Mimic-X: A Large-Scale Motion Dataset via Fast Physics-Based Controller Adaptation

Hongyu Tao[1], Shuaiying Hou[1], Junheng Fang[2], Mingyao Shi[1], Weiwei Xu[1]*

[1]Zhejiang University
[2]Zhejiang Sci-Tech University
3170102625@zju.edu.cn, houshuaiying@zju.edu.cn, fangjh@zstu.edu.cn, shi_my@zju.edu.cn, xww@cad.zju.edu.cn

Abstract

Large and high-quality motion datasets are essential for advancing human motion modeling. However, limitations of existing motion datasets, such as insufficient scale or inadequate quality, significantly hinder the progress of this field. To address these limitations, we introduce Mimic-X, a large-scale (52 hours), physically plausible 3D human motion dataset. To construct Mimic-X, we develop an adaptive option framework that controls a physically simulated character to imitate low-quality motions extracted from a vast collection of online videos. Specifically, we first apply hierarchical clustering to group motions into clusters, and then train option policies to mimic motions sampled from these clusters. Considering the noisy nature of low-quality motions, we utilize a separate encoder for each cluster to map the noisy motions within the cluster into a compact latent space. This significantly enhances the quality of the imitated motions while accelerating the learning process. Subsequently, we employ dynamic programming as a meta-policy to efficiently organize the option policies to generate complete motion clips. Finally, we perform fine-tuning to each motion sequence to further refine motion quality. The proposed adaptive option framework outperforms state-of-the-art human motion recovery methods across various evaluation metrics, demonstrating that motions in Mimic-X exhibit higher quality and greater physical plausibility. Furthermore, experimental results show that Mimic-X enhances the performance of motion generation methods, verifying its effectiveness for motion modeling tasks.

Supp. — https://thyzju17.github.io/projects/Mimic-X/

1 Introduction

Based on the rapid development of deep generative models, state-of-the-art human motion generation methods have shown impressive capabilities in producing a wide range of diverse and complex motions (Jiang et al. 2023; Tevet et al. 2023; Zhang et al. 2024; Guo et al. 2024). Nevertheless, the motions generated by these methods often exhibit different types of artifacts, such as ground penetration, interpenetration, foot sliding, and floating (Li et al. 2024), which hinder the motions' realism and applicability. These issues can be largely attributed to the limitations of training data, as suggested by the data scaling law (Kaplan et al. 2020), which has demonstrated that larger and higher-quality datasets can significantly improve model performance in many domains such as Neural Language Processing (Floridi and Chiriatti 2020) and Embodied-AI (Lin et al. 2024). To address these challenges, enhancing the scale and quality of the human motion dataset is essential. However, constructing such datasets is labor-intensive, time-consuming, and of high cost. Therefore, it's crucial to explore cost-effective approaches to produce extensive, high-fidelity motion datasets and, in turn, develop more robust and powerful deep generative models.

Marker-based motion capture (MoCap) is the most widely used method for generating high-quality human motion datasets (Plappert, Mandery, and Asfour 2016; Mahmood et al. 2019). However, this method requires well-calibrated cameras, precisely placed markers, and controlled indoor environments, limiting its scalability. In contrast, markerless vision-based MoCap methods (Kocabas, Athanasiou, and Black 2020; Karatzas et al. 2024) offer a more scalable solution by capturing human motion from real-world videos available online. Despite this advantage, recovering 3D human motions from 2D videos remains ill-posed, plagued by challenges such as depth ambiguity, self-occlusion, and the lack of camera parameters. As a result, the recovered motions often suffer from physically implausible artifacts like foot sliding, penetration, and floating in the air, as indicated in Fig. 1.

To alleviate physically implausible artifacts, several studies (Peng et al. 2018a,b) have focused on integrating hard physical constraints. Specifically, these methods employ controllers to drive simulated characters that mimic the motions extracted from videos within a physics engine. Although promising, this approach is computationally intensive, requiring hours to generate a single motion clip, which significantly hinders its applicability for constructing large-scale motion datasets. Recently, researchers attempted to use general controllers (Luo et al. 2024; Tessler et al. 2024) for real-time motion imitation. However, due to the inherent complexity and vast diversity of human motions, these controllers perform well on their test datasets but often fail to accurately imitate motions recovered from in-the-wild videos, leading to robotic and unnatural movements.

In this paper, we introduce a novel adaptive option frame-

*Corresponding author

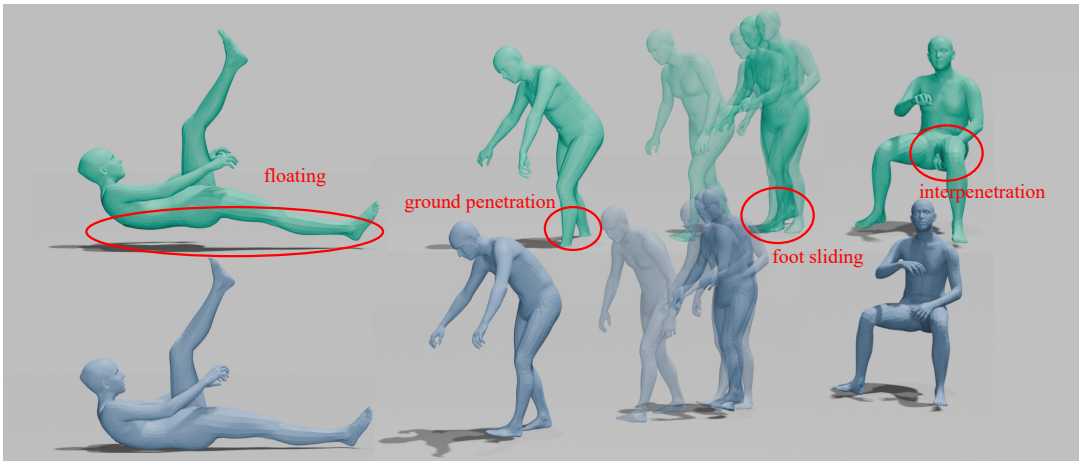

Figure 1: Selected poses from the Motion-X (green) and Mimic-X (blue) datasets, with the latter constructed using our adaptive option framework. The red ellipses highlight artifacts such as floating, ground penetration, foot sliding, and interpenetration. The results demonstrate that our method effectively reduces these artifacts. More results can be found in the accompanying video from 0m23s to 0m38s.

work to address the aforementioned limitations. This framework is designed to efficiently construct a high-quality, physically accurate human motion dataset by mimicking noisy human motions extracted from large-scale in-the-wild videos. Inspired by the traditional option framework (Sutton, Precup, and Singh 1999), where a series of option policies manage specific sub-tasks, enabling the effective execution of more complex tasks, we propose a similar strategy to build our adaptive option framework. Specifically, we first train multiple option policies to mimic motions sampled from distinct subsets of the motion dataset, which we call the pre-training step. We then move on to the adaption step, where we rapidly *adapt* these option policies to each motion sequence by fine-tuning, which significantly reduces the training time for mimicking a single motion sequence from hours to minutes.

The main technical components of our adaptive option framework are as follows: 1) We use an auto-encoder to map the reference motions of the sub-dataset into a compact latent space. This encoding process alleviates the noise in the reference motions, improving the quality of the mimicked motion and reducing the average training time from 12.5 minutes to 9.0 minutes. 2) We employ dynamic programming as the meta-policy to segment a single motion sequence into several sub-sequences, each associated with a specific option policy for imitation. This approach addresses the complexity of mimicking diverse motion types within a single sequence. 3) After segmenting the motion sequence into sub-sequences, we jointly fine-tune the option policies assigned to that motion sequence, which further enhances the fidelity of the imitated motions. With these designs, our framework requires approximately 48 hours for pre-training and an average of 9 minutes for fine-tuning a single motion sequence in Human3.6M (Ionescu et al. 2013) dataset, with the process taking less than 5 minutes for constructing our own dataset. This makes it suitable for rapidly constructing large-scale, high-quality datasets, especially when compared to the state-of-the-art (SOTA) methods that take over an hour to mimic each motion sequence individually.

Utilizing the proposed framework, we construct a large-scale, physically accurate human motion dataset, named Mimic-X, by imitating the motions from the Motion-X dataset (Lin et al. 2023). Our framework outperforms SOTA methods in both motion reconstruction accuracy and physical plausibility, demonstrating that motions in Mimic-X are of high quality. We further evaluate the Mimic-X dataset on several SOTA motion generation methods, and the experimental results verify its significance for motion modeling tasks.

The contributions of this paper can be summarized as follows:

- We present an adaptive option framework that efficiently mimics large-scale human motion datasets from in-the-wild videos. This method can handle complex motions with diverse types and reduces the training time for a single sequence from hours to minutes.
- We construct Mimic-X, a large-scale, physically accurate, and high-quality human motion dataset, consisting 52-hour motions in 30 fps.
- We evaluate Mimic-X on the motion generation task, demonstrating its effectiveness in facilitating the generation of diverse and physically plausible motions.

2 Related Work

2.1 Human Motion Datasets

High-quality human motion datasets (Gross and Shi 2001; Ionescu et al. 2013; Sigal, Balan, and Black 2010; Trumble et al. 2017) are predominantly obtained using optical marker-based motion capture (Mocap) systems. However, optical marker-based Mocap requires well-calibrated cameras, precisely placed markers, and controlled indoor en-

vironments, which limits its scalability. As a compromise, current large human motion datasets are always constructed by collecting small Mocap datasets. The KIT Motion-Language Dataset (Plappert, Mandery, and Asfour 2016) is the first public dataset that pairs human motion data (˜10 hours) with textual descriptions, collecting from two different datasets (Mandery et al. 2015; CMU 2016). The AMASS dataset (Mahmood et al. 2019) aggregates 15 different optical marker-based Mocap datasets, creating a large-scale motion collection (˜40 hours), parameterized using SMPL model (Loper et al. 2023). Since the motions are collected from different motion datasets and then unified into a common representation, the quality of motion data within AMASS varies. Kaufmann et al. (2023) used body-worn, wireless electromagnetic (EM) sensors, along with a handheld iPhone, to record 58 minutes of motion data, called EMDB. However, capturing large amounts of motion data remains challenging due to the need for wearable equipment. In contrast to Mocap-based methods, Lin et al. (2023) developed the largest human motion dataset to date (˜140 hours) by estimating human motion from a vast collection of online wild videos. However, the Motion-X dataset suffers from low motion quality due to the inherent ambiguity of estimating 3D human motion from 2D video data. In this paper, we aim to enhance the motion quality of the Motion-X dataset, creating a refined dataset, which we term Mimic-X.

2.2 Human Motion Recovery

Human motion recovery is about recovering the human pose to precisely align with the input image. Recent approaches in 3D human motion recovery mainly represent human poses using parametric human models, such as the SMPL (Loper et al. 2023) and SMPLX (Pavlakos et al. 2019). Optimization-based methods (Bogo et al. 2016; Pavlakos et al. 2019; Lin et al. 2023) recover the motions by minimizing the re-projection error between projected and detected keypoints. In recent years, neural networks have been increasingly employed to enhance recovery quality. Notable efforts include the use of convolution neural networks (Kanazawa et al. 2019), recurrent neural networks (Luo, Golestaneh, and Kitani 2020; Choi et al. 2021), generative adversarial networks (Kocabas, Athanasiou, and Black 2020), and transformers (Wan et al. 2021; Goel et al. 2023). While these methods achieve precise motion recovery, they only focus on recovering poses in the image coordinates, lacking plausible root translations. WHAM (Shin et al. 2024) directly regresses poses and root translations in the world coordinates at each frame. WHAC (Yin et al. 2025) incorporates visual odometry to estimate camera rotation, and refines the global trajectory using a network. GVHMR (Shen et al. 2024) predicts poses in the gravity coordinates instead of camera coordinates, and directly estimates global translation.

However, the aforementioned approaches often produce physically implausible results. Incorporating physical laws can help mitigate these artifacts. SFV (Peng et al. 2018b) mimics the kinematic motions extracted from video using deepmimic (Peng et al. 2018a) within a physics engine. Similarly, trajectory optimization techniques (Rempe et al. 2020; Gärtner et al. 2022b,a) are employed to mimic kinematic motions. However, they are computationally expensive as deepMimic (Peng et al. 2018a) requires training controllers for each motion sequence from scratch, and trajectory optimization methods involve time-consuming iterative processes. To enable real-time optimization, researchers (Shimada et al. 2020, 2021) focus solely on foot contact and make the contact process differentiable. However, these methods struggle to produce contact-rich motions such as dancing. SimPoE (Yuan et al. 2021) improves motion quality by refining the detected motions and applying residual forces on the root joint to enhance controller performance. Universal controllers (Luo et al. 2024; Tessler et al. 2024) can track the motions recovered from videos in real time by training on large motion datasets. However, they are limited to characters with fixed body shapes and may exhibit jitters when tracking noisy motions.

In this paper, we utilize the motions recovered from videos by these human motion recovery methods as reference and apply an adaptive option framework to mimic these reference motions, thereby creating a large, physically plausible human motion dataset.

3 Pipeline

In this section, we detail our adaptive option framework. As illustrated in Fig. 2, we first extract kinematic motions from videos, obtaining a low-quality reference motion dataset. Then, we cluster the reference motion dataset into several clusters (Section 3.1), and sample motion sequences from these clusters to train option policies using deep reinforcement learning (Section 3.2). This clustering strategy alleviates the challenge of mimicking large-scale motions that exhibit high diversity. Finally, we fine-tune the option policies to mimic each motion sequence, further refining the motion quality (Section 3.3). During fine-tuning, we use dynamic programming as the meta-policy to select and arrange the policies. This training-free meta-policy improves motion quality while significantly reducing training time. Since extracting motions from videos is outside the scope of this paper, details are provided in the supplementary material.

3.1 Motion Clustering

Motion clustering partitions the motions into C subsets to reduce the complexity of modeling the entire dataset, and each subset contains similar motions. Previous works (Onuma, Faloutsos, and Hodgins 2008; Won, Gopinath, and Hodgins 2020) employ k-means clustering over the velocity-based and acceleration-based feature vectors computed on the motion sequence in the dataset. We adopt a similar approach and define the motion features $\mathcal{F}$, as follows:

$$\mathcal{F} = \{\mathbf{R}^h, \mathbf{R}_{\perp}^{avel}, \mathbf{X}_{\perp}^{vel}, \mathbf{X}_{\parallel}^{vel}, \mathbf{X}^{acc}\} \tag{1a}$$

$$\mathbf{R}^h = \frac{1}{N}\sum_i^N r_i^h, \mathbf{R}_{\perp}^{avel} = \frac{1}{N}\sum_i^N \|r_{\perp,i}^{avel}\|^2 \tag{1b}$$

$$\mathbf{X}_{\perp\parallel}^{vel} = \frac{1}{N}\sum_i^N \|x_{\perp\parallel,i}^{vel}\|^2, \mathbf{X}^{acc} = \frac{1}{N}\sum_i^N \|x_i^{acc}\|^2 \tag{1c}$$

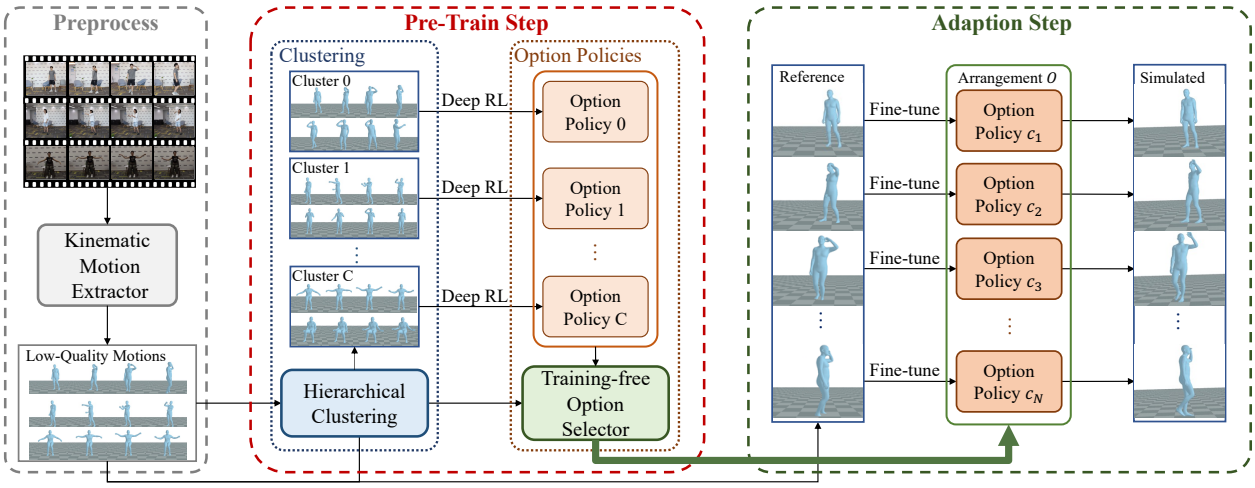

Figure 2: System overview. Our system comprises three main steps: preprocessing, pre-training, and adaptation. In the preprocessing, we extract kinematic motion data from videos, resulting in a low-quality motion dataset. During the pre-train step, we apply hierarchical clustering to group the low-quality dataset into several clusters, and train option policies to mimic motions sampled from the clusters using deep reinforcement learning. Finally, in the adaptation step, an option selector arranges the option policies to mimic the reference motion. After fine-tuning for several minutes, we obtain high-quality simulated motions.

where $\mathbf{R}^h \in \mathbb{R}$ represents the average height of the root joint, with r_i^h being the height of the root joint at the i_{th} frame, $\mathbf{R}_\perp^{avel} \in \mathbb{R}$ being the average rotational kinetic energy of the root joint around the vertical axis, and $r_{\perp,i}^{avel}$ being the angular velocity of the root joint around the vertical axis; $\mathbf{X}_\perp^{vel} \in \mathbb{R}^J$ and $\mathbf{X}_\parallel^{vel} \in \mathbb{R}^J$ are the average kinetic energies in planar and vertical directions, respectively, with J being the joint number, $x_\perp^{vel}$ and $x_\parallel^{vel}$ being the velocities of joints in the root coordinate system; $\mathbf{X}^{acc} \in \mathbb{R}^J$ is an approximation of energy expenditure, with x_i^{acc} being the acceleration of joints in the root coordinate system. For simplicity, we omit the joint index j in Equation 1c. Compared to previous motion features (Onuma, Faloutsos, and Hodgins 2008; Won, Gopinath, and Hodgins 2020), the addition of $\mathbf{R}^h$ and $\mathbf{R}_\perp^{avel}$ helps to better distinguish motions such as lying, twisting, and jumping.

To ensure that the motions in each cluster can be imitated by a single policy, we constrain the cluster radius, which is defined as the largest Euclidean distance between any two motion features within the cluster. To enforce this constraint, we employ a hierarchical k-means approach, which iteratively applies k-means to clusters with a radius greater than l_{max}. In our experiment, we set $l_{max} = 40$, and provide an evaluation in Section 4.

3.2 Option Policies

We train C option policies, denoted by $\pi_{1:C}$, to enable a simulated character to mimic motions sampled from each cluster, as shown in the pre-train step of Fig. 2. We use the proximal policy optimization (PPO) (Schulman et al. 2017) to learn the option policies. During training, an agent interacts with the environments according to the control policy π_c to mimic the reference motion $m = x_{1:N}$ of N frames, which is sampled from cluster c. Here, x_t refers to the pose at the t_{th} frame. At time step t, the agent observes the state s_t and samples an action a_t according to the policy $a_t \sim \pi_c(a_t|s_t)$. The agent then applies the action, causing the environment to transition to the next state s_{t+1} and receive a reward r_t, which measures the difference between the simulated character and the current reference motions. After iterating for N steps, the agent produces the simulated motion $\hat{x}_{1:N}$ from the state sequence $s_{1:N}$. Our objective is to learn the control policy π_c that aligns $x_{1:N}$ and $\hat{x}_{1:N}$ as closely as possible. The following parts of this section provide a detailed description of the state s_t, action a_t, reward r_t, and policy π_c used in our learning process.

States. The state at time step t, $s_t \in \mathbb{R}^{570}$, consists of the reference pose $\mathcal{P}_{t+1}$, body shape parameters β, the current simulated pose $\hat{\mathcal{P}}_t$, and the pose difference between current pose and the simulated pose $\mathcal{P}_{t+1} - \hat{\mathcal{P}}_t$, where $\hat{\cdot}$ means that the variable is derived from the simulated character. The pose $\mathcal{P}_t$ is defined as $\mathcal{P}_t = \{r_t^h, r_t^a, \dot{r}_t^p, \dot{r}_t^a, x_t^a, \dot{x}_t^a, e_t^p\}$, where r_t^h is the height of the root from the ground, r_t^a is the root rotation, $\dot{r}_t^p$ & $\dot{r}_t^a$ are the root velocity & angular velocity in the local root coordinate system, x_t^a denotes the local rotations of all joints, $\dot{x}_t^a$ denotes the angular velocities of all joints, and e_t^p denotes the local positions of the end-effectors (hands and feet). Following (Peng et al. 2022), rotations are represented in a 6D format, and angular velocities are represented in the axis-angle format. Additionally, we include the root position offset in the pose difference $\mathcal{P}_{t+1} - \hat{\mathcal{P}}_t$ to indicate the deviations in the root position.

Actions. The action at time step t, $a_t \in \mathbb{R}^{55}$, is represented by rotations in the target pose of proportional-derivative (PD) policies. Following (Peng et al. 2022), the rotations are represented by quaternions' exponential maps for 3D spherical joints and angulars for hinge joints (knees and elbows).

Rewards. At each time step, the reward is defined as $r_t = r_t^m \cdot r_t^{2d}$. The mimic reward r_t^m quantifies the difference between the simulated pose $\hat{\mathcal{P}}_t$ and the reference pose $\mathcal{P}_t$ as follows:

$$\begin{aligned} r_t^m = & exp(-w_r\|r_t^p - \hat{r}_t^p\|^2)exp(-w_r\|r_t^a - \hat{r}_t^a\|^2) \\ & exp(-w_a\|x_t^a - \hat{x}_t^a\|^2)exp(-w_{av}\|\dot{x}_t^a - \hat{\dot{x}}_t^a\|^2) \\ & exp(-w_e\|e_t^p - \hat{e}_t^p\|^2) \end{aligned} \quad (2)$$

where r_t^p is the root position. The projection reward r_t^{2d} measures the difference between the 2D projection of the simulated pose and the detected keypoints:

$$r_t^{2d} = exp(-w_{2d}\|\Pi(SMPLX(\hat{x}_t^a, \hat{r}_t^p, \beta)) - x_t^{2d}\|^2) \quad (3)$$

where x_t^{2d} is the detected 2D keypoints, $SMPLX(\cdot)$ is the forward process of SMPLX (Pavlakos et al. 2019), $\Pi(\cdot)$ selects the corresponding joints and projects the joints into the image space.

Policies. We employ distinct policies to mimic motions across different motion clusters. Following (Peng et al. 2022; Li et al. 2024), each policy is implemented as a neural network that map the state s_t to a Gaussian distribution over actions, $\pi_c(a_t|s_t) = \mathcal{N}(\mu_{\pi_c}(s_t), \Sigma_{\pi_c})$, where c denotes the cluster index. The mean $\mu_{\pi_c}(s_t)$ depends on the current state s_t, while the covariance matrix Σ_{π_c} is fixed and diagonal. Since the reference motions are usually noisy, we employ an auto-encoder to map the motions within each cluster into a more compact latent space, and train the policies on the learned latent representations, to accelerate the learning process. We show that this approach significantly enhances the training of the policies (see Section 4). Additionally, we utilize gradient penalty (Chen et al. 2024) to encourage smoother and less jittery motions.

3.3 Adaption Step

In this section, we describe how to mimic the motion sequence $m = \mathcal{P}_{1:N}$ using C option policies $\pi_{1:C}$. Mimicking a single motion sequence with a single option policy is challenging, as the sequence may contain various types of motions. Therefore, segmentation is required to divide the sequence into sub-sequences, each of which is then assigned an optimal option policy to mimic.

We propose a simple yet efficient option selector based on dynamic programming to generate an arrangement $\mathcal{O} = o_{1:N}$, where o_t is the identifier of the option policy to apply at frame t. The procedure begins by simultaneously controlling C agents, where each agent is driven by its respective option policy to mimic the motion. If the reward of any agent falls below a threshold σ, the agent's state is reset to the one with the highest reward among its current states. This results in C reward trajectories $r_{1:N}^{1:C}$. The next step is to determine the optimal arrangement $\mathcal{O}^*$ that maximizes the total reward. Specifically, we aim to solve the following optimization problem:

$$\begin{aligned} \mathcal{O}^* &= \arg\max_{\mathcal{O}} R(\mathcal{O}) \\ &= \arg\max_{\mathcal{O}} \sum_{t}^{N} r_t^{o_t} + (1 - \delta_{o_t, o_{t+1}}) \cdot w_p \end{aligned} \quad (4)$$

where $R(\mathcal{O})$ is the cumulative reward for arrangement $\mathcal{O}$, $r_t^{o_t}$ is the reward at time step t when option o_t is applied, $\delta_{\cdot,\cdot}$ is the Kronecker delta function, N is the total number of frames in the reference motion, and C is the number of option policies. The negative penalty term w_p is added to discourage frequent transitions, since rapid transitions between different policies can cause artifacts like jitters and unnatural poses. We solve this problem using dynamic programming. The state transition equation is as follows:

$$\begin{aligned} dp(t, c) = & \max_{c'} dp(t-1, c') + r_t^{c'} \\ & + (1 - \delta_{c,c'}) \cdot w_p, (t = 2, .., N) \end{aligned} \quad (5)$$

where the state function $dp(t, c)$ represents the maximum cumulative reward when executing policy π_c at frame t, initialized with $dp(1, c) = r_1^c$. Here, c' iterates over all policy indices. After populating the state function $dp(\cdot, \cdot)$, the optimal solution $\mathcal{O}^*$ can be recovered by backtracking from the final state $\max_c dp(N, c)$ to the initial state $dp(1, \cdot)$.

After obtaining an imitated motion sequence, we fine-tune the control policies for the sequence according to its corresponding arrangement $\mathcal{O}^*$ until the reward reaches a predefined threshold. This fine-tuning process significantly enhances performance. Quantitative results on this enhancement, the associated time cost, and the role of the option selector are discussed in the evaluation section.

4 Experiments and Evaluations

We perform all experiments on a single NVIDIA V100 GPU. As for the physics engine, we employ NVIDIA's Isaac Gym. For Mimic-X, there are 16 control policies in the pretraining step, requiring approximately 2 days to train. In the adaptation step, it takes an average of 9 minutes to mimic a specific motion sequence, and less than 5 minutes to construct our Mimic-X dataset.

Our Mimic-X dataset is constructed from three subsets of Motion-X, namely AIST, IDEA400, and fitness, totaling 52.3 hours of motion data at 30 fps. The motions in Mimic-X exhibit superior physical plausibility compared to those in Motion-X, as demonstrated in Fig. 1. More comparison results and information about the Mimic-X dataset can be found in the supplementary material.

In the following, we first evaluate the quality of the imitated motion produced by our pipeline. Subsequently, we demonstrate the effectiveness of our Mimic-X dataset in enhancing the performance of motion generation models.

4.1 Evaluations of Motion Quality

The evaluation is conducted on the Human3.6M (Ionescu et al. 2013) dataset, which is commonly used for human motion recovery. Following the protocol in (Ci et al. 2019), we utilize motions of subjects S1, S5, S6, S7, and S8 as the training set, S9 and S11 as the test set. We also evaluate on AIST (Tsuchida et al. 2019) dataset, which contains challenging dance motions. We randomly select 20 sequences as the test set.

Dataset	Model	MRPE↓	MPJPE↓	PA-MPJPE↓	Vel↓	Accel↓	FS↓	GP↓	Float↓
Human3.6M	VIBE (Kocabas, Athanasiou, and Black 2020)	151	56	44	8.8	12.6	10.5	6.5	0.6
	GVHMR (Shen et al. 2024)	296	*45*	*32*	**2.8**	**2.5**	*0.8*	6.7	72.6
	PhysCap (Shimada et al. 2020)	183	97	65	7.2	-	-	-	-
	SimPoE (Yuan et al. 2021)	-	57	42	-	6.7	3.4	1.6	-
	PhysAware (Shimada et al. 2021)	-	77	58	4.5	-	-	-	-
	TrajOp (Gärtner et al. 2022b)	143	84	56	9.0	-	-	-	-
	DiffPhy (Gärtner et al. 2022a)	-	82	56	6.7	-	-	-	-
	MaskedMimic (Tessler et al. 2024)	*95*	60	49	*3.6*	*2.7*	**0.4**	**1.0**	**0.0**
	Ours	**59**	**33**	**26**	3.8	*2.7*	1.3	*1.2*	**0.0**
AIST	VIBE (Kocabas, Athanasiou, and Black 2020)	335	95	61	20.3	27.1	49.3	58.8	0.7
	GVHMR (Shen et al. 2024)	256	83	*48*	*8.5*	6.2	2.1	9.9	46.1
	MaskedMimic (Tessler et al. 2024)	*103*	*70*	50	8.8	**5.7**	**0.6**	*0.6*	**0.0**
	Ours	**86**	**54**	**41**	**7.4**	*6.0*	*1.1*	**0.5**	**0.0**

Table 1: Evaluations of motion recovery quality. The best results are highlighted as **1st**, *2nd*, and 3rd.

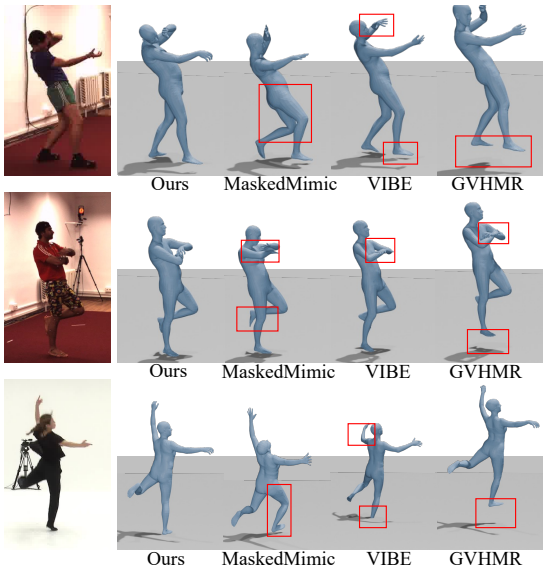

Figure 3: Visualization of recovered poses. MaskedMimic fails to mimic the pose accurately; VIBE shows floating, interpenetration, and ground penetration; GVHMR suffers from interpenetration and severe floating. All the artifacts are indicated by red boxes. Our results are more accurate and physically plausible (from 2m06s to 2m36s in the accompanying video).

Metrics We evaluate the results from two perspectives: motion recovery accuracy and physical plausibility. To assess motion recovery accuracy, we report the mean root position error (MRPE), mean per-joint position error (MPJPE), and Procrustes-aligned mean per-joint position error (PA-MPJPE). To evaluate physical plausibility, we introduce four metrics: velocity error (Vel), acceleration error (Accel), foot sliding (FS), ground penetration (GP), and floating (Float). Detailed formulations of these metrics are provided in the supplementary material.

Comparison to state-of-the-art methods We compare our method against SOTA 3D human motion recovery methods, including both kinematics-based approaches (VIBE (Kocabas, Athanasiou, and Black 2020) and GVHMR (Shen et al. 2024)), and physics-based approaches (PhysCap (Shimada et al. 2020), SimPoE (Yuan et al. 2021), PhysAware (Shimada et al. 2021), TrajOp (Gärtner et al. 2022b), DiffPhy (Gärtner et al. 2022a), and MaskedMimic (Tessler et al. 2024)). Implementation details can be found in the supplementary material.

Method	MPRE↓	MPJPE↓	PA-MPJPE↓	Vel↓	Accel↓	FT↓
scratch	358.6	156.6	79.4	4.2	2.5	181.4
w/o ft	375.1	142.6	50.0	5.5	3.3	-
w/o enc	91.3	48.4	31.4	4.1	2.9	12.5
w/o dp	246.6	111.1	42.7	4.2	2.6	15.5
r=20	60.8	34.4	27.6	3.8	2.6	14.5
r=40(ours)	59.3	33.3	26.3	3.8	2.7	9.0
r=50	80.9	43.6	29.3	3.9	2.7	14.0
Whole1×	60.4	35.6	27.7	3.8	2.6	15.6
Whole4×	60.0	33.0	26.5	3.7	2.6	27.1

Table 2: Ablation studies. FT represents the average fine-tuning time in minutes, while for *scratch* it means the average training time.

From the results shown in Table 1, we can observe that our method significantly outperforms SOTA methods in motion recovery accuracy, as indicated by MRPE, MPJPE, and PA-MPJPE. This improvement can be largely attributed to the fine-tuning step. For the metrics Vel, Accel, and FS, our method achieves results comparable to MaskedMimic and GVHMR, demonstrating that it can produce smooth movements with minimal jitters and small foot sliding. In terms of ground penetration (GP) and floating (Float), our method shows low values on both metrics, indicating higher physical plausibility. Furthermore, our method outperforms both kinematics-based and physics-based approaches. In particular, VIBE and GVHMR show the poorest performance in physical plausibility. This arises from their lack of consideration for ground interactions, which are explicitly addressed in the physics-based methods. The large floating and ground penetration in GVHMR, shown in Fig. 3, are primarily due to accumulated errors in the predicted root translation. Additionally, we found that MaskedMimic fails to mimic some motions, as shown in Fig. 3. This observation reveals the difficulty of mimicking diverse motions with a universal controller. More comparison results are shown in the supplementary material.

Method	Physical Metrics			R Precision↑			FID↓	MultiModal Dist↓
	FS↓	GP↓	Float↓	Top1	Top2	Top3		
T2M-GPT	$9.371^{\pm.014}$	$2.109^{\pm.008}$	$20.881^{\pm.349}$	$\mathbf{0.491^{\pm.002}}$	$\mathbf{0.679^{\pm.002}}$	$0.774^{\pm.002}$	$\mathbf{0.169^{\pm.006}}$	$3.112^{\pm.008}$
T2M-GPT-Motion-X	$9.221^{\pm.020}$	$2.931^{\pm.011}$	$18.819^{\pm.229}$	$0.490^{\pm.003}$	$0.677^{\pm.003}$	$0.772^{\pm.002}$	$0.234^{\pm.009}$	$3.135^{\pm.011}$
T2M-GPT-Mimic-X	$\mathbf{9.085^{\pm.017}}$	$\mathbf{1.120^{\pm.009}}$	$\mathbf{17.828^{\pm.292}}$	$0.488^{\pm.002}$	$0.677^{\pm.002}$	$\mathbf{0.775^{\pm.002}}$	$0.190^{\pm.008}$	$\mathbf{3.101^{\pm.009}}$
ReMoDiffuse	$28.995^{\pm.074}$	$2.627^{\pm.074}$	$36.267^{\pm.125}$	$\mathbf{0.498^{\pm.002}}$	$0.674^{\pm.003}$	$0.772^{\pm.003}$	$\mathbf{0.127^{\pm.005}}$	$3.052^{\pm.009}$
ReMoDiffuse-Motion-X	$26.381^{\pm.042}$	$2.713^{\pm.090}$	$29.369^{\pm.209}$	$0.482^{\pm.003}$	$0.665^{\pm.003}$	$0.761^{\pm.002}$	$0.227^{\pm.006}$	$3.133^{\pm.011}$
ReMoDiffuse-Mimic-X	$\mathbf{24.588^{\pm.044}}$	$\mathbf{1.677^{\pm.023}}$	$\mathbf{27.525^{\pm.312}}$	$\mathbf{0.498^{\pm.003}}$	$\mathbf{0.681^{\pm.002}}$	$\mathbf{0.773^{\pm.001}}$	$0.158^{\pm.006}$	$\mathbf{3.028^{\pm.009}}$

Table 3: Quantitative evaluation on the HumanML3D, Motion-X, and Mimic-X dataset. The best results are highlighted as **1st**.

For fair comparison, we also finetune MaskedMimic on each motion sequences in the test dataset. Details can be found in the supplementary material.

Ablation Studies To evaluate the contribution of each component in our method, we perform several ablation studies. In the following, we will detail these experiments.

Auto-encoder Module. In this experiment, denoted as *w/o enc*, we remove the auto-encoder module. As shown in Table 2, the auto-encoder module significantly reduces the average fine-tuning time in our framework, from 12.5 minutes to 9.0 minutes. Additionally, we observe that this module helps mimic the motion more accurately and alleviates motion jitters, as indicated by the lower Vel and Accel values.

Option Arrangement. To evaluate the effectiveness of our proposed dynamic programming option selector, we replace it with a naive clustering strategy: assigning each motion sequence with single cluster according to the motion feature. This ablation study is referred to as *w/o dp*. From the results in Table 2, we can observe that our dynamic programming-based approach not only improves performance but also reduces the average fine-tuning time.

Clustering. In our experiments, we cluster the training set of the Human3.6M dataset into 4 clusters using a cluster radius threshold of $r_{max} = 40$. Therefore, our model size is 94.4MB (23.6MB for each control network). To assess the impact of this clustering approach, we conduct the following experiments: 1) We use a controller 4 times larger to mimic the entire Human3.6M dataset, referred to as *Whole4*$\times$; 2) We cluster the dataset using different cluster radius thresholds of 20, 50, and 70, resulting in variant referred to as $r = 20$, $r = 50$, and *Whole1*$\times$. The results in Table 2 illustrate that using a larger controller to mimic the entire dataset is challenging, since the fine-tuning time for *Whole4*$\times$ is three times as long as our model. We also find that clustering the dataset into an appropriate size achieves the highest mimic quality and the shortest fine-tuning time. Too many or too few clusters will lead to poor results or increased fine-tuning time.

Other Evaluations. To evaluate the effectiveness of our method in reducing the training time required to mimic the entire dataset, we report the training time for mimicking each motion sequence separately from scratch, referred to as *scratch*. Additionally, we present the results before fine-tuning, referred to as *w/o ft*. The results in Table 2 demonstrate that our adaptive option framework drastically reduces the time required to mimic various motions, from an average of 181.4 minutes to 9.0 minutes. Although approximately 2 days of pre-training are required for the option controllers, fine-tuning significantly accelerates dataset construction, particularly when handling thousands of motion sequences. We also observe that without fine-tuning, our controllers perform poorly. More details can be found in the supplementary material.

4.2 Impact on Text-driven Motion Generation

To evaluate the impact of the Mimic-X dataset on text-driven motion generation, we train and evaluate two different methods: T2M-GPT (Zhang et al. 2023a), a transformer-based approach, and ReMoDiffuse (Zhang et al. 2023b), a diffusion-based approach. Specifically, we train these models on the Mimic-X and HumanML3D (Guo et al. 2022) datasets, referred to as T2M-GPT-Mimic-X and ReMoDiffuse-Mimic-X. We also train the models on the corresponding subset of Motion-X (Lin et al. 2023) and HumanML3D datasets, termed T2M-GPT-Motion-X and ReMoDiffuse-Motion-X. For fair comparison, we re-train T2M-GPT using the preprocessed HumanML3D dataset from ReMoDiffuse (Zhang et al. 2023b) and evaluate on the corresponding test dataset.

In addition to the original metrics used in their papers, we report physical metrics described in section 4.1. As shown in Table 3, the models trained with our Mimic-X always rank in the top 1 on the physical metrics, and get comparable results on generation metrics. These results indicate that integrating a large-scale, physically plausible human motion dataset does not degrade the model performance in terms of generation metrics, and can improve the physical plausibility (visual results can be found in the accompanying video).

5 Conclusion

In this paper, we present a large-scale, physically plausible human motion dataset called Mimic-X, which is constructed by mimicking the largest existing motion dataset, Motion-X. To build this dataset efficiently, we propose an adaptive option framework, which can fast adapt the pre-train option controllers to mimic each motion sequence in a physics engine. Comprehensive experiments demonstrate that the motions in Mimic-X exhibit higher quality and greater physical plausibility than existing datasets, and they significantly benefit motion modeling tasks such as motion generation.

Limitation. Our adaptive option framework fails to mimic motions extracted from videos with dramatic camera movements, as disentangling human motion from camera motion remains an open issue.

Acknowledgements

We thank the reviewers for their efforts in reviewing this paper. Weiwei Xu is partially supported by NSFC grant No. 92570206 and No. 62421003. Shuaiying Hou is partially supported by the Postdoctoral Fellowship Program of CPSF under Grant No. GZC20252516 and Yongjiang Innovation Project No. 2025Z062. And we also thank the support provided by the Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

References

Bogo, F.; Kanazawa, A.; Lassner, C.; Gehler, P.; Romero, J.; and Black, M. J. 2016. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, 561–578. Springer.

Chen, Z.; He, X.; Wang, Y.-J.; Liao, Q.; Ze, Y.; Li, Z.; Sastry, S. S.; Wu, J.; Sreenath, K.; Gupta, S.; et al. 2024. Learning smooth humanoid locomotion through lipschitz-constrained policies. *arXiv preprint arXiv:2410.11825*.

Choi, H.; Moon, G.; Chang, J. Y.; and Lee, K. M. 2021. Beyond static features for temporally consistent 3d human pose and shape from a video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1964–1973.

Ci, H.; Wang, C.; Ma, X.; and Wang, Y. 2019. Optimizing network structure for 3d human pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2262–2271.

CMU. 2016. CMU Graphics Lab Motion Capture Database. [Online; Accessed October 5, 2016].

Floridi, L.; and Chiriatti, M. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30: 681–694.

Gärtner, E.; Andriluka, M.; Coumans, E.; and Sminchisescu, C. 2022a. Differentiable dynamics for articulated 3d human motion reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13190–13200.

Gärtner, E.; Andriluka, M.; Xu, H.; and Sminchisescu, C. 2022b. Trajectory optimization for physics-based reconstruction of 3d human pose from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13106–13115.

Goel, S.; Pavlakos, G.; Rajasegaran, J.; Kanazawa, A.; and Malik, J. 2023. Humans in 4D: Reconstructing and tracking humans with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14783–14794.

Gross, R.; and Shi, J. 2001. The CMU Motion of Body (MoBo) Database. Technical Report CMU-RI-TR-01-18, Pittsburgh, PA.

Guo, C.; Mu, Y.; Javed, M. G.; Wang, S.; and Cheng, L. 2024. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1900–1910.

Guo, C.; Zou, S.; Zuo, X.; Wang, S.; Ji, W.; Li, X.; and Cheng, L. 2022. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5152–5161.

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2013. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7): 1325–1339.

Jiang, B.; Chen, X.; Liu, W.; Yu, J.; Yu, G.; and Chen, T. 2023. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36: 20067–20079.

Kanazawa, A.; Zhang, J. Y.; Felsen, P.; and Malik, J. 2019. Learning 3d human dynamics from video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5614–5623.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling Laws for Neural Language Models. arXiv:2001.08361.

Karatzas, S.; Papageorgiou, G.; Lazari, V.; Bersimis, S.; Fousteris, A.; Economou, P.; and Chassiakos, A. 2024. A text analytic framework for gaining insights on the integration of digital twins and machine learning for optimizing indoor building environmental performance. *Developments in the Built Environment*, 100386.

Kaufmann, M.; Song, J.; Guo, C.; Shen, K.; Jiang, T.; Tang, C.; Zárate, J. J.; and Hilliges, O. 2023. Emdb: The electromagnetic database of global 3d human pose and shape in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14632–14643.

Kocabas, M.; Athanasiou, N.; and Black, M. J. 2020. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5253–5263.

Li, Z.; Luo, M.; Hou, R.; Zhao, X.; Liu, H.; Chang, H.; Liu, Z.; and Li, C. 2024. Morph: A Motion-free Physics Optimization Framework for Human Motion Generation. *arXiv preprint arXiv:2411.14951*.

Lin, F.; Hu, Y.; Sheng, P.; Wen, C.; You, J.; and Gao, Y. 2024. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*.

Lin, J.; Zeng, A.; Lu, S.; Cai, Y.; Zhang, R.; Wang, H.; and Zhang, L. 2023. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 36: 25268–25280.

Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; and Black, M. J. 2023. SMPL: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 851–866.

Luo, Z.; Cao, J.; Merel, J.; Winkler, A.; Huang, J.; Kitani, K. M.; and Xu, W. 2024. Universal Humanoid Motion Representations for Physics-Based Control. In *The Twelfth International Conference on Learning Representations*.

Luo, Z.; Golestaneh, S. A.; and Kitani, K. M. 2020. 3d human motion estimation via motion compression and refinement. In *Proceedings of the Asian Conference on Computer Vision*.

Mahmood, N.; Ghorbani, N.; Troje, N. F.; Pons-Moll, G.; and Black, M. J. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5442–5451.

Mandery, C.; Terlemez, Ö.; Do, M.; Vahrenkamp, N.; and Asfour, T. 2015. The KIT whole-body human motion database. In *2015 International Conference on Advanced Robotics (ICAR)*, 329–336. IEEE.

Onuma, K.; Faloutsos, C.; and Hodgins, J. K. 2008. FMDistance: A Fast and Effective Distance Function for Motion Capture Data. *Eurographics (Short Papers)*, 7.

Pavlakos, G.; Choutas, V.; Ghorbani, N.; Bolkart, T.; Osman, A. A.; Tzionas, D.; and Black, M. J. 2019. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10975–10985.

Peng, X. B.; Abbeel, P.; Levine, S.; and Van de Panne, M. 2018a. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4): 1–14.

Peng, X. B.; Guo, Y.; Halper, L.; Levine, S.; and Fidler, S. 2022. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4): 1–17.

Peng, X. B.; Kanazawa, A.; Malik, J.; Abbeel, P.; and Levine, S. 2018b. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions On Graphics (TOG)*, 37(6): 1–14.

Plappert, M.; Mandery, C.; and Asfour, T. 2016. The kit motion-language dataset. *Big data*, 4(4): 236–252.

Rempe, D.; Guibas, L. J.; Hertzmann, A.; Russell, B.; Villegas, R.; and Yang, J. 2020. 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*, 71–87. Springer.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shen, Z.; Pi, H.; Xia, Y.; Cen, Z.; Peng, S.; Hu, Z.; Bao, H.; Hu, R.; and Zhou, X. 2024. World-Grounded Human Motion Recovery via Gravity-View Coordinates. In *SIGGRAPH Asia 2024 Conference Papers*, 1–11.

Shimada, S.; Golyanik, V.; Xu, W.; Pérez, P.; and Theobalt, C. 2021. Neural monocular 3d human motion capture with physical awareness. *ACM Transactions on Graphics (ToG)*, 40(4): 1–15.

Shimada, S.; Golyanik, V.; Xu, W.; and Theobalt, C. 2020. Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics (ToG)*, 39(6): 1–16.

Shin, S.; Kim, J.; Halilaj, E.; and Black, M. J. 2024. Wham: Reconstructing world-grounded humans with accurate 3d motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2070–2080.

Sigal, L.; Balan, A. O.; and Black, M. J. 2010. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1): 4–27.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211.

Tessler, C.; Guo, Y.; Nabati, O.; Chechik, G.; and Peng, X. B. 2024. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions on Graphics (TOG)*, 43(6): 1–21.

Tevet, G.; Raab, S.; Gordon, B.; Shafir, Y.; Cohen-or, D.; and Bermano, A. H. 2023. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*.

Trumble, M.; Gilbert, A.; Malleson, C.; Hilton, A.; and Collomosse, J. P. 2017. Total capture: 3D human pose estimation fusing video and inertial sensors. In *BMVC*, volume 2, 1–13. London, UK.

Tsuchida, S.; Fukayama, S.; Hamasaki, M.; and Goto, M. 2019. AIST Dance Video Database: Multi-Genre, Multi-Dancer, and Multi-Camera Database for Dance Information Processing. In *ISMIR*, volume 1, 6.

Wan, Z.; Li, Z.; Tian, M.; Liu, J.; Yi, S.; and Li, H. 2021. Encoder-decoder with multi-level attention for 3d human shape and pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13033–13042.

Won, J.; Gopinath, D.; and Hodgins, J. 2020. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Transactions on Graphics (TOG)*, 39(4): 33–1.

Yin, W.; Cai, Z.; Wang, R.; Wang, F.; Wei, C.; Mei, H.; Xiao, W.; Yang, Z.; Sun, Q.; Yamashita, A.; et al. 2025. Whac: World-grounded humans and cameras. In *European Conference on Computer Vision*, 20–37. Springer.

Yuan, Y.; Wei, S.-E.; Simon, T.; Kitani, K.; and Saragih, J. 2021. Simpoe: Simulated character control for 3d human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7159–7169.

Zhang, J.; Zhang, Y.; Cun, X.; Zhang, Y.; Zhao, H.; Lu, H.; Shen, X.; and Shan, Y. 2023a. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14730–14740.

Zhang, M.; Cai, Z.; Pan, L.; Hong, F.; Guo, X.; Yang, L.; and Liu, Z. 2024. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, M.; Guo, X.; Pan, L.; Cai, Z.; Hong, F.; Li, H.; Yang, L.; and Liu, Z. 2023b. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 364–373.