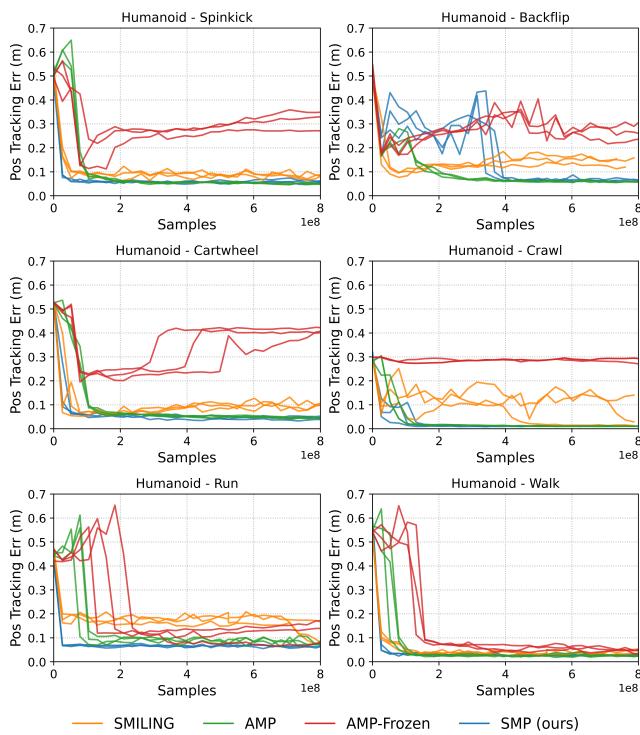


Fig. 8. Learning curves for single-clip imitation tasks over three random seeds. To evaluate the robustness of SMP, we also train the diffusion models used for motion priors with different random seeds for each experimental run. Our framework demonstrates consistently good performance across seeds.

transitions and behaviors beyond the reference data. In this speed following experiment, the dataset contains only three motion clips that move at different speeds. But the policy then learns to adapt those motions to move at a wide range of target speeds, as shown in Figure 5(d). The character naturally adjusts their gait to match the target speed, walking at slower paces and shifting into jogging and running as the speed increases. Although the reference dataset

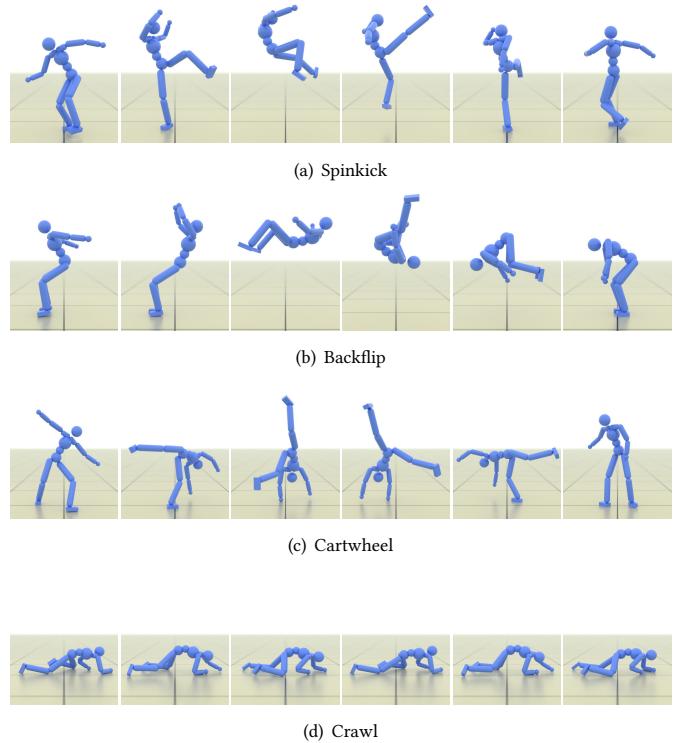


Fig. 9. Visual snapshots of humanoid characters trained via SMP imitating diverse skills, including highly dynamic and contact-rich motions.

only contains motions at three discrete speeds, the policies learn to modulate the frequency and stride within each gait, producing continuous variations that preserve the style of the original motions. Moreover, the policies exhibit smooth and expressive transitions between gaits not present in the dataset, including building from walk to jog, snapping down from run to walk, and bursting from walk into a sprint. In Figure 7, the learning curves show that the motion prior also improves sample efficiency compared to the baseline without any prior rewards (w/o Prior).

8.6 Benchmark: Single-Clip Imitation

To further evaluate the effectiveness of SMP at imitating behaviors from reference motion clips, we compare it with AMP [Peng et al. 2021], AMP-Frozen, and SMILING [Wu et al. 2025a] on a series of single-clip imitation tasks. The policies are trained using the prior reward r_t^{smp} only. Comparison with a motion tracking method, DeepMimic [Peng et al. 2018], is included for reference. Imitation performance is assessed using the position tracking error e_t^{POS} . Unlike motion-tracking methods such as DeepMimic, which are explicitly designed for precise replication of reference motions, SMP, AMP, and SMILING focus on imitating the general style of the motions. Consequently, these methods do not synchronize the policy with the reference trajectory. To fairly evaluate all methods, dynamic time warping (DTW) is applied for SMP, AMP, and SMILING to temporally align the simulated motion with the reference using position tracking error as the cost function [Sakoe and Chiba

Table 5. Comparison between computing the SDS objective using a single randomly sampled timestep versus ensemble over a fixed set of timesteps. Policies trained with random sampled timestep exhibit higher errors, particularly on challenging skills such as the backflip.

Skill	Position Tracking Error [m]	
	Random	Ensemble
Run	0.062 \pm 0.000	0.067 \pm 0.001
Cartwheel	0.058 \pm 0.015	0.043 \pm 0.005
Backflip	0.195 \pm 0.006	0.069 \pm 0.008

1978]. Additionally, pose termination is disabled for DeepMimic to ensure consistency, as it does not apply to non-synchronized approaches.

Quantitative results are summarized in Table 4, with learning curves provided in Figure 8. While AMP exhibits strong imitation performance, it requires retaining the reference motion data throughout policy training. In comparison, SMP accurately reproduces a diverse range of skills, including highly dynamic ones, and demonstrates imitation accuracy and sample efficiency comparable to AMP, all without relying on reference data during training. Across skills, SMP consistently outperforms SMILING and AMP-Frozen. AMP-Frozen, which attempts to remove AMP’s data dependency by simply substituting a pre-trained discriminator, proves inadequate for motion imitation and results in severely degraded behaviors. DeepMimic benefits from explicit phase synchronization and attains lower tracking errors on certain motions. However, SMP still performs competitively. DeepMimic’s performance degrades on challenging motions such as spinkick and cartwheel, where disabling pose termination leads some runs to converge to suboptimal local solutions.

9 Ablations

Our system is one of the first frameworks to construct a *reusable* motion prior based on score-matching, that can achieve motion fidelity comparable to state-of-the-art adversarial imitation learning methods. In this section, we identify the key design choice of *ensemble score-matching* , which enables more stable training and higher-quality results. We compare our ensemble score-matching approach (“Ensemble”), which averages multiple SDS evaluations over a fixed set of diffusion timesteps, against the typical strategy used in prior work, which evaluates SDS at a single randomly sampled timestep (“Random”) [Luo et al. 2024; Poole et al. 2022; Wu et al. 2025a]. The comparison is conducted on single-clip imitation tasks of varying difficulty, including Run, Cartwheel, and Backflip. As shown in Table 5, the single-timestep SDS objective performs comparably on simple skills but struggles on more challenging behaviors. For instance, when imitating a backflip, policies trained with a single SDS evaluation tend to collapse to stationary standing or fall backward instead of executing a complete flip. In contrast, ensembling SDS evaluations across a fixed set of diffusion timesteps provides a more consistent and informative reward signal, which enables effective policy training even for complex acrobatic motions.

10 Discussion and Limitations

In this work, we presented *Score-Matching Motion Priors (SMP)* , a *reusable* and *modular* motion prior for physics-based character animation based on score-matching. Once trained, the learned motion prior can be reused to train control policies for diverse tasks, guiding the policies towards natural behaviors that match the reference distribution. Our priors can be effectively used without the need to retain the original motion dataset. Test-time diffusion techniques, such as classifier-free guidance, can be applied to shape the base motion prior and produce novel stylistic priors that enable agents to perform tasks in specific styles. We demonstrate the effectiveness of our method across a diverse variety of settings, ranging from single-character behaviors to human-object interactions. SMPs can be effectively constructed from a wide spectrum of different datasets, with as few as 3 seconds of motion clips to relatively large-scale (20-hour) motion datasets.

In our experiments, we demonstrate that reinforcement learning with score distillation sampling (SDS) objectives can produce motions of comparable quality to adversarial imitation learning, without the need to continuously update the prior during policy training. SMP also demonstrates higher sample efficiency than adversarial priors in most scenarios, which may be in part due to the more stable stationary reward function from SMP compared to adversarial reward models. However, as with many mode-seeking objectives, policies trained with SMP are susceptible to mode-collapse, leading to policies that only reproduce a limited subset of behaviors in the original dataset. While our work primarily focuses on applications to humanoid motion control, we believe SMP can also be applied to other control problems. We hope this work opens new directions toward building general motion priors that enable more versatile controllers for physics-based character animation and robotics beyond motion tracking.

Acknowledgments

This work was supported by Sony Interactive Entertainment, NSERC (RGPIN-2015-04843), and the National Research Council Canada (AI4D-166). We would like to thank Michiel van de Panne, Amit H. Bermano, and Alejandro Escontrela for their insights and discussions.

References

- Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. 2019. DReCon: data-driven responsive control of physics-based characters. *ACM Transactions On Graphics (TOG)* 38, 6 (2019), 1–11.
- Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. 2023. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*.
- Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. 2010. Generalized Biped Walking Control. *ACM Transactions on Graphics* 29, 4 (2010), Article 130.
- Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.
- Alejandro Escontrela, Xue Bin Peng, Wenhao Yu, Tingnan Zhang, Atil Iscen, Ken Goldberg, and Pieter Abbeel. 2022. Adversarial motion priors make good substitutes for complex reward functions. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 25–32.
- Petros Faloutsos, Michiel Van de Panne, and Demetri Terzopoulos. 2001. Composable controllers for physics-based character animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 251–260.
- F Sebastian Grassia. 1998. Practical parameterization of rotations using the exponential map. *Journal of graphics tools* 3, 3 (1998), 29–48.

- Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilijas Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morris Doulaty, Akshay Erappalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebrselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Wesli Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghai Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. 2022. Egō4D: Around the World in 3,000 Hours of Egocentric Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 18995–19012.
- Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian LaForte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. 2023. threestudio: A unified framework for 3D content generation. <https://github.com/threestudio-project/threestudio>.
- Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 60–1.
- Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016).
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
- Jessica K Hodgins, Wayne L Wooten, David C Brogan, and James F O’Brien. 1995. Animating human athletics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 71–78.
- Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.
- Daniel Holden, Jun Saito, and Taku Komura. 2016. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.
- Bo-Ruei Huang, Chun-Kai Yang, Chun-Mai Lai, Dai-Jie Wu, and Shao-Hua Sun. 2024b. Diffusion Imitation from Observation. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 137190–137217. https://proceedings.neurips.cc/paper_files/paper/2024/file/f7faa46b563c2e5343a728c85bace833-Paper-Conference.pdf
- Xiaoyu Huang, Yufeng Chi, Ruofeng Wang, Zhongyu Li, Xue Bin Peng, Sophia Shao, Borivoje Nikolic, and Koushil Sreenath. 2024a. DiffuseLoco: Real-Time Legged Locomotion Control with Diffusion from Offline Datasets. *ArXiv* abs/2404.19264 (2024). <https://api.semanticscholar.org/CorpusID:269457018>
- Xiaoyu Huang, Takara Truong, Yunbo Zhang, Fangzhou Yu, Jean Pierre Sleiman, Jessica Hodgins, Koushil Sreenath, and Farbod Farshidian. 2025. Diffuse-CLoC: Guided Diffusion for Physics-based Character Look-ahead Control. *arXiv preprint arXiv:2503.11801* (2025).
- Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. 2022. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991* (2022).
- Yanqin Jiang, Chaohui Yu, Chenjie Cao, Fan Wang, Weinming Hu, and Jin Gao. 2024. Animate3D: Animating Any 3D Model with Multi-view Video Diffusion. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 125879–125906. https://proceedings.neurips.cc/paper_files/paper/2024/file/e3b53f89136b1bc69a5714ea465f01b6-Paper-Conference.pdf
- Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. 2024. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 24174–24184.
- Chun-Mai Lai, Hsiang-Chun Wang, Ping-Chun Hsieh, Yu-Chiang Frank Wang, Min-Hung Chen, and Shao-Hua Sun. 2024. Diffusion-Reward Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 95456–95487. https://proceedings.neurips.cc/paper_files/paper/2024/file/ad47b1801557e4be37d30baf623de426-Paper-Conference.pdf
- Yoonsang Lee, Sungyun Kim, and Jehee Lee. 2010. Data-driven biped control. In *ACM SIGGRAPH 2010 papers*. 1–8.
- Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. 2024. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6517–6526.
- Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 300–309.
- Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. 2020. Character controllers using motion vae’s. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 40–1.
- Libin Liu and Jessica Hodgins. 2017. Learning to schedule control fragments for physics-based characters using deep q-learning. *ACM Transactions on Graphics (TOG)* 36, 3 (2017), 1–14.
- Libin Liu, KangKang Yin, Michiel van de Panne, and Baining Guo. 2012. Terrain runner: control, parameterization, composition, and planning for highly dynamic motions. *ACM Trans. Graph.* 31, 6 (2012), 154–1.
- Libin Liu, KangKang Yin, Michiel Van de Panne, Tianjia Shao, and Weiwei Xu. 2010. Sampling-based contact-rich motion control. In *ACM SIGGRAPH 2010 papers*. ACM New York, NY, USA, 1–10.
- Calvin Luo, Mandy He, Zilai Zeng, and Chen Sun. 2024. Text-Aware Diffusion for Policy Learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://openreview.net/forum?id=nK6OnCpd3n>
- Ian Mason, Sebastian Starke, and Taku Komura. 2022. Real-time style modelling of human locomotion via feature-wise transformations and local motion phases. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 5, 1 (2022), 1–18.
- Uldarico Muico, Yongjoon Lee, Jovan Popović, and Zoran Popović. 2009. Contact-aware nonlinear control of dynamic characters. In *ACM SIGGRAPH 2009 papers*. 1–9.
- Teng Pang, Bingzheng Wang, Guoqiang Wu, and Yilong Yin. 2025. DPR: Diffusion Preference-based Reward for Offline Reinforcement Learning. *arXiv e-prints*, Article arXiv:2503.01143 (March 2025), arXiv:2503.01143 pages. arXiv:2503.01143 [stat.ML] doi:10.48550/arXiv.2503.01143
- Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. 2019. Learning predict-and-simulate policies from unorganized human motion data. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–11.
- 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 (TOG)* 37, 4 (2018), 1–14.
- Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. 2017. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *Acm transactions on graphics (tog)* 36, 4 (2017), 1–13.
- 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 (ToG)* 40, 4 (2021), 1–20.
- Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).
- Marc H Raibert and Jessica K Hodgins. 1991. Animation of dynamic legged locomotion. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. 349–358.
- Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. 2021. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 11488–11499.
- Jiawei Ren, Mingyuan Zhang, Cunjun Yu, Xiao Ma, Liang Pan, and Ziwei Liu. 2023. InsActor: Instruction-driven Physics-based Characters. *NeurIPS* (2023).
- H. Sakoe and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, 1 (1978), 43–49. doi:10.1109/TASSP.1978.1163055
- John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438* (2015).
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- Agon Serifi, Ruben Grandia, Espen Knoop, Markus Gross, and Moritz Bächer. 2024. Robot motion diffusion model: Motion generation for robotic characters. In *SIGGRAPH asia 2024 conference papers*. 1–9.
- Dana Sharon and Michiel van de Panne. 2005. Synthesis of controllers for stylized planar biped walking. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2387–2392.
- Yi Shi, Jingbo Wang, Xuekun Jiang, Bingkun Lin, Bo Dai, and Xue Bin Peng. 2024. Interactive character control with auto-regressive motion diffusion models. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–14.
- Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems* 32 (2019).
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=PxTIG12RRHS>

- Sebastian Starke, Paul Starke, Nicky He, Taku Komura, and Yuting Ye. 2024. Categorical codebook matching for embodied character controllers. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–14.
- Richard S Sutton, Andrew G Barto, et al. 1998. *Reinforcement learning: An introduction*. Vol. 1. MIT press Cambridge.
- 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. *arXiv preprint arXiv:2410.03441* (2024).
- Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. 2023. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*. <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*. 1–10.
- Pascal Vincent. 2011. A connection between score matching and denoising autoencoders. *Neural computation* 23, 7 (2011), 1661–1674.
- Kevin Wampler, Zoran Popović, and Jovan Popović. 2014. Generalizing locomotion style to new animals with inverse optimal regression. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.
- Bingzheng Wang, Guoqiang Wu, Teng Pang, Yan Zhang, and Yilong Yin. 2024. DiffAIL: Diffusion Adversarial Imitation Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 14 (Mar. 2024), 15447–15455. doi:10.1609/aaai.v38i14.29470
- Jack M Wang, David J Fleet, and Aaron Hertzmann. 2009. Optimizing walking controllers. In *ACM SIGGRAPH Asia 2009 papers*. 1–8.
- Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. 2023. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems* 36 (2023), 8406–8441.
- Andrew Witkin and Michael Kass. 1988. Spacetime constraints. *ACM Siggraph Computer Graphics* 22, 4 (1988), 159–168.
- Jungdam Won, Jongho Park, Kwanyu Kim, and Jehee Lee. 2017. How to train your dragon: example-guided control of flapping flight. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–13.
- Runzhe Wu, Yiding Chen, Gokul Swamy, Kianté Brantley, and Wen Sun. 2025a. Diffusing States and Matching Scores: A New Framework for Imitation Learning. In *The Thirteenth International Conference on Learning Representations*. <https://openreview.net/forum?id=kWRKNDU6uN>
- Yan Wu, Korrawe Karunratanakul, Zhengyi Luo, and Siyu Tang. 2025b. UniPhys: Unified Planner and Controller with Diffusion for Flexible Physics-Based Character Control. *arXiv preprint arXiv:2504.12540* (2025).
- Michael Xu, Yi Shi, KangKang Yin, and Xue Bin Peng. 2025. Parc: Physics-based augmentation with reinforcement learning for character controllers. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 1–11.
- Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. 2022. Controlvae: Model-based learning of generative controllers for physics-based characters. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–16.
- Heyuan Yao, Zhenhua Song, Yuyang Zhou, Tenglong Ao, Baoquan Chen, and Libin Liu. 2024. Moconvq: Unified physics-based motion control via scalable discrete representations. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–21.
- KangKang Yin, Kevin Loken, and Michiel Van de Panne. 2007. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 105–es.
- Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédéric Durand, William T. Freeman, and Taesung Park. 2024. One-step Diffusion with Distribution Matching Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6613–6623.
- He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. 2018. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (ToG)* 37, 4 (2018), 1–11.
- Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2022. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001* (2022).
- Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5745–5753.
- Qingxu Zhu, He Zhang, Mengting Lan, and Lei Han. 2023. Neural categorical priors for physics-based character control. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–16.