PDP: Physics-Based Character Animation via Diffusion Policy

TAKARA TRUONG, Stanford University, USA
MICHAEL PISENO, Stanford University, USA
ZHAOMING XIE, Stanford University, USA
C. KAREN LIU, Stanford University, USA

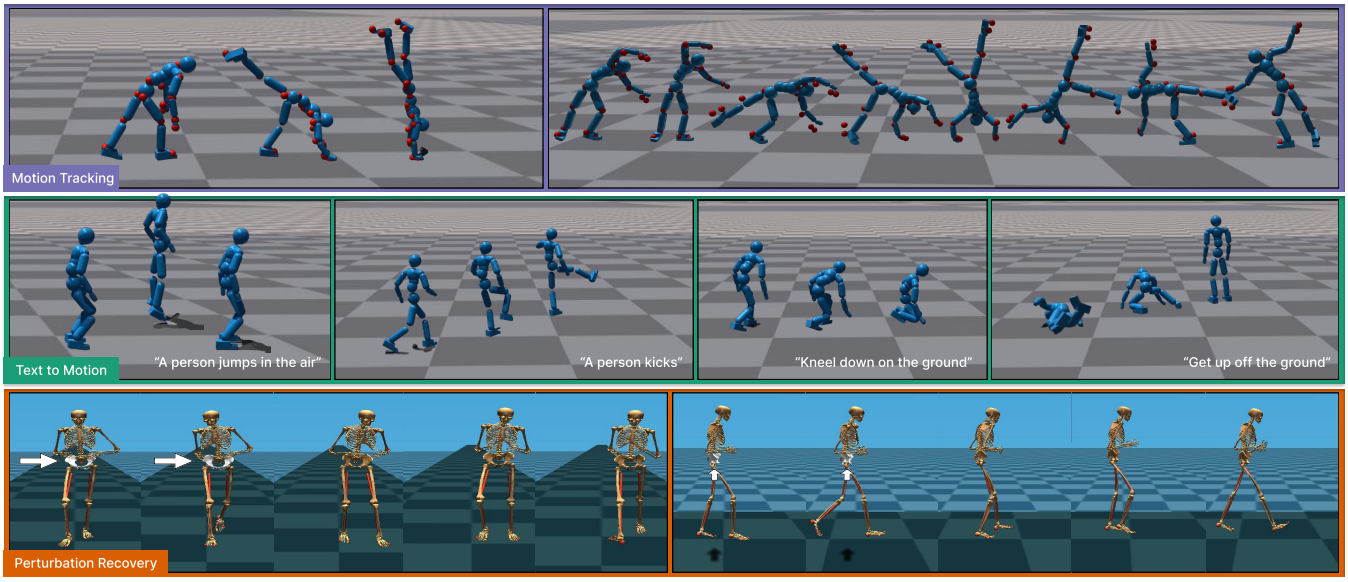


Fig. 1. PDP performs well across a diverse range of physics-based application domains. Top: Motion tracking. PDP is capable of tracking difficult and highly dynamic motions such as handstands and cartwheels. Middle: Text-to-motion. PDP is also capable of following user-provided text instructions. Bottom: Robustness to perturbations. PDP learns robust recovery strategies from random perturbations.

Generating diverse and realistic human motion that can physically interact with an environment remains a challenging research area in character animation. Meanwhile, diffusion-based methods, as proposed by the robotics community, have demonstrated the ability to capture highly diverse and multi-modal skills. However, naively training a diffusion policy often results in unstable motions for high-frequency, under-actuated control tasks like bipedal locomotion due to rapidly accumulating compounding errors, pushing the agent away from optimal training trajectories. The key idea lies in using RL policies not just for providing optimal trajectories but for providing corrective actions in sub-optimal states which gives the policy a chance to correct for errors caused by environmental stimulus, model errors, or numerical errors in simulation. Our method, Physics-Based Character Animation via Diffusion Policy (PDP), combines reinforcement learning (RL) and behavior cloning (BC) to create a robust diffusion policy for physics-based character animation. We demonstrate PDP on perturbation recovery, universal motion tracking, and physics-based text-to-motion synthesis.

Additional Key Words and Phrases: character animation, reinforcement learning, diffusion models

ACM Reference Format:
Takara Truong, Michael Piseno, Zhaoming Xie, and C. Karen Liu. 2024. PDP: Physics-Based Character Animation via Diffusion Policy. *ACM Trans. Graph.* 1, 1 (September 2024), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Developing a framework capable of generating diverse human movements that can traverse and interact with the environment is a crucial objective in character animation with broad applications in robotics, exoskeletons, virtual/augmented reality, and video games. Many of these applications demand not only a diverse range of human kinematic poses, but also the physical actions needed to achieve them. Previous works have demonstrated that physics-based control tasks can be formulated as a Markov Decision Process and solved through a Reinforcement Learning (RL) algorithm, or as a regression problem and solved using supervised learning techniques such as Behavior Cloning (BC). Despite the achievements observed in dynamic motor skills learning through both RL and BC methodologies,

Authors' addresses: Takara Truong, Stanford University, Stanford, Ca, 94305, USA, takaraet@stanford.edu; Michael Piseno, Stanford University, Stanford, CA, USA, mpiseno@stanford.edu; Zhaoming Xie, Stanford University, Stanford, CA, USA, zxieaa@gmail.com; C. Karen Liu, Stanford University, Stanford, CA, USA, karenliu@cs.stanford.edu.

ACM 0730-0301/2024/9-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

they encounter difficulties in effectively capturing the diversity and multi-model characteristics inherent in human motions.

To address this issue, recent works have explored various generative models. While Conditional Variational Autoencoders (C-VAEs) and Generative Adversarial Networks (GANs) have been used to capture humanoid skills, VAE-based models suffer from sensitive trade-off between diversity and robustness, while GAN-based methods often suffer from mode collapse without additional objectives [Dou et al. 2023]. Although diffusion models have been used to generate varied kinematic human motion [Tevet et al. 2023; Tseng et al. 2022], their application in high-frequency control domains is relatively unexplored. Recent work in robotics shows that BC combined with diffusion models can effectively learn diverse and multi-modal actions for real-world execution [Chi et al. 2023; Huang et al. 2024]. However, naively training a diffusion-based BC policy is ineffective for physics-based character animation due to compounding errors in high-frequency, under-actuated control tasks, exacerbating the domain shift problem. This issue is especially prominent in bipedal locomotion, where accumulated errors can quickly lead to falling. Can we combine the strengths of RL and diffusion-based BC policy, such that a physically simulated character can perform a diverse set of tasks robustly against distribution shifts due to disturbances in the environment, error in model prediction, or numerical errors in simulation?

We introduce PDP, a novel method that learns a robust diffusion policy for physics-based character animation, addressing the noted challenges. PDP leverages diffusion policies [Song and Ermon 2020] and large-scale motion datasets to learn diverse and multimodal motor skills through supervised learning and diffusion models. To overcome sensitivity to domain shifts, PDP uses expert RL policies to gather physically valid sequences of observations and actions. However, using RL for data collection alone does not resolve domain shift sensitivity.

Our key insight is that RL policies provide not only optimal trajectories but more importantly corrective actions from sub-optimal states. We employ a sampling strategy from robotics literature [Xie et al. 2020], collecting noisy-state clean-action paired trajectories to train the diffusion policy. We find that the choice of pairing noisy state with clean actions is a critical detail that contributes to producing a robust policy, outperforming the standard clean-state-clean-action trajectory collection and noisy-state-noisy-action sampling strategies for domain randomization. Additionally, we can now pool together data collected by small-task RL policies which can be efficiently trained, and leave the learning of diverse tasks on large-scale datasets to supervised learning.

PDP is a versatile method applicable to various motion synthesis tasks and agnostic to training datasets. We evaluate PDP on locomotion control under large physical perturbations, universal motion tracking, and physics-based text-to-motion synthesis, using different motion capture datasets for each application. Our model captures the multi-modality of human push recovery behavior, outperforming VAE-based methods and deterministic multi-layer perceptron networks on the Bump'em dataset [Werling et al. 2023]. It can also track 98.9% of all AMASS motions and [Mahmood et al. 2019] generate motion from textual descriptions [Guo et al. 2022]. Our contributions are as follows:

- We present a method of robust BC that scales to large motion datasets without the need for complex training architectures and can easily adapt to new skills.
- We analyze the effect of different sampling strategies for data augmentation on model performance.
- We introduce physics-based models that support locomotion control, motion tracking, and text-to-motion tasks.

2 RELATED WORK

2.1 Physics-Based Character Animation

In physics-based character animation, the central challenge is developing systems that learn a diverse range of realistic motions. Such motions manifest in motion tracking, motion generation, and task-oriented applications that require consideration of physical interactions.

Methods for learning individual or relatively similar motions, such as walking, running, or jumping are well-established [Peng et al. 2018, 2021]. However, these methods often do not suffice to complete tasks that involve multiple skills. Other methods have been successful in learning motion tracking policies capable of multiple skills using model-free RL [Bergamin et al. 2019; Park et al. 2019] and model-based RL [Fussell et al. 2021]. However, scaling to large diverse datasets is challenging. UniCon [Wang et al. 2020] introduced a motion tracking controller that scales to large and diverse motion datasets by using a novel constrained multi-objective reward function. Other works propose a mixture of experts [Won et al. 2020a], where different experts specialize in different skills. PHC [Luo et al. 2023] proposes an iterative approach to learning a large number of skills sequentially.

Another challenge in physics-based character animation is capturing diversity in motion data for use in downstream tasks. Human behaviors are multimodal, meaning a range of plausible behaviors can be employed in the same situation. A common method of capturing diversity in motions is to employ a Variational Autoencoder (VAE) to learn a latent space of skills, then sampling from the VAE prior to produce a wide range of motions [Merel et al. 2018; Won et al. 2022; Yao et al. 2022, 2023; Zhu et al. 2023]. These latent motion representations can then be used for downstream tasks such as motion generation [Luo et al. 2024] or object iterations [Merel et al. 2020]. Adversarial methods have also been proposed for capturing motion diversity [Dou et al. 2023; Peng et al. 2022], which combine a diversity reward and adversarial reward that encourage the policy to mimic the motion distribution.

Robustness issues also arise in behavior cloning methods where error accumulation can easily push the policy out of distribution. One method for improving policy robustness is to continually roll out the current policy, collecting on-policy data to train a student in the next learning iteration, as in DAgger [Ross et al. 2011]. Alternatively, robustness can be achieved by injecting perturbations into the state-action pairs in the training dataset, effectively expanding the distribution of states seen during training, similar to DASS [Xie et al. 2020].

2.2 Diffusion Models for Motion Synthesis and Robotics

Similar to VAEs, Diffusion models represent another category of generative AI and have exhibited success in the domain of kinematic motion synthesis, showcasing the capability of generating diverse and intricate human motion patterns [Tevet et al. 2023; Tseng et al. 2023]. Recently, Diffusion Policy [Chi et al. 2023] has effectively applied diffusion models to robotic manipulation tasks, human-robot collaborative endeavors [Ng et al. 2024], and tasks involving following language instructions [Zhang et al. 2022]. These models have primarily concentrated on high-level motion planning with a limited action space, such as forecasting the end-effector trajectory. While effective in low-frequency environments, the application of Diffusion Policy to high-frequency scenarios where minor inaccuracies in model predictions could result in failure, such as in physics-based character animation, remains relatively unexplored. Concurrent work, DiffuseLoco [Huang et al. 2024], is similar to PDP, employing a diffusion model to distill an offline dataset of multimodal skills, however they focus on simple locomotion gaits due to their policy being deployed on a real robot.

3 METHODS

Our method consists of three stages. First, we train a set of expert policies, each specialized in a small task but together completing a wide variety of motion tracking tasks in a physics simulator. Second, we generate state-action trajectories from the trained policies stochastically to build a dataset with noisy-state and clean-action trajectories. Lastly, we train a diffusion model via Behavior Cloning (BC) to obtain a single policy that can perform all tasks. Fig. 2 gives an overview of our system.

3.1 Expert Policy Training

We aim to obtain a control policy $\pi_{\text{PDP}} : \mathcal{O} \times \mathcal{T} \rightarrow \mathcal{A}$ to control a humanoid character, where $\mathcal{O}$ is the set of observations that describes the state of the character, $\mathcal{T}$ is the set of tasks, and $\mathcal{A}$ is the set of actions used to control the humanoid character. Such control policies can be trained via reinforcement learning. However, when the set $\mathcal{T}$ is large, it may be challenging to train a single policy to master all tasks, while it is relatively easy to train policies that specialize in a subset of tasks. We can divide the task set $\mathcal{T}$ into subset $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$, where $\bigcup_i \mathcal{T}_i = \mathcal{T}$, and train a expert policy $\pi_{\mathcal{T}_i}$ for each $\mathcal{T}_i$. The strategy for dividing the task is not critical, as long as it results in a set of policies that can generate desired state-action trajectories.

3.2 Stochastic Data Collection

In the second stage, we utilize the expert policies to generate a dataset for BC. For each task $\mathcal{T}_i$, we create a dataset $\mathcal{D}_{\mathcal{T}_i}$ by rolling out policy $\pi_{\mathcal{T}_i}$ and collecting trajectories. Specifically, we sample a motion task $\tau \in \mathcal{T}_i$, and run the policy to generate a sequence $\{\boldsymbol{o}_0, \boldsymbol{a}_0, \boldsymbol{o}_1, \boldsymbol{a}_1, \ldots, \boldsymbol{o}_N, \boldsymbol{a}_N\}$, where $\boldsymbol{a}_t = \pi_{\mathcal{T}_i}(\boldsymbol{o}_t, \tau) + \epsilon$ is a noisy version of optimal action proposed by the expert policy. The tuples $(\boldsymbol{o}_t, \tau, \pi_{\mathcal{T}_i}(\boldsymbol{o}_t, \tau))$ which correspond to the observation, task/goal information, and action are added to the dataset $\mathcal{D}_{\mathcal{T}_i}$. We repeat the data collection process until a maximum number of data points are collected, and use $\mathcal{D} = \bigcup \mathcal{D}_{\mathcal{T}_i}$ as the dataset for BC. Note that

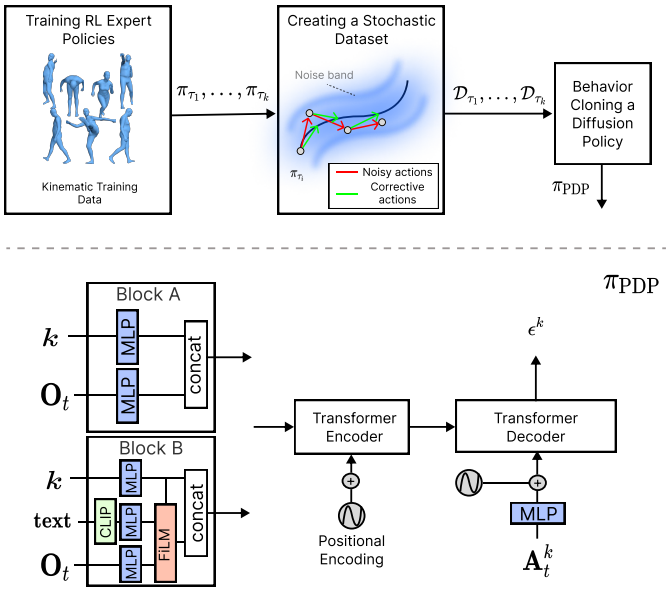


Fig. 2. PDP Overview. Top: First, we train expert RL policies $\pi_{\mathcal{T}_i}$ on tasks $\mathcal{T}_i$. We use $\pi_{\mathcal{T}_i}$ to create a dataset of noisy-state clean-actions. We then use BC to train a diffusion model. Bottom: Our model is a transformer encoder-decoder architecture. Block-B is used for text-conditioned applications, while other applications use Block A. Note that these applications are trained separately on their own distilled dataset.

the optimal action, not the noisy action a_t, is stored in $\mathcal{D}_{\mathcal{T}_i}$ and can be thought of as a corrective action from a noisy observation. This important detail, inspired by the DASS strategy proposed by [Xie et al. 2020], results in a training set that consists of sequences of noisy-state and clean-action pairs. This allows the collected data to cover a wider range of observation space compared to naively collecting clean optimal state-action trajectories, effectively creating a "noise band" around the clean trajectories. Our method extends DASS by further widening the noise band. Specifically, we generate short recovery episodes by initializing the character with a random root position and orientation offset from its original motion and allow it to recover to the original motion over several timesteps. This approach applies the noise band not only to the joints but also to the character's overall pose, helping to mitigate drift over time.

Another potential option for sampling is to collect noisy-state and noisy-action pairs, a common domain randomization practice in robotics to battle the sim-to-real gap. We found this randomization strategy produces less robust policies for character animation.

3.3 Behavior Cloning with Diffusion Policy

We parameterize our policy as a diffusion model. Here we give an overview of the diffusion model and our choice of architecture.

3.3.1 Diffusion model. We employ Diffusion Policy [Chi et al. 2023], which models the action distribution conditioned on the observations as a denoising process using a Denoising Diffusion Probabilistic Model (DDPM) [Ho et al. 2020]. Given a dataset of sequences

$\mathcal{D}$ collected using the method described previously, the denoising process is learned by a noise-prediction network $\epsilon_\theta(\boldsymbol{A}_t^k, \boldsymbol{O}_t, \boldsymbol{\tau}_t, k)$, where $\boldsymbol{A}_t^k$ is an action sequence sampled from $\mathcal{D}$ with added Gaussian noise and k is the diffusion step. The diffusion model is conditioned on $\boldsymbol{O}_t$, the corresponding observation sequence. τ is the necessary task or goal information, and θ is the set of learned model parameters. As in Diffusion Policy, $\boldsymbol{A}_t^k$ is a length T sequence of actions beginning at timestep t and $\boldsymbol{O}_t$ is a length T sequences of observations ending at timestep t. Sampling is then achieved through a denoising process known as Stochastic Langevin Dynamics [Welling and Teh 2011] starting from pure random noise.

$$\boldsymbol{A}_t^{k-1} = \alpha(\boldsymbol{A}_t^k - \gamma\epsilon_\theta(\boldsymbol{A}_t^k, \boldsymbol{O}_t, \boldsymbol{\tau}_t, k) + \mathcal{N}(0, \sigma^2\boldsymbol{I})), \tag{1}$$

where α, γ and σ are hyper-parameters of the denoising process. The noise-prediction model is learned in a self-supervised manner using the mean squared error objective

$$\mathcal{L} = MSE(\epsilon^k, \epsilon_\theta(\boldsymbol{A}_t^0 + \epsilon^k, \boldsymbol{O}_t, \boldsymbol{\tau}_t, k)), \tag{2}$$

where ϵ^k is the noise applied at to the action sequence at diffusion step k and $\boldsymbol{A}_t^0$ is the clean action sequence.

3.3.2 Model Architecture. Figure 2 depicts our model architecture. We adopt a similar architecture to the time-series diffusion transformer proposed by Diffusion Policy [Chi et al. 2023], with slight modifications depending on the application. For locomotion control and motion tracking, task information is included in the observation. For text-to-motion, the conditioning for the action sequence is computed as follows: a raw text prompt is encoded using the CLIP ViT-B/32 model [Radford et al. 2021] , then passed through an MLP text encoder. The observation $\boldsymbol{O}t$ is also fed through an MLP encoder. The diffusion step k is embedded into the same space and added to the text embedding. This result is fed through a Feature-wise Linear Modulation (FiLM) layer [Perez et al. 2017], which applies a learned element-wise scale and shift transformation to the embedding of $\boldsymbol{O}t$. Finally, the diffusion step embedding is concatenated with the FiLM layer result to produce our condition, serving as the input to the transformer encoder. This conditioning method is represented in Block B of Figure 2. The transformer decoder then takes an embedding of the noisy action sequence $\boldsymbol{A}_t^k$ along with the encoder result and predicts the noise applied to the action ϵ^k. For motion tracking tasks, we condition our transformer encoder using Block A of Figure 2, which is similar to Block B but without the CLIP text.

4 EXPERIMENTS

We demonstrate the generality and effectiveness of PDP by applying it to three distinctive applications using three different datasets: locomotion control under large physical perturbations using addbiomechanics dataset [Werling et al. 2023], universal motion tracking using AMASS [Mahmood et al. 2019], and physics-based text-to-motion synthesis using the KIT subset of AMASS and HumanML3D [Guo et al. 2022] for text labels. Experimental details for each application are described in this section, with results, comparisons, and ablation studies presented in the next section. Note that although each application uses the same model architecture in Figure 2 (except for Block A and B), they are trained separately with different distilled datasets.

4.1 Perturbation Recovery

The goal of the perturbation recovery task is to train a single diffusion policy that is capable of capturing the wide range of human responses to perturbations. Being able to model and simulate this behavior is important for studying human robustness [Jensen et al. 2023] and for designing better exo-skeleton or prosthetic systems [Hodossy and Farina 2023].

4.1.1 Dataset. We use the Bump'em dataset, a subset of the Addbiomechanics dataset [Werling et al. 2023], which consists of recovery motions of human participants being physically pushed while walking on a treadmill. Participants are perturbed in the same stance with varying forces and directions applied to the hip through a parallel tethered robot [Tan et al. 2020]. The recorded motions demonstrate that participants exhibit a diverse set of recovery strategies even under the same perturbation and initial stance, which makes this dataset particularly well suited for studying how well a model can capture the multi-modality of human behaviour.

For this task, trials from one participant were used. Perturbations were collected as the subject walked forward on a treadmill at a fixed speed, impacted at left toe-off stance in four directions (front, left, right, back) with magnitudes of 7.5% and 15% body weight, resulting in 16 total motions; two were dropped due to poor motion quality.

4.1.2 Experimental Details. We use a 25-joint skeletal model from Addbiomechanics [Werling et al. 2023], optimized for the specific participant's inertia and joint lengths. The environment is simulated with Mujoco and consists of the simulated treadmill and the skeletal model. In this environment, the observation space is defined in the world frame. The RL agent's observation consists of the bodies center of mass positions $\boldsymbol{x}^p \in \mathbb{R}^{3B}$ and linear velocities $\dot{\boldsymbol{x}}^p \in \mathbb{R}^{3B}$, as well as the bodies rotation $\boldsymbol{x}^r \in \mathbb{R}^{3\times3\times B}$ During RL training, the agent receives the same perturbation experienced by the human during the trial and optimizes tracking the human response through the collected reference motion following [Peng et al. 2018].

After training expert RL policies for each motion, we collect new observations for PDP, by including a binary signal $\boldsymbol{p}$ that indicates if the human is being perturbed, without detailing the force magnitude or direction. This signal is helpful for the diffusion policy to differentiate between normal walking and perturbation recovery; otherwise, the policy would react to non-existent perturbations. This addition is justified, as humans can discern disturbances directly. Furthermore, this inclusion provides no predictive advantage, similar to providing foot contact information. The full perturbation recovery observation can then be defined as a tuple $(\boldsymbol{x}^p, \boldsymbol{x}^r, \dot{\boldsymbol{x}}^p, \dot{\boldsymbol{x}}^r, \boldsymbol{p})$. 15 motions were sampled using the various noisy/clean state action strategies, and used to train PDP. Each model was trained on a single RTX 2080 Ti GPU for approximately 1.5 hours.

Sampling Strategy		Tracking Task					Perturbation Task	
State	**Action**	Success (%) ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{vel} ↓	E_{acc} ↓	Success ID (%) ↑	FPC ↓
Clean	Clean	68.8	57.1	33.1	8.77	5.55	3.36	-
Noisy	Noisy	64.5	61.7	41.4	16.3	17.6	59.5	7.94
Noisy*	**Clean***	93.5	49.9	31.6	8.25	5.55	100.0	2.77

Table 1. Performance with different sampling strategies. The Tracking task was trained on the KIT subset of the AMASS train set and evaluated on the AMASS test set. The Perturbation task was trained on the Bump'em dataset. *Indicates our method.

4.2 Universal Motion Tracking

The goal of the universal motion tracking task is to train a single diffusion policy capable of controlling the character to track any given reference motion under physics simulation.

4.2.1 Dataset. We use the AMASS dataset, and use the same train/test splits as PHC [Luo et al. 2023], where the sub-datasets: Transitions Mocap and SSM synced are used for testing and the rest are placed in the training set. Since our method is agnostic to the RL policies themselves, we utilize pre-trained motion tracking controller PHC [Luo et al. 2023] to track most of the motions and train individual policies for challenging motions where PHC fails [Peng et al. 2018]. With PHC and a few specialized small policies, our training set covers most motions in AMASS [Mahmood et al. 2019] and KIT [Plappert et al. 2016]. Following the same practice in PHC, we exclude motions that are infeasible in our physics simulators, such as leaning on tables.

4.2.2 Experimental Details. We use the humanoid model from [Luo et al. 2023], which follows the SMPL kinematic structure with $J = 23$ spherical joints. The reference motion includes the linear position $\boldsymbol{x}^p_{ref} \in \mathbb{R}^{3J}$ and 6d rotation $\boldsymbol{x}^r_{ref} \in \mathbb{R}^{6J}$ of each joint. The linear and angular velocities are found by finite difference $\dot{\boldsymbol{x}}^p_{ref} \in \mathbb{R}^{3J}$, $\dot{\boldsymbol{x}}^r_{ref} \in \mathbb{R}^{3J}$ respectively. The full motion tracking observation can then be defined as a tuple $(\Delta\boldsymbol{x}^p, \Delta\boldsymbol{x}^r, \Delta\dot{\boldsymbol{x}}^p, \Delta\dot{\boldsymbol{x}}^r, \boldsymbol{x}^p_{ref}, \boldsymbol{x}^r_{ref})$, where $\Delta\boldsymbol{x}^p = \boldsymbol{x}^p_{ref} - \boldsymbol{x}^p$ and the other Δ terms are similarly defined. All quantities are measured in the character frame, where the origin is placed at the root of the character, the x-axis aligns with the character's facing direction, and the z-axis points upward.

We train PDP with a history of 4 observations and 1 action prediction horizon. For ablation experiments using KIT, we train on 4 *NVIDIA A100* GPUs for approximately 24 hours. For experiments using AMASS, we train on 8 *NVIDIA V100* GPUs for about 70 hours.

4.3 Text-to-Motion

In the physics-based text-to-motion application, the goal is to train a diffusion policy to generate motions conditioned on a natural language text prompt.

4.3.1 Dataset. For training data, we use KIT dataset and the text annotations from HumanML3D [Guo et al. 2022]. The task vector is generated by passing the text annotation through the CLIP embedding [Radford et al. 2021]. We use the same pre-trained tracking controller, PHC, to obtain the observations and actions for training.

4.3.2 Experimental Details. The observation space used to train PDP with the text-to-motion task is different from the tracking task. We use the joint position $\boldsymbol{x}^p \in \mathbb{R}^{3J}$ and linear velocities $\dot{\boldsymbol{x}}^p \in \mathbb{R}^{3J}$, as well as the joint rotation $\boldsymbol{x}^r \in \mathbb{R}^{6J}$ and rotational velocities $\dot{\boldsymbol{x}}^r \in \mathbb{R}^{3J}$. All quantities are measured in the character frame as defined in the motion tracking task. We use a history of 4 observations and 12 action prediction horizon. We train on 4 *NVIDIA A100* GPUs for approximately 32 hours.

5 RESULTS

The experiments are designed to answer the following questions.

(1) Does the proposed sampling strategy, noisy-state-clean-action, outperform alternatives sampling strategies?
(2) For the application of perturbation recovery during locomotion, can PDP achieve both robust and diverse control policy?
(3) For universal motion tracking, how does PDP compare to the state-of-the-art RL policy and how important is it to use a generative model for this task?
(4) Can we train a physics-based text-to-motion policy using PDP?

5.1 Sampling Strategy

We examine the impact of sampling strategies on the tracking and perturbation tasks. For the tracking task, training was conducted using the KIT dataset, with evaluations performed on the test set employed in the AMASS-Test split. Table 1 records the quantitative results of different sampling strategies.

The clean-state clean-action approach has the lowest performance, with a success rate of 3.36% for perturbation and 68.8% for tracking. In comparison, the noisy-state noisy-action strategy significantly improved the perturbation task to 66.9%, but the tracking performance dropped to 64.5%. A possible explanation for the drop in performance in the tracking task but increased performance in the perturbation task is the scale of datasets. The clean-state clean-action strategy restricted the perturbation dataset to just 14 unique trajectories, limiting its diversity. In contrast, the tracking task included 3,626 examples, making it less dependent on additional data from noisy sampling. Thus, random sampling strategies were likely more beneficial for the perturbation task as they introduced necessary variability lacking in the original dataset.

Noisy-state clean-action strategy outperforms other strategies, achieving a perfect success rate of 100% in the perturbation task and 93.5% success rate in the tracking task, see Table 1. These results underscore the importance of selecting the right sampling strategy. By visiting out-of-distribution states and collecting the optimal action,

	AMASS-Train*					AMASS-Test*				
Method	Success (%) ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{vel} ↓	E_{acc} ↓	Success (%) ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{vel} ↓	E_{acc} ↓
MLP	98.8	37.3	26.5	4.6	3.0	97.8	47.3	30.9	8.0	5.9
PHC	98.9	37.5	26.9	4.9	3.3	96.4	47.4	30.9	9.1	6.8
PDP (Ours)	98.9	36.8	26.2	4.7	3.3	97.1	46.2	30.2	8.0	5.7

Table 2. Motion tracking results on AMASS train and test datasets.

this approach generates higher quality demonstrations, leading to better generalization and robustness.

5.2 Perturbation Recovery

We compare the performance of PDP to two other baselines, a C-VAE and an MLP. The selection of these approaches was based on their reliance on supervised learning principles. The C-VAE introduces an alternative generative model framework, whereas the MLP serves as a deterministic alternative. The C-VAE uses a similar setup as [Won et al. 2022] where state and next state are fed into an encoder to produce a latent code. A decoder takes a randomly sampled latent vector alongside the current state to produce the current action. Both the C-VAE and MLP follow the same architecture as PDP with minor algorithm-dependent adjustments such as excluding the diffusion timestep embedding.

5.2.1 Robustness. Robustness is measured by the successful completion of an episode, defined as the agent not falling within 6 seconds of the perturbation, a period that allows for several gait cycles to complete post-perturbation. Perturbations are categorized into In-Distribution (ID), which are the same as those in the training data, and Out-of-Distribution (OOD) perturbations, which determines the policy's capacity to effectively handle unforeseen perturbations by adjusting aspects such as timing, intensity, and direction of impact. We sample OOD perturbations as follows: We choose a random perturbation to cover all gait phases by randomly choosing an impact timing within [0, 2] seconds (equivalent to 2.5 gait cycles overlap), a random force magnitude between 7.5% and 15% of body weight which represents the extrema of the forces used in the Bump'em dataset, and a random force direction that is parallel to the ground.

Table 3 shows the ID and OOD performance of each baseline. All three models can handle ID perturbations, achieving a success rate of 100%. However, when faced with OOD perturbations, C-VAE and PDP methods exhibit notably higher performance, with C-VAE achieving a success rate of 91.3% and PDP achieving a success rate of 96.3%. Handling OOD impacts is challenging because the model may not have seen them before, especially the impact timing, as all training examples occur at the left toe-off. The OOD distribution performance results could be attributed to the multi-modal nature of the dataset, where using an MLP would result in policies that return the average of the response recorded in the dataset, causing the policy to fail, while C-VAE and PDP can synthesize a more tailored response from the multimodal distribution.

Two important hyper-parameters in our method are the choice of noise level for creating the stochastic dataset and the action prediction horizon. Note that the action prediction horizon refers to the number of actions being predicted during training. For inference, all policies re-plan after every step of the simulation. Table 4 shows the performance across different choices of noise level and horizon. Importantly, we see that 0 noise (equivalent to clean-state clean-action) performs extremely poorly with an ID success rate of just 3.36%. Adding noise increases robustness significantly before eventually slightly harming performance. For the action prediction horizon, we note that lower horizons yield better performance, with horizon 1 achieving the best ID and OOD success rates of 100.0% and 96.3%, respectively. Larger horizons see a dramatic drop in performance. We speculate that this is because actions closer to the current timestep are more important for stability, but our loss function does not weight this importance.

5.2.2 Foot Placement Correctness. Foot placement holds significant importance in understanding perturbation response and balance [Perry and Srinivasan 2017; Rebula et al. 2013]. A reliable model should ideally mirror the human response or closely approximate it while also accounting for multi-modality. Figure 4a illustrates both the actual foot placements of the participant and the distribution of foot placements obtained through a noisy sampling procedure. When examining foot placement correctness for impacts within the distribution, a lower variance in foot placement should be anticipated. This is gauged by assessing the variance in the L2 distance between each foot placement and the nearest ground truth data point. We design a metric based on this called foot placement correctness (FPC) to answer how spread apart the foot positions are from the policy compared to the closest ground truth position:

$$\text{FPC} = \frac{1}{N}\sum_{i=1}^{N}\left(\min_{j\in\{1,2,\ldots,M\}}\sqrt{(x_i-\bar{x}_j)^2+(y_i-\bar{y}_j)^2}\right) \tag{3}$$

where (x_i, y_i) refers to the foot contact position from a single policy rollout and $(\bar{x}_j, \bar{y}_j)$ refers to the ground truth foot contact positions.

PDP achieves a much lower FPC score, 2.79 compared to the top performing C-VAE model $\beta = .001$, 4.97 as shown in Table 3, signifying its proficiency in generating action state sequences that closely align with human responses. This is also demonstrated in Figure 4 where it is clear that PDP aligns more closely to the ground truth foot contact positions than the C-VAE.

Method	Beta Value	Success ID (%) ↑	Success OOD (%) ↑	FPC ↓
C-VAE	0.0001	98.1	91.0	5.09
	0.001	100.0	91.3	4.97
	0.01	100.0	71.0	4.26
	0.1	99.8	61.3	5.27
	1.0	99.8	59.0	5.28
MLP	-	100.0	81.0	-
PDP	-	100.0	96.3	2.79

Table 3. Baseline comparisons on the Bump-em Dataset. ID stands for in-distribution, OOD stands for out-of-distribution, and FPC stands for foot placement correctness.

Noise Level	Horizon	Success ID (%) ↑	Success OOD (%) ↑	FPC ↓
0.0	6	3.36	0.0	-
0.08	6	97.9	75.0	2.26
0.12	6	100.0	90.0	2.77
0.16	6	90.5	83.0	2.98
0.12	1	100.0	96.3	2.79
0.12	6	100.0	90.0	2.77
0.12	9	65.0	19.7	3.40
0.12	12	6.5	3.6	-

Table 4. Noise Level and Horizon Ablation for Bump'em Perturbation Task. FPC is measured on the left foot contact positions.

C-VAE also struggles to capture multi-modality effectively. This is exemplified in the right impact, where the C-VAE prioritizes modeling one mode while PDP is able to generate both modes. Figure 3 shows two responses by the diffusion policy for the rightwards impact. Figure 4 also displays the initial foot contact positions in response to the perturbation from both models. Balancing the reconstruction loss and the KL loss in C-VAE results in a significant trade-off in capturing the multi-modality and the variance in foot contact. Allowing the model to exhibit more variance enables better mode capture. Conversely, reducing the significance of the latent code leads to the model collapsing to a specific response with reduced variance. The comparison between Figure 4b and 4c may seem counter-intuitive in that increasing the β term decreases variability. Nevertheless, this issue of posterior collapse is linked to C-VAE and is further discussed in the Discussion section.

5.3 Motion Tracking

We demonstrate that PDP is capable of reliably tracking a significant portion of the motion in AMASS. Our method achieves a 96.4% success rate on the AMASS* test dataset, where failure is defined as in PHC [Luo et al. 2023]: an episode is considered a failure if at any point during evaluation the joints are, on average, more than 0.5 meters from the reference motion. In addition to the success rate, we also adopt metrics used by [Luo et al. 2023], specifically mean per-joint position error (E_{mpjpe}) and global mean per-joint position error ($E_{\text{g-mpjpe}}$), which assess the model's accuracy in matching the reference motion in local and global frames. We also measure the error in velocity (E_{vel}) and acceleration (E_{acc}) between the simulated character and the motion capture data. As shown in Table 2, PDP matches or outperforms PHC in all metrics.

Given the same architecture, we also compare with a regression model using MLP without diffusion, and find that MLP outperforms both PDP and PHC. This result is not entirely surprising because the benefit of generative models for the motion tracking application is not obvious in this task, as the action for tracking a particular reference pose from a particular state is not multi-modal but could be used as a motion prior.

5.4 Text-to-Motion

Figure 1 (middle) shows our results in the text-to-motion domain. We demonstrate that PDP is capable of following diverse text commands in natural language, such as jumping and kicking commands. Evaluating a model that employs auto-regressive inference during simulation with a limited history for composite actions presents significant challenges. Specifically, a prompt such as "walk then jump" cannot be effectively executed because the agent lacks the necessary memory of the initial action. To address this, we evaluate the model using a set of 42 action text prompts from the dataset, such as "a person dances" and "a person walks forward." Each prompt is considered successful if the model performs the specified action without falling.

Diffusion models excel in handling multi-modal distributions, which may not significantly benefit the motion tracking task. However, in the text-to-motion application, where capturing multi-modality is crucial, diffusion models (PDP) significantly outperforms MLP achieving a success rate of 57.1% compared to 11.9%, respectively.

6 DISCUSSION

Robust Locomotion Policies. Given the recent robotics community's interest in developing robust humanoid locomotion policies, our findings are particularly relevant. [Kaymak et al. 2023; Li et al. 2021; Singh et al. 2023]. While a single optimal strategy might suffice for a specific impact or perturbation, our results indicate that an MLP, which cannot capture different modes, lacks robustness to out-of-distribution (OOD) perturbations. In contrast, our diffusion model effectively stores a variety of strategies, providing it with a broader base of information to draw from when an OOD impact occurs. This capability could enhance locomotion policies' ability to handle diverse and unpredictable real-world perturbations.

C-VAE Posterior Collapse in Perturbation Recovery Task. Tuning the Beta value in C-VAE models presents significant challenges, primarily due to the posterior collapse problem. Increasing the β value forces the latent distribution to align more closely with a normal distribution and can cause the model to disregard the latent vector and rely solely on the observation, effectively reducing the model to function like an MLP. We find that diffusion models cover the distribution of initial foot contact positions more effectively while requiring less tuning, making this model a preferred choice.

PDP and MLP Tracking Task. Previous literature has shown the difficulty of producing a single and reliable motion tracker [Luo et al. 2023; Won et al. 2020b]. Our method can train a model directly through supervised learning and exceed the performance of more

complex hierarchical RL policies. Furthermore, This capability facilitates the creation of a pre-trained tracking controller that can be swiftly adapted to new datasets. This feature is notably distinct from conventional RL-based methods [Luo et al. 2023; Won et al. 2020a,b] which necessitates finetuning low-level controllers or training completely new ones. Additionally, as hierarchical systems accumulate new low-level controllers, the composer faces increasing complexity, often necessitating a complete system retrain to reduce the number of low-level controllers. In contrast, our method only requires standard finetuning procedures to integrate a new dataset after training the local experts, enabling more efficient scalability with additional motion datasets.

Text2Motion Challenges. Our system's capability extends to the text-to-motion task, demonstrating smooth transitions between text prompts despite a limited range of transitions.We hypothesize this effectiveness is partly due to the noisy sampling approach. By creating a band around each clean trajectory, we inadvertently increase the likelihood of intersecting with the state of another motion, facilitating more transitions.

However, our empirical observations indicate that text-to-motion does not perform at the same level as kinematic motion generation models. This discrepancy can likely be attributed to several factors. First, the model must balance performing the motion and maintaining equilibrium. When losing balance, it compensates with small corrective steps, disrupting the original motion. Secondly, while combining two distinct motions can be close in kinematic space, like superimposing root rotation onto a jump to create a jump-and-turn motion, achieving these motions may be significantly different in skill space. That is, the agent faces a challenge in determining how to manipulate the feet to rotate the root while simultaneously executing the jump.

Limitations. Despite its advantages, our approach has notable limitations. The primary constraint is the speed of the denoising process. Compared to the inference time of the MLP baseline, the diffusion model takes K times longer, where K is the number of denoising steps. This can make it challenging for applications that require high frequency control. Recent methods can reduce the number of inference steps required [Huang et al. 2024; Yin et al. 2023].

Another limitation lies in the trade-off inherent in diffusion-based policies when predicting over longer horizons. Although capturing multi-modality necessitates considering future actions, focusing on predicting multiple actions and weighting them equally can dilute the importance of the immediate action, thereby reducing robustness. This tension between long-horizon prediction for diversity and the accuracy of immediate actions is a crucial challenge. Future work could explore adaptive weighting schemes during training that balance effective long-horizon prediction with the precision of immediate actions.

7 CONCLUSION AND FUTURE WORK

We present a novel framework for physics-based character control that leverages the capability of diffusion models to capture diverse behaviors. Our proposed sampling strategy, noisy-state-clean-action, significantly outperforms alternative sampling strategies. For the perturbation recovery task, our method effectively captures the distribution of human responses and demonstrates robustness to both in-distribution and out-of-distribution perturbations. In universal motion tracking, our method surpasses the state-of-the-art performance, including our baseline using a non standard non-generative model. Additionally, we showcase our methods ability to synthesize motion conditioned on text. Future work could look into speeding up the inference by using methods like [Gu and Dao 2023; Yin et al. 2023], or leveraging the large pre-trained motion tracker for downstream RL tasks.

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)*.

Zhiyang Dou, Xuelin Chen, Qingnan Fan, Taku Komura, and Wenping Wang. 2023. C·ASE: Learning Conditional Adversarial Skill Embeddings for Physics-based Characters. *arXiv preprint arXiv:2309.11351* (2023).

Levi Fussell, Kevin Bergamin, and Daniel Holden. 2021. SuperTrack: motion tracking for physically simulated characters using supervised learning. *ACM Trans. Graph.* 40, 6, Article 197 (dec 2021), 13 pages. https://doi.org/10.1145/3478513.3480527

Albert Gu and Tri Dao. 2023. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752 [cs.LG]

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. *CoRR* abs/2006.11239 (2020). arXiv:2006.11239 https://arxiv.org/abs/2006.11239

Balint K Hodossy and Dario Farina. 2023. Shared Autonomy Locomotion Synthesis with a Virtual Powered Prosthetic Ankle. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 31 (2023), 4738–4748.

Xiaoyu Huang, Yufeng Chi, Ruofeng Wang, Zhongyu Li, Xue Bin Peng, Sophia Shao, Borivoje Nikolic, and Koushil Sreenath. 2024. DiffuseLoco: Real-Time Legged Locomotion Control with Diffusion from Offline Datasets. arXiv:2404.19264 [cs.RO]

Alexis Jensen, Thomas Chatagnon, Niloofar Khoshsiyar, Daniele Reda, Michiel Van De Panne, Charles Pontonnier, and Julien Pettré. 2023. Physical Simulation of Balance Recovery after a Push. In *Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction and Games* (<conf-loc>, <city>Rennes</city>, <country>France</country>, </conf-loc>) *(MIG '23)*. Association for Computing Machinery, New York, NY, USA, Article 23, 11 pages. https://doi.org/10.1145/3623264.3624448

Çağrı Kaymak, Ayşegül Uçar, and Cüneyt Güzeliş. 2023. Development of a new robust stable walking algorithm for a humanoid robot using deep reinforcement learning with multi-sensor data fusion. *Electronics* 12, 3 (2023), 568.

Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. 2021. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2811–2817.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. 2024. Universal Humanoid Motion Representations for Physics-Based Control. arXiv:2310.04582 [cs.CV] https://arxiv.org/abs/2310.04582

Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. 2023. Perpetual Humanoid Control for Real-time Simulated Avatars. arXiv:2305.06456 [cs.CV]

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture as Surface Shapes. In *International Conference on Computer Vision*. 5442–5451.

Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. 2018. Neural probabilistic motor primitives for humanoid control. *arXiv preprint arXiv:1811.11711* (2018).

Josh Merel, Saran Tunyasuvunakool, Arun Ahuja, Yuval Tassa, Leonard Hasenclever, Vu Pham, Tom Erez, Greg Wayne, and Nicolas Heess. 2020. Catch & Carry: Reusable Neural Controllers for Vision-Guided Whole-Body Tasks. arXiv:1911.06636 [cs.AI] https://arxiv.org/abs/1911.06636

Eley Ng, Ziang Liu, and Monroe Kennedy. 2024. Diffusion Co-Policy for Synergistic Human-Robot Collaborative Tasks. *IEEE Robotics and Automation Letters* 9, 1 (2024), 215–222. https://doi.org/10.1109/LRA.2023.3330663

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 Trans. Graph.* 37, 4, Article 143 (July 2018), 14 pages. 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 (TOG)* 41, 4 (2022), 1–17.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. *CoRR* abs/2104.02180 (2021). arXiv:2104.02180 https://arxiv.org/abs/2104.02180

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. 2017. FiLM: Visual Reasoning with a General Conditioning Layer. *CoRR* abs/1709.07871 (2017). arXiv:1709.07871 http://arxiv.org/abs/1709.07871

Jennifer A Perry and Manoj Srinivasan. 2017. Walking with wider steps changes foot placement control, increases kinematic variability and does not improve linear stability. *Royal Society open science* 4, 9 (2017), 160627.

Matthias Plappert, Christian Mandery, and Tamim Asfour. 2016. The KIT motion-language dataset. *Big data* 4, 4 (2016), 236–252.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. *CoRR* abs/2103.00020 (2021). arXiv:2103.00020 https://arxiv.org/abs/2103.00020

John R Rebula, Lauro V Ojeda, Peter G Adamczyk, and Arthur D Kuo. 2013. Measurement of foot placement and its variability with inertial sensors. *Gait & posture* 38, 4 (2013), 974–980.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 15)*, Geoffrey Gordon, David Dunson, and Miroslav Dudík (Eds.). PMLR, Fort Lauderdale, FL, USA, 627–635. https://proceedings.mlr.press/v15/ross11a.html

Rohan P Singh, Zhaoming Xie, Pierre Gergondet, and Fumio Kanehiro. 2023. Learning bipedal walking for humanoids with current feedback. *IEEE Access* (2023).

Yang Song and Stefano Ermon. 2020. Improved Techniques for Training Score-Based Generative Models. *CoRR* abs/2006.09011 (2020). arXiv:2006.09011 https://arxiv.org/abs/2006.09011

Guan Rong Tan, Michael Raitor, and Steven H. Collins. 2020. Bump'em: an Open-Source, Bump-Emulation System for Studying Human Balance and Gait. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 9093–9099. https://doi.org/10.1109/ICRA40945.2020.9197105

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

Jonathan Tseng, Rodrigo Castellon, and C. Karen Liu. 2022. EDGE: Editable Dance Generation From Music. arXiv:2211.10658 [cs.SD]

Jonathan Tseng, Rodrigo Castellon, and Karen Liu. 2023. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 448–458.

Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. 2020. UniCon: Universal Neural Controller For Physics-based Character Motion. arXiv:2011.15119 [cs.GR]

Max Welling and Yee W Teh. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 681–688.

Keenon Werling, Nicholas A Bianco, Michael Raitor, Jon Stingel, Jennifer L Hicks, Steven H Collins, Scott L Delp, and C Karen Liu. 2023. AddBiomechanics: Automating model scaling, inverse kinematics, and inverse dynamics from human motion data through sequential optimization. *Plos one* 18, 11 (2023), e0295152.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2020a. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 33–1.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2020b. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph.* 39, 4, Article 33 (aug 2020), 12 pages. https://doi.org/10.1145/3386569.3392381

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2022. Physics-based character controllers using conditional VAEs. *ACM Trans. Graph.* 41, 4, Article 96 (jul 2022), 12 pages. https://doi.org/10.1145/3528223.3530067

Zhaoming Xie, Patrick Clary, Jeremy Dao, Pedro Morais, Jonanthan Hurst, and Michiel Panne. 2020. Learning locomotion skills for cassie: Iterative design and sim-to-real. In *Conference on Robot Learning*. PMLR, 317–329.

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* 41, 6 (Nov. 2022), 1–16. https://doi.org/10.1145/3550454.3555434

Heyuan Yao, Zhenhua Song, Yuyang Zhou, Tenglong Ao, Baoquan Chen, and Libin Liu. 2023. MoConVQ: Unified Physics-Based Motion Control via Scalable Discrete Representations. *arXiv preprint arXiv:2310.10198* (2023).

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. 2023. One-step Diffusion with Distribution Matching Distillation. arXiv:2311.18828 [cs.CV]

Edwin Zhang, Yujie Lu, William Yang Wang, and Amy Zhang. 2022. Lad: Language augmented diffusion for reinforcement learning. In *Second Workshop on Language and Reinforcement Learning*.

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.

A HYPER-PARAMETERS

A.1 PDP

PDP Hyper-parameters	AMASS tasks	Bump-em task
Transformer Dim	512	256
Num Heads	8	8
Num Layers	10	8
Attention Dropout	0.3	0.3
Optimizer	AdamW	
Weight Decay	1e-3	
Learning Rate	1e-4	
Learning Rate Warm-up:	1000	
β-schedule	cosine	
Diffusion Steps	100	

A.2 Expert Tracking Policies

Tracking Policy Hyper-parameters	AMASS tasks	Bump-em task
Simulator	Isaac Lab	Mujoco
Optimizer	Adam	AdamW
Actor Learning Rate	2e-5	1e-5
Critic Learning Rate	2e-5	1e-4
Exploration noise	.055	.082
Num Envs	500	100
Horizon Length	32	50
Mini-epochs	6	10
Batch size	16000	1250
Actor Hidden Layers	[1024,512]	[1024, 1024]
Critic Hidden Layers	[1024,512]	[128, 128]
γ	.99	
λ	.95	
clip	.2	

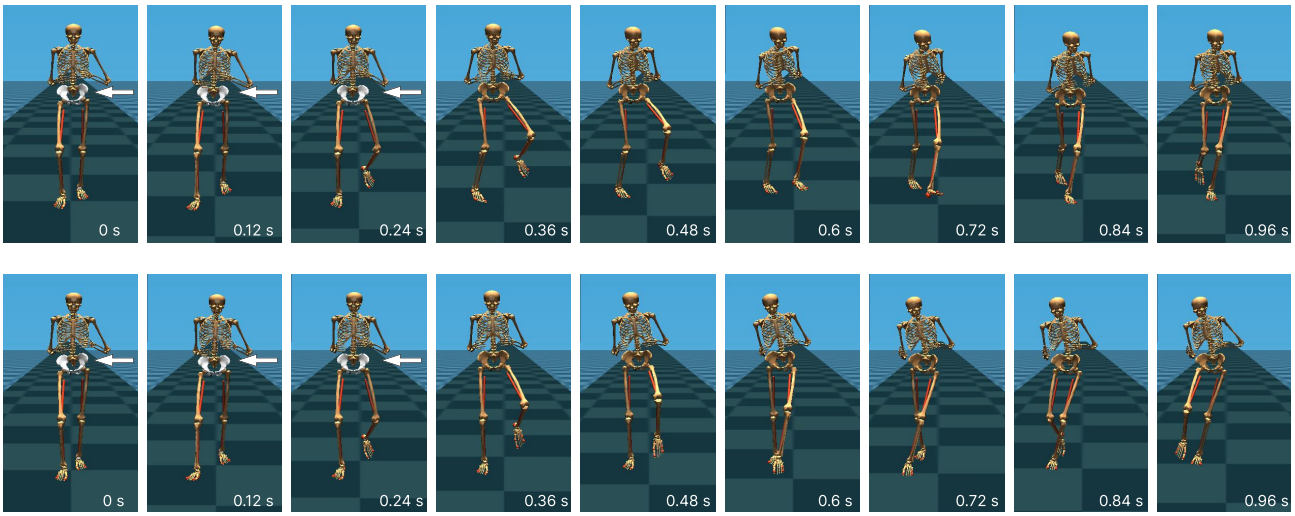
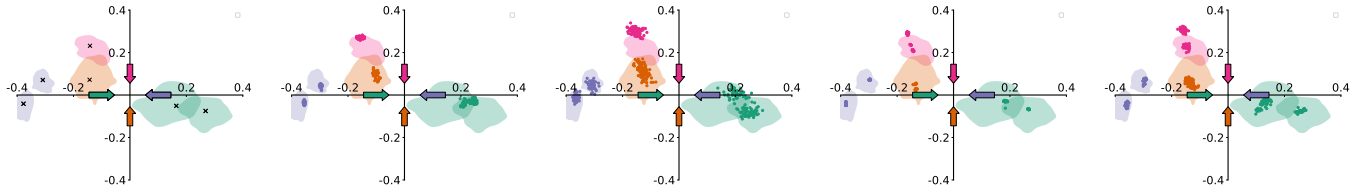


Fig. 3. PDP rollouts for a 15% body weight perturbation where the white pelvis and arrows indicate where the force is applied and the direction, respectively. Each row demonstrates a unique mode of recovery from the same perturbation.



(a) Ground truth distribution (b) VAE β 0.01 (c) VAE β 0.001 (d) PDP horizon 1 (e) PDP horizon 6

Fig. 4. Global left foot contact positions after 15% body weight perturbation in meters. +Y and +X align with the character's forward and right directions, respectively. The different colored arrows represent the directions that the force is applied on the person. The shaded areas represent foot contacts in the training distribution with noise level 0.12. The black X's represent the ground truth foot contacts of the human participant. All policies were trained on the stochastic dataset with noise level 0.12.