Robot Motion Diffusion Model: Motion Generation for Robotic Characters

AGON SERIFI, ETH Zürich, Switzerland and Disney Research, Switzerland
RUBEN GRANDIA, Disney Research, Switzerland
ESPEN KNOOP, Disney Research, Switzerland
MARKUS GROSS, ETH Zürich, Switzerland and Disney Research, Switzerland
MORITZ BÄCHER, Disney Research, Switzerland

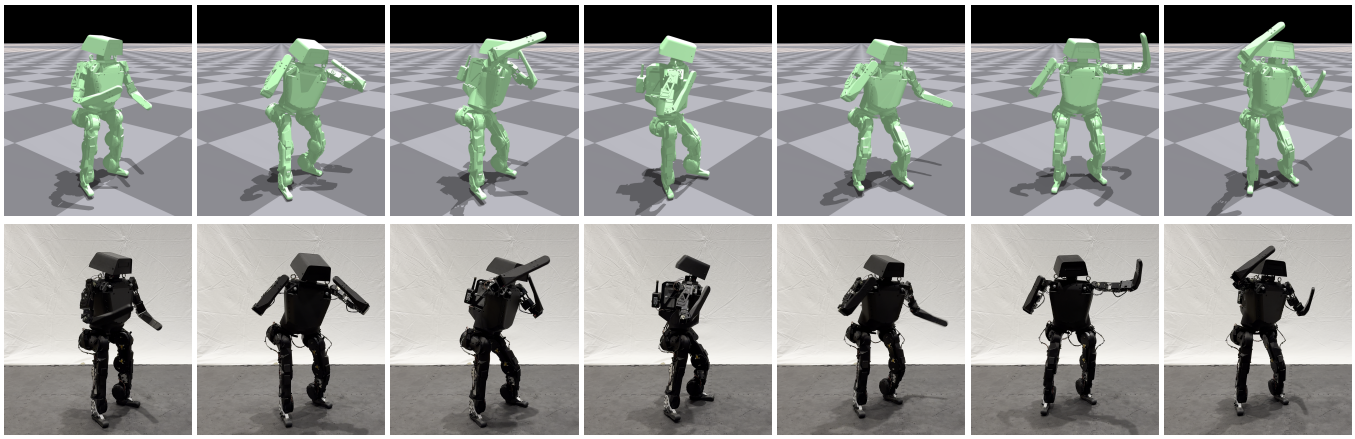


Fig. 1. Robot Motion Diffusion Model (RobotMDM) generates motions that are physics-aware and respect character limits. Our method enables the seamless integration of kinematic motion generators with physics-based character control and can be deployed on robots. The example shows a robot performing the prompt "a person who performed a right-handed uppercut."

Recent advancements in generative motion models have achieved remarkable results, enabling the synthesis of lifelike human motions from textual descriptions. These kinematic approaches, while visually appealing, often produce motions that fail to adhere to physical constraints, resulting in artifacts that impede real-world deployment. To address this issue, we introduce a novel method that integrates kinematic generative models with physics-based character control. Our approach begins by training a reward surrogate to predict the performance of the downstream non-differentiable control task, offering an efficient and differentiable loss function. This reward model is then employed to fine-tune a baseline generative model, ensuring that the generated motions are not only diverse but also physically plausible for real-world scenarios. The outcome of our processing is the Robot Motion Diffusion Model (RobotMDM), a text-conditioned kinematic diffusion model that interfaces with a reinforcement learning-based tracking controller. We demonstrate the effectiveness of this method on a challenging humanoid robot, confirming its practical utility and robustness in dynamic environments.

CCS Concepts: • **Computing methodologies** → **Reinforcement learning**; **Learning from demonstrations**; **Physical simulation**; *Animation*; • **Computer systems organization** → **Robotics**.

Additional Key Words and Phrases: physics-based characters, robotics, motion synthesis, motion control, reinforcement learning, animation

ACM Reference Format:
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 (SA Conference Papers '24), December 03–06, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3680528.3687626

Authors' Contact Information: Agon Serifi, ETH Zürich, Switzerland and Disney Research, Switzerland, agon.serifi@inf.ethz.ch; Ruben Grandia, Disney Research, Switzerland, ruben.grandia@disneyresearch.com; Espen Knoop, Disney Research, Switzerland, espen.knoop@disneyresearch.com; Markus Gross, ETH Zürich, Switzerland and Disney Research, Switzerland, grossm@inf.ethz.ch; Moritz Bächer, Disney Research, Switzerland, moritz.baecher@disneyresearch.com.

SA Conference Papers '24, December 03–06, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687626

1 Introduction

The automated generation of realistic motions based on high-level user input is a highly relevant task in physics-based character animation and robotics. Traditionally, computer animation has emphasized kinematic-based approaches, which are well-suited for animated film and video games where visual storytelling takes precedence. Recent advances in generative models have demonstrated the ability to synthesize diverse and visually appealing motions when trained

on large datasets [Tevet et al. 2023]. However, these kinematic-based generated motions do not strictly satisfy the constraints of a physics-based environment. As a result, the motions often contain artifacts such as floating, foot sliding, self-collisions, violations of joint limits, and dynamic imbalance, making it challenging to deploy these models in the real world. Although robust motion tracking controllers exist [Peng et al. 2018; Wang et al. 2020], the resulting motion is inherently limited by the quality of the provided target motion. We therefore identify the need to *align* the output of kinematic generative models with the downstream task of tracking these motions with a physics-based or robotic character.

Evaluating the performance of a controlled character on generated motions requires long-horizon simulations, which are computationally expensive and non-differentiable. Even if a differentiable simulation is available, the highly non-linear nature of the articulated rigid body system and the contact dynamics results in poorly behaved gradients [Hämäläinen et al. 2020; Suh et al. 2022]. Drawing inspiration from Reinforcement Learning from Human Feedback (RLHF) [Christiano et al. 2017], we propose to train a reward surrogate that predicts the expected performance of the downstream task. This provides a differentiable and computationally efficient loss function to fine-tune the generative model. During deployment, we interface the fine-tuned generative kinematic model with the existing tracking controller.

This processing contrasts the direct training of a generative controller [Juravsky et al. 2022], which typically results in a controller with a latent space that can be sampled. However, since these controllers are trained with Reinforcement Learning (RL), they are typically constrained to shallow Multilayer Perceptrons (MLPs) and do not scale well to large datasets. By decoupling the problem of generating motion from tracking motion, we can utilize more advanced networks and specialized training strategies. This approach, which combines strong kinematic motion generators with imitation-driven physics-based controllers, directly scales to larger datasets. In this work, we build on a text-conditioned diffusion-based approach [Tevet et al. 2023], although our fine-tuning strategy is applicable to generative models in general.

Succinctly, our contributions include:

- A fine-tuning method for generative kinematic motion models that uses a reward surrogate, offering a computationally efficient, differentiable estimate of the downstream task.
- A demonstration of RobotMDM, a text-conditioned kinematic diffusion model that interfaces with an RL-based tracking controller, deployed on a real-world robot.

2 Related Work

Kinematic Motion Synthesis. Motion generation has been a pivotal area of research within computer graphics, primarily focusing on synthesizing realistic and context-aware motion for animated characters. The underlying goal of motion synthesis is to learn a controllable latent manifold from which natural motions can be drawn. In recent years, many neural architectures and different motion representations have been investigated [Chandran et al. 2022; Harvey et al. 2020; Holden et al. 2017, 2016, 2015; Lee et al. 2018; Ling et al. 2020; Rempe et al. 2021; Starke et al. 2022, 2019, 2020]. This branch of work has been primarily used in animated character control, where the user provides simple control signals such as walking direction and velocity.

With the increased availability of unified large-scale motion capture datasets, such as AMASS [Mahmood et al. 2019], and comprehensive text and action annotations [Guo et al. 2022, 2020; Plappert et al. 2016; Punnakkal et al. 2021], progress has been made in generating expressive and diverse motions from more complex control signals. Guo et al. [2022] use an autoencoder combined with a recurrent neural network and text embeddings to generate motion sequences. In transformer-based extensions, a motion encoder and a text encoder are either trained jointly [Petrovich et al. 2022], or the motion latent space is aligned with a pre-trained language-image model such as CLIP [Radford et al. 2021]. This alignment exploits the rich semantic space of languages, enabling even the translation of cultural references into motions [Tevet et al. 2022]. More recently, T2M-GPT [Zhang et al. 2023] and MotionGPT [Jiang et al. 2024] formulate text-to-motion as a translation problem.

Diffusion models [Ho et al. 2020] have also been successfully adapted to the motion domain [Chen et al. 2023; Tevet et al. 2023; Zhang et al. 2024]. Beyond producing remarkably high-quality motion, these models inherit several key properties of diffusion models, such as supporting many-to-many generation and motion editing. Much research has leveraged Motion Diffusion Models (MDM [Tevet et al. 2023]) as foundational frameworks for motion synthesis. PriorMDM [Shafir et al. 2023] fine-tunes MDM to control the position of end effectors. Similarly, GMD [Karunratanakul et al. 2023] predicts a trajectory based on the given text prompt, which then guides the diffusion process. OmniControl [Xie et al. 2024] enables dense spatial control over any joints of the character. Extending this framework, DNO [Karunratanakul et al. 2024] introduces an optimization process where a differentiable objective is defined and used to optimize the input noise so that the resulting motion minimizes the objective. Although highly effective, this method depends on the differentiability of the objective. PhysDiff [Yuan et al. 2023] utilizes a pre-trained MDM and projects the motion onto a physically plausible state using a tracking controller and simulation. However, the evaluation of multiple simulations at runtime makes the method computationally expensive. Furthermore, the used control policy can apply non-physical residual forces to the character to preserve the semantic meaning of complex motions, aiming to remove visual artifacts such as ground penetration and floating. In contrast, we aim at incorporating physical understanding directly into the sampling process of MDM, and generate motions that are feasible without artificial external forces.

Physics-Based Character Control. Breakthroughs in deep reinforcement learning [Sutton and Barto 2018] have led to impressive imitation results that capture a single motion [Peng et al. 2018], or a handful of similar motions [Bergamin et al. 2019; Park et al. 2019; Wang et al. 2020], with small policy networks. To imitate large-scale and versatile motion datasets, additional mechanisms to stabilize the training process are required. A common approach uses a pool of expert policies, where each policy is responsible for a skill class [Won et al. 2020] or learns to deal with progressively more difficult motions [Luo et al. 2023]. Alternatively, pre-trained

motion embeddings [Serifi et al. 2024] may provide an additional information source during imitation.

Besides imitation, the field has also studied latent spaces for generative tasks. One goal is to reuse imitation policies in high-level tasks, where an additional policy learns to navigate the latent space so that the generated motion reaches a goal. This latent space is either learned jointly with the imitation objective [Dou et al. 2023; Feng et al. 2023; Gehring et al. 2023; Peng et al. 2022; Tessler et al. 2023; Won et al. 2022; Yao et al. 2022; Zhu et al. 2023], or is exploited in a post-processing step [Luo et al. 2024; Merel et al. 2018]. To generate motions from text, Juravsky et al. [2022] propose to align the motion latent space with the latent space of a text encoder, similar to the kinematic counterparts [Tevet et al. 2022]. Albeit promising, the combined learning of policy and text conditioning suffers from the sample inefficiency of reinforcement learning, which restricts scalability to datasets of a few minutes and limits versatility. More recently, high-level neural networks based on transformers [Yao et al. 2024] and diffusion models [Ren et al. 2023] have been trained to navigate the pre-trained latent space of low-level policies. Despite their ability to model complex language-to-motion relations, current methods lack an understanding and notion of feasibility, resulting in invalid states or unnatural transitions.

Marginalized Critics. Marginalized critics, which predict the expected return of an RL agent based solely on context rather than both context and current state, have been utilized to shape a training curriculum that oversamples underperforming scenarios [Won and Lee 2019; Xie et al. 2020]. In this work, we demonstrate that this critic formulation can also be applied outside the context of RL, serving as a loss function in a second learning problem. By leveraging the critic as a surrogate for the physical feasibility of generated motions, our method yields motion generators that better align with the physical character and control requirements.

3 Method

We assume the availability of a control policy, conditioned on a reference motion, and a generative model that produces kinematic motions. In this work, we train a VMP policy [Serifi et al. 2024], and a MDM generative model [Tevet et al. 2023] on our dataset. From there, our method consists of three parts (see Fig. 2): training a reward surrogate for the motion tracking task, aligning a generative model with this reward, and sequencing the generative model with the tracking controller during deployment.

Motion Representation. Motions of duration n are encoded with a $n \times (7 + 2j)$ matrix M, where j presents the number of joints. This matrix includes measurements for root height, root linear velocity (xy-plane), root angular velocity (about z-axis), root pose (3-dimensional), and joint positions and velocities. This representation is consistently applied across all stages of the method. Furthermore, motion data is normalized to the local heading frame of the character, where the x-axis aligns with the heading direction and the z-axis points upward. This normalization strategy decouples each frame from its absolute position and orientation in global coordinates, thereby facilitating a more efficient utilization of the data resources. Matrix m_t is a subset of rows from matrix M corresponding to either a single frame or motion window. If a motion is shorter, we pad the matrix with zero columns, restricting evaluations of loss or reward functions to the number of non-zero columns.

3.1 Critic Training

Conditional Reinforcement Learning. The goal in physics-based motion tracking is to translate kinematic motion inputs to physical actions in an environment. Using reinforcement learning [Sutton and Barto 2018], a policy is trained through interaction with a simulated environment, maximizing the expected return over a period of time. The policy is a probability function, $\pi(a_t|s_t, m_t)$, where a_t is the action taken, s_t is the observed state at time t, and m_t represents the kinematic motion input to the policy[1]. The environment reacts to the action by transitioning to the next state, s_{t+1}, and providing a scalar reward $r_t = r(s_t, a_t, s_{t+1}, m_t)$. The reward reflects how accurately the resulting physical motion tracks the kinematic input. See [Serifi et al. 2024] for a detailed specification of the reward and termination conditions.

During training, we initialize an episode by randomly choosing a motion and a starting frame from the dataset. We then shift by one frame within the same motion clip to retrieve the next reference. We continue this process until we reach the end of a clip, randomly jumping to a new clip if the episode has not terminated yet. Additionally, we use domain randomization to increase the robustness of the policy and randomize rigid body masses and friction coefficients to avoid overfitting to a single set of simulation parameters, introducing random disturbance forces also. To further reduce the sim-to-real gap, we add actuator models [Grandia et al. 2024] to the simulator.

Critic Training. After training, the parameters of the actor are frozen and the same environment is used to learn a function that predicts the performance of the actor given a motion reference. Concretely, we aim to estimate the expected discounted cumulative reward given the current motion reference,

$$v(m) = \mathbb{E}_{\substack{s_{0:\infty} \\ a_{0:\infty} \\ m_{1:\infty}}} \left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| m_0 = m, \pi \right], \tag{1}$$

where the expectation is evaluated over state-action trajectories and future motion references, and $\gamma \in [0, 1]$ is the discount factor. Variable r_t is the reward at time t, for which we choose the same reward as during RL. In principle, the reward function could be altered at this stage. The estimate (1) is closely related to the value function used during RL,

$$v^{\mathrm{RL}}(s, m) = \mathbb{E}_{\substack{s_{1:\infty} \\ a_{0:\infty} \\ m_{1:\infty}}} \left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s, m_0 = m, \pi \right]. \tag{2}$$

However, the RL value function has access to the current state of the character while $v(m)$ does not. Our reward surrogate can, therefore, be understood as the *averaged* value function over the distribution of states, thus establishing a differentiable link between kinematic motion and expected reward. Due to this similarity, we refer to the reward surrogate as *critic*.

[1]The policy used in this work uses a latent representation of the provided reference motion. This mapping from motion to latent space is considered part of the policy $\pi(a_t|s_t, m_t)$ itself.

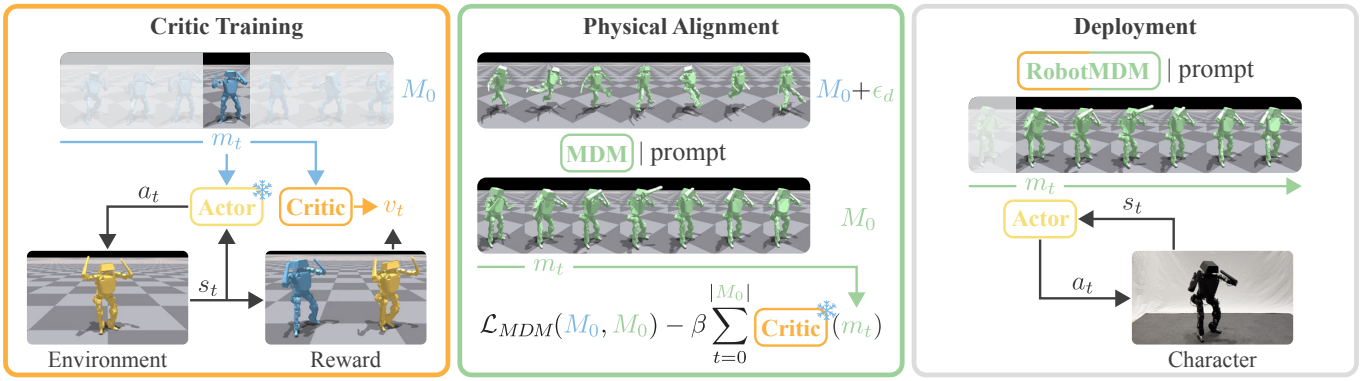


Fig. 2. **Overview.** RobotMDM leverages a pre-trained imitation policy (Actor) and a pre-trained Motion Diffusion Model (MDM) in a two-stage process. In the first stage (Critic Training), a Critic is trained using motions from the dataset to evaluate the Actor's performance, creating a differentiable surrogate for expected future rewards conditioned on motion input, and linking kinematic inputs to physical feasibility. In the second stage (Physical Alignment), the learned Critic is used to fine-tune the MDM, aligning it with the character's limits and ensuring physical feasibility. The final result is RobotMDM, a method capable of generating physics-aware motions, suitable for deployment on real-world systems (Deployment).

Observing that the proposed critic is an RL critic with partial observations, we apply standard value function estimation algorithms to train a network, $v^\theta(m)$, that directly approximates Eq. (1). We use the approach from PPO [Schulman et al. 2017], which estimates a value function target using truncated Generalized Advantage Estimation (GAE, [Schulman et al. 2016]), corresponding to a truncated TD(λ) estimate [Sutton and Barto 2018]. Given a finite roll-out of the current policy of length T, and a current set of parameters θ, an updated value function estimate, $\hat{v}_t$, is computed as

$$\hat{v}_t = v_t^\theta + \sum_{t'=t}^{T-1} (\gamma\lambda)^{(t'-t)}\, \delta_{t'}, \tag{3}$$

where $\delta_{t'}$ is the TD error at time t', given by

$$\delta_t = r_t + \gamma v_{t+1}^\theta - v_t^\theta. \tag{4}$$

With the collected batch, the critic's parameters are updated according to a square loss function

$$\min_\theta \sum \|\hat{v}_t - v_t^\theta\|_2^2. \tag{5}$$

The training process is outlined in Alg. 1.

Algorithm 1: Critic Training

Input: π: control policy; $\mathcal{D}$: set of target motions m

$v^\theta \leftarrow$ init MLP with θ parameters
$\mathcal{B} \leftarrow \emptyset$ init replay buffer
/* collecting trajectories */
while $\mathcal{B}$ *not full* **do**
  $m \leftarrow$ sample motion window from $\mathcal{D}$
  $s_0 \leftarrow$ set character to random pose
  $\tau \leftarrow \emptyset$ init empty trajectory
  for $t = 0, \ldots, T$ **do**
    simulate one step using $a_t \sim \pi(a_t|s_t, m_t)$
    $r_t \leftarrow$ compute reward
    record (m_t, r_t) in τ
  end
  store τ in $\mathcal{B}$
end
/* critic updates */
for *each* τ *in* $\mathcal{B}$ **do** ▷ batch processing
  for *each* $(m_t, r_t)_{t=0}^T$ *in* τ **do**
    $\hat{v}_t \leftarrow$ compute value function estimate ▷ (3), (4)
  end
  $\theta \leftarrow \theta - \eta_c \nabla_\theta \left(\sum_{t=0}^{T-1} (\hat{v}_t - v_t^\theta(m_t))^2 \right)$ ▷ (5)
end

3.2 Physics-Aligned Generative Model

Training the generative model consists of a kinematic pre-training step, followed by fine-tuning. Before discussing our fine-tuning, we briefly recap the training of the generative model, which is a text-conditioned diffusion model in our case.

Denoising Diffusion Probabilistic Model. The diffusion process begins with a clean motion sequence, denoted as M_0, and progressively adds noise, resulting in a noisy sequence M_d at each step d. This process can be mathematically expressed as $q(M_d|M_0) = \mathcal{N}(M_d; \sqrt{\alpha_d}M_0, (1-\alpha_d)I)$, with α_d representing a noise schedule that determines the intensity of the added noise [Ho et al. 2020]. Essentially, the diffusion process creates a process from clean motion to increasingly distorted motion. The objective of the motion diffusion model is to learn the reverse process: how to denoise a sequence and gradually reconstruct the clean motion from noisy states. This is done by training the model to predict the clean motion M_0 using a parameterized function $p^\phi(M_d, d, c)$,

$$\mathcal{L}_{MDM} = \|M_0 - p^\phi(M_d, d, c)\|_2^2, \tag{6}$$

where c represents additional conditions like text prompts or other contextual information. By providing these conditions, the model can generate specific types of motions. This loss is minimized on randomly sampled motion-context pairs and random diffusion steps d. During inference, a random noise motion is sampled from a

standard Gaussian distribution $M_D \sim \mathcal{N}(0, I)$, and D diffusion steps are applied to generate a clean motion M_0. Note that this process is not aware of physical properties that would be needed for true-to-life motion simulation.

RobotMDM. Given a pre-trained motion diffusion model and a critic, we propose to use the critic as an additional loss to fine-tune the diffusion model. We therefore generate a motion $M = p^{\phi}(M_d, d, c)$ and use the critic, with frozen parameters, to evaluate the expected performance for that motion. We maintain the standard MDM loss functions to ensure the model generates motions according to the data distribution and textual conditioning. However, we now also use the negative sum of critic values to indicate feasibility

$$\mathcal{L}_{RobotMDM} = \mathcal{L}_{MDM} - \beta \sum_{t=0}^{|M|} v^{\theta}(m_t), \tag{7}$$

where we sum over all the motion windows m_t contained in the generated motion. With this loss function, the MDM is trained to shape motions into more realistic examples without losing contextual accuracy, achieving higher critic values, which indicate that the policy can track the motion more accurately.

4 Evaluation and Results

Character. We evaluate our method on a bipedal robot with 20 degrees of freedom. The robot stands 0.84 m tall and has a mass of 16.2 kg. In simulation, we operate on a torque-controlled system [Grandia et al. 2024]; the policy outputs actuator positions that serve as inputs for the proportional-derivative (PD) controllers at each joint. We built a physical replication of the robot where the two legs, each with 5 DoFs, are equipped with Unitree A1 actuators, while its neck and arms use Dynamixel XH540-V150-R actuators. We estimate the robot's state by using input from an onboard IMU and a motion capture setup.

Data. In our experiments, we use the textually-annotated AMASS subset [Mahmood et al. 2019] of the HumanML3D dataset [Guo et al. 2022]. This dataset is a collection of human mocap data. To retarget the motions to our robotic character, we use the inverse kinematic formulation by Schumacher et al. [2021]. After removing motions shorter than two seconds and mirroring them, we end up with 27112 motions annotated with 70958 textual descriptions and a total length of ~55 h. We use the same train-test split as HumanML3D. Note that after retargeting, the dataset necessarily introduces artifacts due to the mismatch in topology and degrees of freedom between the SMPL body model [Loper et al. 2015] and our character.

Training Details. Tab. 1 provides a summary of the most important training parameters used for critic training and our physical alignment. For the pre-trained actor, we use parameters reported in [Serifi et al. 2024]. During critic training, we use a fixed learning rate η_c and reduce γ to 0.9 to focus on short- and medium-term rewards. The critic is a small MLP with 3 hidden layers of size 256. As the generative backbone, we train MDM [Tevet et al. 2023] on the retargeted dataset for 3 million steps. To stabilize the training, we apply Exponential Moving Averaging (EMA, [Kingma and Ba 2014]) over the model weights, following the implementation of Nichol et al. [2021]. The original MDM used 1000 diffusion steps (referred to as MDM-1K). With EMA, comparable results can be achieved in just 50 diffusion steps (referred to as MDM) [Karunratanakul et al. 2024]. We use a single EMA rate of 0.9999. The model parameters and loss function remain as reported in [Tevet et al. 2023]. RobotMDM is fine-tuned for an additional 400k steps, equivalent to 12 hours of training, using objective (7) and learning rate η_f.

Table 1. Training Parameters.

Critic Training		**Physical Alignment**	
Parameter	Value	Parameter	Value
Batch size	8192×32	Batch size	256
Layers	3×256	EMA rate	0.9999
η_c	$3 \cdot 10^{-4}$	η_f	0.003
γ	0.9	Fine-tuning steps	400k
λ	0.95	β	0.001
		D	50

4.1 Kinematic Motion Generation

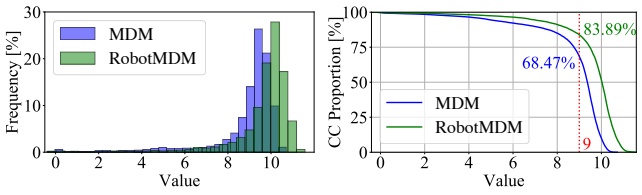
We first evaluate the motion generation capability of the aligned RobotMDM, compared to the baseline MDM and PhysDiff. The results are summarized in Tab. 2, where the first row evaluates the performance metrics on the dataset itself. For PhysDiff, we perform a single projection step. We note that in the original PhysDiff method, the controller used during projection applies external forces to the root of the character. Given the goal of deploying the motion on a real robot, we use a controller without external forces in our evaluation of PhysDiff. To evaluate the motion quality and diversity, different metrics were proposed in previous work [Guo et al. 2022, 2020]: the *Frèchet Inception Distance* (FID [Heusel et al. 2017]) measures the disparity between the feature distribution of the dataset and generated motions by utilizing an inception network. *R-Precision* compares the ground truth text description and 31 random text descriptions by measuring the Euclidean distance between the text embedding and generated motion embedding. Top-3 accuracy is reported. *MultiModal dist.* evaluates mode coverage by measuring the Euclidean distance between motion features and text features. To evaluate *Diversity*, we compare the variance between generated motions and original motions and rank the methods based on closeness to the dataset variance. The *MultiModality* measures the diversity (variance) of generated motions, conditioned on the same text prompt. Finally, we propose the *Realism* score, which reports the accumulated value estimated by the critic network, $\sum_{t=0}^{|M|} v^{\theta}(m_t)$, which can be taken to indicate the feasibility of the motion.

RobotMDM significantly improves the Realism score while preserving similar levels of performance across other metrics compared to MDM. Note that the dataset itself has a much lower Realism score, primarily due to noise and retargeting artifacts arising from the discrepancy between our character and the human skeleton. Compared to the dataset, motions generated by MDM are smoother and have fewer artifacts, which consequently leads to higher Realism scores. Although PhysDiff enhances the Realism of generated motions, the use of a projection using the simulated control policy results in

Table 2. **Kinematic Motion Generation.** Comparative evaluation of various kinematic motion generation methods across multiple metrics for quality, diversity, and feasibility. **Best** and second best (excluding the dataset itself). ± indicates the 95% confidence interval.

Method	R-Precision, top 3 ↑	FID ↓	MultiModal Dist ↓	Diversity →	Multi-modality ↑	Realism ↑
Dataset	$0.696^{\pm.003}$	$0.002^{\pm.000}$	$3.799^{\pm.014}$	$8.958^{\pm.102}$	-	$6.774^{\pm.002}$
MDM-1K	$0.675^{\pm.013}$	$0.688^{\pm.090}$	$3.840^{\pm.039}$	$\mathbf{8.952^{\pm.060}}$	$\mathbf{2.355^{\pm.148}}$	$8.392^{\pm.036}$
MDM	$0.680^{\pm.008}$	$\mathbf{0.415^{\pm.045}}$	$\mathbf{3.831^{\pm.028}}$	$9.074^{\pm.135}$	$2.068^{\pm.067}$	$8.730^{\pm.018}$
PhysDiff (1-step)	$0.482^{\pm.007}$	$10.401^{\pm.089}$	$5.500^{\pm.025}$	$6.546^{\pm.037}$	$1.890^{\pm.113}$	$8.951^{\pm.034}$
RobotMDM (ours)	$\mathbf{0.684^{\pm.007}}$	$0.472^{\pm.023}$	$3.835^{\pm.020}$	$9.170^{\pm.064}$	$2.087^{\pm.101}$	$\mathbf{9.562^{\pm.017}}$

↑: higher is better; ↓: lower is better; →: closer to dataset is better.



(a) Realism Distribution. (b) CCDF.

Fig. 3. Comparison of MDM and RobotMDM methods for 10000 randomly-generated motions. **(a)** Distribution of Realism scores. RobotMDM shows a shift towards higher values, indicating improvements in feasibility. **(b)** Complementary Cumulative Distribution Function of the motion Realism values shown in (a). RobotMDM motions demonstrate significantly higher values. Anecdotally, values above 9.0 correspond to well-tracked motions.

a decreased performance in other metrics. While PhysDiff effectively eliminates ground penetrations, it heavily depends on the controller's tracking accuracy. We hypothesize that the absence of external helper forces in our control policy necessitates large projection steps and reduces the effectiveness of this projection strategy, ultimately leading to less versatility. Rather than projecting the motions, RobotMDM learns a strategy to circumvent infeasible motions, thereby sustaining quality and diversity. Additionally, in contrast to PhysDiff, RobotMDM does not add any extra computational overhead to MDM during motion generation.

4.2 Physical Alignment

In the following, we compare generated motions, with a focus on feasibility. The dataset includes motions involving object interactions, such as sitting on a chair, which are generally not feasible without the presence of the object. Additionally, the dataset was collected from human subjects, and kinematically retargeting these motions to our character does not guarantee that the character can perform them. Therefore, simply learning the motion distribution from the dataset is insufficient for achieving physical realism.

Fig. 4 shows two examples where RobotMDM refines the motions to enhance realism. The first example depicts a leg kick where the MDM-generated motion is imbalanced and features an excessively strong high kick. Given the character's inability to capture such dynamic motions, our method moderates the strength and stabilizes the movement, making it more realistic. The second example in Fig. 4 demonstrates a motion involving sitting. In MDM, the character

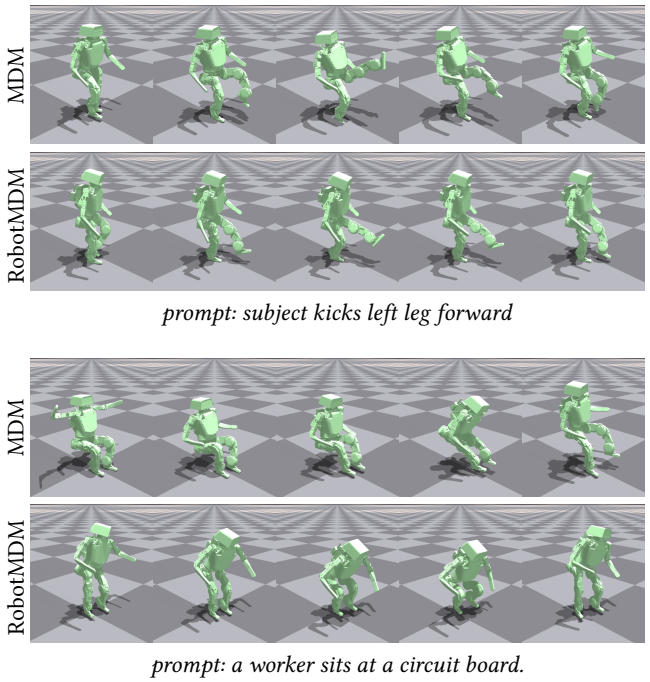
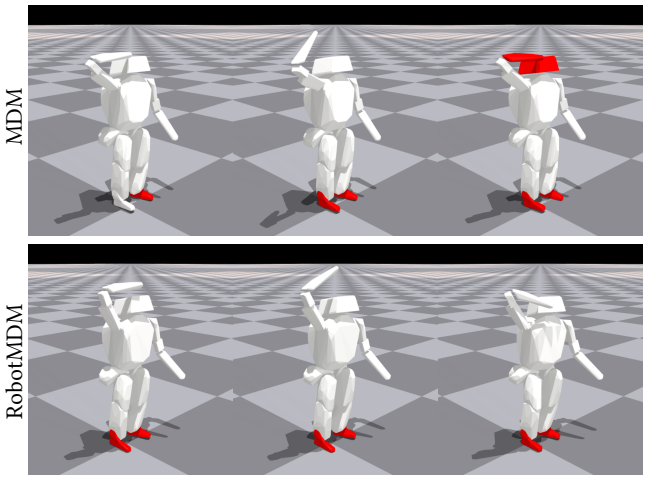


Fig. 4. **Realistic Motion Generation.** Aligning the motion diffusion model with physical knowledge results in more realistic motions within the character's limits while preserving the context. This results in a less extreme kick where the character also remains more balanced, or a sitting motion that is feasible in the absence of a chair.

appears to sit in mid-air, as similar motions are present in the dataset. Our method recognizes the infeasibility of this action and adjusts accordingly, resulting in a version where the character squats down instead of leaning back.

Another issue, resulting from the retargeting process, is that the dataset contains motions with self-collisions. MDM therefore produces motions where body parts intersect. Such intersections lead to a lower reward, as the policy will not be able to track them accurately. Consequently, the fine-tuned RobotMDM avoids these intersections. Fig. 5 visualizes the character's collision bodies during a waving motion, where bodies are colored red during collisions (e.g. when the feet contact the floor). MDM results in a collision



prompt: a figure raises their right hand in a sweeping motion

Fig. 5. **Collision Avoidance.** Collisions between bodies results in a lower reward, because they are not accurately tracked by the policy. The aligned RobotMDM naturally circumvents collisions.

between the head and arm, whereas RobotMDM successfully avoids such issues while preserving the expressiveness of the motion.

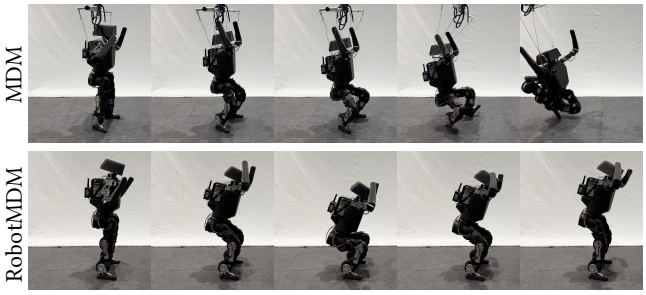
4.3 Physics-Based Motion Tracking

Next, we assess the tracking performance of the control policy. A total of 10000 motions, each 10 s long, are generated for each method based on test prompts, resulting in 30 hours of motion. We evaluate the performance using 2048 simulation episodes, each lasting 30 s. During these episodes, the policy attempts to imitate the target motions, starting from a randomly selected frame. If the selected motion ends or the character is terminated, a new motion is sampled. We summarize tracking performance results in Tab. 3. Poses generated by RobotMDM are tracked with greater accuracy, particularly the lower body pose. Motions generated by MDM are more frequently infeasible, often featuring overly expressive leg movements. Additionally, root rotation errors are nearly halved as the motions created by RobotMDM are more balanced, requiring smaller corrections to the root pose. Furthermore, both the linear and angular velocities are more closely aligned with what is feasible on the robot, enhancing the overall realism and functionality of the generated motions. Figs. 1 and 7 show the tracking of expressive motions on the real robot, and additional results are provided in the supplementary video.

The ability to generate motions can also be leveraged to enhance the motion tracking policy. While the original tracking policy was trained on the retargeted dataset, we retrain the policy based on the generated motions from MDM and RobotMDM, resulting in a tailored tracking policy for each motion generator. We find that even after specializing the policy on the motions generated by MDM, the generated motions remain hard to execute stably, confirming that the motions are indeed infeasible. Fig. 6 presents an example of the post-training performance: The character performs a lifting motion. RobotMDM-generated motions for this prompt are feasible, while

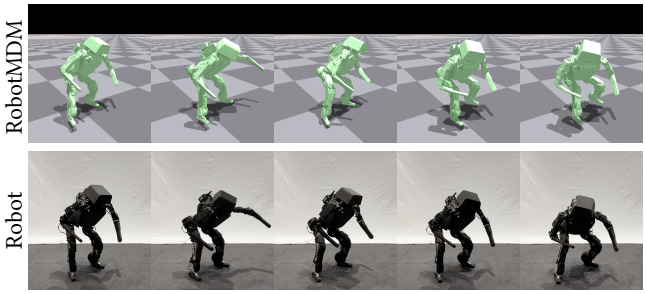
Table 3. **Motion Tracking.** Evaluation of tracking performance across linear and angular root velocity, root rotation, and upper and lower body Degrees of Freedom (DoFs) tracking, measured over 2048 simulations of 30-second references from motions generated by MDM and RobotMDM.

	Tracking Error				
Input	lin. vel. [m/s]	ang. vel. [rad/s]	root rot. [°]	upper DoFs [°]	lower DoFs [°]
MDM	4.90	0.29	4.13	10.88	16.11
RobotMDM	3.43	0.23	2.34	9.36	11.44



prompt: the person lifts a dumbbell over his head.

Fig. 6. **Robot Control.** MDM motions are difficult to track on a real robot system—the lack of balance and non-physicality lead to poor target matching. The same prompt for RobotMDM yields better robot motions.



prompt: a person sneakily crouches while moving laterally.

Fig. 7. **Robot Precision.** The generated motions can be accurately tracked on a real-world robot.

those from MDM cause the robot to lose balance. See the supplementary video for similar results. The clear differences between the two types of generated motions underscore the potential benefits of iteratively improving both the control policy and the motion generator. By enhancing these components in tandem, we can further boost the precision and reliability of the robot's performance.

5 Conclusion

In this work, we combine kinematic motion generation with physics-based character control. Our method aligns a motion diffusion model with the physical constraints of the character without compromising versatility and diversity, and can be deployed on real-world robots.

Limitations. While the generated motions align more closely with the desired objectives, there is no hard constraint on motion feasibility. Prompts that significantly deviate from what the character can realistically do might require substantial adjustment of the motion to become feasible, leading to a conflict with the original text prompt. As shown in the video, when prompted to make the robot swim, RobotMDM struggles to completely resolve this conflict. We can, therefore, not guarantee its reliability in performance-critical environments, since there remains a risk that the model generates motions that violate constraints or are infeasible. Finally, despite significant advancements in the past year, the need for a character-specific dataset, often retargeted from human data, does not cover the full expressiveness that robotic hardware could support. We believe that further bridging the gap between kinematic and physics-based approaches through large-scale synthetic datasets generated from physics-aware motion generators will ultimately lead to enhanced policies and robot capabilities.

References

Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. 2019. DReCon: Data-Driven Responsive Control of Physics-Based Characters. *ACM Trans. Graph.* 38, 6, Article 206 (nov 2019), 11 pages. https://doi.org/10.1145/3355089.3356536

Prashanth Chandran, Gaspard Zoss, Markus Gross, Paulo Gotardo, and Derek Bradley. 2022. Facial Animation with Disentangled Identity and Motion using Transformers. *Computer Graphics Forum* 41, 8 (2022), 267–277. https://doi.org/10.1111/cgf.14641

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. 2023. Executing your Commands via Motion Diffusion in Latent Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18000–18010.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems* 30 (2017).

Zhiyang Dou, Xuelin Chen, Qingnan Fan, Taku Komura, and Wenping Wang. 2023. C·ASE: Learning Conditional Adversarial Skill Embeddings for Physics-Based Characters. In *SIGGRAPH Asia 2023 Conference Papers*. Association for Computing Machinery, New York, NY, USA, Article 2, 11 pages. https://doi.org/10.1145/3610548.3618205

Yusen Feng, Xiyan Xu, and Libin Liu. 2023. MuscleVAE: Model-Based Controllers of Muscle-Actuated Characters. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) *(SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 3, 11 pages. https://doi.org/10.1145/3610548.3618137

Jonas Gehring, Deepak Gopinath, Jungdam Won, Andreas Krause, Gabriel Synnaeve, and Nicolas Usunier. 2023. Leveraging Demonstrations with Latent Space Priors. *Transactions on Machine Learning Research* (2023). https://openreview.net/forum?id=OzGIu4T4Cz

Ruben Grandia, Espen Knoop, Michael A. Hopkins, Georg Wiedebach, Jared Bishop, Steven Pickles, David Müller, and Moritz Bächer. 2024. Design and Control of a Bipedal Robotic Character. In *Proceedings of Robotics: Science and Systems*. Delft, the Netherlands.

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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5142–5151. https://doi.org/10.1109/CVPR52688.2022.00509

Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. 2020. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*. 2021–2029.

Perttu Hämäläinen, Juuso Toikka, Amin Babadi, and C Karen Liu. 2020. Visualizing movement control optimization landscapes. *IEEE Transactions on Visualization and Computer Graphics* 28, 3 (2020), 1648–1660.

Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust Motion In-Betweening. *ACM Trans. Graph.* 39, 4, Article 60 (aug 2020), 12 pages. https://doi.org/10.1145/3386569.3392480

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-functioned neural networks for character control. *ACM Trans. Graph.* 36, 4 (2017).

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.

Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. 2015. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 technical briefs*. 1–4.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. 2024. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems* 36 (2024).

Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. 2022. PADL: Language-Directed Physics-Based Character Control. In *SIGGRAPH Asia 2022 Conference Papers*. Association for Computing Machinery, New York, NY, USA, Article 19, 9 pages.

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. 2024. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1334–1345.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Kyungho Lee, Seyoung Lee, and Jehee Lee. 2018. Interactive character animation by learning multi-objective control. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–10.

Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. 2020. Character Controllers Using Motion VAEs. *ACM Trans. Graph.* 39, 4, Article 40 (aug 2020), 12 pages. https://doi.org/10.1145/3386569.3392422

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris M. Kitani, and Weipeng Xu. 2024. Universal Humanoid Motion Representations for Physics-Based Control. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=OrOd8PxOO2

Zhengyi Luo, Jinkun Cao, Alexander W. Winkler, Kris Kitani, and Weipeng Xu. 2023. Perpetual Humanoid Control for Real-time Simulated Avatars. In *International Conference on Computer Vision (ICCV)*.

Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF 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. In *International Conference on Learning Representations*.

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved Denoising Diffusion Probabilistic Models. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 8162–8171. https://proceedings.mlr.press/v139/nichol21a.html

Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. 2019. Learning predict-and-simulate policies from unorganized human motion data. *ACM Trans. Graph.* 38, 6 (Nov. 2019), 1–11. https://doi.org/10.1145/3355089.3356501

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).

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 Trans. Graph.* 41, 4 (2022).

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

Matthias Plappert, Christian Mandery, and Tamim Asfour. 2016. The KIT Motion-Language Dataset. *Big Data* 4, 4 (dec 2016), 236–252. https://doi.org/10.1089/big.2016.0028

Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. 2021. BABEL: Bodies, Action and Behavior with English Labels. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*. 722–731.

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. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 8748–8763.

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 *International Conference on Computer Vision (ICCV)*.
Jiawei Ren, Mingyuan Zhang, Cunjun Yu, Xiao Ma, Liang Pan, and Ziwei Liu. 2023. InsActor: Instruction-driven Physics-based Characters. *NeurIPS* (2023).
John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1506.02438
John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
Christian Schumacher, Espen Knoop, and Moritz Bächer. 2021. A Versatile Inverse Kinematics Formulation for Retargeting Motions Onto Robots With Kinematic Loops. *IEEE Robotics and Automation Letters* 6, 2 (2021), 943–950. https://doi.org/10.1109/LRA.2021.3056030
Agon Serifi, Ruben Grandia, Espen Knoop, Markus Gross, and Moritz Bächer. 2024. VMP: Versatile Motion Priors for Robustly Tracking Motion on Physical Characters. *Computer Graphics Forum (in proceedings SCA)* (2024). https://doi.org/10.1111/cgf.15175
Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. 2023. Human Motion Diffusion as a Generative Prior. In *The Twelfth International Conference on Learning Representations*.
Sebastian Starke, Ian Mason, and Taku Komura. 2022. DeepPhase: periodic autoencoders for learning motion phase manifolds. *ACM Trans. Graph.* 41, 4 (July 2022), 1–13. https://doi.org/10.1145/3528223.3530178
Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. 2019. Neural state machine for character-scene interactions. *ACM Transactions on Graphics* 38, 6 (2019), 178.
Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. 2020. Local Motion Phases for Learning Multi-Contact Character Movements. *ACM Trans. Graph.* 39, 4, Article 54 (aug 2020), 14 pages. https://doi.org/10.1145/3386569.3392450
Hyung Ju Suh, Max Simchowitz, Kaiqing Zhang, and Russ Tedrake. 2022. Do differentiable simulators give better policy gradients?. In *International Conference on Machine Learning*. PMLR, 20668–20696.
Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. 2023. CALM: Conditional Adversarial Latent Models for Directable Virtual Characters. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3588432.3591541
Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. 2022. Motionclip: Exposing human motion generation to clip space. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*. Springer, 358–374.
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
Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. 2020. UniCon: Universal Neural Controller For Physics-based Character Motion. arXiv:2011.15119 [cs.GR]
Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2020. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph.* 39, 4 (Aug. 2020), 33:1–33:12. 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 (July 2022), 1–12. https://doi.org/10.1145/3528223.3530067
Jungdam Won and Jehee Lee. 2019. Learning body shape variation in physics-based characters. *ACM Trans. Graph.* 38, 6, Article 207 (nov 2019), 12 pages. https://doi.org/10.1145/3355089.3356499
Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. 2024. OmniControl: Control Any Joint at Any Time for Human Motion Generation. In *The Twelfth International Conference on Learning Representations*.
Zhaoming Xie, Hung Yu Ling, Nam Hee Kim, and Michiel van de Panne. 2020. ALLSTEPS: Curriculum-driven Learning of Stepping Stone Skills. *Computer Graphics Forum* 39, 8 (2020), 213–224. https://doi.org/10.1111/cgf.14115 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14115
Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. 2022. ControlVAE: Model-Based Learning of Generative Controllers for Physics-Based Characters. *ACM Trans. Graph.* 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. 2024. MoConVQ: Unified Physics-Based Motion Control via Scalable Discrete Representations. *ACM Trans. Graph.* 43, 4, Article 144 (jul 2024), 21 pages. https://doi.org/10.1145/3658137
Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. 2023. PhysDiff: Physics-Guided Human Motion Diffusion Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. 2023. T2M-GPT: Generating Human Motion from Textual Descriptions with Discrete Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2024. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).
Qingxu Zhu, He Zhang, Mengting Lan, and Lei Han. 2023. Neural Categorical Priors for Physics-Based Character Control. *ACM Trans. Graph.* 42, 6, Article 178 (dec 2023), 16 pages. https://doi.org/10.1145/3618397