UniTracker: Learning Universal Whole-Body Motion Tracker for Humanoid Robots

Kangning Yin[1,2*], Weishuai Zeng[2,3*], Ke Fan[1,2], Zirui Wang[2,4],Qiang Zhang[5],

Zheng Tian[6], Jingbo Wang[2], Jiangmiao Pang[2], Weinan Zhang[1,2]

[1]Shanghai Jiao Tong Univeristy, [2]Shanghai Artificial Intelligence Laboratory,
[3]Peking University, [4]Zhejiang University,
[5]The Hong Kong University of Science and Technology (Guangzhou), [6]ShanghaiTech University

https://yinkangning0124.github.io/Humanoid-UniTracker/

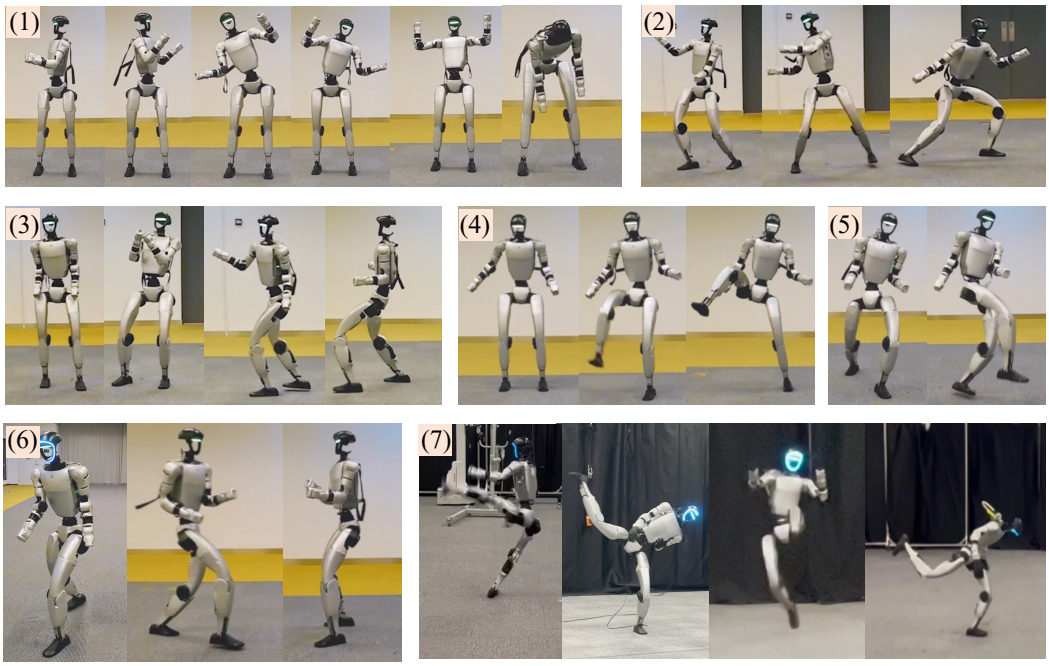

Figure 1: We deploy our UniTracker on a real humanoid robot, enabling it to perform a diverse range of motions, including (1)stretching, (2)traditional Chinese martial arts, (3)dancing, (4)high kicks, (5)ball-kicking, (6)various other common motions, and (7)challenging motions by fast adaption.

Abstract: Achieving expressive and generalizable whole-body motion control is essential for deploying humanoid robots in real-world environments. In this work, we propose UniTracker, a three-stage training framework that enables robust and scalable motion tracking across a wide range of human behaviors. In the first stage, we train a teacher policy with privileged observations to generate high-quality actions. In the second stage, we introduce a Conditional Variational Autoencoder (CVAE) to model a universal student policy that can be deployed directly on real hardware. The CVAE structure allows the policy to learn a global latent representation of motion, enhancing generalization to unseen behaviors and addressing the limitations of standard MLP-based policies under partial observations. Unlike pure MLPs that suffer from drift in global attributes like orientation, our CVAE-student policy incorporates global intent during training by aligning a partial-observation prior to the full-observation encoder. In the third stage, we in-

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

troduce a fast adaptation module that fine-tunes the universal policy on harder motion sequences that are difficult to track directly. This adaptation can be performed both for single sequences and in batch mode, further showcasing the flexibility and scalability of our approach. We evaluate UniTracker in both simulation and real-world settings using a Unitree G1 humanoid, demonstrating strong performance in motion diversity, tracking accuracy, and deployment robustness.

Keywords: Whole-Body Humanoid Robot Tracker, Imitation Learning, Reinforcement Learning

1 Introduction

Humanoid robots have garnered growing interest in the robotics community for their human-like morphology, which equips them with the potential to perform a wide range of tasks traditionally carried out by humans. To function effectively in real-world, human-centric environments, these robots must exhibit not only physical versatility but also robust and expressive motor control. Among the key enablers of such capabilities is whole-body control, which coordinates multiple joints and limbs to perform complex tasks while ensuring stability, expressiveness and adaptability.

Recent research has explored various control interfaces tailored to humanoid whole-body controller. These can be broadly categorized into dense and sparse control signals. Dense signals, such as teleoperation [1, 2, 3, 4, 5, 6], offline motion datasets [7, 8, 9, 10, 11, 12, 13], and video-based motion estimation [14, 15], provide rich trajectory-level information. In contrast, sparse signals such as high-level task commands and VR-based guidance [16, 17] offer minimal information and often lead to reduced motion quality. In this work, we focus on **universal whole-body motion tracking**, where the input is a reference motion sequence and the goal is to track it robustly and expressively using a single policy.

A widely adopted paradigm for learning motion tracking policies is the teacher-student framework. In this paradigm, a teacher policy is initially trained using full privileged observations to precisely track reference motions within simulation environments. Subsequently, a student policy is learned to imitate the teacher policy based solely on the partial observations available during deployment. Despite its effectiveness, existing implementations of this framework, particularly those utilizing simple MLP-based DAgger architectures [18] encounter three significant limitations. First, they frequently fail to preserve the diversity inherent in the original reference motions during the distillation process, leading to behavior that is averaged and less expressive. Second, constrained by their limited representational capacity, such models often exhibit poor generalization to unseen motion sequences. Third, the absence of global contextual information during training gives rise to issues such as orientation drift and broader inconsistencies in global behavior, which become particularly pronounced when the policies are deployed on real-world humanoid robots.

To address the aforementioned limitations of existing teacher-student frameworks, we introduce UniTracker, a unified and expressive whole-body tracking framework that integrates a Conditional Variational Autoencoder (CVAE) [19] into the student policy architecture. By explicitly modeling a structured latent space conditioned on future motion references, UniTracker enables the policy to generate diverse and high-fidelity behaviors even under partial observations. From a probabilistic standpoint, the latent variable captures the inherent ambiguity in the mapping from observations to actions, allowing the policy to model a distribution over plausible motor behaviors rather than collapsing to a single deterministic output. This capability enhances motion expressiveness and significantly improves generalization across diverse and unseen movement patterns.

In addition, the CVAE-based framework also effectively addresses the challenge of missing global context—often manifested as orientation drift and other global inconsistencies during deployment. To this end, we employ task-aware feature modeling during training: the encoder is trained using privileged, globally informative observations to infer a structured latent representation, while a prior network is concurrently trained based only on the partial observations available at deployment time.

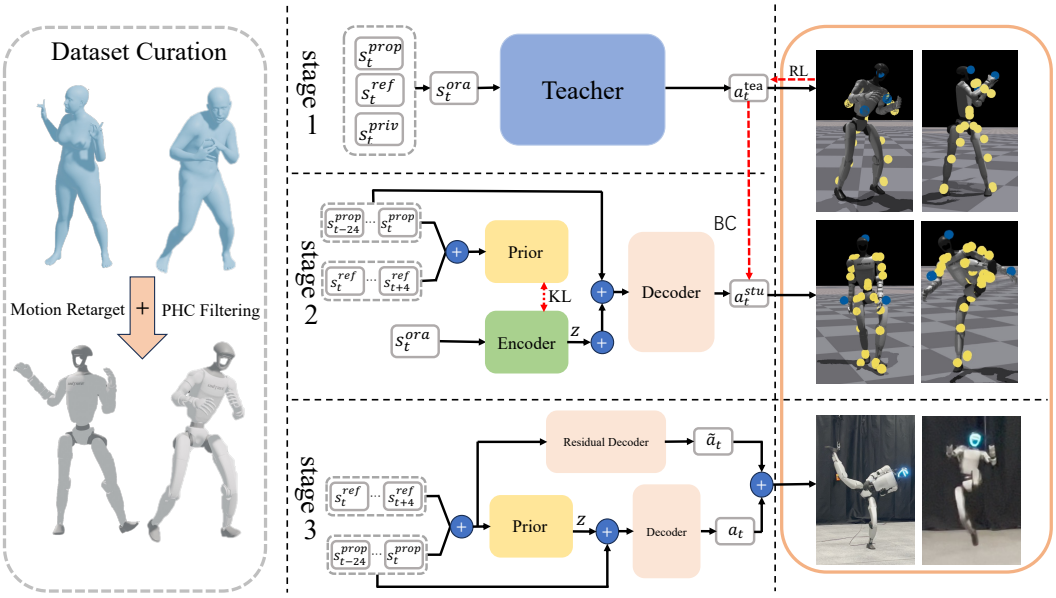

Figure 2: **An overview of UniTracker**: In Stage 1, we train a teacher policy using oracle states via goal-conditioned reinforcement learning. In Stage 2, we distill the policy into a deployable form using a CVAE-based DAgger framework. In Stage 3, we introduce a fast adaptation module for handling challenging motion sequences, implemented using a residual decoder. The training dataset is derived from the AMASS dataset, filtered by PHC to remove physically infeasible motions.

The two distributions are aligned via a KL divergence objective. As a result, although the final deployed policy operates under partial observability, it benefits from a latent space informed by global context during training. This implicit incorporation of global information leads to more coherent and globally consistent behaviors in real-world settings.

While this CVAE-based universal policy demonstrates strong performance across a wide range of motions, it is neither necessary nor realistic to expect it to perfectly track all possible reference sequences—particularly those that are rare, highly dynamic, or lie far outside the training distribution. To accommodate such challenging cases, we introduce a fast adaptation phase that fine-tunes the universal policy in a task-specific manner. Leveraging the expressiveness and generality of the base policy, this adaptation process enables rapid specialization with minimal training time. Moreover, our framework supports both single-sequence adaptation and batch-mode adaptation, allowing for scalable refinement when dealing with multiple difficult motions. This final phase complements the universal policy by extending its practical applicability to edge cases, and highlights the modularity and flexibility of our overall three-stage training framework.

We extensively evaluate UniTracker in both simulated environments and real-world deployment scenarios. Experiments conducted on a 29-DoF Unitree G1 humanoid demonstrate that our policy is capable of tracking over 8,100 diverse motion sequences—including highly dynamic behaviors—using a single unified network. Compared to strong teacher-student baselines that do not incorporate CVAE-based modeling, UniTracker consistently achieves superior performance in terms of tracking accuracy, robustness, and generalization to unseen motions.

Our main contributions are summarized as follows: i)**A three-stage training framework for universal whole-body tracking:** We design a modular pipeline comprising a privileged teacher for data generation, a CVAE-based student policy for deployment under partial observations, and a lightweight fast adaptation phase for motion-specific fine-tuning. ii)**Diversity-aware and gloabl context-integrated policy via CVAE modeling:** We employ a Conditional Variational Autoencoder to capture motion diversity and encode global context, enabling expressive behaviors and reducing global inconsistencies by aligning a globally-informed encoder with a partial-observation prior. iii)**Fast adaptation on top of a universal policy:** We introduce a rapid fine-tuning mecha-

nism that adapts the universal policy to challenging or out-of-distribution motions, supporting both single-sequence and batch-level adaptation with minimal overhead. iv)**Extensive validation on real-world hardware:** We demonstrate that UniTracker robustly tracks over 8,100 human motions on a 29-DoF humanoid using a single policy, outperforming strong teacher-student baselines in accuracy, robustness, and generalization.

2 Method

2.1 Problem Formulation

We formulate the problem of humanoid robot whole-body motion tracking as a goal-conditioned reinforcement learning (RL) task, where a policy π is trained to track reference motions at the whole-body level. The state s_t comprises the robot's proprioceptive information s_t^p and the goal s_t^g which specifies the target state for all body parts. The reward function $r_t = R(s_t^p, s_t^g)$, defined in terms of the agent's proprioception and goal state, yields dense signals to guide policy optimization. To better focus on motion tracking at the whole-body level, we fix the wrist joints of our 29-degree-of-freedom (DoF) Unitree G1 robot[20], reducing the action space to 23 dimensions. The action $a_t \in \mathbb{R}^{23}$ specifies target joint positions, which are executed via a PD controller to actuate the robot. For policy optimization, we employ Proximal Policy Optimization (PPO) [21] to maximize the expected cumulative discounted reward $\mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$.

The remainder of this section is organized as follows. Section 2.2 introduces the construction of a high-quality humanoid motion dataset for policy training. Section 2.3 describes the training of an oracle policy in simulation, which is designed both to maximize the expressiveness of motion tracking. Section 2.4 details the distillation of the oracle policy into a deployable student using a CVAE-based framework. Section 2.5 presents our fast adaptation strategy, which enables rapid fine-tuning based on the universal tracker. An overview of the proposed universal whole-body motion tracking framework is provided in Figure 2.

2.2 Humanoid Motion Dataset Curation

A large-scale humanoid motion dataset serves as fuel for training a universal motion tracker. Our dataset is primarily derived from the publicly available AMASS [22] dataset, filtered to exclude interactions and sequences shorter than 10 frames. This yields a final training set of 11,313 human motions, represented using the SMPL [23] parameters. The SMPL model parameterizes the human body through shape parameters $\beta \in \mathbb{R}^{10}$, pose parameters $\theta \in \mathbb{R}^{24\times 3}$ and root translation $p \in \mathbb{R}^3$. S denotes the SMPL function, where $S(\beta, \theta, p) : \beta, \theta, p \rightarrow \mathbb{R}^{6980\times 3}$ maps the SMPL parameters to the positions of vertices of a triangular human mesh.

To bridge the embodiment gap between SMPL human model and humanoid robots, we employ a two-stage retargeting approach inspired by H2O [4]. First, we carefully select 16 corresponding body links and optimize the shape parameter β' for humanoid robots by minimizing the distances between selected links in the rest pose. Second, leveraging the optimized β' alongside the original pose θ and translation p from dataset, we perform gradient descent over the humanoid robot's root translation, root orientation and joint positions to minimize the distances between selected links throughout the whole sequence. Additional regularization terms are added to avoid aggressive behaviors and ensure temporal smoothness.

2.3 Oracle Policy Training in Simulation

Oracle State Space Design. We train an oracle motion tracking policy $\pi^{orcale}(a_t|s_t^{p-oracle}, s_t^{g-oracle})$ with all the state information accessible in simulators. The proprioception is defined as $s_t^{p-oracle} \triangleq [p_t, q_t, \theta_t, \dot{p}_t, \dot{q}_t, \omega_t, a_{t-1}]$, which contains the humanoid rigid-body position p_t, orientation θ_t, linear velocity $\dot{p}_t$, angular velocity ω_t, joint position q_t, joint velocity $\dot{q}_t$ and previous action a_{t-1}. The goal state is defined as

$s_t^{g-oracle} \triangleq [\hat{p}_{t+1} - p_t, \hat{q}_{t+1} - q_t, \hat{\theta}_{t+1} \ominus \theta_t, \hat{v}_{t+1} - v_t, \hat{\omega}_{t+1} - \omega_t, \hat{p}_{t+1} - p_t^{root}, \hat{\theta}_{t+1} \ominus \theta_t^{root}]$, which contains the one-frame difference between the reference pose $(\hat{p}_{t+1}, \hat{q}_{t+1}, \hat{\theta}_{t+1}, \hat{v}_{t+1}, \hat{\omega}_{t+1})$ and the current pose. p_t^{root} refers to the root translation and θ_t^{root} refers to the root orientation of the current pose. All these states are rotated to the local coordinate of the current pose.

Reward Design. We formulate the reward r_t as a weighted sum of three components: 1) task rewards for motion tracking, 2)regularization, and 3) penalty, detailed in Table 3. We apply curriculum learning to the regularization terms and penalty terms such that the policy could better focus on the motion tracking task itself and gradually take the penalty and regularization into account for more reasonable behaviors.

Early Termination and Reference State Initialization. In the early stage of training, the agent is prone to falling, resulting in the collection of invalid data that hinders effective learning. To address this issue, we follow prior works [24, 25] and introduce two early termination conditions: 1) orientation: the projected gravity on x or y axis exceeds 0.8; 2) tracking tolerance: the average link distance between the robot and reference motions is further than 0.5m. Proper task initialization is also crucial for RL training. We employ the Reference State Initialization [24, 9] framework, where the starting point of the reference motion is randomly sampled for the policy to track. The robot's initial state, including the root position, orientation, linear and angular velocities, as well as joint positions and velocities, is then derived from the corresponding reference pose. This initialization strategy substantially enhances motion tracking training by enabling the policy to learn different motions phases in parallel, rather being constrained to a sequential learning process.

Domain Randomization. Domain randomization has been proved to be a critical source of robustness and generalization for successful sim-to-real transfer [11, 4, 6]. Existing domain randomization terms can be broadly categorized into two classes: asset property randomization and environmental dynamic randomization. Asset property randomization, focusing on parameters such as friction coefficients, center of mass offsets, and link masses, aims to prevent the policy from overfitting to a specific asset configuration. In contrast, environmental dynamic randomization, focusing on PD gains, applied torques, or other external perturbations (e.g., pushing the robot), introduces dynamic variations to simulate the environmental uncertainties encountered during real-world deployment. While previous works typically apply both forms of randomization concurrently throughout training, we propose decoupling them to better balance the tradeoff between motion tracking expressiveness and robustness. Hence, in the first stage, we exclusively randomize asset properties to maximize the policy's ability to accurately track diverse motions. This approach not only builds an upper bound for subsequent policies but also yields a proxy agent capable of generating high-quality action signals in simulators.

2.4 Hierarchical Controller via Online Distillation

Deployable State Space Design. Since certain privileged information in the oracle state space is unavailable in real-world deployment, we define a deployable state space based on data accessible on humanoid robots. The proprioception is defined as $s_t^{p-deploy} \triangleq [q_{t-25:t}, \dot{q}_{t-25:t}, w_{t-25:t}^{root}, g_{t-25:t}, a_{t-25:t-1}]$ where q_t and $\dot{q}_t$ denote joint positions and velocities, w_t^{root} refers to the root angular velocity, g_t is the gravity vector and a_{t-1} is the previous action. These terms are stacked over the past 25 steps to form the proprioceptive input. The goal state is defined as $s_t^{g-deploy} \triangleq [\hat{h}_{t+1}, \hat{\theta}_{t+1}^{root} \ominus \theta_t^{root}, \hat{v}_{t+1}^{root}, \hat{w}_{t+1}^{root} - w_t^{root}, \hat{p}_{t+1} - \hat{p}_{t+1}^{root}]$, where $\hat{h}_{t+1}$ is the reference pose height, $\hat{\theta}_{t+1}^{root}$ and θ_t^{root} are the reference and current root orientations, $\hat{v}_{t+1}^{root}$ and $\hat{w}_{t+1}^{root}$ are the reference root linear and angular velocities, w_t^{root} is the current root angular velocity, $\hat{p}_{t+1}$ and $\hat{p}_{t+1}^{root}$ are the reference rigid body and root positions. The first four terms are rotated into the current pose's local coordinate while the last term is rotated into the reference pose's local frame.

Modeling Diversity with CVAE Our target is to obtain a deployable policy $\pi^{deploy}(a_t|s_t^{p-deploy}, s_t^{g-deploy})$ which maintains the expressiveness of the oracle policy to the maximum. Given the deployable state space is partial and may lead to motion ambiguity, We model the deployable policy as a conditional variational autoencoder (CVAE) to model the diversity.

Specifically, we have a variational encoder $\varepsilon(z_t|s_t^{p-orcale}, s_t^{g-oracle})$ that computes the latent code distribution based on the same observation with the teacher policy, a decoder $D(a_t|s_t^{p-deploy}, z_t)$ that produces action. We employ a learned conditional prior $\rho(z_t|s_t^{p-deploy}, s_t^{g-deploy})$ following prior works which allows the model to learn different distributions based on proprioception. Using the evidence lower bound, we have the objective function as:

$$\begin{aligned} \log P(a_t|s_t^{p-d}, s_t^{g-d}) \geq E_{\varepsilon(z|s_t^{p-o}, s_t^{g-o})}[\log D(a_t|s_t^{p-d}, z_t) \\ - D_{KL}(\varepsilon(z_t|s_t^{p-o}, s_t^{g-o})||\rho(z_t|s_t^{p-d}, s_t^{g-d}))] \end{aligned} \tag{1}$$

where the prior, encoder and decoder are all modeled as diagonal Gaussian distribution. Following prior works, we model the encoder as a residual to the prior and have:

$$\begin{aligned} \rho(z_t|s_t^{p-d}, s_t^{g-d}) &= N(\mu^\rho(s_t^{p-d}, s_t^{g-d}), \sigma^\rho(s_t^{p-d}, s_t^{g-d})) \\ \varepsilon(z_t|s_t^{p-o}, s_t^{g-o}) &= N(\mu^\rho(s_t^{p-d}, s_t^{g-d}) + \mu^\varepsilon(s_t^{p-o}, s_t^{g-o}), \\ &\qquad \sigma^\varepsilon(s_t^{p-o}, s_t^{g-o})) \\ D(a_t|s_t^{p-d}, z_t) &= N(\mu^D(s_t^{p-d}, z_t), \sigma^D(s_t^{p-d}, z_t)) \end{aligned} \tag{2}$$

To optimize the loss function, we use the action supervision signal provided by the oracle policy and learn the deployable policy in an online distillation fashion. The loss function derived from the objective above can be formulated as:

$$L = L_{action} + \beta L_{KL} \tag{3}$$

where $L_{action} = ||a_t^{deploy} - a_t^{oracle}||_2^2$ and L_{KL} is the KL divergence between the prior and encoder.

2.5 Fast Adaption on top of a Universal Policy

In this stage, we focus on motion references that the universal policy fails to reliably track. These motions typically exhibit two challenging characteristics: they lie far outside the training data distribution, and they involve highly dynamic behaviors that approach the physical limits of the humanoid robot.

To address these cases, we introduce a lightweight residual decoder $D(\widetilde{a_t}|{s_t}^{p-deploy}, {s_t}^{g-deploy})$ appended to the universal motion tracking framework, to enable fast adaptation to challenging motions. During training, we initialize the model with the universal policy checkpoint and freeze all existing parameters. Only the residual decoder is fine-tuned using reinforcement learning, the reward is the same as Section 2.3. The input to the residual decoder is identical to that of the prior network. We directly use explicit motion references instead of latent representations for two main reasons. First, such out-of-distribution motions may not be adequately captured by the learned CVAE prior. Second, the goal at this stage is to quickly overfit to specific motion sequences rather than to maintain generalization. The final action applied to the environment is obtained by summing the output of the universal policy and the residual correction:

$$a_t^{final} = a_t + \widetilde{a_t}, \tag{4}$$

where a_t is the action predicted by the frozen universal policy, $\widetilde{a_t}$ is the residual generated by the residual decoder.

3 Experimental Results

3.1 Experiment Setup

We evaluate UniTracker in both simulated and real-world environments. In simulation, policies are trained using IsaacGym [26] with 8192 parallel environments under domain randomization [27]. The training data is derived from the AMASS dataset [22] and filtered by PHC [25]. We further validate our approach through sim-to-sim transfer by deploying the trained policies in MuJoCo, and

Methods	All AMASS Train Dataset				Successful AMASS Train Dataset		
	SR↑	MPKPE↓	Vel-Dist↓	Acc-Dist↓	MPKPE↓	Vel-Dist↓	Acc-Dist↓
(a) Compare with Baselines							
Dagger without CVAE	88.21	84.79	5.60	2.97	84.70	5.03	2.41
Train from Scratch	72.11	103.94	8.78	6.33	98.61	8.47	6.31
Ours	**91.83**	**82.62**	**4.27**	**1.83**	**78.95**	**3.65**	**1.39**
(b) Ablation with Architecture Design							
Actor with Explicit Reference	88.29	85.31	5.76	3.03	84.96	4.58	2.15
Actor without Explicit Reference	**91.83**	**82.62**	**4.27**	**1.83**	**78.95**	**3.65**	**1.39**
(c) Ablation with KL Residual							
KL without Residual	85.49	91.24	6.02	4.68	90.79	6.01	4.23
KL with Residual	**91.83**	**82.62**	**4.27**	**1.83**	**78.95**	**3.65**	**1.39**
(d) Ablation with KL Coefficient							
KL Coef = 1.0	81.18	97.23	7.75	6.04	96.53	5.39	2.41
KL Coef = 0.1	**91.83**	**82.62**	**4.27**	**1.83**	**78.95**	**3.65**	**1.39**
KL Coef = 0.01	86.10	89.88	7.99	5.47	88.01	5.23	2.37
KL Coef = 0.001	80.55	105.56	9.21	7.98	104.63	5.38	2.40
(e) Ablation with Future Window Size							
Window Size = 1	90.77	85.98	5.20	3.12	85.00	4.79	2.91
Window Size = 5	**91.83**	82.62	**4.27**	**1.83**	78.95	**3.65**	**1.39**
Window Size = 10	91.28	**82.14**	4.77	1.95	**78.78**	3.90	1.87
Window Size = 20	90.29	82.96	5.01	1.78	80.24	4.33	1.48
(f) Ablation with Latent Dimension							
Latent Dimension = 32	88.82	87.25	4.79	2.36	86.52	4.67	2.21
Latent Dimension = 64	**91.83**	**82.62**	4.27	**1.83**	**78.95**	**3.65**	**1.39**
Latent Dimension = 128	90.54	83.75	4.46	2.14	82.92	4.45	2.08
Latent Dimension = 256	90.72	83.99	**4.21**	1.97	83.71	4.19	1.77
(g) Analysis of CVAE diversity							
Decoder with Stochastic Latent	91.77	83.21	4.59	1.82	79.07	3.72	**1.33**
Decoder with Deterministic Latent	**91.83**	**82.62**	**4.27**	**1.83**	**78.95**	**3.65**	1.39

Table 1: **Baselines Comparison and Ablation Studies** We compare our method with two baseline approaches and observe that it consistently outperforms both. Additionally, we conduct ablation studies on architectural design, KL residuals, KL coefficient, future window size, and latent dimensionality. Finally, we analyze the diversity of CVAE comparing the stochastic latent and the deterministic latent. The final configuration is selected based on the best overall performance observed across these settings.

conduct extensive ablation and comparative studies. Performance is assessed using four key metrics: Success Rate (SR), Mean Per Keypoint Position Error (MPKPE), Velocity Distance(Vel-Dist) and Acceleration Distance(Acc-Dist). SR reflects the overall viability and stability of the policy; MPKPE respectively quantify the accuracy of keypoint tracking in the world coordinate frame; Vel-Dist quantifies the difference in joint velocities between the reference motion and the executed motion of the robot; Acc-Dist quantifies the difference in joint acceleration between the reference motion and the executed motion of the robot. For real-world evaluation, we deploy UniTracker on the Unitree G1 humanoid robot, which stands 1.3 meters tall and has 29 degrees of freedom. We control 23 DoFs by locking the 6 wrist joints.

3.2 Baselines

What kind of motion tracker yields the best tracking performance? We include two baselines: (1) a universal policy trained from scratch without the teacher-student architecture, and (2) a teacher-student framework using DAgger in the second stage without a CVAE. Direct comparisons with prior

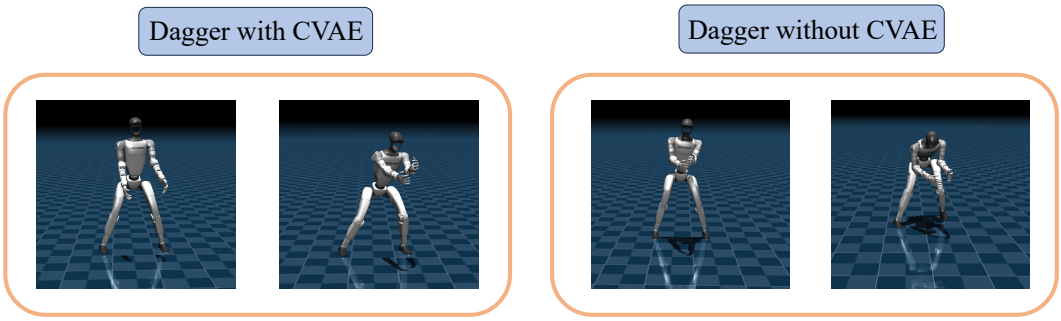

Figure 3: Generalization Capability of UniTracker in MuJoCo

methods are not included, as their implementations are not publicly available. Experimental results in Table 1 show that UniTracker consistently outperforms our baselines across all evaluation metrics.

3.3 Ablation Studies

What model architecture best balances diversity, expressiveness and robustness? To answer this question, we first investigated whether the RL actor (the decoder in the CVAE) should explicitly take the reference motion as part of its input. We evaluated two variants: one where the actor input is $({s_t}^{p-deploy}, {s_t}^{g-deploy}, z_t)$ and another where the reference motion is excluded. The results show that when the actor receives the reference motion directly, the influence of the latent variable z vanishes. In this case, the behavior of the model closely resembles that of a standard DAgger setup, as the strong reference input causes the policy to ignore the latent guidance. We further evaluated a pure DAgger setup with an MLP-based actor in the second stage. While this design performs well on motions seen during training, it degrades noticeably on out-of-distribution (OOD) motions. The strong reliance on explicit reference inputs reduces the policy's robustness and its ability to generalize to unseen or novel motion sequences.

In Figure 3, we evaluate the generalization capability of UniTracker in MuJoCo by testing it on a challenging lateral squat motion, which demands strong balance control from the robot. Although UniTracker cannot precisely track all keypoints of the reference motion, it successfully produces a plausible lateral squat behavior while maintaining balance. In contrast, the DAgger baseline without the CVAE fails to remain stable and results in the robot falling, highlighting the importance of latent motion modeling for generalization to unseen motions. We have uploaded more real-world OOD videos on our website, showcasing the generalization ability of our policy.

In Table 2, we evaluate the robustness of our policy by incrementally adding noise to the observations. In this table, a noise level of 0 indicates no observation noise, while higher levels correspond to increasing noise intensity. We observe that UniTracker consistently outperforms the variant without CVAE in both Success Rate (SR) and Mean Per Keypoint Position Error (MPkPE). Moreover, UniTracker exhibits a slower degradation in performance as noise increases, indicating stronger robustness under observation perturbations.

In Figure 5, we evaluate the global awareness of our CVAE-based framework using a walking sequence in the MuJoCo simulator. The results show that, under our framework, the robot is able to walk in a straight line while closely following the reference trajectory. In contrast, the framework without CVAE gradually deviates from the reference path, demonstrating the superiority of our approach in maintaining global consistency.

What hyperparameter choices lead to the best performance in whole-body motion tracking? We conduct ablation studies on three key hyperparameters in our framework: the CVAE latent dimension, the future window size of the reference motion used in the CVAE prior, and the weight of the KL loss term. For the latent dimension, we experiment with values of 32, 64, 128, and 256. We observe that a dimension of 64 yields the best performance. Larger latent sizes make the CVAE harder to train and do not lead to further improvements. Results are shown in Table 1 For the future window, which controls how many future frames of reference motion are included in the prior input,

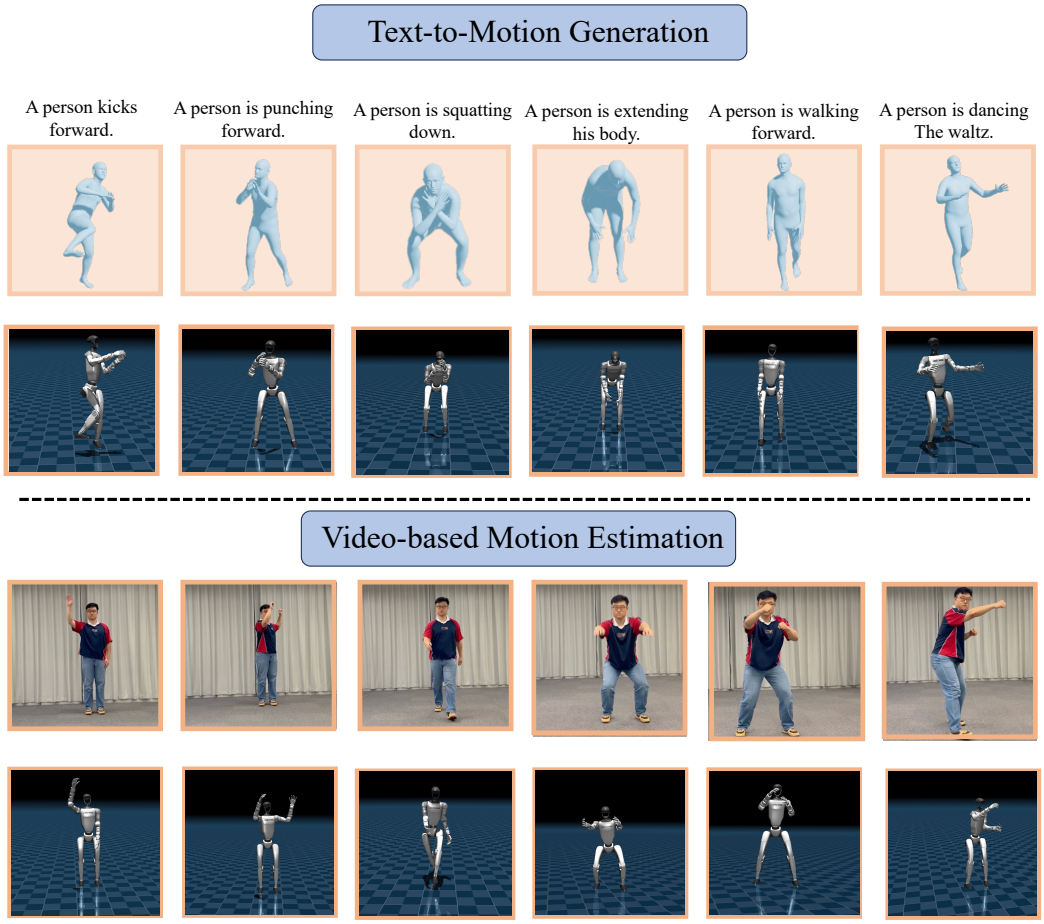

Figure 4: **The Outcome of Downstream Applications in mujoco:** We evaluate text-to-motion generation and video-based motion estimation in the muJoCo simulator. The results demonstrate that our policy can effectively track both types of reference motions

we test window sizes of 1, 5, 10, and 20 frames. The results indicate that using 5 future frames provides the best trade-off between responsiveness and stability. Lastly, we tune the KL loss coefficient with values of 1.0, 0.1, 0.01, and 0.001. We find that setting the weight to 0.1 leads to the best policy performance, effectively balancing latent regularization and reconstruction quality.

3.4 Downstream Applications

We evaluate two downstream applications of our framework: motion generation and video-based motion estimation. For motion generation, we use the MDM [28] model to produce SMPL-based [23] motion sequences conditioned on text input. These motions are then retargeted to the G1 humanoid robot for tracking. Both MuJoCo [29] simulation and real-world results demonstrate that our policy can accurately track motions generated from text prompts. For video estimation, we record a long motion sequence using a monocular camera, and convert it to SMPL format using GVHMR [30]. The resulting motion is retargeted to the G1 robot. Experiments in both MuJoCo and the real world show that our policy successfully tracks the reconstructed motions. These two applications highlight the strong generalization capability of our policy across different types of reference inputs.

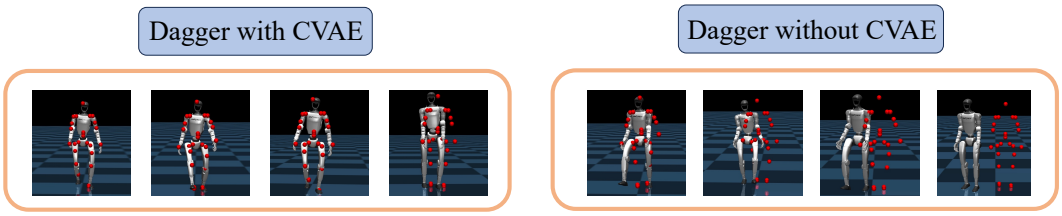

Figure 5: Global Consistency of UniTracker in the MuJoCo Simulator

3.5 Light Weight Fast Adaption

We compared our fast adaptation module against a training-from-scratch baseline. In the latter, the observations of our model remain consistent with those of the second-stage student policy, and the reward function is identical to that used in the fast adaptation module. For evaluation, we selected three challenging motion sequences—Round-Horse Kick, Side Kick, and Martial Art—along with the entire AMASS test dataset. As shown in Figure 6, our fast adaptation approach converges substantially faster than training from scratch in terms of both episode length and cumulative reward. These results underscore the advantages of our second-stage universal policy and highlight the effectiveness of the pretrained model in enabling rapid adaptation.

Methods	**All AMASS Train Dataset**	
	SR↑	MPKPE↓
(a) Noise Level 0		
Dagger without CVAE	88.21	84.79
Ours	**91.83**	**82.62**
(b) Noise Level 1		
Dagger without CVAE	85.79	88.65
Ours	**90.26**	**83.84**
(c) Noise Level 2		
Dagger without CVAE	79.58	93.71
Ours	**86.79**	**87.31**

Table 2: We evaluate the robustness of the policy by incrementally adding noise to the observations.

4 Related Work

4.1 Whole-Body Controller for Humanoid Robots

Whole-body control is essential for enabling humanoid robots to perform a wide variety of complex tasks. Prior to the rise of reinforcement learning, researchers primarily relied on traditional optimization-based control methods for humanoid whole-body control [31, 32, 33, 34, 35, 36, 37]. These approaches typically required explicit mathematical modeling of both the robot and its environment, followed by real-time optimization to compute the robot's next action. However, such methods often struggle to adapt to environmental variations, resulting in limited robustness. Additionally, they impose heavy computational demands during online execution.

To overcome these limitations, reinforcement learning (RL) has emerged as a powerful alternative, offering the ability to learn adaptive, robust control policies directly from interaction with the environment without relying on explicit modeling. Current reinforcement learning–based whole-body controllers for humanoid robots can be categorized by the source of their control signals, including teleoperation [1, 2, 3, 4, 5, 6], offline motion datasets [7, 8, 10, 11], video-based motion estimation [14, 15] [5, 6], and high-level task commands [16, 17]. Teleoperation involves directly controlling the humanoid robot in real time using human input, often through motion capture systems or wearable sensors, allowing the robot to mimic human movements with high fidelity. Representative works in this area include Twist [1] and H2O [4], both of which adopt a two-stage teacher-student framework. The primary difference lies in the design of the policy's observation space. Offline motion datasets consist of pre-collected human or humanoid motion sequences, which are used as references for training control policies through motion imitation. Representative works are Exbody [11] and Exbody2 [10], which begins by carefully curating an offline motion dataset and then decouples upperand lower-body motions as much as possible, aiming to maintain stability in the lower body while encourage diversity and expressiveness in the upper body. In addition, the most recent work, GMT [12], is the first to demonstrate tracking of 8,000 motions using a single unified policy. Video-based motion estimation methods leverage visual input from videos to extract human motion data, which can then be used to guide humanoid robot control policies. This approach

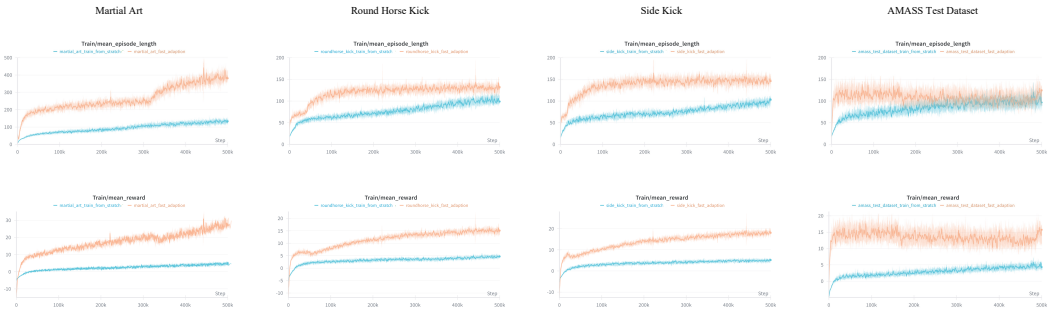

Figure 6: Fast Adaption of Challenging Motions

Name	Function	Weight
tracking keypoints	$exp(\|\|{p_t}^{ref} - p_t\|\|)$	1.6
tracking feet position	$exp(\|\|{p_{t_{feet}}}^{ref} - p_{t_{feet}}\|\|)$	2.1
tracking body rotation	$exp(\|\|r_t - {r_t}^{ref}\|\|)$	0.5
tracking joint position	$exp(\|\|{q_t}^{ref} - q_t\|\|)$	0.75
tracking joint velocity	$exp(\|\|\dot{{q_t}^{ref}} - \dot{q_t}\|\|)$	0.5
tracking body linear velocity	$exp(\|\|\dot{{p_t}^{ref}} - \dot{p_t}\|\|)$	0.5
tracking body angular velocity	$exp(\|\|\dot{r_t} - \dot{{r_t}^{ref}}\|\|)$	0.5
action rate	$-\|\|a_t - a_{t-1}\|\|$	-0.5
torque	$-\tau$	-1e-6
slippage	$-\|\|{v_t}^{foot} * {F_t}^{contact}\|\|$	-1.0
termination	1.0	-200.0

Table 3: Definition of Reward Functions

enables learning from large-scale, diverse motion sources without requiring direct human demonstration. A representative work is VideoMimic [14], which develops a real-to-sim-to-real pipeline to model both the robot and its surrounding environment. Task commands refer to high-level, sparse control signals that specify desired outcomes or goals, such as walking direction or target position, rather than detailed joint-level motions, enabling efficient whole-body control through abstraction. Representative works include Hover [16]and HugWBC [17]. Hover unifies multiple control modes into a single policy, enabling seamless transitions while retaining the strengths of each mode, thus providing a robust and scalable humanoid control solution. HugWBC designs a general task and behavior command space and employs techniques such as symmetrical loss and intervention training. This enables real-world humanoid robots to perform a variety of natural gaits—including walking, jumping, and hopping.

5 Conclusion

In this work, we present UniTracker, a unified and scalable framework for whole-body motion tracking in humanoid robots. Built upon a three-stage training pipeline, our approach begins with a privileged teacher policy that enables high-fidelity motion tracking and effective data curation. We then introduce a CVAE-based student policy that achieves robust deployment under partial observations by modeling motion diversity and implicitly incorporating global context. To further extend the system's adaptability, we propose a lightweight residual decoder for fast adaptation to highly dynamic or out-of-distribution motions. We validate UniTracker extensively in both simulation and real-world settings using a 29-DoF Unitree G1 humanoid. Our method successfully tracks over 8,100 motion sequences with a single policy, outperforming strong teacher-student baselines in terms of accuracy, generalization, and robustness. The results demonstrate the effectiveness of combining generative modeling with hierarchical policy distillation and residual adaptation for expressive and general-purpose humanoid control.

References

[1] Y. Ze, Z. Chen, J. P. Araújo, Z. ang Cao, X. B. Peng, J. Wu, and C. K. Liu. Twist: Teleoperated whole-body imitation system, 2025. URL https://arxiv.org/abs/2505.02833.

[2] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang. Mobile-television: Predictive motion priors for humanoid whole-body control, 2025. URL https://arxiv.org/abs/2412.07773.

[3] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans, 2024. URL https://arxiv.org/abs/2406.10454.

[4] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi. Learning human-to-humanoid real-time whole-body teleoperation, 2024. URL https://arxiv.org/abs/2403.04436.

[5] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon, R. Hoque, L. Paulsen, G. Yang, J. Zhang, S. Yi, G. Shi, and X. Wang. Humanoid policy human policy, 2025. URL https://arxiv.org/abs/2503.13441.

[6] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning, 2024. URL https://arxiv.org/abs/2406.08858.

[7] Y. Shao, X. Huang, B. Zhang, Q. Liao, Y. Gao, Y. Chi, Z. Li, S. Shao, and K. Sreenath. Langwbc: Language-directed humanoid whole-body control via end-to-end learning, 2025. URL https://arxiv.org/abs/2504.21738.

[8] J. Shi, X. Liu, D. Wang, O. Lu, S. Schwertfeger, F. Sun, C. Bai, and X. Li. Adversarial locomotion and motion imitation for humanoid policy learning, 2025. URL https://arxiv.org/abs/2504.14305.

[9] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbab, C. Pan, Z. Yi, G. Qu, K. Kitani, J. Hodgins, L. J. Fan, Y. Zhu, C. Liu, and G. Shi. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills, 2025. URL https://arxiv.org/abs/2502.01143.

[10] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang. Exbody2: Advanced expressive humanoid whole-body control, 2025. URL https://arxiv.org/abs/2412.13196.

[11] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang. Expressive whole-body control for humanoid robots, 2024. URL https://arxiv.org/abs/2402.16796.

[12] Z. Chen, M. Ji, X. Cheng, X. Peng, X. B. Peng, and X. Wang. Gmt: General motion tracking for humanoid whole-body control. *arXiv:2506.14770*, 2025.

[13] W. Xie, J. Han, J. Zheng, H. Li, X. Liu, J. Shi, W. Zhang, C. Bai, and X. Li. Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills, 2025. URL https://arxiv.org/abs/2506.12851.

[14] J. Mao, S. Zhao, S. Song, T. Shi, J. Ye, M. Zhang, H. Geng, J. Malik, V. Guizilini, and Y. Wang. Learning from massive human videos for universal humanoid pose control, 2024. URL https://arxiv.org/abs/2412.14172.

[15] A. Allshire, H. Choi, J. Zhang, D. McAllister, A. Zhang, C. M. Kim, T. Darrell, P. Abbeel, J. Malik, and A. Kanazawa. Visual imitation enables contextual humanoid control, 2025. URL https://arxiv.org/abs/2505.03729.

[16] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, L. Fan, and Y. Zhu. Hover: Versatile neural whole-body controller for humanoid robots, 2025. URL https://arxiv.org/abs/2410.21229.

[17] Y. Xue, W. Dong, M. Liu, W. Zhang, and J. Pang. A unified and general humanoid whole-body controller for versatile locomotion, 2025. URL https://arxiv.org/abs/2502.03206.

[18] S. Ross, G. J. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. URL https://arxiv.org/abs/1011.0686.

[19] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[20] Unitree. Unitree g1 humanoid agent ai avatar, 2024. URL https://www.unitree.com/g1.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[22] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes, 2019. URL https://arxiv.org/abs/1904.03278.

[23] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[24] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne. Deepmimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 37 (4):1–14, July 2018. ISSN 1557-7368. doi:10.1145/3197517.3201311. URL http://dx.doi.org/10.1145/3197517.3201311.

[25] Z. Luo, J. Cao, A. Winkler, K. Kitani, and W. Xu. Perpetual humanoid control for real-time simulated avatars, 2023. URL https://arxiv.org/abs/2305.06456.

[26] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021. URL https://arxiv.org/abs/2108.10470.

[27] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world, 2017. URL https://arxiv.org/abs/1703.06907.

[28] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano. Human motion diffusion model, 2022. URL https://arxiv.org/abs/2209.14916.

[29] E. Todorov, T. Erez, and Y. 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. doi:10.1109/IROS.2012.6386109.

[30] Z. Shen, H. Pi, Y. Xia, Z. Cen, S. Peng, Z. Hu, H. Bao, R. Hu, and X. Zhou. World-grounded human motion recovery via gravity-view coordinates. In *SIGGRAPH Asia Conference Proceedings*, 2024.

[31] L. Sentis and O. Khatib. A whole-body control framework for humanoids operating in human environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2641–2648, 2006. doi:10.1109/ROBOT.2006.1642100.

[32] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim. The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors, 2021. URL https://arxiv.org/abs/2104.09025.

[33] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots*, 40(3):429–455, Mar. 2016. ISSN 0929-5593. doi:10.1007/s10514-015-9479-3.

[34] J. Li and Q. Nguyen. Dynamic walking of bipedal robots on uneven stepping stones via adaptive-frequency mpc, 2022. URL https://arxiv.org/abs/2209.08664.

[35] E. Westervelt, J. Grizzle, and D. Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control*, 48(1):42–56, 2003. doi:10.1109/TAC.2002.806653.

[36] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura. Whole body humanoid control from human motion descriptors. In *2008 IEEE International Conference on Robotics and Automation*, pages 2677–2684, 2008. doi:10.1109/ROBOT.2008.4543616.

[37] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 1, pages 239–246 vol.1, 2001. doi:10.1109/IROS.2001.973365.