This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

POMP: Physics-constrainable Motion Generative Model through Phase Manifolds

Bin Ji[1,2], Ye Pan[2*] Zhimeng Liu[2], Shuai Tan[2], Xiaogang Jin[4], Xiaokang Yang[3]
[1]State Key Laboratory of Micro-Nano Engineering Science, Shanghai Jiao Tong University (SJTU)
[2]John Hopcroft Center, SJTU. [3]MoE Key Lab of Artificial Intelligence, SJTU. [4]Zhejiang University

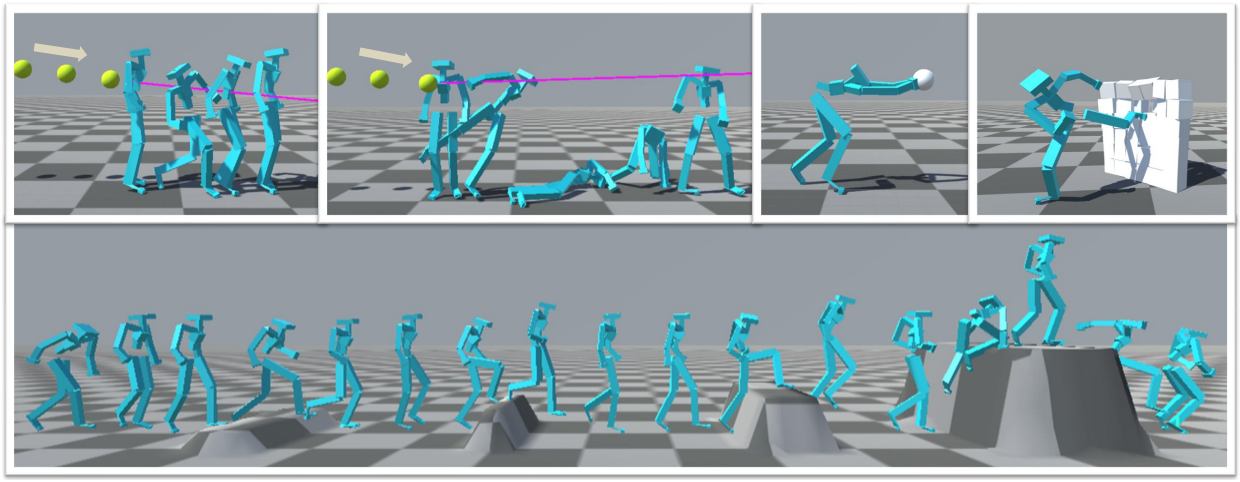


Figure 1. POMP generates physically plausible motions in real time, including locomotion patterns across various terrains and general acyclic behaviors such as collision responses, target tracking, and martial arts.

Abstract

Numerous researches on real-time motion generation primarily focus on kinematic aspects, often resulting in physically implausible outcomes. In this paper, we present POMP ("Physics-cOnstrainable Motion Generative Model through Phase Manifolds"), a kinematics-based framework that synthesizes physically realistic motions by leveraging phase manifolds to align motion priors with physics constraints. POMP operates as a frame-by-frame autoregressive model with three core components: a diffusion-based kinematic module, a simulation-based dynamic module, and a phase encoding module. At each timestep, the kinematic module first generates an initial pose, which is subsequently revised by the dynamic module through a simulation step to incorporate physical constraints. While individual simulation steps induce negligible kinematic distortion, accumulated discrepancies can drive the result beyond the motion prior learned by the kinematic module, leading to failure in subsequent motion generation. To address this, the phase encoding module applies semantic alignment in the phase manifold, projecting the simulated result back to the motion prior. Moreover, we present a pipeline in Unity for generating terrain maps and capturing full-body motion impulses from existing motion capture dataset. The collected terrain topology and motion impulse data facilitate the training of POMP, enabling it to robustly respond to underlying contact forces and applied dynamics. Extensive evaluations demonstrate the efficacy of POMP across various tasks.

1. Introduction

The generation of diverse and realistic virtual humans has drawn widespread interest in the field of computer animation [62, 63, 66–69]. Existing researches mainly focus on two major challenges: (1) improving the generalization of learned models over large-scale datasets, (2) creating realistic human-environment interactions.

To produce a wide range of motion content and styles in large datasets, kinematics-based methods are typically favored. This preference stems from the fact that existing motion datasets primarily capture humanoid poses, allowing kinematics-based approaches to study humanoid motion

[1]Ye Pan is corresponding author.

without considering underlying forces (e.g., gravity, contact force, joint torque). Moreover, recent advancements in generative models like Variational Autoencoders, Normalizing Flows, and Diffusion Models [4, 15, 20, 21, 29, 51, 64, 73, 81] have been easily integrated into kinematic models, greatly enriching their learned motion priors. However, as interactive environments grow more complex, kinematic motion priors often result in physically implausible artifacts like foot sliding, floating and penetration. To overcome these limitations, a promising paradigm incorporates physics principles into the learning process of motion priors. This kind of methods either rely on physics engines and train skill-specific controllers using deep reinforcement learning [33, 34], or design gradient-based optimization frameworks [50, 53] that refine kinematics-based estimates by aligning them with some physics principles. Nonetheless, these approaches often involve significant modeling and computational complexity, making them difficult to train and apply in practice. Recently, Jiang et al. [22] introduce DROP, a plug-in optimization-based simulator that adds dynamics capability to a pre-trained kinematic generative model. While this framework significantly reduces computational complexity by reusing a pre-trained kinematic model, it often generates unrealistic results due to the **domain gap**: the minimalist physics simulator's attempts to enforce physical plausibility on kinematic poses inadvertently compromise kinematic coherence. This inherent trade-off manifests in motions that, while theoretically achievable, deviate significantly from natural human movement patterns. To address the **domain gap** issue, in this paper, we aim to align both kinematic and simulated poses semantically in the phase manifold and generate physically and kinematically plausible motions in complex tasks.

We present POMP, an autoregressive architecture that enforces physical plausibility in real-time motion generation through decoupled kinematic-dynamic integration. POMP operates cyclically through three computational stages: (1) a kinematics module generates a target pose via the learned motion prior; (2) a dynamic module enforces dynamic constraints and produces a simulated pose; (3) a phase encoding module project the simulation pose back to the motion prior and achieve its phase positional encoding. These processed state vectors recursively informs subsequent pose generation. Notably, POMP requires training only for the kinematic generator and the phase encoding module, while the physical simulator functions as a fixed, non-differentiable component. This design offers three key advantages: first, it reduces computational complexity by keeping the resource-intensive dynamic module outside the gradient-based training loop. Second, it addresses the **domain gap** by conducting semantic alignment in the phase manifold. Third, it allows for broader applications, enabling quasi-physical effects without simulator-specific expertise.

The structure of each module is outlined as follows. To exploit the inherent periodic and semantic features of motion, we introduce a phase-guided diffusion model as the kinematic module. Drawing on Mixture of Expert (MoE) architectures used in PFNN [17], MANN [87] and NSM [56], our model employs an Ortho-MoE-based encoder to map motions into a non-linear phase embedding space. In this space, motion feature subdomains are assigned to individual expert branches for specialized learning. The Ortho-MoE module ensures these expert branches learn high-level semantic features in a disentangled and compact way through unsupervised orthogonality enforcement. The linear combination of these bases can represent various motion patterns, which subsequently guide the diffusion-based decoder to generate high-quality target poses in just five inference steps. For the plug-in dynamic module, we opt to use Nvidia PhysX in Unity over alternatives like Mujoco [74] or IsaacGym [39]. This choice is motivated by several advantages: (1). Unity allows a semi-automatic method to match the terrain for MoCap data; (2). PhysX in Unity enables impulse collection of full-body contact force between the character and the environment, which can be fed back to the kinematic module for learning active response to various physical perturbations; (3). Unity is compatible with games and VR/AR applications seeking quasi-physical effects. To mitigate the domain gap, We introduce aligned-PAE, inspired by the concept of periodic autoencoders (PAE) [59], as the phase encoding module. While PAE learns phase manifolds across different motion domains, our approach extends this by establishing correspondence between kinematic and simulated poses within identical motion content. Specifically, aligned-PAE first constructs a unified manifold utilizing only MoCap data and then relocates simulated poses within the same phase ellipse, but with different phase positional encodings.

In summary, our contributions are as follows:

- We introduce POMP, a novel kinematic-based generative model capable of performing active and passive responses to physical disturbances in real time.
- We are the first to tackle the domain gap between kinematic motion priors and simulated poses with a phase-based semantic alignment method.
- we develop a pipeline for terrain reconstruction and full-body motion impulses extraction from MoCap datasets.

2. Related Work

2.1. Kinematic-based motion generation

Creating realistic motions for virtual characters is a fundamental challenge in the computer animation. Advancements in human motion synthesis are accompanied by the development of various types of MoCap datasets, which are crucial for numerous motion-related tasks, such as

single-player motion reconstruction [23, 38, 49], human-human interaction [13, 24, 28, 83], human-object interaction [6, 25, 37, 60, 61, 80] and human-scene interaction [14, 18, 76, 90]. To efficiently learn human motion priors from these large-scale MoCap datasets, recent advancements in character animation primarily focus on deep neural networks. These approaches can be categorized into deterministic models and probabilistic models. Deterministic methods yield a single result for a given input, while probabilistic generative models attempt to describe the distribution of possible outcomes. For deterministic models, the seminal work of PFNN [17] synchronizes locomotion with the timeline using phase-related features, achieving high-quality and real-time motion reconstruction. Following this, the concept of phase has been applied to more intricate tasks like human-scene interactions [56], human-human interactions [57, 58, 65] and motion-in-betweening [55, 70]. However, deterministic predictions of diverse motion patterns often regress to the mean pose. Lately, probabilistic generative models such as Generative Adversarial Networks (GAN) [16, 26], Variational Autoencoders (VAE) [45, 51, 89] and Normalizing Flows [15, 20] have attracted substantial interest within the realm of motion generation to prevent regressing towards a sub-optimal pose. Furthermore, recent advancement in Diffusion Models has significantly advanced the development of motion-related downstream tasks such as text2Motion [4, 10–12, 32, 46, 47], music2Dance [1, 75] and gesture style control [2, 8, 9, 91]. However, as interactive environments become more complex, these kinematic motion priors are prone to artifacts like foot sliding, floating and penetration. To address these limitations, our method post-processes the kinematic result with a plug-in physical simulator that is separate from the training pipeline, ensuring no additional computational complexity.

2.2. Physics-based Motion Simulation

To enhance motion realism, the physics-based paradigm incorporates underlying forces (e.g., gravity, contact force, joint torque) to yield dynamics responses to changes in the human-environment interactions. Deep reinforcement learning has proven effective to simulate a range of motions such as general locomotion [5, 36, 42–44], sports [30, 31, 77, 88], juggling [7, 84] and human-environment interactions [52, 72, 78, 82]. These methods design composite controllers or hierarchical models to manage transitions between a limited number of skills. To extend the adaptability of trained controllers to larger datasets with more motions, some approaches integrate generative models into their policy designs, such as Variational Autoencoders [29, 79, 86] and Generative Adversarial Networks (GANs) [43, 44, 85]. However, their performance improvement comes at the cost of increasing modeling and computational complexity. To reduce this complexity, Jiang et al. [22] reuse an off-the-shelf generative model to predict a kinematic pose, then refine it using a plug-in motion simulator based on projective dynamics. However, this approach often yields unrealistic results due to the domain gap between kinematic motion priors and simulated poses. In contrast, we introduce a phase encoding module that performs semantic alignment, bridging the gap and producing results that are both physically and kinematically plausible.

3. Methodology

POMP operates through a cyclical process, as shown in Fig. 2 (a). The process initiates in the kinematic module, which generates the target pose $\tilde{\boldsymbol{q}}_{\boldsymbol{i}}$ based on the current motion variables $\boldsymbol{x}_{\boldsymbol{i}}$. The target pose is then processed by the dynamic module, where a proportional-derivative (PD) controller calculates the joint force vector $\boldsymbol{\tau}_{\boldsymbol{i}}$. Utilizing $\boldsymbol{\tau}_{\boldsymbol{i}}$, a plug-in physics engine performs forward dynamics, yielding both the physically plausible pose $\boldsymbol{q}_{\boldsymbol{i}}$ and associated full-body contact impulses $\boldsymbol{j}_{\boldsymbol{i+1}}$. The final component, the phase encoding module, extracts the phase variable $\boldsymbol{p}_{\boldsymbol{i}}$ corresponding to the simulated pose $\boldsymbol{q}_{\boldsymbol{i}}$. This phase variable, together with the updated motion variables $\boldsymbol{x}_{\boldsymbol{i+1}}$, is then used as input for the kinematic module in the subsequent iteration of the process.

3.1. Preliminary

Physics-based Character Modeling. We utilize an articulated rigid-body system for collision detection and force sensing. This system comprises 15 box or capsule limb colliders in Unity to handle environmental collisions. Our setup uses fewer collision bodies than actual limbs, demonstrating that this approach is versatile and can be adapted to characters with more complex joint structures. To reduce the computational demands in collision detection and dynamic parameter calculations, limb colliders are simple geometric shapes with low polygon counts, such as boxes and capsules. The model size is based on the CMU database, while its mass is derived from the volume of these colliders using average human body density [48]. To enhance realism, we implement joint limits to make the articulated body's movements more human-like.

Rigid Body Dynamics. To integrate laws of physics into our character, the dynamic module of POMP performs two specific types of calculations: inverse dynamics operated by the motion controller to compute the forces applied to the rigid-body system, and forward dynamics operated by the physics simulator to determine the acceleration response of our character to these applied forces in the environment.

We utilize a PD-servo as our motion controller. At each timestamp, vectors of position, velocity and acceleration are

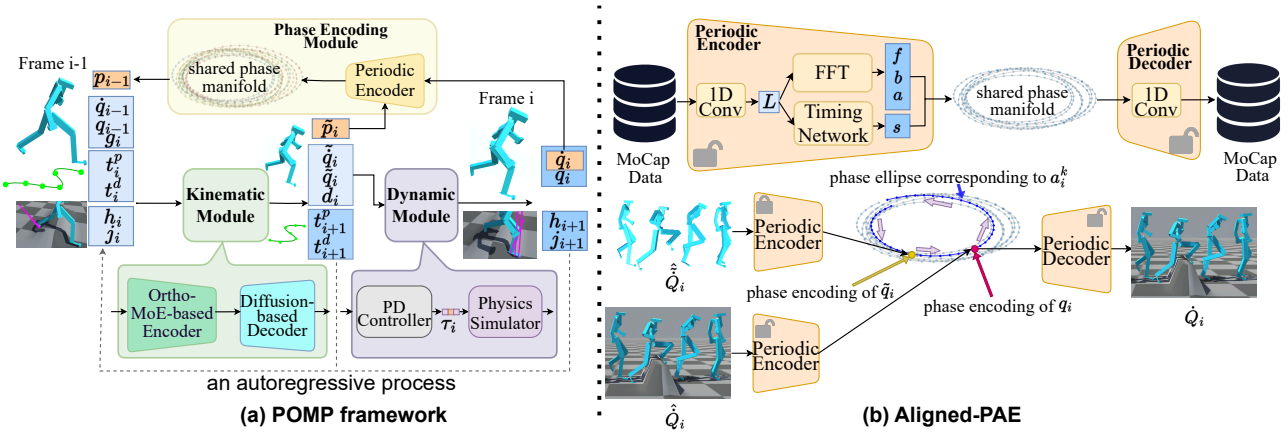


Figure 2. (a) **The overview of POMP.** POMP consists of three components. (1). a kinematic module to generate a target pose based on the previous simulation result; (2). a dynamic module that performs inverse and forward dynamics to mimic the target pose and produce the current simulated pose; (3). a phase encoding module that produces a phase encoding to guide the generation of the next target pose. (b) **Aligned-PAE.** We first learn a shared phase manifold using only the MoCap dataset, then align motion priors and simulation results of the same motion type to different timings within the same phase ellipse.

denoted as $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}} \in \mathbb{R}^n$, respectively. Given a target pose $\tilde{\boldsymbol{q}}$ generated by the kinematic module, the internal joint force vector $\boldsymbol{\tau} \in \mathbb{R}^n$ is calculated as:

$$\boldsymbol{\tau} = \boldsymbol{k_p} \circ (\tilde{\boldsymbol{q}} - \boldsymbol{q}) - \boldsymbol{k_d} \circ \dot{\boldsymbol{q}}, \tag{1}$$

where $\boldsymbol{k_p}$ and $\boldsymbol{k_d}$ are gain parameters of PD controllers and "$\circ$" denotes Hadamard matrix product. The computed torque $\boldsymbol{\tau}$ drives the plug-in physics simulator's forward dynamics to obtain the acceleration vector $\ddot{\boldsymbol{q}}$. $\ddot{\boldsymbol{q}}$ is then used to update the $\mathbf{q}$, $\dot{\mathbf{q}}$ and produces the simulated result.

Motion Representation. Motion variables $\boldsymbol{x_i}$ for frame i consist of four parts: character state, environmental information, phase variables and trajectory location. The character state is defined by positions $\boldsymbol{q_i} \in \mathbb{R}^{3N_J}$ and velocities $\dot{\boldsymbol{q}}_{\boldsymbol{i}} \in \mathbb{R}^{3N_J}$ for N_J body joints, root transformation $\boldsymbol{d_i} \in \mathbb{R}^3$ (covering root velocity on the x-z plane and angular velocity around the y-axis rotation) and one-hot gait vectors $\boldsymbol{g_i} \in \mathbb{R}^{8L}$ indicating the character's gait type (e.g., standing, walking, jogging, crouching, jumping, bumping, lying, getting up). Here, $L = 12$ represents a time window length centered at frame i, sampling past and future motion states every 10 frames. Environmental information comprises surrounding terrain heights $\boldsymbol{h_i} \in \mathbb{R}^{3L}$ and full-body contact impulses $\boldsymbol{j_i} \in \mathbb{R}^{3N_B}$, where N_B is the number of limb colliders. Phase variables serve as both input and output for the kinematic module. Input phase variables $\boldsymbol{p_i} \in \mathbb{R}^{2\times C\times(L+1)}$ contain phase encoding derived from $C = 5$ phase latent channels over the entire time window. The kinematic module, however, only predicts output phase variables for the future half of the time window $\tilde{\boldsymbol{p}}_{\boldsymbol{i}} \in \mathbb{R}^{5\times C\times(0.5L+1)}$, consisting of phase encoding to locate motion in the phase manifold and phase parameters to parameterize latent curves. $\boldsymbol{p_i}$ and $\tilde{\boldsymbol{p}}_{\boldsymbol{i}}$ are extracted through aligned-PAE, with details provided in the following sections. Similarly, trajectory positions $\boldsymbol{t_i^p} \in \mathbb{R}^{2L}$ and directions $\boldsymbol{t_i^d} \in \mathbb{R}^{2L}$ represent the user-controlled trajectory of the entire time window projected onto the 2D x-z plane, while the predicted trajectory variables $\tilde{\boldsymbol{t}}_{\boldsymbol{i}}^{\boldsymbol{p}} \in \mathbb{R}^L$ and $\tilde{\boldsymbol{t}}_{\boldsymbol{i}}^{\boldsymbol{d}} \in \mathbb{R}^L$ cover only the future half.

3.2. Kinematic Module

The kinematic module, illustrated in Fig. 2 (a), utilizes an encoder to encode motion variables $\boldsymbol{x_i} = \{\boldsymbol{p_{i-1}}, \dot{\boldsymbol{q}}_{\boldsymbol{i-1}}, \boldsymbol{q_{i-1}}, \boldsymbol{g_i}, \boldsymbol{t_i^p}, \boldsymbol{t_i^d}, \boldsymbol{h_i}, \boldsymbol{j_i}\} \in \mathbb{R}^{541}$ into the phase domain. This transformation enhances movement synchronization over time [56, 57]. Specifically, the encoder comprises two core modules: a gating network E_g and a 3-layer Ortho-MoE network E_m. E_g takes phase variables $\boldsymbol{p}_{i-1}$ from $\boldsymbol{x}_i$ as input to calculate phase blending coefficients $\{\alpha_k\}_{k=1}^8 \in \mathbb{R}^8$, which enable the allocation of feature subdomains to specialized expert branches within E_m. E_m comprises eight MLP-based branches, with each k-th branch functioning as an expert that learns the motion principle component $\boldsymbol{w_k} \in \mathbb{R}^{1024}$ within its corresponding subdomain. The final phase motion embedding of $\boldsymbol{w} \in \mathbb{R}^{1024}$ is computed as $\boldsymbol{w} = \sum_{k=1}^8 \alpha_k \boldsymbol{w_k}$, representing a linear combination of principle components across eight phase channels. To address the challenge inherent in learning compact and disentangled motion representations across diverse terrain, we implement orthogonality constraints between principle component pairs, defined as $\langle \boldsymbol{w_i}, \boldsymbol{w_j} \rangle = 0$ for $i \neq j$, where $\langle \cdot, \cdot \rangle$ signifies the dot product. This orthogonality constraint ensures optimal encoding of non-linear motion periodicity variations through eight phase channels, yield-

ing a phase motion embedding that effectively preserves essential motion characteristics.

Subsequently, we implement a diffusion model for generating output variables $\boldsymbol{y_i} = \{\tilde{\boldsymbol{p}}_\boldsymbol{i}, \tilde{\dot{\boldsymbol{q}}}_\boldsymbol{i}, \tilde{\boldsymbol{q}}_\boldsymbol{i}, \boldsymbol{d_i}, \tilde{\boldsymbol{t}}^{\boldsymbol{p}}_{\boldsymbol{i+1}}, \tilde{\boldsymbol{t}}^{\boldsymbol{d}}_{\boldsymbol{i+1}}\} \in \mathbb{R}^{388}$. In contrast to previous approaches, POMP executes the diffusion process within the phase domain, conditioned on the motion phase embedding $\boldsymbol{w}$. The motion decoder operation can be formally expressed as:

$$z_0 = D(\boldsymbol{w}, z_N), \tag{2}$$

where $z_N \sim \mathcal{N}(0, \mathbf{I})$ represents Gaussian noise, $z_0 = \{z_k\}_{k=1}^{8} \in \mathbb{R}^{8\times 388}$ is the set of motion codes, $\boldsymbol{y}_i$ is a linear combination of these codes, denoted as $\boldsymbol{y}_i = \sum_{k=1}^{8} \alpha_k \boldsymbol{z_k}$.

The diffusion model begins with a forward process $q(z_{1:N}|z_0)$ to introduce noise into the motion code distribution $q(z_0)$, followed by a reverse process for distribution recovery learning. While the forward process involves N steps, recent advancements through DDIM [54] have accelerated the reverse process to within $dim(\tau)$ steps:

$$\begin{aligned} &q_\lambda(z_{\tau_{n-1}}|z_{\tau_n}) \approx q_\lambda(z_{\tau_{n-1}}|z_{\tau_n}, z_0 = \hat{\mu}_\theta(z_{\tau_n}, \tau_n)) \\ &= \mathcal{N}(z_{\tau_{n-1}}|\tfrac{\sqrt{\bar{\alpha}_{\tau_{n-1}}}}{\sqrt{\bar{\alpha}_{\tau_n}}}\left(z_{\tau_n} - \left(\sqrt{\bar{\beta}_{\tau_n}} - \tfrac{\sqrt{\bar{\alpha}_{\tau_{n-1}}}}{\sqrt{\bar{\alpha}_{\tau_n}}}\sqrt{\bar{\beta}_{\tau_{n-1}} - \lambda^2_{\tau_n}}\right)\cdot\epsilon_\theta, \lambda^2_{\tau_n}\mathbf{I}\right), \end{aligned} \tag{3}$$

$$\hat{\mu}_\theta(z_{\tau_n}, \tau_n) = \frac{1}{\sqrt{\bar{\alpha}_{\tau_n}}}(z_{\tau_n} - \sqrt{\bar{\beta}_{\tau_n}}\epsilon_\theta(z_{\tau_n}, \tau_n)), \tag{4}$$

where α_n and β_n are scalars with $\alpha_n = 1 - \beta_n$, $\bar{\alpha}_n = \prod_{i=1}^{n} \alpha_i$ and $\bar{\beta}_n = 1 - \bar{\alpha}_n$. $\mathbf{I}$ is the identity matrix, $\lambda = (\lambda_1, \cdots, \lambda_N) \in \mathbb{R}^N_{\geq 0}$ represents a set of unspecified non-negative coefficients. $\tau = [\tau_1, \tau_2, ..., \tau_{dim(\tau)}]$ is a subsequence of $[1, ..., N]$ with length $dim(\tau)$, $\epsilon_\theta(z_{\tau_n}, \tau_n)$ is a neural network with learnable parameters θ that predicts noise from z_{τ_n} and step index τ_n. While limited iteration steps in the reverse process can greatly improve computational efficiency, it often compromises motion quality. To speed up the reverse process without sacrificing quality, we draw inspiration from Analytic-DPM [3] by replacing the covariance term $\lambda^2_{\tau_n}\mathbf{I}$ in Eq. (3) with an analytic estimate $(\lambda^2_{\tau_n} + (\sqrt{\bar{\alpha}_{\tau_n - 1}} - \sqrt{\bar{\alpha}_{\tau_n}}\kappa_{\tau_n})^2\bar{\sigma}^2_{\tau_n})\mathbf{I}$, where $\kappa_{\tau_n} = \frac{\sqrt{\bar{\beta}_{\tau_{n-1}} - \lambda^2_{\tau_n}}}{\sqrt{\bar{\beta}_{\tau_n}}}$, $\bar{\sigma}^2_{\tau_n}$ can be estimated using the Monte Carlo method: $\bar{\sigma}^2_{\tau_n} = \frac{\bar{\beta}_{\tau_n}}{\bar{\alpha}_{\tau_n}}\left(1 - \frac{1}{d}\mathbb{E}_{z_{\tau_n}\sim q(z_{\tau_n})}\left[\|\epsilon_\theta\|^2\right]\right)$, and d is the dimension of z_{τ_n}. To take $\boldsymbol{w}$ into account, the reverse step $q_\lambda(Z^c_{n-1}|Z^c_n)$ becomes conditional, $q_\lambda(Z^c_{n-1}|Z^c_n, \boldsymbol{w})$. Therefore, we utilize a neural network to predict the noise conditioned on $\boldsymbol{w}$, expressed as: $\epsilon_\theta(z_{\tau_n}, \tau_n, \boldsymbol{w})$. To ensure compatibility with ONNX format, $\epsilon_\theta(z_{\tau_n}, \tau_n, \boldsymbol{w})$ incorporates simple operations, such as Multi-Layer Perceptron (MLP), 1D convolution (Conv1d), Layer Normalization (LN) and Sigmoid Linear Unit (SiLU).

The kinematic module is trained using the reconstruction loss between the ground truth and predicted output vector:

$$\mathcal{L}_k = \mathcal{L}_{mse}(\hat{\boldsymbol{y}}_\boldsymbol{i}, \boldsymbol{y_i}) \tag{5}$$

3.3. Dynamic Module

The kinematic module demonstrates robust performance in maintaining both semantic and temporal coherence of generated motions. To address physical disturbances, we implement a dynamic module that integrates both inverse and forward dynamics to transform $\boldsymbol{y}_i$ into a physically plausible pose. The module's pipeline initiates with a PD controller that processes three inputs: the target pose $\tilde{\boldsymbol{q}}_i$, previous pose $\boldsymbol{q}_{i-1}$ and velocity $\dot{\boldsymbol{q}}_{i-1}$, to compute the internal joint force vector $\boldsymbol{\tau}_i \in \mathbb{R}^{3N_J}$ as defined in Eq. (1). Subsequently, a plug-in physics simulator computes external force interactions and executes forward dynamics to determine the character's acceleration. The calculated acceleration updates the previous motion state $\{\dot{\boldsymbol{q}}_{\boldsymbol{i-1}}, \boldsymbol{q}_{\boldsymbol{i-1}}\}$ and achieves the final simulated result $\{\dot{\boldsymbol{q}}_\boldsymbol{i}, \boldsymbol{q}_i\}$.

Furthermore, our dynamic module enhances existing Motion Capture (MoCap) data through a two-phase process: terrain reconstruction based on MoCap data, followed by contact force acquisition and pose simulation through the physics simulator. The terrain reconstruction process begins by implementing the terrain fitting approach proposed by Holden et al. [17] to compute the trajectory height $\hat{\boldsymbol{h}}_\boldsymbol{i}$ surrounding the pose $\hat{\tilde{\boldsymbol{q}}}_\boldsymbol{i}$ per frame. We then initialize a zero-valued height map, and use $\hat{\boldsymbol{h}}_\boldsymbol{i}$ of the moving trajectory to gradually "scan" and update the height map, culminating in a comprehensive terrain data. After terrain generation, we import MoCap data into the Unity scene, where the physics simulator performs two functions: (1) collecting ground truth contact impulses $\hat{\boldsymbol{j}}_\boldsymbol{i}$ and (2) generating ground truth simulated poses $\hat{\boldsymbol{q}}_\boldsymbol{i}$. $\hat{\boldsymbol{j}}_\boldsymbol{i}$ are then used to train the kinematic module to adaptively respond to physical disturbances, while $\hat{\boldsymbol{q}}_\boldsymbol{i}$ are used to train the following phase encoding module. To the best of our knowledge, we are the first to utilize simulator in Unity to accomplish the entire process of terrain matching, dynamic feature acquisition, and achieve real-time physics-like effects.

3.4. Phase Encoding Module

The dynamic module of POMP generates a physically plausible pose $\boldsymbol{q}_\boldsymbol{i}$. However, this pose may fall outside the kinematic motion prior, creating a domain gap that potentially disrupts motion continuity in the generation of the next target pose $\tilde{\boldsymbol{q}}_{\boldsymbol{i+1}}$ from $\boldsymbol{q}_\boldsymbol{i}$. Given POMP's frame-by-frame autoregressive architecture, this error could accumulate, potentially leading to motion generation failure. To address the domain gap, we propose to semantically align kinematic and dynamic poses within a shared phase manifold. Inspired by dance performances, where musical accompaniment ensures temporal synchronization among dancers, we view the phase manifold as a "music repository", with individual phase ellipses representing distinct "songs" and the phase encoding serving as the "rhythm" within each "song".

Our initial step involves modeling a 1D manifold with

five channels. We train the phase manifold using only MoCap data to effectively capture the continuous and cyclical nature of various motions. As is shown in Fig. 2 (b), given a ground truth kinematic velocity sequence $\hat{\dot{Q}}_i \in \mathbb{R}^{\mathcal{T}\times 3N_J}$ centered at $\hat{\dot{\boldsymbol{q}}}_{\boldsymbol{i}}$, the periodic encoder first utilizes a 1D convolutional layer E_p to encode the sequence into a lower-dimensional embedding, denoted as $\hat{\tilde{L}}_i = E_p(\hat{\dot{Q}}_i) \in \mathbb{R}^{\mathcal{T}\times 5}$, where the time window $\mathcal{T}^{1s}_{-1s} = 121$. This embedding is defined by four parameters: amplitude $\tilde{\boldsymbol{a}}_{\boldsymbol{i}} \in \mathbb{R}^5$, frequency $\tilde{\boldsymbol{f}}_{\boldsymbol{i}} \in \mathbb{R}^5$, offset $\tilde{\boldsymbol{b}}_{\boldsymbol{i}} \in \mathbb{R}^5$, and phase shift $\tilde{\boldsymbol{s}}_{\boldsymbol{i}} \in \mathbb{R}^5$. Following the work of PAE [59], we apply a differentiable real Fast Fourier Transform (FFT) layer to extract $\{\tilde{\boldsymbol{a}}_{\boldsymbol{i}}, \tilde{\boldsymbol{f}}_{\boldsymbol{i}}, \tilde{\boldsymbol{b}}_{\boldsymbol{i}}\}$, while using an MLP to predict $\tilde{\boldsymbol{s}}_{\boldsymbol{i}}$ alone. The embedding is then parameterized as $\tilde{L}_i = \tilde{\boldsymbol{a}}_{\boldsymbol{i}} \cdot sin(2\pi \cdot (\tilde{\boldsymbol{f}}_{\boldsymbol{i}} \cdot \mathcal{T} - \tilde{\boldsymbol{s}}_{\boldsymbol{i}})) + \tilde{\boldsymbol{b}}_{\boldsymbol{i}}$. This reconstructed embedding is subsequently mapped back to the input velocity curves using a 1D deconvolution layer D_p, resulting in $\tilde{\dot{Q}}_i = D_p(\tilde{L}_i)$. The Network is trained unsupervised using:

$$\mathcal{L}_p^{(1)} = \mathcal{L}_{mse}(\tilde{\dot{Q}}_i, \hat{\tilde{\dot{Q}}}_i). \tag{6}$$

After training, phase encoding $\boldsymbol{pe_i} \in \mathbb{R}^{10}$ is computed by $\boldsymbol{pe_i}^{(2t)} = \boldsymbol{a_i}^{(t)} \cdot cos(2\pi \cdot \boldsymbol{s_i}^{(t)})$ and $\boldsymbol{pe_i}^{(2t+1)} = \boldsymbol{a_i}^{(t)} \cdot sin(2\pi \cdot \boldsymbol{s_i}^{(t)})$, where t is the channel index.

Then we introduce a novel training strategy to bridge the domain gap between $\tilde{\dot{\boldsymbol{q}}}_{\boldsymbol{i}}$ and $\dot{\boldsymbol{q}}_{\boldsymbol{i}}$ by: (1). identifying the appropriate "song" (phase ellipse) for $\tilde{\dot{\boldsymbol{q}}}_{\boldsymbol{i}}$'s motion type from the pre-established "music repository" (phase manifold); (2). aligning $\dot{\boldsymbol{q}}_{\boldsymbol{i}}$ to the appropriate "rhythm" (phase encoding) within this phase ellipse. The intuition is that even if the simulated result diverges from the kinematic motion prior, it retains the same motion type as the target pose, albeit with different timing. Within the phase manifold, $\boldsymbol{a}$ encodes different motion types, where motions of the same type theoretically map to the phase ellipse corresponding to the same $\boldsymbol{a}$. Unlike VQ-PAE [27], which learns a discrete amplitude space using vector quantization, our proposed aligned-PAE learns a continuous amplitude space, ensuring smooth transitions between motion types. Additionally, $\boldsymbol{f}$ represents movement frequency, $\boldsymbol{b}$ denotes range of motion, and $\boldsymbol{s}$ indicates timing within the ellipse. The implementation is as follows: $\hat{\tilde{\dot{Q}}}_i \in \mathbb{R}^{\mathcal{T}\times 3N_j}$ is first fed into a pre-trained periodic encoder to generate the amplitude parameter $\boldsymbol{a}^{\boldsymbol{k}}_{\boldsymbol{i}}$. Next, the ground truth simulated velocity sequence $\hat{\dot{Q}}_i \in \mathbb{R}^{\mathcal{T}\times 3N_j}$ centered at $\hat{\dot{\boldsymbol{q}}}_{\boldsymbol{i}}$ is input into aligned-PAE to train both the periodic encoder and decoder as described in the initial step. Unlike the initial step, this training process also aligns the kinematic and simulated poses within the same ellipse. Therefore, the loss function employed in this stage can be expressed as follows:

$$\mathcal{L}_p^{(2)} = \mathcal{L}_{mse}(\dot{Q}_i, \hat{\dot{Q}}_i) + \mathcal{L}_{mse}(\boldsymbol{a}^{\boldsymbol{k}}_{\boldsymbol{i}}, \boldsymbol{a}^{\boldsymbol{s}}_{\boldsymbol{i}}) \tag{7}$$

where $\boldsymbol{a}^{\boldsymbol{s}}_{\boldsymbol{i}}$ is the amplitude of $\dot{Q}_i$. During inference, $\tilde{\boldsymbol{p}}_{\boldsymbol{i}}$ contains two components: $\boldsymbol{pe} \in \mathbb{R}^{2\times C\times(0.5L+1)}$ and $\{\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{b}}\} \in \mathbb{R}^{3\times C\times(0.5L+1)}$. These components allow us to reconstruct $\tilde{L}_i$ and compute $\tilde{\dot{Q}}_i$ as $\tilde{\dot{Q}}_i = D_p(\tilde{L}_i)$. Since simulated poses for future frames are unavailable, $\dot{Q}_i$ is constructed with three parts: simulated poses $\{\dot{q}_{i-60}, ..., \dot{q}_i\}$, artificial simulated poses $\{\dot{q}'_{i+1}, ..., \dot{q}'_{i+20}\}$ and target poses $\{\tilde{\dot{q}}_{i+21}, ..., \tilde{\dot{q}}_{i+60}\}$. Artificial simulated poses are derived using the residual acceleration $\Delta a = (\dot{q}_i - \tilde{\dot{q}}_i)/\Delta t$. Δa is introduced by the simulator to correct artifacts like foot sliding and penetration. Given that human response latency is about 0.3s, we apply Δa only to the following 20 frames.

4. Experiments

4.1. Datasets and Implementation Details

POMP is evaluated over three datasets that vary in terms of size and motion diversity:

1. **human-terrain interaction dataset** is based on the terrain-fitting dataset [17], containing over 4 million frames representing diverse motion types across various terrains. We further expand the dataset by incorporating terrain height maps and motion impulses and sampling about 90,000 additional frames of more motion types such as falling, getting up and rebalancing.
2. **100STYLE** comprises over 4 million frames of stylized locomotion data, encompassing 100 distinct styles [41].
3. **AMASS** is a large-scale motion capture database. We follow the same filtering process as PHC [34] to obtain a filter dataset with about 4 million frames.

We implement POMP using Pytorch, setting $N_J = 31$, $N_B = 15$. The forward diffusion process consists of $N = 1000$ steps, with the non-negative coefficient computed as $\lambda_n = \frac{\bar{\beta}_{n-1}}{\beta_n}\beta_n$. The reverse process involves $dim(\tau) = 5$ iterative steps. We first train the phase encoding module and then freeze its weights and jointly train the kinematic module. For training, we use the AdamWR optimizer with cosine annealing warm-restart scheduling, setting the iterations to 10 and the restart factor to 2.0. Both the learning rate and weight decay are initialized to 10^{-4}, with a batch size set of 1024 and a dropout rate of 0.3. Training and testing are conducted on an RTX 3090 graphics card.

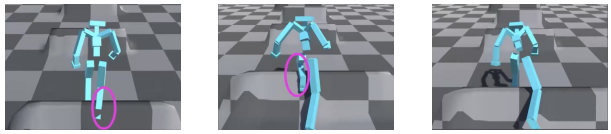
4.2. Physical Responses Across Different Motions

In Fig. 1, we illustrate POMP's ability to generate physically realistic motions across various tasks. More details are available in the supplementary video.

Bump Reactions. POMP can react to collisions of varying intensities. For minor impacts that don't affect the center of mass (CoM), affected limbs gradually recover their original motion patterns under the influence of the initial τ_i. As collision forces increase to displace the CoM, and compromise balance, POMP actively adjusts τ_i to restore

Models	FMD ↓	Foot Skating ↓	Runtime ↓	#Params ↓	#Styles ↑
PFNN	10.25	0.03	**7.55**	**2.39M**	*
POMP-MoE	1.42	0.08	42.42	22.45M	*
Ours(human-terrain)	**0.92**	**0.01**	42.42	22.45M	*
Mason et al.	6.98	0.27	16.31	4.32M	58
StyleERD	2.31	0.57	22.85	**0.48M**	7
RTSM	3.15	0.20	8.30	18.42M	100
Motion Puzzle	9.44	0.41	95.20	37.16M	100
Ours(100STYLE)	**1.02**	**0.06**	51.31	55.46M	100

Table 1. Quantitative comparisons with baselines and SOTA kinematic real-time methods.



(a) w/o dynamic module (b) w/o PEM (c) POMP

Figure 3. **Qualitative evaluation of DM/PEM components.** (a) pose penetration artifacts without DM; (b) motion generation failures due to domain gap without PEM; (c) successful obstacle crossing through phase manifold alignment.

stability. In case of an overwhelming force that knocks the character down, POMP temporarily loses control. Once grounded, POMP will control the character to get back up.

Target Tracking. POMP effectively mitigates the effect of τ_i on the target joint and proportionally reduce its influence on the related limbs. Then, an external force f is applied to the target end effector (e.g., left/right hand) to pull it toward the position constraint. This approach ensures that f influences the relevant limbs while maintaining the overall motion continuity under the control of τ_i.

Complex human-scene interactions. As shown in Fig. 1, POMP not only enables the synthesis of physically realistic locomotion across diverse terrains but also supports the generation of acyclic behaviors, such as martial arts sequences.

4.3. Ablation Study

We evaluate POMP's core components: ortho-MoE, dynamic module (DM), phase encoding module (PEM) and Aligned-PAE. Two types of metrics are used: (1). motion quality, assessed by Fréchet Motion Distance (FMD) [70] for distribution similarity, foot skating [87] and penetration [50] metrics for physical artifacts; (2). model complexity, assessed by the number of model parameters and runtime.

Ortho-MoE. To evaluate the capacity of Ortho-MoE in learning motion representations, we conduct experiments using the standard MoE, denoted as POMP-MoE. The evaluation is performed on the human-terrain dataset, which encompasses diverse aperiodic motions. As shown in Tab. 1, the results indicate that integrating Ortho-MoE into POMP enhances motion quality without introducing additional computational cost or increased model parameters.

Dynamic Module. Fig. 3 (a) highlights the critical role

	PFNN	POMP-wo-DM	POMP
Penetration	14.4%	12.7%	**0%**

Table 2. Quantitative evaluation on DM component.

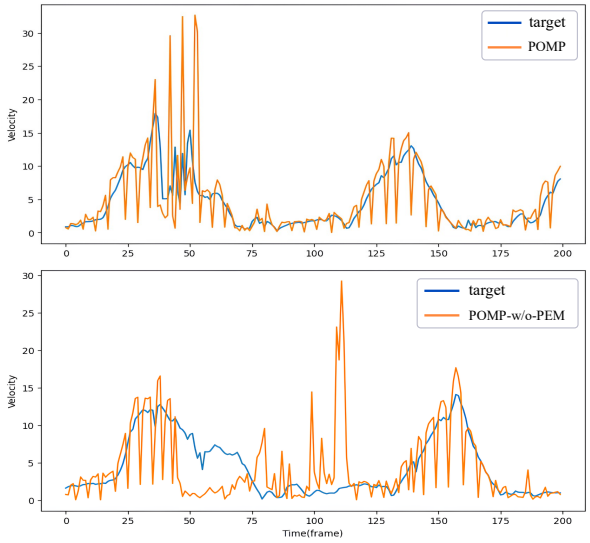


Figure 4. **Evaluation on PEM.** The kinematic module generate the target pose for the dynamic module to follow. The upper plot demonstrates velocity adaptation under PEM-guided kinematic-dynamic alignment, while the lower plot exhibits unconstrained velocity drift in the absence of PEM constraints.

of the dynamic module in our framework. The comparative analysis reveals that omitting the dynamic module leads to generated poses with pronounced interpenetration artifacts. To quantify this, we evaluate foot penetrations on complex terrains, as detailed in Tab. 2. The results indicate that POMP-wo-DM offers little advantage over SOTA method. In contrast, POMP effectively prevents penetrations by employing a physics simulator in the dynamic module.

PEM. Fig. 3 (b)-(c) illustrate PEM's role during motion mode transitions. In this experiment, a terrain obstacle introduced at 40th frame triggers a walking-to-traversal transition. We further quantify PEM's efficacy in this test through velocity trajectory alignment between simulated and target poses. As shown in Fig. 4, PEM enables projection of the previous frame's simulation results into the motion prior manifold, allowing the kinematic module to adaptively refine target poses and maintain velocity trajectory alignment during motion transitions. While discrete sampling in the physics simulator induces trajectory oscillations, PEM ensures stable alignment. In contrast, PEM ablation leads to progressive divergence (frames 40-110), as the kinematic module fails to timely adjusting the target pose according to the simulated pose that gradually moves away from the learned motion prior.

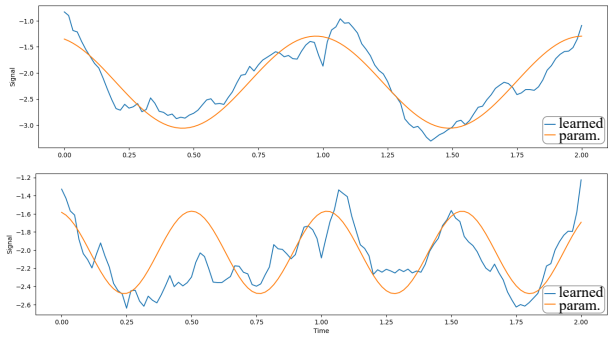


Figure 5. **Evaluation on aligned-PAE.** This visualization depicts a representative channel of the phase manifold. The upper panel presents the learned phase embedding curve and its parameterized representation during "normal walking" . The lower panel quantifies the variations in phase frequencies and phase shifts during the gait transition from "walking" to "obstacle crossing".

Aligned-PAE. In this stage, we assess the representational capacity of Aligned-PAE. Fig. 5 displays the learned and parameterized low-dimensional embedding curves for two action modes in phase manifold: "steady-state walking" and "walking-to-obstacle-crossing transition". The obstacle interaction initiates at 60th frame. During the initial 60 frames, learned embedding curves of two motion types exhibit similar trajectories, reflecting their shared "walking" state. However, once encountering the obstacle at 60 frame, the learned curve of the second mode changes rapidly. This change is evident in the parameterized curves: the increased frequency and phase shift of second mode relative to the steady-walking one. The experimental results confirm that the aligned PAE can indeed learn the phase characteristics of different action modes.

4.4. Comparative Performance

We perform a quantitative evaluation to compare POMP with state-of-the-art (SOTA) real-time motion generation methods including PFNN [17], Mason et al. [40], StyleERD [71], RTSM [41] and Motion Puzzle [19].

Tab. 1 presents the quantitative comparison results. POMP outperforms all baseline methods in terms of motion quality. Although StyleERD has fewer model parameters, its capability is restricted to generating only seven types of flat-ground motions. In contrast, POMP produces 100 types of motions on complex terrains and can further enable unseen motion style editing via target tracking. Although PFNN achieves computational efficiency with faster runtime and reduced parameter count, it underperforms in the FMD metric due to its deterministic nature, often yielding motions that regress to the mean pose and lack detail. POMP addresses this limitation through its phase-guided diffusion model, which effectively captures the authentic kinematic distribution of human motions. Despite requiring more computational resources and model parameters, as is shown in the supplementary video, POMP's ONNX format can run in real time on an 4060 GPU, making the trade-off between performance and model complexity worthwhile.

Models	train success ↑	MPJPE ↓	test success ↑	MPJPE ↓
Ours	**100%**	**21.5**	**100%**	**29.8**
PHC	**100%**	26.6	99.2%	36.1
MaskedMimic	99.4%	32.9	99.2%	35.1
PULSE	99.8%	39.2	97.1%	54.1

Table 3. Quantitative comparisons with physics-based methods.

Furthermore, we conduct a comparative evaluation between POMP and physics-based methods on the AMASS dataset using two evaluation metrics: (1). the success metric proposed by [34], which classifies a trial as unsuccessful if any simulated joint exceeds a 0.5-meter deviation from the target pose; (2). MPJPE (Mean Per Joint Position Error, in millimeters), which quantifies the precision of joint position alignment. The comparative methods include PHC [34], PULSE [35] and Maskedmimic [72], with detailed results presented in Tab. 3. POMP demonstrates superior performance in both generalization capability and motion quality compared to these RL-based approaches. Notably, POMP achieves remarkable training efficiency, requiring only 17.2 hours on a single 3090 GPU, compared to PHC's one-week training on an A100 GPU [34] and Maskedmimic's two-week training on four A100 GPUs [72]. Additional qualitative comparisons are provided in the supplementary video.

5. Conclusion

We present POMP, a framework to generate physically realistic motion in real time. POMP combines kinematic and dynamic models in a straightforward yet effective manner. Instead of integrating physics simulation into a gradient-based training loop, POMP first uses a kinematic generative model to provide a target pose for the physics simulator. The simulation result is then fed back into the kinematic module for subsequent motion generation. Since simulated poses can deviate from the kinematic motion prior, directly using them to predict the next target pose may cause motion generation failures. To address this, we introduce a phase encoding module that aligns simulated poses and kinematic motion priors in a share phase manifold, using the phase encoding to guide the kinematic module in generating the next target pose. Additionally, we extend this framework in Unity to handle various tasks from locomotion patterns across various terrains to more general acyclic behaviors such as bump reactions, target tracking and martial arts. Extensive experiments demonstrate POMP's superiority.

6. Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC, NO. 62102255,62472285), Shanghai Municipal Science and Technology Major Project (No. 2021SHZDZX0102),and the Open Project Program of the State Key Laboratory of CAD&CG (Grant No. A2305) Zhejiang University. Xiaogang Jin was supported by the National Natural Science Foundation of China (Grant: 62472373)

References

[1] Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–20, 2023. 3

[2] Tenglong Ao, Zeyi Zhang, and Libin Liu. Gesturediffuclip: Gesture diffusion model with clip latents. *ACM Transactions on Graphics (TOG)*, 42(4):1–18, 2023. 3

[3] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022. 5

[4] German Barquero, Sergio Escalera, and Cristina Palmero. Seamless human motion composition with blended positional encodings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 457–469, 2024. 2, 3

[5] 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

[6] Samarth Brahmbhatt, Chengcheng Tang, Christopher D Twigg, Charles C Kemp, and James Hays. Contactpose: A dataset of grasps with object contact and hand pose. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 361–378. Springer, 2020. 3

[7] Jason Chemin and Jehee Lee. A physics-based juggling simulation using reinforcement learning. In *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pages 1–7, 2018. 3

[8] Junming Chen, Yunfei Liu, Jianan Wang, Ailing Zeng, Yu Li, and Qifeng Chen. Diffsheg: A diffusion-based approach for real-time speech-driven holistic 3d expression and gesture generation. In *CVPR*, 2024. 3

[9] Kiran Chhatre, Nikos Athanasiou, Giorgio Becherini, Christopher Peters, Michael J Black, Timo Bolkart, et al. Emotional speech-driven 3d body animation via disentangled latent diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1942–1953, 2024. 3

[10] Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. *arXiv preprint arXiv:2405.11126*, 2024. 3

[11] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. *arXiv preprint arXiv:2404.19759*, 2024.

[12] 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. 3

[13] Wen Guo, Xiaoyu Bie, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Multi-person extreme motion prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13053–13064, 2022. 3

[14] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael Black. Stochastic scene-aware motion prediction. In *Proceedings of the International Conference on Computer Vision 2021*, 2021. 3

[15] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 2, 3

[16] Alejandro Hernandez, Jurgen Gall, and Francesc Moreno-Noguer. Human motion prediction via spatio-temporal inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7134–7143, 2019. 3

[17] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017. 2, 3, 5, 6, 8

[18] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16750–16761, 2023. 3

[19] Deok-Kyeong Jang, Soomin Park, and Sung-Hee Lee. Motion puzzle: Arbitrary motion style transfer by body part. *ACM Transactions on Graphics*, 41(3):1–16, 2022. 8

[20] Bin Ji, Ye Pan, Yichao Yan, Ruizhao Chen, and Xiaokang Yang. Stylevr: Stylizing character animations with normalizing flows. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2, 3

[21] Bin Ji, Ye Pan, Zhimeng Liu, Shuai Tan, and Xiaokang Yang. Sport: From zero-shot prompts to real-time motion generation. *IEEE Transactions on Visualization & Computer Graphics*, (01):1–13, 2025. 2

[22] Yifeng Jiang, Jungdam Won, Yuting Ye, and C Karen Liu. Drop: Dynamics responses from human motion prior and projective dynamics. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023. 2, 3

[23] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[24] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart

Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 3

[25] Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C. Karen Liu. Controllable human-object interaction synthesis. In *ECCV*, 2024. 3

[26] Peizhuo Li, Kfir Aberman, Zihan Zhang, Rana Hanocka, and Olga Sorkine-Hornung. Ganimator: Neural motion synthesis from a single sequence. *ACM Transactions on Graphics (TOG)*, 41(4):138, 2022. 3

[27] Peizhuo Li, Sebastian Starke, Yuting Ye, and Olga Sorkine-Hornung. Walkthedog: Cross-morphology motion alignment via phase manifolds. In *SIGGRAPH, Technical Papers*, 2024. 6

[28] Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-human motion generation under complex interactions. *International Journal of Computer Vision*, pages 1–21, 2024. 3

[29] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion vaes. *ACM Transactions on Graphics (TOG)*, 39(4):40–1, 2020. 2, 3

[30] 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):1–14, 2017. 3

[31] 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

[32] Shunlin Lu, Ling-Hao Chen, Ailing Zeng, Jing Lin, Ruimao Zhang, Lei Zhang, and Heung-Yeung Shum. Humantomato: Text-aligned whole-body motion generation. *arxiv:2310.12978*, 2023. 3

[33] Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. Dynamics-regulated kinematic policy for egocentric pose estimation. *Advances in Neural Information Processing Systems*, 34:25019–25032, 2021. 2

[34] Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023. 2, 6, 8

[35] Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. *arXiv preprint arXiv:2310.04582*, 2023. 8

[36] Zhengyi Luo, Jinkun Cao, Alexander W. Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *International Conference on Computer Vision (ICCV)*, 2023. 3

[37] Xintao Lv, Liang Xu, Yichao Yan, Xin Jin, Congsheng Xu, Shuwen Wu, Yifan Liu, Lincheng Li, Mengxiao Bi, Wenjun Zeng, et al. Himo: A new benchmark for full-body human interacting with multiple objects. *arXiv preprint arXiv:2407.12371*, 2024. 3

[38] 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. 3

[39] 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. *arXiv preprint arXiv:2108.10470*, 2021. 2

[40] Ian Mason, Sebastian Starke, He Zhang, Hakan Bilen, and Taku Komura. Few-shot learning of homogeneous human locomotion styles. In *Computer Graphics Forum*, pages 143–153. Wiley Online Library, 2018. 8

[41] Ian Mason, Sebastian Starke, and Taku Komura. Real-time style modelling of human locomotion via feature-wise transformations and local motion phases. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 5(1):1–18, 2022. 6, 8

[42] 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

[43] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021. 3

[44] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022. 3

[45] 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

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

[47] Mathis Petrovich, Michael J. Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *International Conference on Computer Vision (ICCV)*, 2023. 3

[48] Stanley Plagenhoef, F Gaynor Evans, and Thomas Abdelnour. Anatomical data for analyzing human motion. *Research quarterly for exercise and sport*, 54(2):169–178, 1983. 3

[49] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. Babel: Bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 722–731, 2021. 3

[50] Davis Rempe, Leonidas J Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. Contact and human dynamics from monocular video. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 71–87. Springer, 2020. 2, 7

[51] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11488–11499, 2021. 2, 3

[52] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13756–13766, 2023. 3

[53] 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):1–16, 2020. 2

[54] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5

[55] Paul Starke, Sebastian Starke, Taku Komura, and Frank Steinicke. Motion in-betweening with phase manifolds. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 6(3):1–17, 2023. 3

[56] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019. 2, 3, 4

[57] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)*, 39 (4):54–1, 2020. 3, 4

[58] Sebastian Starke, Yiwei Zhao, Fabio Zinno, and Taku Komura. Neural animation layering for synthesizing martial arts movements. *ACM Transactions on Graphics (TOG)*, 40 (4):1–16, 2021. 3

[59] Sebastian Starke, Ian Mason, and Taku Komura. Deepphase: periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 2, 6

[60] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 581–600. Springer, 2020. 3

[61] Omid Taheri, Vasileios Choutas, Michael J Black, and Dimitrios Tzionas. Goal: Generating 4d whole-body motion for hand-object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13263–13273, 2022. 3

[62] Shuai Tan, Bin Ji, Mengxiao Bi, and Ye Pan. Edtalk: Efficient disentanglement for emotional talking head synthesis (supplementary material). 1

[63] Shuai Tan, Bin Ji, and Ye Pan. Emmn: Emotional motion memory network for audio-driven emotional talking face generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22146–22156, 2023. 1

[64] Shuai Tan, Biao Gong, Yutong Feng, Kecheng Zheng, Dandan Zheng, Shuwei Shi, Yujun Shen, Jingdong Chen, and Ming Yang. Mimir: Improving video diffusion models for precise text understanding. *arXiv preprint arXiv:2412.03085*, 2024. 2

[65] Shuai Tan, Biao Gong, Xiang Wang, Shiwei Zhang, Dandan Zheng, Ruobing Zheng, Kecheng Zheng, Jingdong Chen, and Ming Yang. Animate-x: Universal character image animation with enhanced motion representation. *arXiv preprint arXiv:2410.10306*, 2024. 3

[66] Shuai Tan, Bin Ji, Yu Ding, and Ye Pan. Say anything with any style. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5088–5096, 2024. 1

[67] Shuai Tan, Bin Ji, and Ye Pan. Flowvqtalker: High-quality emotional talking face generation through normalizing flow and quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26317–26327, 2024.

[68] Shuai Tan, Bin Ji, and Ye Pan. Style2talker: High-resolution talking head generation with emotion style and art style. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5079–5087, 2024.

[69] Shuai Tan, Bin Ji, Mengxiao Bi, and Ye Pan. Edtalk: Efficient disentanglement for emotional talking head synthesis. In *European Conference on Computer Vision*, pages 398–416. Springer, 2025. 1

[70] Xiangjun Tang, Linjun Wu, He Wang, Bo Hu, Xu Gong, Yuchen Liao, Songnan Li, Qilong Kou, and Xiaogang Jin. Rsmt: Real-time stylized motion transition for characters. *arXiv preprint arXiv:2306.11970*, 2023. 3, 7

[71] Tianxin Tao, Xiaohang Zhan, Zhongquan Chen, and Michiel van de Panne. Style-erd: Responsive and coherent online motion style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6593–6603, 2022. 8

[72] Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *arXiv preprint arXiv:2409.14393*, 2024. 3, 8

[73] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[74] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012. 2

[75] Jonathan Tseng, Rodrigo Castellon, and C Karen Liu. Edge: Editable dance generation from music. *arXiv preprint arXiv:2211.10658*, 2022. 3

[76] Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. Towards diverse and natural scene-aware 3d human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20460–20469, 2022. 3

[77] Jiashun Wang, Jessica Hodgins, and Jungdam Won. Strategy and skill learning for physics-based table tennis animation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3

[78] Jingbo Wang, Zhengyi Luo, Ye Yuan, Yixuan Li, and Bo Dai. Pacer+: On-demand pedestrian animation controller in driving scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 718–728, 2024. 3

[79] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Physics-based character controllers using conditional vaes. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022. 3

[80] Yan Wu, Jiahao Wang, Yan Zhang, Siwei Zhang, Otmar Hilliges, Fisher Yu, and Siyu Tang. Saga: Stochastic whole-body grasping with contact. In *European Conference on Computer Vision*, pages 257–274. Springer, 2022. 3

[81] Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. Motionstreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. *arXiv preprint arXiv:2503.15451*, 2025. 2

[82] Zhaoming Xie, Hung Yu Ling, Nam Hee Kim, and Michiel van de Panne. Allsteps: curriculum-driven learning of stepping stone skills. In *Computer Graphics Forum*, pages 213–224. Wiley Online Library, 2020. 3

[83] Liang Xu, Xintao Lv, Yichao Yan, Xin Jin, Shuwen Wu, Congsheng Xu, Yifan Liu, Yizhou Zhou, Fengyun Rao, Xingdong Sheng, et al. Inter-x: Towards versatile human-human interaction analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22260–22271, 2024. 3

[84] Pei Xu, Xiumin Shang, Victor Zordan, and Ioannis Karamouzas. Composite motion learning with task control. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023. 3

[85] Pei Xu, Kaixiang Xie, Sheldon Andrews, Paul G Kry, Michael Neff, Morgan McGuire, Ioannis Karamouzas, and Victor Zordan. Adaptnet: Policy adaptation for physics-based character control. *ACM Transactions on Graphics (TOG)*, 42(6):1–17, 2023. 3

[86] Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. Controlvae: Model-based learning of generative controllers for physics-based characters. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 3

[87] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018. 2, 7

[88] Haotian Zhang, Ye Yuan, Viktor Makoviychuk, Yunrong Guo, Sanja Fidler, Xue Bin Peng, and Kayvon Fatahalian. Learning physically simulated tennis skills from broadcast videos. *ACM Trans. Graph.*, 2023. 3

[89] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14730–14740, 2023. 3

[90] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. Couch: Towards controllable human-chair interactions. In *European Conference on Computer Vision*, pages 518–535. Springer, 2022. 3

[91] Zeyi Zhang, Tenglong Ao, Yuyao Zhang, Qingzhe Gao, Chuan Lin, Baoquan Chen, and Libin Liu. Semantic gesticulator: Semantics-aware co-speech gesture synthesis. *ACM Transactions on Graphics (TOG)*, 43(4):1–17, 2024. 3