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
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 physical 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 trade-offs, and we demonstrate significant improvements over physics-based post-processing.

2. Related Work

Denosing 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 c such as text or an action label, we aim to generate a physically-plausible human motion \( \mathbf{x}^{1:H} \), clean motion \( \mathbf{x}^{1:H} \) of length \( H \). Each pose \( \mathbf{x}^{H} \in \mathbb{R}^{J \times D} \) 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 \( \mathbf{x}^{1:H}_{\text{noisy}} \), PhysDiff models the denoising distribution \( q(\mathbf{x}^{1:H}_{\text{noisy}} | \mathbf{x}^{1:H}, \mathcal{P}_\pi, c) \) that denoises the motion from diffusion timestep \( t \) to \( s \) \( (s < t) \). Iteratively applying the model denoises the motion into a clean motion \( \mathbf{x}^{1:H}_{\text{clean}} \), which becomes the final output \( \mathbf{x}^{1:H} \). 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 \( \pi \) 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. Then we 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 \( c \).
Note that we can always train diffusion models with some condition \( c \); even for the unconditional case, we can condition the model on a universal null token \( \emptyset \) [26].

Let \( p_0(x) \) denote the data distribution, and define a series of time-dependent distributions \( p_t(x_t) \) by injecting i.i.d. Gaussian noise to samples from \( p_0 \), i.e., \( p_t(x_t|x) = \mathcal{N}(x, \sigma_t^2\mathbb{I}) \), where \( \sigma_t \) defines a series of noise levels that is increasing over time such that \( \sigma_0 = 0 \) and \( \sigma_T \) 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]:

\[
\frac{dx}{dt} = -\left( \beta_t + \delta_t \right) \sigma_t \nabla_x \log p_t(x) dt + \sqrt{2\beta_t} \sigma_t d\omega_t, \tag{1}
\]

where \( \nabla_x \log p_t(x) \) is the score function, \( \omega_t \) is the standard Wiener process, and \( \beta_t \) 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 \( \nabla_x \log p_t(x_t) \) is that it recovers the minimum mean squared error (MMSE) estimator of \( x \) given \( x_t \) [76, 16, 68]:

\[
\hat{x} := \mathbb{E}[x|x_t] = x_t + \sigma_t^2 \nabla_x \log p_t(x_t), \tag{2}
\]

where we can essentially treat \( \hat{x} \) as a “denoised” version of \( x_t \). Since \( x_t \) and \( \sigma_t \) are known during sampling, we can obtain \( \nabla_x \log p_t(x_t) \) from \( x_t \), and vice versa.

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

\[
\mathbb{E}_{x \sim p_0(x), t \sim \pi(t), \epsilon \sim p(\epsilon)}[\lambda(t)\|x - D(x + \sigma t \epsilon, t, c)\|^2] \tag{3}
\]

where \( D \) is the denoiser that depends on the noisy data, the time \( t \) and the condition \( c, \epsilon \sim \mathcal{N}(0, \mathbb{I}), p(\epsilon) \) 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 \( \hat{x} \) 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 \( x_t \), which is described in Algorithm 1. Intuitively, the sample \( x_s \) at time \( s \) is generated from a Gaussian distribution; its mean is a linear interpolation between \( x_t \) and the denoised result \( \hat{x} \), 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., \( x^1:H \). To incorporate the condition \( c \) during sampling, one can employ classifier-based or classifier-free guidance [13, 26].

### Algorithm 1 DDIM sampling algorithm

1. **Input:** Denoiser \( D \), sample \( x_t \) at time \( t \), target time \( s \), condition \( c \), hyperparameter \( \eta \in [0, 1] \).
2. Compute the denoised result \( \hat{x} := D(x_t, t, c) \).
3. Obtain variance \( v_s \) as a scalar that depends on \( \eta \).
4. Obtain mean \( \mu_s \) as a linear combination of \( \hat{x} \) and \( x_t \):

\[
\mu_s := \hat{x} + \frac{\sqrt{\sigma_t^2 - v_s}}{\sigma_t} (x_t - \hat{x})
\]

5. Draw sample \( x_s \sim \mathcal{N}(\mu_s, v_s \mathbb{I}) \).

### 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 \( x^1:H \) 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 $P_{\pi}: \mathbb{R}^{H \times J \times D} \rightarrow \mathbb{R}^{H \times J \times D}$ as a module that maps the original motion $\tilde{x}_{1:H}$ to a physically-plausible one, denoted as $\hat{x}_{1:H}$. We incorporate the proposed physics-based projection $P_{\pi}$ 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.

1: **Input**: Denoiser $D$, sample $x_{1:H}^t$ at time $t$, condition $c$, target time $s$, physics-based projection $P_{\pi}, \eta \in [0, 1]$.
2: Compute the denoised motion $\hat{x}_{1:H} := D(x_{1:H}^t, t, c)$.
3: if projection is performed at time $t$ then
4: $\hat{x}_{1:H} := P_{\pi}(\hat{x}_{1:H})$  # Physics-Based Projection
5: else
6: $\hat{x}_{1:H} := x_{1:H}^t$
7: end if
8: # The remaining part is similar to DDIM
9: Obtain variance $v_s$ as a scalar that depends on $\eta$.
10: Obtain mean $\mu_s$:
11: $\mu_s := \bar{x}_{1:H} + \frac{\sigma_s^2 - v_s}{\sigma_t} (x_{1:H}^t - \hat{x}_{1:H})$
12: Draw sample $x_{1:H} = \mathcal{N}(\mu_s, v_s I)$.

Scheduling Physics-based Projection. Due to the use of physics simulation, the projection $P_{\pi}$ 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 diffuse noise level is high. This is because the denoiser $D$ by design gives us $\tilde{x}_{1:H} = \mathbb{E}[x_{1:H} | x_{1:H}^t]$, i.e., the mean motion of $x_{1:H}$ given the current noisy motion $x_{1:H}^t$, and it is close to the mean of the training data for high noise levels when the condition $x_{1:H}^t$ 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 $P_{\pi}$. It is tasked with projecting the denoised motion $\tilde{x}_{1:H}$ of a diffusion step, which disregards the laws of physics, into a physically-plausible motion $x_{1:H} = P_{\pi}(\tilde{x}_{1:H})$. The projection is achieved by learning a motion imitation policy $\pi$ that controls a simulated character to mimic the denoised motion $\tilde{x}_{1:H}$ in a physics simulator. The resulting motion $\hat{x}_{1:H}$ 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 $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, R, \gamma)$ 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 $\pi(a^h | s^h)$, which models the distribution of choosing an action $a^h \in \mathcal{A}$ given the current state $s^h \in \mathcal{S}$. The state $s^h$ consists of the character’s physical state (e.g., joint angles, velocities, positions) as well as the next pose $\bar{x}_{h+1}$ from the input motion. Including $\bar{x}_{h+1}$ in the state informs the policy $\pi$ to choose an action $a^h$ that can mimic $\bar{x}_{h+1}$ in the simulator. Starting from an initial state $s^1$, the agent iteratively samples an action $a^h$ from the policy $\pi$ and the simulator with transition dynamics $T(s^h+1 | s^h, a^h)$ generates the next state $s^{h+1}$, from which we can extract the simulated pose $\bar{x}_{h+1}$. By running the policy for $H$ steps, we can obtain the physically-simulated motion $\hat{x}_{1:H}$. 

Training. During training, a reward $r^h$ is also assigned to the character based on how well the simulated motion $\hat{x}_{1:H}$ aligns with the ground-truth motion $\tilde{x}_{1:H}$. Note that the motion imitation policy $\pi$ is trained on large motion capture datasets where high-quality ground-truth motion is available. We use reinforcement learning (RL) to learn the policy $\pi$, 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 $\hat{x}_{1:H}$ to match the ground truth $\tilde{x}_{1:H}$. Here, we use $\cdot$ to denote ground-truth quantities. The
reward \( r^h \) at each timestep consists of four sub-rewards:

\[
r^h = w_\phi r^\phi_h + w_\psi r^\psi_h + w_j r^j_h + w_q r^q_h,
\]

\[
r^\phi_h = \exp \left[ -\alpha_\phi \left( \sum_{i=1}^{J} \| \hat{o}^h_i \otimes \tilde{o}^h_i \|_2^2 \right) \right],
\]

\[
r^\psi_h = \exp \left[ -\alpha_\psi \| \hat{p}^h - \tilde{p}^h \|_2^2 \right],
\]

\[
r^j_h = \exp \left[ -\alpha_j \left( \sum_{i=1}^{J} \| p^h_i - \tilde{p}^h_i \|_2^2 \right) \right],
\]

\[
r^q_h = \exp \left[ -\alpha_q \left( \sum_{i=1}^{J} \| q^h_i - \tilde{q}^h_i \|_2^2 \right) \right].
\]

where \( w_\phi, w_\psi, w_j, w_q, \alpha_\phi, \alpha_\psi, \alpha_j, \alpha_q \) are weighting factors. The pose reward \( r^\phi_h \) measures the difference between the local joint rotations \( o^h_i \) and the ground truth \( \hat{o}^h_i \), where \( \otimes \) denotes the relative rotation between two rotations, and \( \| \cdot \|_2 \) computes the rotation angle. The velocity reward \( r^\psi_h \) measures the mismatch between joint velocities \( \hat{v}^h \) and the ground truth \( \tilde{v}^h \), which are computed via finite difference. The joint position reward \( r^j_h \) encourages the 3D world joint positions \( \hat{p}^h_i \) to match the ground truth \( \tilde{p}^h_i \). Finally, the joint rotation reward \( r^q_h \) measures the difference between the global joint rotations \( q^h_i \) and the ground truth \( \tilde{q}^h_i \).

**States.** The agent state \( s^h \) consists of the character’s current physical state, the input motion’s next pose \( \hat{x}^{h+1} \), and a character attribute vector \( \psi \). 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 \( \hat{x}^{h+1} \), the state \( s^h \) contains the difference of \( \hat{x}^{h+1} \) 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 \( \psi \) 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 \( \pi(a^h | s^h) = \mathcal{N}(\mu_\theta(s^h), \Sigma) \) where the mean action \( \mu_\theta \) is output by a simple multi-layer perceptron (MLP) network with parameters \( \theta \), and \( \Sigma \) 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?

**Evaluation Metrics.** For text-to-motion generation, we first use two standard metrics suggested by Guo et al. [22]: \( \text{FID} \) measures the distance between the generated and ground-truth motion distributions; \( R\text{-Precision} \) assesses the relevancy of the generated motions to the input text. For action-to-motion generation, we replace R-Precision with an \( \text{Accuracy} \) 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: \( \text{Penetrate} \) measures ground penetration; \( \text{Float} \) measures floating; \( \text{Skate} \) measures foot sliding; \( \text{Phys-Err} \) is an overall physical error metric that sums the three metrics (all in \( \text{mm} \)) 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.
Results. In Table 1, we compare our method to the SOTA methods: JL2P [1], Text2Gesture [9], T2M [22], Motion-Diffuse [92], and MDM [77]. Due to the plug-and-play nature of our method, we design two variants of PhysDiff using Motion-Diffuse (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.
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-

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

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

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.

**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 Phys-Diff 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


[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


