PhysDiff: Physics-Guided Human Motion Diffusion Model

Ye Yuan Jiaming Song Umar Iqbal Arash Vahdat Jan Kautz
NVIDIA
https://nvlabs.github.io/PhysDiff

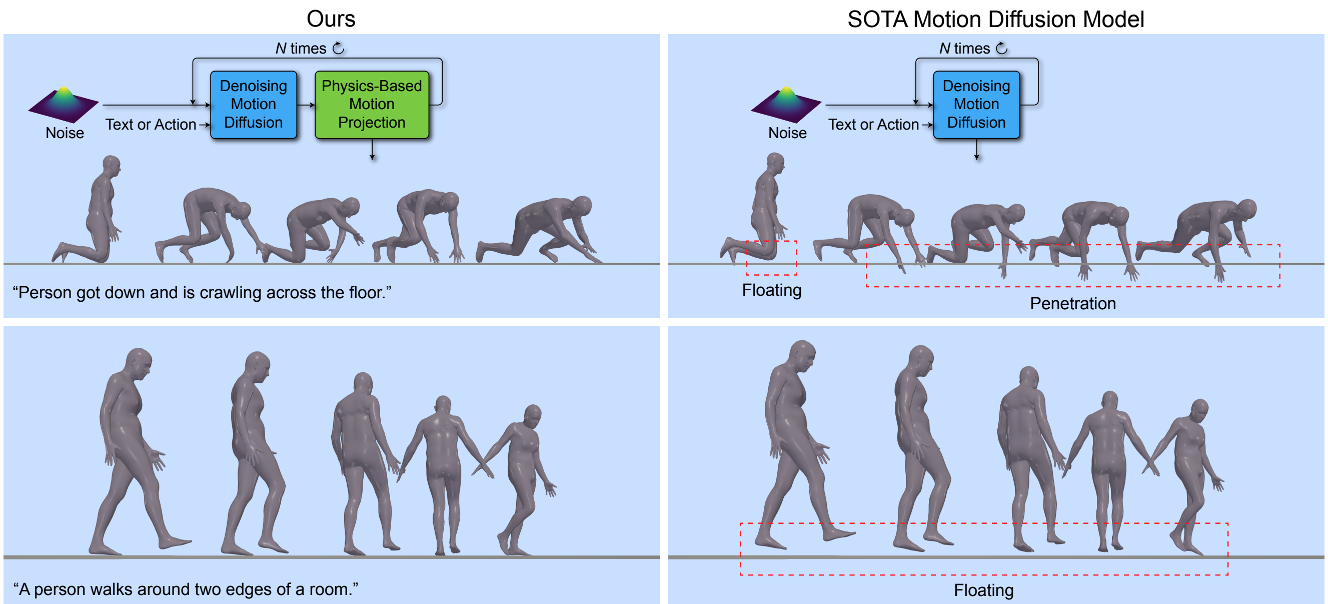

Figure 1. Our PhysDiff model generates physically-plausible motions using a physics-based motion projection in the diffusion process, eliminating artifacts such as floating, ground penetration, and foot sliding, often observed with state-of-the-art models.

Abstract

Denoising diffusion models hold great promise for generating diverse and realistic human motions. However, existing motion diffusion models largely disregard the laws of physics in the diffusion process and often generate physically-implausible motions with pronounced artifacts such as floating, foot sliding, and ground penetration. This seriously impacts the quality of generated motions and limits their real-world application. To address this issue, we present a novel physics-guided motion diffusion model (PhysDiff), which incorporates physical constraints into the diffusion process. Specifically, we propose a physics-based motion projection module that uses motion imitation in a physics simulator to project the denoised motion of a diffusion step to a physically-plausible motion. The projected motion is further used in the next diffusion step to guide the denoising diffusion process. Intuitively, the use of physics in our model iteratively pulls the motion toward a physically-plausible space, which cannot be achieved by simple post-processing. Experiments on large-scale human motion datasets show that our approach achieves state-of-the-art motion quality and improves physical plausibility drastically (>78% for all datasets).

1. Introduction

Deep learning-based human motion generation is an important task with numerous applications in animation, gaming, and virtual reality. In common settings such as text-to-motion synthesis, we need to learn a conditional generative model that can capture the multi-modal distribution of human motions. The distribution can be highly complex due to the high variety of human motions and the intricate interaction between human body parts. Denoising diffusion models [73, 25, 74] are a class of generative models that are especially suited for this task due to their strong ability to model complex distributions, which has been demonstrated extensively in the image generation domain [67, 62, 66, 13]. These models have exhibited strong mode coverage often indicated by high test likelihood [75, 33, 79]. They also

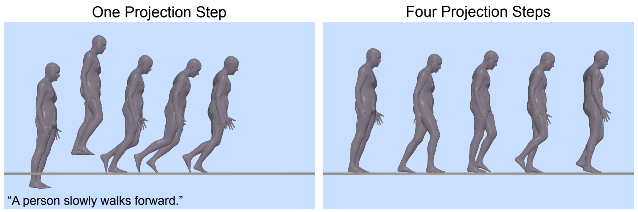

Figure 2. (**Left**) Performing one physics-based projection step (post-processing) at the end yields unnatural motion since the motion is too physically-implausible to correct. (**Right**) Our approach solves this issue by iteratively applying physics and diffusion.

have better training stability compared to generative adversarial networks (GANs [19]) and better sample quality compared to variational autoencoders (VAEs [34]) and normalizing flows [72, 79, 78, 4]. Motivated by this, recent works have proposed motion diffusion models [77, 92] which significantly outperform standard deep generative models in motion generation performance.

However, existing motion diffusion models overlook one essential aspect of human motion – the underlying laws of physics. Even though diffusion models have a superior ability to model the distribution of human motion, they still have no explicit mechanisms to enforce physical constraints or model the complex dynamics induced by forces and contact. As a result, the motions they generate often contain pronounced artifacts such as floating, foot sliding, and ground penetration. This severely hinders many real-world applications such as animation and virtual reality, where humans are highly sensitive to the slightest clue of physical inaccuracy [63, 27]. In light of this, a critical problem we need to address is making human motion diffusion models physics-aware.

To tackle this problem, we propose a novel physics-guided motion diffusion model (PhysDiff) that instills the laws of physics into the denoising diffusion process. Specifically, PhysDiff leverages a physics-based motion projection module (details provided later) that projects an input motion to a physically-plausible space. During the diffusion process, we use the motion projection module to project the denoised motion of a diffusion step into a physically-plausible motion. This new motion is further used in the next diffusion step to guide the denoising diffusion process. Note that it may be tempting to only add the physics-based projection at the end of the diffusion process, i.e., using physics as a post-processing step. However, this can produce unnatural motions since the final denoised kinematic motion from diffusion may be too physically-implausible to be corrected by physics (see Fig. 2 for an example) and the motion may be pushed away from the data distribution. Instead, we need to embed the projection in the diffusion process and apply physics and diffusion iteratively to keep the motion close to the data distribution while moving toward the physically-plausible space (see Sec. 4.4).

The physics-based motion projection module serves the vital role of enforcing physical constraints in PhysDiff, which is achieved by motion imitation in a physics simulator. Specifically, using large-scale motion capture data, we train a motion imitation policy that can control a character agent in the simulator to mimic a vast range of input motions. The resulting simulated motion enforces physical constraints and removes artifacts such as floating, foot sliding, and ground penetration. Once trained, the motion imitation policy can be used to mimic the denoised motion of a diffusion step to output a physically-plausible motion.

We evaluate our model, PhysDiff, on two tasks: text-to-motion generation and action-to-motion generation. Since our approach is agnostic to the specific instantiation of the denoising network used for diffusion, we test two state-of-the-art (SOTA) motion diffusion models (MDM [77] and MotionDiffuse [92]) as our model's denoiser. For text-to-motion generation, our model outperforms SOTA motion diffusion models significantly on the large-scale HumanML3D [22] benchmark, reducing physical errors by more than 86% while also improving the motion quality by more than 20% as measured by the Frechet inception distance (FID). For action-to-motion generation, our model again improves the physics error metric by more than 78% on HumanAct12 [23] and 94% on UESTC [29] while also achieving competitive FID scores.

We further perform extensive experiments to investigate various schedules of the physics-based projection, *i.e.*, at which diffusion timesteps to perform the projection. Interestingly, we observe a trade-off between physical plausibility and motion quality when varying the number of physics-based projection steps. Specifically, while more projection steps always lead to better physical plausibility, the motion quality increases before a certain number of steps and decreases after that, *i.e.*, the resulting motion satisfies the physical constraints but still may look unnatural. This observation guides us to use a balanced number of physics-based projection steps where both high physical plausibility and motion quality is achieved. We also find that adding the physics-based projection to late diffusion steps performs better than early steps. We hypothesize that motions from early diffusion steps may tend toward the mean motion of the training data and the physics-based projection could push the motion further away from the data distribution, thus hampering the diffusion process. Finally, we also show that our approach outperforms physics-based post-processing (single or multiple steps) in motion quality and physical plausibility significantly.

Our contributions are summarized as follows:

- We present a novel physics-guided motion diffusion model that generates physically-plausible motions by instilling the laws of physics into the diffusion process. Its plug-and-play nature makes it flexible to use with different kinematic diffusion models.

- We propose to leverage human motion imitation in a physics simulator as a motion projection module to enforce physical constraints.
- Our model achieves SOTA performance in motion quality and drastically improves physical plausibility on large-scale motion datasets. Our extensive analysis also provides insights such as schedules and tradeoffs, and we demonstrate significant improvements over physics-based post-processing.

2. Related Work

Denoising Diffusion Models. Score-based denoising diffusion models [73, 25, 75, 74] have achieved great successes in various applications such as image generation [67, 62, 66, 72, 79], text-to-speech synthesis [35], 3D shape generation [95, 45, 91], machine learning security [54], as well as human motion generation [77, 92, 65]. These models are trained via denoising autoencoder objectives that can be interpreted as score matching [80], and generate samples via an iterative denoising procedure that may use stochastic updates [6, 5, 15, 94] which solve stochastic differential equations (SDEs) or deterministic updates [74, 43, 41, 93, 30, 14] which solve ordinary differential equations (ODEs).

To perform conditional generation, the most common technique is classifier(-free) guidance [13, 26]. However, it requires training the model specifically over paired data and conditions. Alternatively, one could use pretrained diffusion models that are trained only for unconditional generation. For example, SDEdit [52] modifies the initialization of the diffusion model to synthesize or edit an existing image via colored strokes. In image domains, various methods solve linear inverse problems by repeatedly injecting known information to the diffusion process [44, 12, 75, 11, 32, 31]. A similar idea is applied to human motion diffusion models in the context of motion infilling [77]. In our case, generating physically-plausible motions with diffusion models has a different set of challenges. First, the constraint is specified through a physics simulator, which is non-differentiable. Second, the physics-based projection itself is relatively expensive to compute, unlike image-based constraints which use much less compute than the diffusion model in general. As a result, we cannot simply apply the physics-based projection to every step of the sampling process.

Human Motion Generation. Early work on motion generation adopts deterministic human motion modeling which only generates a single motion [17, 38, 18, 51, 56, 20, 2, 81, 50, 86]. Since human motions are stochastic in nature, more work has started to use deep generative models which avoid the mode averaging problem common in deterministic methods. These methods often use GANs or VAEs to generate motions from various conditions such as past motions [7, 83, 3, 88, 87], key frames [24], music [37, 96, 36], text [1, 9, 22, 61], and action labels [23, 60, 10]. Recently, denoising diffusion models [73, 25, 74] have emerged as a new class of generative models that combine the advantages of standard generative models. Therefore, several motion diffusion models [77, 92, 65] have been proposed which demonstrate SOTA motion generation performance. However, existing motion diffusion models often produce physically-implausible motions since they disregard physical constraints in the diffusion process. Our method addresses this problem by guiding the diffusion process with a physics-based motion projection module.

Physics-Based Human Motion Modeling. Physics-based human motion imitation is first applied to learning locomotion skills such as walking, running, and acrobatics with deep reinforcement learning (RL) [39, 40, 53, 57, 58]. RL-based motion imitation has also been used to learn user-controllable policies for character animation [8, 55, 82]. For 3D human pose estimation, recent work has adopted physics-based trajectory optimization [90, 64, 71, 70] and motion imitation [85, 86, 28, 89, 84, 46, 47] to model human dynamics. Unlike previous work, we explore the synergy between physics simulation and diffusion models, and show that applying physics and diffusion iteratively can generate more realistic and physically-plausible motions.

3. Method

Given some conditional information $\boldsymbol{c}$ such as text or an action label, we aim to generate a physically-plausible human motion $\boldsymbol{x}^{1:H} = \{\boldsymbol{x}^h\}_{h=1}^H$ of length H. Each pose $\boldsymbol{x}^h \in \mathbb{R}^{J \times D}$ in the generated motion is represented by the D-dimensional features of J joints, which can be either the joint positions or angles. We propose a physics-guided denoising diffusion model (PhysDiff) for human motion generation. Starting from a noisy motion $\boldsymbol{x}_T^{1:H}$, PhysDiff models the denoising distribution $q(\boldsymbol{x}_s^{1:H}|\boldsymbol{x}_t^{1:H}, \mathcal{P}_\pi, \boldsymbol{c})$ that denoises the motion from diffusion timestep t to s ($s < t$). Iteratively applying the model denoises the motion into a clean motion $\boldsymbol{x}_0^{1:H}$, which becomes the final output $\boldsymbol{x}^{1:H}$. A critical component in the model is a physics-based motion projection module $\mathcal{P}_\pi$ that enforces physical constraints. It leverages a motion imitation policy π to mimic the denoised motion of a diffusion step in a physics simulator and uses the simulated motion to further guide the diffusion process. An overview of our PhysDiff model is provided in Fig. 3. In the following, we first introduce the physics-guided motion diffusion process in Sec. 3.1. We then describe the details of the physics-based motion projection $\mathcal{P}_\pi$ in Sec. 3.2.

3.1. Physics-Guided Motion Diffusion

Motion Diffusion. To simplify notations, here we sometimes omit the explicit dependence over the condition $\boldsymbol{c}$.

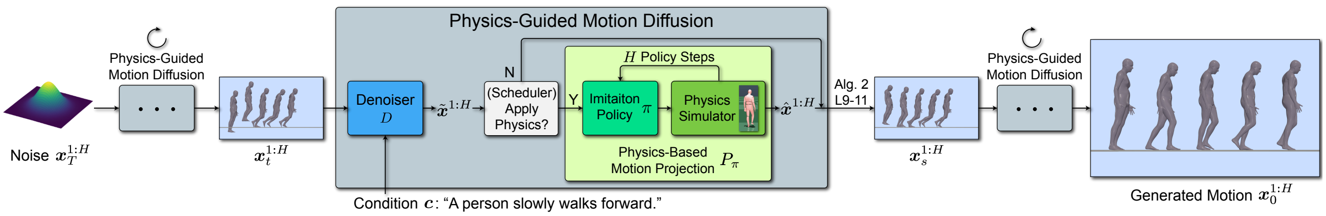

Figure 3. **Overview of PhysDiff.** Each physics-guided diffusion step denoises a motion from timestep t to s, where physics-based motion projection is used to enforce physical constraints. The projection is achieved using a motion imitation policy to control a character in a physics simulator. A scheduler controls when the physics-based projection is applied. The denoiser can be any motion-denoising network.

Note that we can always train diffusion models with some condition $\boldsymbol{c}$; even for the unconditional case, we can condition the model on a universal null token $\varnothing$ [26].

Let $p_0(\boldsymbol{x})$ denote the data distribution, and define a series of time-dependent distributions $p_t(\boldsymbol{x}_t)$ by injecting *i.i.d.* Gaussian noise to samples from p_0, *i.e.*, $p_t(\boldsymbol{x}_t|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}, \sigma_t^2 \mathbf{I})$, where σ_t defines a series of *noise levels* that is increasing over time such that $\sigma_0 = 0$ and σ_T for the largest possible T is much bigger than the data's standard deviation. Generally, diffusion models draw samples by solving the following stochastic differential equation (SDE) from $t = T$ to $t = 0$ [21, 30, 94]:

$$\mathrm{d}\boldsymbol{x} = -(\beta_t + \dot{\sigma}_t)\sigma_t \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\mathrm{d}t + \sqrt{2\beta_t}\sigma_t \mathrm{d}\omega_t, \quad (1)$$

where $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ is the score function, ω_t is the standard Wiener process, and β_t controls the amount of stochastic noise injected in the process; when it is zero, the SDE becomes and ordinary differential equation (ODE). A notable property of the score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ is that it recovers the minimum mean squared error (MMSE) estimator of $\boldsymbol{x}$ given $\boldsymbol{x}_t$ [76, 16, 68]:

$$\tilde{\boldsymbol{x}} := \mathbb{E}[\boldsymbol{x}|\boldsymbol{x}_t] = \boldsymbol{x}_t + \sigma_t^2 \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t), \quad (2)$$

where we can essentially treat $\tilde{\boldsymbol{x}}$ as a "denoised" version of $\boldsymbol{x}_t$. Since $\boldsymbol{x}_t$ and σ_t are known during sampling, we can obtain $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ from $\tilde{\boldsymbol{x}}$, and vice versa.

Diffusion models approximate the score function with the following denoising autoencoder objective [80]:

$$\mathbb{E}_{\boldsymbol{x}\sim p_0(\boldsymbol{x}), t\sim p(t), \epsilon\sim p(\epsilon)}[\lambda(t)\|\boldsymbol{x} - D(\boldsymbol{x} + \sigma_t\epsilon, t, \boldsymbol{c})\|_2^2] \quad (3)$$

where D is the denoiser that depends on the noisy data, the time t and the condition $\boldsymbol{c}$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $p(t)$ is a distribution from which time is sampled, and $\lambda(t)$ is the loss weighting factor. The optimal solution to D would be one that recovers the MMSE estimator $\tilde{\boldsymbol{x}}$ according to Eq. (2). For a detailed characterization of the training procedure, we refer the reader to Karras *et al*. [30].

After the denoising diffusion model has been trained, one can apply it to solve the SDE / ODE in Eq. (1). A particular approach, DDIM [74], performs a one-step update from time t to time s ($s < t$) given the sample $\boldsymbol{x}_t$, which is described in Algorithm 1. Intuitively, the sample $\boldsymbol{x}_s$ at time s is generated from a Gaussian distribution; its mean is a linear interpolation between $\boldsymbol{x}_t$ and the denoised result $\tilde{\boldsymbol{x}}$, and its variance depends on a hyperparameter $\eta \in [0, 1]$. Specifically, $\eta = 0$ corresponds to denoising diffusion probabilistic models (DDPM [25]) and $\eta = 1$ corresponds to denoising diffusion implicit models (DDIM [74]). We find $\eta = 0$ produces better performance for our model. Since the above sampling procedure is general, it can also be applied to human motion data, *i.e.*, $\boldsymbol{x}^{1:H}$. To incorporate the condition $\boldsymbol{c}$ during sampling, one can employ classifier-based or classifier-free guidance [13, 26].

Algorithm 1 DDIM sampling algorithm

Input: Denoiser D, sample $\boldsymbol{x}_t$ at time t, target time s, condition $\boldsymbol{c}$, hyperparameter $\eta \in [0, 1]$.
Compute the denoised result $\tilde{\boldsymbol{x}} := D(\boldsymbol{x}_t, t, \boldsymbol{c})$.
Obtain variance v_s as a scalar that depends on η.
Obtain mean μ_s as a linear combination of $\tilde{\boldsymbol{x}}$ and $\boldsymbol{x}_t$:

$$\mu_s := \tilde{\boldsymbol{x}} + \frac{\sqrt{\sigma_s^2 - v_s}}{\sigma_t}(\boldsymbol{x}_t - \tilde{\boldsymbol{x}})$$

Draw sample $\boldsymbol{x}_s \sim \mathcal{N}(\mu_s, v_s \boldsymbol{I})$.

Applying Physical Constraints. Existing diffusion models for human motions are not necessarily trained on data that complies with physical constraints, and even if they are, there is no guarantee that the produced motion samples are still physically realizable, due to the approximation errors in the denoiser networks and the stochastic nature of the sampling process. While one may attempt to directly correct the final motion sample to be physically-plausible, the physical errors in the motion might be so large that even after such a correction, the motion is still not ideal (see Fig. 2 for a concrete example and Sec. 4.4 for comparison).

To address this issue, we exploit the fact that diffusion models produce intermediate estimates of the desired outcome, *i.e.*, the denoised motion $\tilde{\boldsymbol{x}}^{1:H}$ of each diffusion step. In particular, we may apply physical constraints not only to the final step of the diffusion process, but to the intermediate steps as well. Concretely, we propose a physics-based motion projection $\mathcal{P}_\pi : \mathbb{R}^{H\times J\times D} \rightarrow \mathbb{R}^{H\times J\times D}$ as a module that maps the original motion $\tilde{\boldsymbol{x}}^{1:H}$ to a physically-plausible one, denoted as $\hat{\boldsymbol{x}}^{1:H}$. We incorporate the proposed physics-based projection $\mathcal{P}_\pi$ into the denoising diffusion sampling procedure, where the one-step update from time t to time s is described in Algorithm 2. The process differs from the DDIM sampler mostly in terms of performing the additional physics-based projection; as we will show in the experiments, this is a simple yet effective approach to enforcing physical constraints. Notably, our model, PhysDiff, is agnostic to the specific instantiation of the denoiser D, which is often implemented with various network architectures. The physics-based projection in PhysDiff is also only applied during diffusion sampling (inference), which makes PhysDiff generally compatible with different pre-trained motion diffusion models. In other words, PhysDiff can be used to improve the physical plausibility of existing diffusion models without retraining.

Algorithm 2 PhysDiff sampling algorithm for motion.

Input: Denoiser D, sample $\boldsymbol{x}_t^{1:H}$ at time t, condition $\boldsymbol{c}$, target time s, physics-based projection $\mathcal{P}_\pi$, $\eta \in [0, 1]$.
Compute the denoised motion $\tilde{\boldsymbol{x}}^{1:H} := D(\boldsymbol{x}_t^{1:H}, t, \boldsymbol{c})$.
if projection is performed at time t **then**
    $\hat{\boldsymbol{x}}^{1:H} := \mathcal{P}_\pi(\tilde{\boldsymbol{x}}^{1:H})$ # Physics-Based Projection
else
    $\hat{\boldsymbol{x}}^{1:H} := \tilde{\boldsymbol{x}}^{1:H}$
end if
# The remaining part is similar to DDIM
Obtain variance v_s as a scalar that depends on η.
Obtain mean μ_s:

$$\mu_s := \hat{\boldsymbol{x}}^{1:H} + \frac{\sqrt{\sigma_s^2 - v_s}}{\sigma_t}(\boldsymbol{x}_t^{1:H} - \hat{\boldsymbol{x}}^{1:H})$$

Draw sample $\boldsymbol{x}_s^{1:H} \sim \mathcal{N}(\mu_s, v_s \boldsymbol{I})$.

Scheduling Physics-based Projection. Due to the use of physics simulation, the projection $\mathcal{P}_\pi$ is rather expensive, and it is infeasible to perform the projection at every diffusion timestep. Therefore, if we have a limited number of physics-based projection steps to be performed, we need to prioritize certain timesteps over others. Here, we argue that we should not perform physics projection when the diffusion noise level is high. This is because the denoiser D by design gives us $\tilde{\boldsymbol{x}}^{1:H} = \mathbb{E}[\boldsymbol{x}^{1:H}|\boldsymbol{x}_t^{1:H}]$, *i.e.*, the mean motion of $\boldsymbol{x}^{1:H}$ given the current noisy motion $\boldsymbol{x}_t^{1:H}$, and it is close to the mean of the training data for high noise levels when the condition $\boldsymbol{x}_t^{1:H}$ contains little information. Empirically, the mean motion often has little body movement due to mode averaging while still having some root translations and rotations, which is clearly physically-implausible. Correcting such a physically-incorrect motion with the physics-based projection would push the motion further away from the data distribution and hinder the diffusion process. In Sec. 4.3, we perform a systematic study that validates this hypothesis and reveals a favorable scheduling strategy that balances sample quality and efficiency.

3.2. Physics-Based Motion Projection

An essential component in the physics-guided diffusion process is the physics-based motion projection $\mathcal{P}_\pi$. It is tasked with projecting the denoised motion $\tilde{\boldsymbol{x}}^{1:H}$ of a diffusion step, which disregards the laws of physics, into a physically-plausible motion $\hat{\boldsymbol{x}}^{1:H} = \mathcal{P}_\pi(\tilde{\boldsymbol{x}}^{1:H})$. The projection is achieved by learning a motion imitation policy π that controls a simulated character to mimic the denoised motion $\tilde{\boldsymbol{x}}^{1:H}$ in a physics simulator. The resulting motion $\hat{\boldsymbol{x}}^{1:H}$ from the simulator is considered physically-plausible since it obeys the laws of physics.

Motion Imitation Formulation. The task of human motion imitation [57, 88] can be formulated as a Markov decision process (MDP). The MDP is defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \gamma)$ of states, actions, transition dynamics, a reward function, and a discount factor. A character agent acts in a physics simulator according to a motion imitation policy $\pi(\boldsymbol{a}^h|\boldsymbol{s}^h)$, which models the distribution of choosing an action $\boldsymbol{a}^h \in \mathcal{A}$ given the current state $\boldsymbol{s}^h \in \mathcal{S}$. The state $\boldsymbol{s}^h$ consists of the character's physical state (*e.g.*, joint angles, velocities, positions) as well as the next pose $\tilde{\boldsymbol{x}}^{h+1}$ from the input motion. Including $\tilde{\boldsymbol{x}}^{h+1}$ in the state informs the policy π to choose an action $\boldsymbol{a}^h$ that can mimic $\tilde{\boldsymbol{x}}^{h+1}$ in the simulator. Starting from an initial state $\boldsymbol{s}^1$, the agent iteratively samples an action $\boldsymbol{a}^h$ from the policy π and the simulator with transition dynamics $\mathcal{T}(\boldsymbol{s}^{h+1}|\boldsymbol{s}^h, \boldsymbol{a}^h)$ generates the next state $\boldsymbol{s}^{h+1}$, from which we can extract the simulated pose $\hat{\boldsymbol{x}}^{h+1}$. By running the policy for H steps, we can obtain the physically-simulated motion $\hat{\boldsymbol{x}}^{1:H}$.

Training. During training, a reward r^h is also assigned to the character based on how well the simulated motion $\hat{\boldsymbol{x}}^{1:H}$ aligns with the ground-truth motion $\bar{\boldsymbol{x}}^{1:H}$. Note that the motion imitation policy π is trained on large motion capture datasets where high-quality ground-truth motion is available. We use reinforcement learning (RL) to learn the policy π, where the objective is to maximize the expected discounted return $J(\pi) = \mathbb{E}_\pi\left[\sum_h \gamma^h r^h\right]$ which translates to mimicking the ground-truth motion as closely as possible. We adopt a standard RL algorithm (PPO [69]) to solve for the optimal policy. In the following, we will elaborate on the design of rewards, states, actions, and the policy.

Rewards. The reward function is designed to encourage the simulated motion $\hat{\boldsymbol{x}}^{1:H}$ to match the ground truth $\bar{\boldsymbol{x}}^{1:H}$. Here, we use $\bar{\cdot}$ to denote ground-truth quantities. The reward r^h at each timestep consists of four sub-rewards:

$$r^h = w_{\mathrm{p}} r^h_{\mathrm{p}} + w_{\mathrm{v}} r^h_{\mathrm{v}} + w_{\mathrm{j}} r^h_{\mathrm{j}} + w_{\mathrm{q}} r^h_{\mathrm{q}}\,, \tag{4}$$

$$r^h_{\mathrm{p}} = \exp\left[-\alpha_{\mathrm{p}}\left(\textstyle\sum_{j=1}^{J} \|\boldsymbol{o}^h_j \ominus \bar{\boldsymbol{o}}^h_j\|^2\right)\right], \tag{5}$$

$$r^h_{\mathrm{v}} = \exp\left[-\alpha_{\mathrm{v}} \|\boldsymbol{v}^h - \bar{\boldsymbol{v}}^h\|^2\right], \tag{6}$$

$$r^h_{\mathrm{j}} = \exp\left[-\alpha_{\mathrm{j}}\left(\textstyle\sum_{j=1}^{J} \|\boldsymbol{p}^h_j - \bar{\boldsymbol{p}}^h_j\|^2\right)\right], \tag{7}$$

$$r^h_{\mathrm{q}} = \exp\left[-\alpha_{\mathrm{q}}\left(\textstyle\sum_{j=1}^{J} \|\boldsymbol{q}^h_j - \bar{\boldsymbol{q}}^h_j\|^2\right)\right]. \tag{8}$$

where $w_{\mathrm{p}}, w_{\mathrm{v}}, w_{\mathrm{j}}, w_{\mathrm{q}}, \alpha_{\mathrm{p}}, \alpha_{\mathrm{v}}, \alpha_{\mathrm{j}}, \alpha_{\mathrm{q}}$ are weighting factors. The pose reward r^h_{p} measures the difference between the local joint rotations $\boldsymbol{o}^h_j$ and the ground truth $\bar{\boldsymbol{o}}^h_j$, where $\ominus$ denotes the relative rotation between two rotations, and $\|\cdot\|$ computes the rotation angle. The velocity reward r^h_{v} measures the mismatch between joint velocities $\boldsymbol{v}^h$ and the ground truth $\bar{\boldsymbol{v}}^h$, which are computed via finite difference. The joint position reward r^h_{j} encourages the 3D world joint positions $\boldsymbol{p}^h_j$ to match the ground truth $\bar{\boldsymbol{p}}^h_j$. Finally, the joint rotation reward r^h_{q} measures the difference between the global joint rotations $\boldsymbol{q}^h_j$ and the ground truth $\bar{\boldsymbol{q}}^h_j$.

States. The agent state $\boldsymbol{s}^h$ consists of the character's current physical state, the input motion's next pose $\tilde{\boldsymbol{x}}^{h+1}$, and a character attribute vector $\boldsymbol{\psi}$. The character's physical state includes its joint angles, joint velocities, and rigid bodies' positions, rotations, and linear and angular velocities. For the input pose $\tilde{\boldsymbol{x}}^{h+1}$, the state $\boldsymbol{s}^h$ contains the difference of $\tilde{\boldsymbol{x}}^{h+1}$ w.r.t. the agent in joint angles as well as rigid body positions and rotations. Using the difference informs the policy about the pose residual it needs to compensate for. All the features are computed in the character's heading coordinate to ensure rotation and translation invariance. Since our character is based on the SMPL body model [42], the attribute $\boldsymbol{\psi}$ includes the gender and SMPL shape parameters to allow the policy to control different characters.

Actions. We use the target joint angles of proportional derivative (PD) controllers as the action representation, which enables robust motion imitation as observed in prior work [59, 88]. We also add residual forces [88] in the action space to stabilize the character and compensate for missing contact forces required to imitate motions such as sitting.

Policy. We use a parametrized Gaussian policy $\pi(\boldsymbol{a}^h|\boldsymbol{s}^h) = \mathcal{N}(\boldsymbol{\mu}_\theta(\boldsymbol{s}^h), \boldsymbol{\Sigma})$ where the mean action $\boldsymbol{\mu}_\theta$ is output by a simple multi-layer perceptron (MLP) network with parameters θ, and $\boldsymbol{\Sigma}$ is a fixed diagonal covariance matrix.

4. Experiments

We perform experiments on two standard human motion generation tasks: text-to-motion and action-to-motion generation. In particular, our experiments are designed to answer the following questions: (1) Can PhysDiff achieve SOTA motion quality and physical plausibility? (2) Can PhysDiff be applied to different kinematic motion diffusion models to improve their motion quality and physical plausibility? (3) How do different schedules of the physics-based projection impact motion generation performance? (4) Can PhysDiff outperform physics-based post-processing?

Method	FID↓	R-Precision↑	Penetrate↓	Float↓	Skate↓	Phys-Err↓
J2LP [1]	11.020	0.486	-	-	-	-
Text2Gesture [9]	7.664	0.345	-	-	-	-
T2M [22]	1.067	0.740	11.897	7.779	2.908	22.584
MDM [77]	0.544	0.611	11.291	18.876	1.406	31.572
MotionDiffuse (MD) [92]	0.630	**0.782**	20.278	6.450	3.925	30.652
PhysDiff w/ MD (Ours)	0.551	0.780	**0.898**	**1.368**	**0.423**	**2.690**
PhysDiff w/ MDM (Ours)	**0.433**	0.631	0.998	2.601	0.512	4.111

Table 1. Text-to-motion results on HumanML3D [22].

Method	FID↓	Accuracy↑	Penetrate↓	Float↓	Skate↓	Phys-Err↓
Action2Motion [23]	0.338	0.917	-	-	-	-
ACTOR [60]	0.120	0.955	8.939	14.479	2.277	25.695
INR [10]	**0.088**	0.973	7.055	13.212	0.928	22.096
MDM [77]	0.100	**0.990**	5.600	6.703	1.075	13.377
PhysDiff w/ MDM (Ours)	0.096	0.983	**0.689**	**2.002**	**0.159**	**2.850**

Table 2. Action-to-motion results on HumanAct12 [23].

Method	FID↓	Accuracy↑	Penetrate↓	Float↓	Skate↓	Phys-Err↓
ACTOR [60]	23.43	0.911	8.441	9.737	1.073	19.251
INR [10]	15.00	0.941	5.999	4.633	0.741	11.373
MDM [77]	**12.81**	0.950	13.077	13.912	1.383	28.371
PhysDiff w/ MDM (Ours)	13.27	**0.956**	**0.874**	**0.201**	**0.389**	**1.463**

Table 3. Action-to-motion results on UESTC [29].

Evaluation Metrics. For text-to-motion generation, we first use two standard metrics suggested by Guo *et al*. [22]: *FID* measures the distance between the generated and ground-truth motion distributions; *R-Precision* assesses the relevancy of the generated motions to the input text. For action-to-motion generation, we replace R-Precision with an *Accuracy* metric, which measures the accuracy of a trained action classifier over the generated motion. Additionally, we also use four physics-based metrics to evaluate the physical plausibility of generated motions: *Penetrate* measures ground penetration; *Float* measures floating; *Skate* measures foot sliding; *Phys-Err* is an overall physical error metric that sums the three metrics (all in *mm*) together. Please refer to the supplementary materials for details.

Implementation Details. Our model uses 50 diffusion steps with classifier-free guidance [26]. We test PhysDiff with two SOTA motion diffusion models, MDM [77] and MotionDiffuse [92], as the denoiser D. By default, MDM is the denoiser of PhysDiff for qualitative results. We adopt IsaacGym [49] as the physics simulator for motion imitation. More details are given in the supplementary materials.

4.1. Text-to-Motion Generation

Data. We use the HumanML3D [22] dataset, which is a textually annotated subset of two large-scale motion capture datasets, AMASS [48] and HumanAct12 [23]. It contains 14,616 motions annotated with 44,970 textual descriptions.

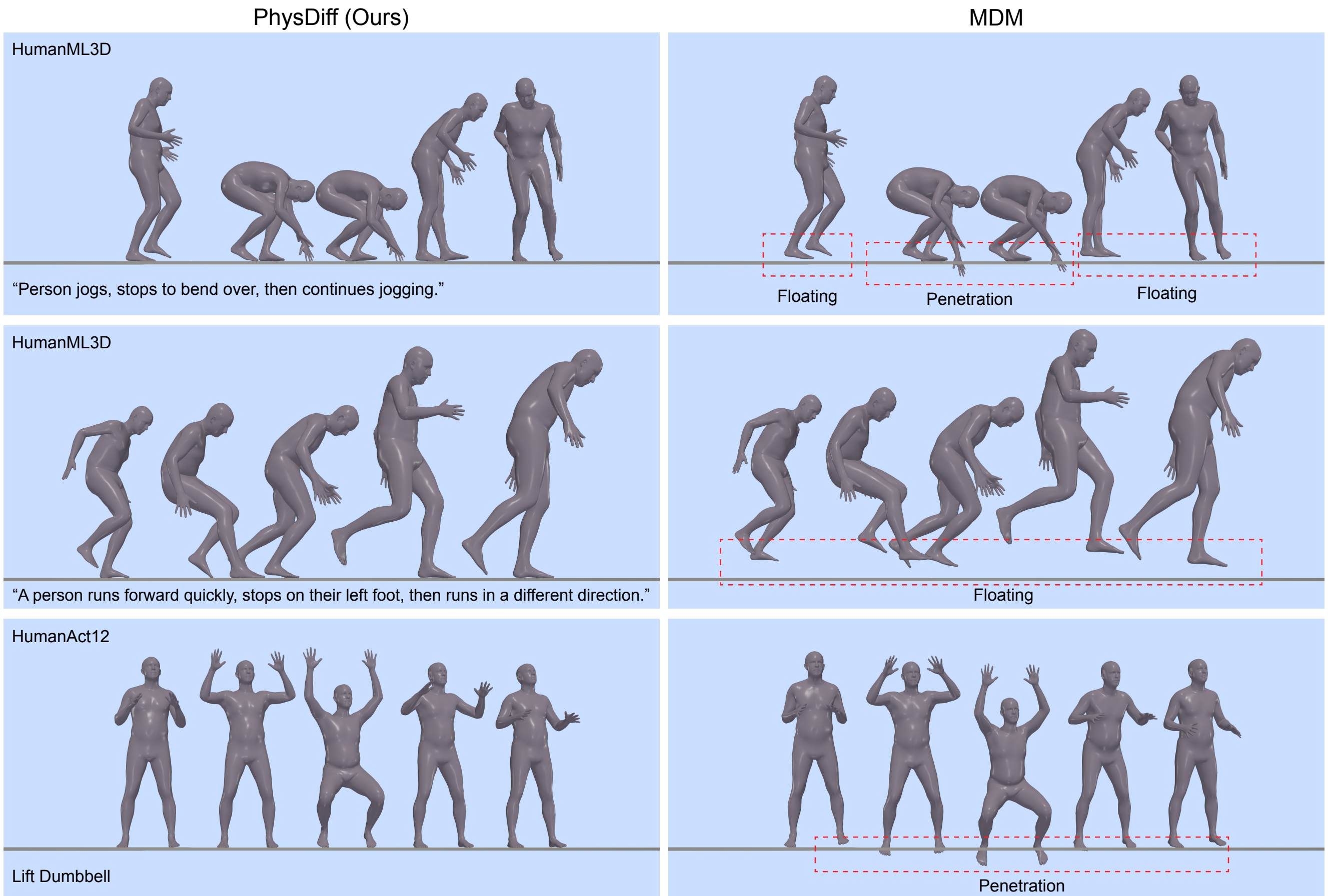

Figure 4. Visual comparison of PhysDiff against the SOTA, MDM [77], on HumanML3D, HumanAct12, and UESTC. PhysDiff reduces physical artifacts such as floating and penetration significantly. Please refer to the supplementary video for more qualitative comparison.

Results. In Table 1, we compare our method to the SOTA methods: JL2P [1], Text2Gesture [9], T2M [22], MotionDiffuse [92], and MDM [77]. Due to the plug-and-play nature of our method, we design two variants of PhysDiff using MotionDiffuse (MD) and MDM. PhysDiff with MDM achieves SOTA FID and also reduces Phys-Err by more than 86% compared to MDM. Similarly, PhysDiff with MD achieves SOTA in physics-based metrics while maintaining high R-Precision and improving FID significantly. We also provide qualitative comparison in Fig. 4, where we can clearly see that PhysDiff substantially reduces physical artifacts such as penetration and floating. Please also refer to the supplementary video for more qualitative results.

4.2. Action-to-Motion Generation

Data. We evaluate on two datasets: HumanAct12 [23], which contains around 1200 motion clips for 12 action categories; UESTC [29], which consists of 40 action classes, 40 subjects, and 25k samples. For both datasets, we use the sequences provided by Petrovich *et al.* [60].

Results. Tables 2 and 3 summarize the results on HumanAct12 and UESTC, respectively, where we compare PhysDiff against the SOTA methods: MDM [77], INR [10], Action2Motion [23], and ACTOR [60]. The results show that our method achieves competitive FID on both datasets while drastically improving Phys-Err (by 78% on HumanAct12 and 94% on UESTC). Please refer to Fig. 4 and the supplementary video for qualitative comparison, where we show that PhysDiff improves the physical plausibility of generated motions significantly.

4.3. Schedule of Physics-Based Projection

We perform extensive experiments to analyze the schedule of the physics-based projection, *i.e.*, at which timesteps we perform the projection in the diffusion process.

Number of Projection Steps. Since the physics-based projection is relatively expensive to compute, we first investigate whether we can reduce the number of projection steps without sacrificing performance. To this end, we vary the number of projection steps performed during diffusion from 50 to 0, where the projection steps are gradually removed from earlier timesteps and applied consecutively.

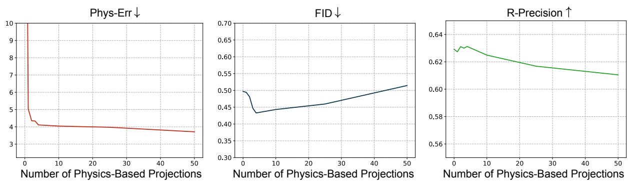

Figure 5. Effect of varying the number of physics-based projection steps for text-to-motion generation on HumanML3D [22].

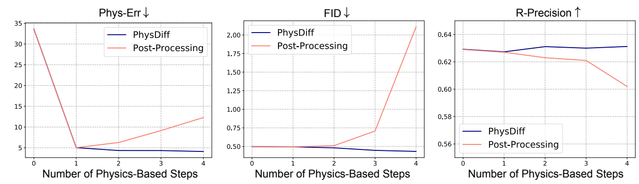

Figure 6. Comparison with post-processing, *i.e.*, applying physics-based projections after the diffusion process.

Schedule	FID ↓	R-Precision ↑	Penetrate ↓	Float ↓	Skate ↓	Phys-Err ↓
Uniform 4	0.473	0.630	0.979	3.479	0.463	4.921
Start 3, End 1	0.510	0.623	**0.918**	3.173	**0.459**	4.550
Start 2, End 2	0.503	0.623	**0.918**	2.723	0.492	4.133
End 4, Space 3	0.469	0.630	0.990	3.226	0.473	4.689
End 4, Space 2	0.469	0.630	0.990	3.004	0.476	4.470
End 4, Space 1	**0.433**	**0.631**	0.998	**2.601**	0.512	**4.111**

Table 4. Projection schedule comparison on HumanML3D [22].

We plot the curves of FID, R-Precision, and Phys-Err in Fig. 5. As can be seen, Phys-Err keeps decreasing with more physics-based projection steps, which indicates more projection steps always help improve the physical plausibility of PhysDiff. Interestingly, both the FID and R-Precision first improve (FID decreases and R-Precision increases) and then deteriorate when increasing the number of projection steps. This suggests that there is a trade-off between physical plausibility and motion quality when more projection steps are performed at the early diffusion steps. We hypothesize that motions generated at the early diffusion steps are denoised to the mean motion of the dataset (with little body movement) and are often not physically-plausible. As a result, performing the physics-based projection at these early steps can push the generated motion away from the data distribution, thus hindering the diffusion process.

Placement of Projections Steps. Fig. 5 indicates that four physics-based projection steps yield a good trade-off between physical plausibility and motion quality. Next, we investigate the best placement of these projection steps in the diffusion process. We compare three groups of schedules: (1) *Uniform* N, which spreads the N projection steps evenly across the diffusion timesteps *i.e.*, for 50 diffusion steps and $N = 4$, the projection steps are performed at $t \in \{0, 15, 30, 45\}$; (2) *Start* M*, End* N, which places M consecutive projection steps at the beginning of the diffusion process and N projection steps at the end; (3) *End* N*, Space* S, which places N projections steps with time spacing S at the end of the diffusion process (*e.g.*, for $N = 4, S = 3$, the projections steps are performed at $t \in \{0, 3, 6, 9\}$). We summarize the results in Table 4. We can see that the schedule *Start* M*, End* N has inferior FID and R-Precision since more physics-based projection steps are performed at early diffusion steps, which is consistent with our findings in Fig. 5. The schedule *Uniform* N works better in terms of FID and R-Precision but has worse Phys-Err. This is likely because too many non-physics-based diffusion steps between the physics-based projections undo the effect of the projection and reintroduce physical errors. This is also consistent with *End* 4*, Space* 3 being worse than *End* 4*, Space* 1 since the former has more diffusion steps between the physics-based projections. Hence, the results suggest that it is better to schedule the physics-based projection steps consecutively toward the end. This guides us to use *End* 4*, Space* 1 for baseline comparison.

Inference Time. Due to the use of physics simulation, PhysDiff is 2.5x slower than MDM (51.6s vs. 19.6s) to generate a single motion, where both methods use 1000 diffusion steps. The gap closes when increasing the batch size to 256 where PhysDiff is only 1.7x slower (280.3s vs. 471.3s), as the physics simulator benefits more from parallelization.

4.4. Comparing against Post-Processing

To demonstrate the synergy between physics and diffusion, we compare PhysDiff against a post-processing baseline that applies one or more physics-based projection steps to the final kinematic motion from diffusion. As shown in Fig. 6, multiple post-processing steps cannot enhance motion quality or physical plausibility; instead, they deteriorate them. This is because the final kinematic motion may be too physically implausible for the physics to imitate, *e.g.*, the human may lose balance due to wrong gaits. Repeatedly imitating these implausible motions could amplify the problem and lead to unstable simulation. PhysDiff overcomes this issue by iteratively applying diffusion and physics to recover from bad simulation states and move closer to the data distribution.

5. Conclusion and Future Work

In this paper, we proposed a novel physics-guided motion diffusion model (PhysDiff) which instills the laws of physics into the diffusion process to generate physically-plausible human motions. To achieve this, we proposed a physics-based motion projection module that uses motion imitation in physics simulation to enforce physical constraints. Our approach is agnostic to the denoising network and can be used to improve SOTA motion diffusion models without retraining. Experiments on large-scale motion data demonstrate that PhysDiff achieves SOTA motion quality and substantially improves physical plausibility.

Due to physics simulation, the inference speed of PhysDiff can be two-to-three times slower than SOTA models. Future work could speed up the model with a faster physics simulator or improve the physics-based projection to reduce the number of required projection steps.

References

[1] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *2019 International Conference on 3D Vision (3DV)*, pages 719–728. IEEE, 2019. 3, 6, 7

[2] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7144–7153, 2019. 3

[3] Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Lars Petersson, and Stephen Gould. A stochastic conditioning scheme for diverse human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5223–5232, 2020. 3

[4] Jyoti Aneja, Alex Schwing, Jan Kautz, and Arash Vahdat. A contrastive learning approach for training variational autoencoder priors. *Advances in Neural Information Processing Systems*, 34:480–493, 2021. 2

[5] Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu, and Bo Zhang. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In *International Conference on Machine Learning*, 2022. 3

[6] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: An analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022. 3

[7] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1418–1427, 2018. 3

[8] Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. Drecon: data-driven responsive control of physics-based characters. *ACM Transactions on Graphics (TOG)*, 38(6):1–11, 2019. 3

[9] Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 1–10. IEEE, 2021. 3, 6, 7

[10] Pablo Cervantes, Yusuke Sekikawa, Ikuro Sato, and Koichi Shinoda. Implicit neural representations for variable length human motion generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 356–372. Springer, 2022. 3, 6, 7

[11] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, August 2021. 3

[12] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-Closer-Diffuse-Faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. *arXiv preprint arXiv:2112.05146*, December 2021. 3

[13] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021. 1, 3, 4

[14] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. GENIE: Higher-order denoising diffusion solvers. In *Advances in Neural Information Processing Systems*, 2022. 3

[15] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped Langevin diffusion. In *International Conference on Learning Representations*, 2022. 3

[16] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. 4

[17] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015. 3

[18] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*, pages 458–466. IEEE, 2017. 3

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[20] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, Lee Giles, and Alexander G Ororbia. A neural temporal model for human motion prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12116–12125, 2019. 3

[21] Ulf Grenander and Michael I Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(4):549–581, 1994. 4

[22] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. 2, 3, 6, 7, 8

[23] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. 2, 3, 6, 7

[24] Chengan He, Jun Saito, James Zachary, Holly Rushmeier, and Yi Zhou. Nemf: Neural motion fields for kinematic animation. *arXiv preprint arXiv:2206.03287*, 2022. 3

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. 1, 3, 4

[26] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 3, 4, 6

[27] Ludovic Hoyet, Rachel McDonnell, and Carol O'Sullivan. Push it real: Perceiving causality in virtual interactions. *ACM Transactions on Graphics (TOG)*, 31(4):1–9, 2012. 2

[28] Mariko Isogawa, Ye Yuan, Matthew O'Toole, and Kris M Kitani. Optical non-line-of-sight physics-based 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7013–7022, 2020. 3

[29] Yanli Ji, Feixiang Xu, Yang Yang, Fumin Shen, Heng Tao Shen, and Wei-Shi Zheng. A large-scale rgb-d database for arbitrary-view human action recognition. In *Proceedings of the 26th ACM international Conference on Multimedia*, pages 1510–1518, 2018. 2, 6, 7

[30] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022. 3, 4

[31] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022. 3

[32] Bahjat Kawar, Gregory Vaksman, and Michael Elad. SNIPS: Solving noisy inverse problems stochastically. *arXiv preprint arXiv:2105.14951*, May 2021. 3

[33] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in Neural Information Processing Systems*, 34:21696–21707, 2021. 1

[34] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[35] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. 3

[36] Buyu Li, Yongchi Zhao, Shi Zhelun, and Lu Sheng. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1272–1279, 2022. 3

[37] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412, 2021. 3

[38] Zimo Li, Yi Zhou, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. *arXiv preprint arXiv:1707.05363*, 2017. 3

[39] Libin Liu and Jessica Hodgins. Learning to schedule control fragments for physics-based characters using deep q-learning. *ACM Transactions on Graphics (TOG)*, 36(3):29, 2017. 3

[40] Libin Liu and Jessica Hodgins. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 3

[41] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022. 3

[42] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 6

[43] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, 2022. 3

[44] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[45] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3D point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3

[46] Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. Dynamics-regulated kinematic policy for egocentric pose estimation. In *Advances in Neural Information Processing Systems*, 2021. 3

[47] Zhengyi Luo, Shun Iwase, Ye Yuan, and Kris Kitani. Embodied scene-aware human pose estimation. In *Advances in Neural Information Processing Systems*, 2022. 3

[48] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 6

[49] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 6

[50] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9489–9497, 2019. 3

[51] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017. 3

[52] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 3

[53] Josh Merel, Yuval Tassa, Sriram Srinivasan, Jay Lemmon, Ziyu Wang, Greg Wayne, and Nicolas Heess. Learning human behaviors from motion capture by adversarial imitation. *arXiv preprint arXiv:1707.02201*, 2017. 3

[54] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning*, 2022. 3

[55] Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. Learning predict-and-simulate policies from unorganized human motion data. *ACM Transactions on Graphics (TOG)*, 38(6):1–11, 2019. 3

[56] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. *arXiv preprint arXiv:1805.06485*, 2018. 3

[57] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 3, 5

[58] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018. 3

[59] Xue Bin Peng and Michiel van de Panne. Learning locomotion skills using deeprl: Does the choice of action space matter? In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 1–13, 2017. 6

[60] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. 3, 6, 7

[61] Mathis Petrovich, Michael J. Black, and Gül Varol. TEMOS: Generating diverse human motions from textual descriptions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3

[62] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 3

[63] Paul SA Reitsma and Nancy S Pollard. Perceptual metrics for character animation: sensitivity to errors in ballistic motion. In *ACM SIGGRAPH 2003 Papers*, pages 537–542, 2003. 2

[64] Davis Rempe, Leonidas J. Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. Contact and human dynamics from monocular video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[65] Zhiyuan Ren, Zhihong Pan, Xin Zhou, and Le Kang. Diffusion motion: Generate text-guided 3d human motion by diffusion model. *arXiv preprint arXiv:2210.12315*, 2022. 3

[66] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 3

[67] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1, 3

[68] Saeed Saremi and Aapo Hyvärinen. Neural empirical bayes. *Journal of machine learning research: JMLR*, 20(181):1–23, 2019. 4

[69] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 5

[70] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, Patrick Pérez, and Christian Theobalt. Neural monocular 3d human motion capture with physical awareness. *ACM Transactions on Graphics (ToG)*, 40(4):1–15, 2021. 3

[71] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics (TOG)*, 39(6), dec 2020. 3

[72] Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2C: Diffusion-decoding models for few-shot conditional generation. In *Advances in Neural Information Processing Systems*, 2021. 2, 3

[73] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015. 1, 3

[74] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 1, 3, 4

[75] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 3

[76] Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981. 4

[77] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 2, 3, 6, 7

[78] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020. 2

[79] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, 2021. 1, 2, 3

[80] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. 3, 4

[81] Borui Wang, Ehsan Adeli, Hsu-kuang Chiu, De-An Huang, and Juan Carlos Niebles. Imitation learning for human pose prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7124–7133, 2019. 3

[82] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Transactions on Graphics (TOG)*, 39(4):33–1, 2020. 3

[83] Xinchen Yan, Akash Rastogi, Ruben Villegas, Kalyan Sunkavalli, Eli Shechtman, Sunil Hadap, Ersin Yumer, and Honglak Lee. Mt-vae: Learning motion transformations to generate multimodal human dynamics. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 265–281, 2018. 3

[84] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13167–13178, 2022. 3

[85] Ye Yuan and Kris Kitani. 3d ego-pose estimation via imitation learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 735–750, 2018. 3

[86] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time pd control. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10082–10092, 2019. 3

[87] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 346–364. Springer, 2020. 3

[88] Ye Yuan and Kris Kitani. Residual force control for agile human behavior imitation and extended motion synthesis. In *Advances in Neural Information Processing Systems*, 2020. 3, 5, 6

[89] Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. Simpoe: Simulated character control for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[90] Petrissa Zell, Bastian Wandt, and Bodo Rosenhahn. Joint 3d human motion capture and physical analysis from monocular videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 17–26, 2017. 3

[91] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. LION: Latent point diffusion models for 3D shape generation. In *Advances in Neural Information Processing Systems*, 2022. 3

[92] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 2, 3, 6, 7

[93] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022. 3

[94] Qinsheng Zhang, Molei Tao, and Yongxin Chen. gDDIM: Generalized denoising diffusion implicit models. *arXiv preprint arXiv:2206.05564*, 2022. 3, 4

[95] Linqi Zhou, Yilun Du, and Jiajun Wu. 3D shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 3

[96] Wenlin Zhuang, Congyi Wang, Jinxiang Chai, Yangang Wang, Ming Shao, and Siyu Xia. Music2dance: Dancenet for music-driven dance generation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(2):1–21, 2022. 3