Humanoid Diffusion Controller

Yunshen Wang[1,3*], **Shaohang Zhu**[1,4*], **Jingze Zhang**[1,5], **Jiaxin Li**[1,6], **Yixuan Li**[1,6], **Tengyu Liu**[1,2], **Siyuan Huang**[1,2]
[1] State Key Laboratory of General Artificial Intelligence, BIGAI
[2] Joint Laboratory of Embodied AI and Humanoid Robots, BIGAI & UniTree Robotics
[3] Beijing University of Posts and Telecommunications [4] Xi'an Jiaotong University
[5] Tsinghua University [6] Beijing Institute of Technology
[*] Equal contribution
https://humanoid-diffusion-controller.github.io

Abstract: We introduce the Humanoid Diffusion Controller (**HDC**), the first diffusion-based generative controller for real-time whole-body control of humanoid robots. Unlike conventional online reinforcement learning (RL) approaches, **HDC** learns from large-scale offline data and leverages a Diffusion Transformer to generate temporally coherent action sequences. This design provides high expressiveness, scalability, and temporal smoothness. To support training at scale, we propose an effective data collection pipeline and training recipe that avoids costly online rollouts while enabling robust deployment in both simulated and real-world environments. Extensive experiments demonstrate that **HDC** outperforms state-of-the-art online RL methods in motion tracking accuracy, behavioral quality, and generalization to unseen motions. These findings underscore the feasibility and potential of large-scale generative modeling as a scalable and effective paradigm for generalizable and high-quality humanoid robot control.

Keywords: Humanoid Robots, Diffusion Models, Whole-body Control

1 Introduction

Humanoid whole-body control plays a crucial role in enabling robots to perform complex, coordinated motions across a wide range of applications, including assistive robotics, industrial automation, and entertainment. Achieving robust, dynamic, and versatile control of humanoid systems is a fundamental step toward deploying robots in real-world, unstructured environments. Existing model-based controllers [1–3] achieve strong performance but rely on simplified dynamics models due to the high computational cost of full-body modeling, limiting their scalability to diverse motions and generalizability to unseen environments.

Recent progress in humanoid whole-body control has been primarily driven by online reinforcement learning (RL) [4–16], enabling the development of policies capable of handling complex, high-dimensional whole-body tasks. However, these approaches face several critical limitations: 1) Their reliance on online rollouts limits scalability to larger models and datasets, hindering improvements in generalization and the development of more versatile control policies. 2) These methods [11–15] typically model the action at each timestep as a Gaussian distribution conditioned on the current observation and goal state, which encourages the policy to predict the most likely action but limits its ability to capture richer multimodal action distributions inherent in humanoid whole-body behaviors. 3) The use of single-step action predictions at every timestep often leads to jittery motions and the accumulation of errors over long horizons [17, 18].

To tackle these limitations, we propose the *Humanoid Diffusion Controller* (**HDC**), the first diffusion-based generative model for real-time whole-body control of humanoid robots. **HDC** draws inspiration from recent advances in large-scale generative modeling, which have demonstrated strong performance, generalization, and adaptability across various domains [19–24]. By framing

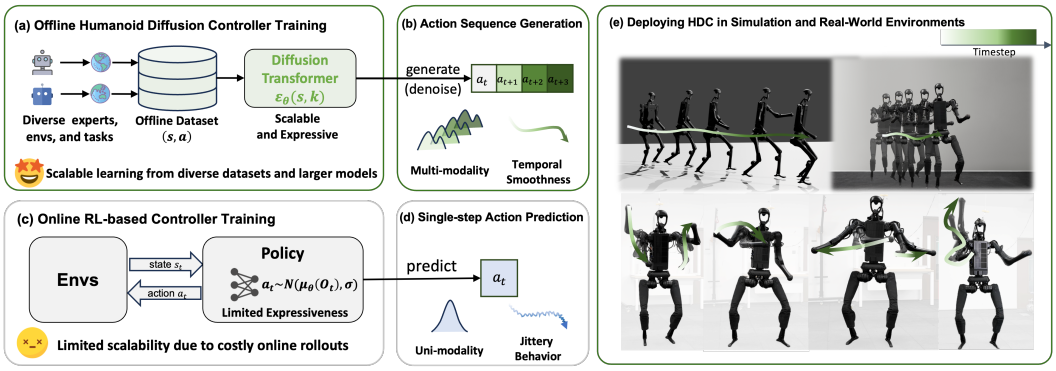

Figure 1: **Overview of HDC**. (a) **HDC** performs large-scale generative learning using expressive models. (b) **HDC** frames whole-body control as action sequence generation, improving multi-modality and temporal smoothness. (c) Online RL-based controllers require interaction with the environment during training, limiting scalability. (d) Prior controllers predict single-step, unimodal actions, resulting in limited expressiveness and jitter. (e) Deployment results of **HDC** in both simulation and the real world.

whole-body control as an action sequence generation task, **HDC** leverages a diffusion-based architecture to learn from large-scale offline data, enabling three key properties of learned humanoid controller:

- Expressiveness: **HDC** excels at capturing the multimodal nature of human whole-body actions by leveraging diffusion models' capacity to represent complex, high-dimensional distributions learned from large-scale datasets.
- Scalability: **HDC** uses Diffusion Transformers (DiTs)[25] as a scalable backbone, enabling scalable performance improvements by learning from vast amounts of data.
- Temporal Smoothness: Unlike methods that predict single-step action[11–16], **HDC** generates action sequences by sampling from a high-dimensional space, resulting in smooth and coherent motions over time and effectively mitigating jittery behaviors and compounding errors.

To unleash the full potential of **HDC** 's expressiveness and scalability, we train it using large-scale offline data, without requiring online simulation during training. However, applying offline learning to the control of humanoid robots—characterized by complex dynamics, high degrees of freedom, and stringent physical feasibility requirements—remains a significant challenge. To overcome this, we propose an effective data collection pipeline and training recipe that allows **HDC** to learn directly from vast offline datasets, achieving precise and robust whole-body control. In contrast to approaches that require continuous interaction with simulation during training [11, 13–15], our offline paradigm facilitates superior scalability in terms of both dataset size and model complexity.

Notably, to meet the real-time demands of physical robot control, our diffusion noise scheduler is designed with a linearly increasing noise level during both training and inference [26, 27]. This enables **HDC** to generate partially denoised action trajectories and execute immediate actions, allowing for faster sampling and control.

In summary, our contributions are as follows:

- We introduce **HDC**, a Diffusion Transformer-based **real-time** humanoid whole-body controller with **expressiveness, scalability, and temporally smoothness**. Its powerful model capacity and advanced architecture enable robust, precise, and seamless deployment in both simulation and real-world environments.
- We propose an effective data collection pipeline and training strategy that allows **HDC** to learn humanoid whole-body control policies entirely from offline data, eliminating the need for online simulation during training. This offline paradigm further facilitates **scalable learning from diverse datasets using larger models**.
- Extensive experiments across both simulated and real-world environments demonstrate that **HDC** consistently outperforms state-of-the-art online RL-based controllers in whole-body motion track-

ing. These results underscore the effectiveness of expressive generative models trained offline for achieving robust, generalizable, and high-quality humanoid control.
- This work shows that learning humanoid whole-body control from large-scale offline data using expressive generative models is not only feasible but also highly effective, paving the way for future scaling of model and data sizes. Furthermore, it positions **HDC** as a promising foundation model for general-purpose humanoid control.

2 Related Work

2.1 Whole-Body Control for Humanoids via Reinforcement Learning

Whole-body control of humanoid robots remains a longstanding challenge. Recent advances in sim-to-real reinforcement learning have enabled effective transfer of control policies from simulation to real-world robots [4–16, 28, 29]. RL-based methods have significantly improved humanoid locomotion by enabling stable and robust movement execution [4–10, 12, 28, 30]. Building on this, recent work has extended RL to whole-body control using point-based targets and motion commands, allowing for diverse loco-manipulation behaviors [11, 13, 15, 29]. However, these approaches rely on continual interaction with simulation [31, 32], which limits the scalability of both models and datasets, constraining the full potential of deep networks.

2.2 Diffusion Models for Robotic Planning and Control

Due to their fascinating properties, diffusion models [33, 34] have gained increasing traction in robotics, with applications in high-level trajectory planning, visuomotor control, and sequential decision making [18, 35, 36]. However, existing work [18, 37, 38] has primarily focused on tasks with low-dimensional action spaces, low replanning frequency, and inherently stable dynamics. The effectiveness of diffusion models for low-level, high-frequency control tasks remains largely unexplored. While DiffuseLoco[39] takes a step in this direction by applying diffusion models to multi-skill learning on quadruped robots, the range of skills demonstrated is still limited, falling short of fully showcasing the expressive capacity of diffusion-based policies.

2.3 Large-Scale Learning for Humanoid Robotics

The trend of scaling up model capacity, data volume, and computational resources has fueled breakthroughs across various domains of deep learning, giving rise to foundation models with remarkable generalization and multi-task abilities[19, 22, 23, 25, 40]. In humanoid robotics, GR00T N1[41] and Humanoid-X [42] have advanced scalable learning by focusing on upper-body control tasks through vision-language-action modeling and large-scale human motion datasets. MaskedMimic[43] explores motion generation for virtual characters by formulating character control as a motion inpainting problem, though it remains confined to simulation without real-world transfer. However, the application of large-scale generative modeling to humanoid whole-body control remains largely unexplored. Our proposed **HDC** addresses this gap by enabling scalable, high-quality whole-body control using expressive diffusion models trained purely from large-scale offline data.

3 Humanoid Diffusion Controller

In this section, we provide an overview of **HDC**, a model designed to generate action sequences in real time and to learn whole-body control policies from offline datasets at scale.

3.1 Humanoid Whole-body Control

In humanoid whole-body control, actions are typically generated based on proprioceptive observations and a target goal state. We define the proprioceptive state at time step t as: $\mathbf{s}_t \triangleq [\mathbf{q}_t,\ \dot{\mathbf{q}}_t,\ \boldsymbol{\omega}_t,\ \mathbf{g}_t]$, where $\mathbf{q}_t$ and $\dot{\mathbf{q}}_t$ denote the joint positions and velocities (degrees of freedom), $\boldsymbol{\omega}_t$ is the root angular velocity, and $\mathbf{g}_t$ is the gravity vector expressed in the root frame.

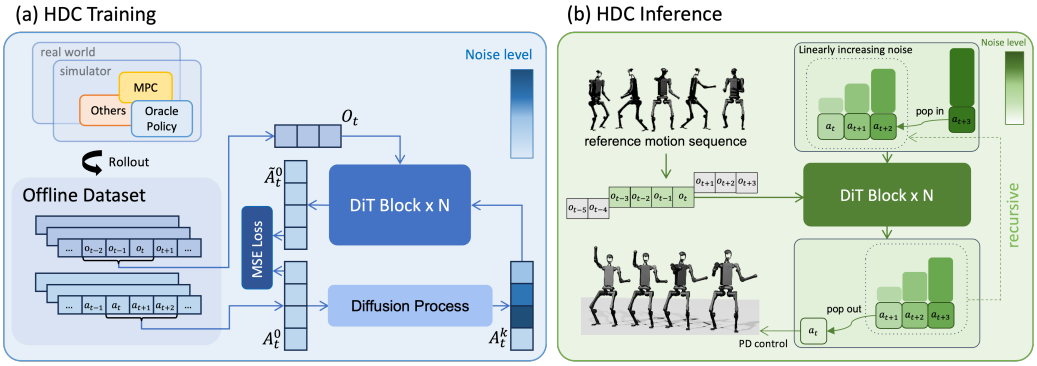

Figure 2: **HDC** training and inference. (a) During training, future action sequence $\mathbf{A}_t$ and past observation sequence $\mathbf{O}_t$ are sampled from offline data. Each action in $\mathbf{A}_t$ is perturbed with a different noise level, and the DiT model, conditioned on $\mathbf{O}_t$, learns to denoise the entire action sequence.(b) During inference, each element in the action buffer is initialized with linearly increasing noise. The denoiser iteratively denoise the buffer until the first action becomes clean, which is then executed and removed from the buffer. A new purely noisy action is appended to the end, and the process repeats recursively.

The control objective is to drive the robot to follow a desired kinematic trajectory specified by the goal state: $\mathbf{g}_t \triangleq \left[\hat{\mathbf{p}}_t^{\text{kp}},\ \dot{\hat{\mathbf{p}}}_t^{\text{kp}},\ \hat{\mathbf{p}}_t^{\text{kp}} - \mathbf{p}_t^{\text{kp}}\right]$, where $\hat{\mathbf{p}}_t^{\text{kp}}$ and $\dot{\hat{\mathbf{p}}}_t^{\text{kp}}$ are the desired positions and linear velocities of selected body keypoints, and $\hat{\mathbf{p}}_t^{\text{kp}} - \mathbf{p}_t^{\text{kp}}$ represents the positional error with respect to the current keypoint positions $\mathbf{p}_t^{\text{kp}}$. This formulation is consistent with prior works [11, 13].

Given the observation $\mathbf{o}_t = [\mathbf{s}_t, \mathbf{g}_t]$, which concatenates the current proprioceptive state $\mathbf{s}_t$ with the goal state $\mathbf{g}_t$, the high-level controller outputs an action $\mathbf{a}_t$ representing the desired joint positions. These targets are then tracked by a low-level PD controller that actuates the robot's joints.

Traditional RL-based Control. Previous online reinforcement learning approaches [11–15] formulate the learning problem as goal-conditioned RL for a Markov Decision Process (MDP). They train the control policy in physical simulator based on the tracking reward and learn the action as Gaussian distribution $\mathbf{a}_t = \pi(\mathbf{s}_t, \mathbf{g}_t) \sim \mathcal{N}(\mu, \sigma)$. As shown in Fig. 1, these models rely on online rollouts and can only make single-step action prediction, limiting their capability to learn diverse modalities and make time-consistent predictions.

3.2 Humanoid Whole-Body Control with Generative Models

HDC proposes to learn the humanoid control with expressive generative models, as shown in Fig. 1. Specifically, it formulates **whole-body control as a conditional action sequence generation task**. Therefore, we train a diffusion-based generative model to learn the distribution of action sequences $\mathbf{A}_t = [\mathbf{a}_t, \ldots, \mathbf{a}_{t+T_a-1}]$ of length T_a into the future, conditioned on the past T_o timesteps of the goal state $\mathbf{G}_t = [\mathbf{g}_{t-T_o+1}, \ldots, \mathbf{g}_t]$ and proprioception $\mathbf{S}_t = [\mathbf{s}_{t-T_o+1}, \ldots, \mathbf{s}_t]$. The diffusion process then iteratively denoises the action sequence using Stochastic Langevin Dynamics[44]:

$$\mathbf{A}_t^{k-1} \leftarrow \alpha_k(\mathbf{A}_t^k - \gamma_k \epsilon_\theta(\mathbf{A}_t^k; \mathbf{G}_t, \mathbf{S}_t, k) + \mathcal{N}(0, \sigma_k^2)), \tag{1}$$

where $\mathbf{A}_t^k$ denotes the action sequence at the k^{th} iteration, and $\boldsymbol{\epsilon}_\theta(\mathbf{A}_t^k, \mathbf{G}_t, \mathbf{S}_t, k)$ represents the predicted noise by the denoising model ϵ_θ, parameterized by θ, conditioned on the current noisy action sequence $\mathbf{A}_t^k$, the goal state $\mathbf{G}_t$, the proprioceptive state $\mathbf{S}_t$, and the denoising timestep k. The term $\mathcal{N}(0, \sigma_k^2)$ denotes Gaussian noise sampled from the DDPM scheduler. The scheduler is governed by three hyperparameters: α_k, which controls the rate of noise injection at each step; γ_k, which regulates the denoising strength; and σ_k, which determines the sampling stochasticity. Each action sequence begins at diffusion step $k = N$ (pure noise) and is gradually refined through successive denoising iterations until $k = 0$, which corresponds to a clean, executable action.

3.2.1 Partially Denoised Action Sequence Generation for Real-time Control

To generate the action sequence for humanoid whole-body control, we begin by sampling an initial noisy action sequence, $\mathbf{A}_t^N$ from Gaussian noise, and the DDPM [33] is then conditioned on goal state $\mathbf{G}_t$ and proprioception $\mathbf{S}_t$, and undergoes K iterations of denoising steps using Eq. (1). However, the K-iteration sampling process of vanilla DDPM, especially when using large models like transformers, struggles to meet the real-time control frequency demands. We also find that using accelerated sampling methods like DDIM [45] leads to degrade accuracy during sampling, which underperform in humanoid whole-body control, as shown in Section 4.3.

Inspired by Diffusion Forcing [27] and Streaming Diffusion Policy [26], we adopt a **Partially Denoised Action Sequence Generation** approach. In each denoising iteration, **HDC** produces a partially denoised action sequence with varying levels of noise corruption. The immediate action to be executed is noise-free, while subsequent actions progressively incorporate more noise and uncertainty. Specifically, In **HDC**, the action $\mathbf{A}_{t,i}$ at action timestep $i (0 \leq i \leq T_a - 1)$ is denoised at a separate noise level k_i from $\mathbf{k} = [k_0, k_1, \ldots, k_{T_a-1}]$:

$$\mathbf{A}_t^{\mathbf{k}-1} \leftarrow \alpha_{\mathbf{k}}(\mathbf{A}_t^{\mathbf{k}} - \gamma_{\mathbf{k}}\epsilon_\theta(\mathbf{A}_t^{\mathbf{k}}; \mathbf{G}_t, \mathbf{S}_t, \mathbf{k}) + \mathcal{N}(0, \sigma_{\mathbf{k}}^2)), \tag{2}$$

with $\alpha_{\mathbf{k}} = [\alpha_{k_0}, \alpha_{k_1}, \ldots, \alpha_{k_{T_a-1}}]$. The coefficients $\gamma_{\mathbf{k}}, \sigma_{\mathbf{k}}$ are constructed similarly [26]. Each action starts at a diffusion level of $k_i = N$ corresponding to pure noise, with $k_i = 0$ corresponding to a clean action. In our implementation, the action buffer stores a sequence of actions, each associated with increasing noise levels:

$$\mathbf{k} = \left[\frac{N}{T_a}, \frac{2N}{T_a}, \ldots, N\right], \tag{3}$$

The future actions $\mathbf{A}_{t,i}$ for $i > 1$ in the action buffer are used to generate actions in future observations. This design allows recursive sampling: after each control step, we perform a partial denoising update on the buffer using Eq. (2), denoising each action by $\frac{N}{T_a}$. The first action in the buffer (now at $k_0 = 0$) becomes noise-free and is executed. It is then removed from the buffer, and a new action—initialized with pure noise—is appended to the end (illustrated in Fig. 2).

This recursive sampling mechanism **achieves a T_a-fold speedup** by avoiding full trajectory sampling at each step. At the same time, it naturally models the increasing uncertainty of future actions. This enables **HDC** to run at real-time control frequencies ranging from 80 to 200 Hz, supporting stable and responsive humanoid control.

3.2.2 Training HDC

Building upon the principles of **Partially Denoised Action Sequence Generation**, we have the flexibility to assign a unique variance to each element in

$$\epsilon^{\mathbf{k}} = [\epsilon_{k_0}, \ldots, \epsilon_{k_{T_a-1}}]^{\mathrm{T}}, \tag{4}$$

corresponding to each action in the action sequence. This leads to the denoising loss:

$$\mathcal{L} = \mathrm{MSE}(\epsilon^{\mathbf{k}}, \epsilon_\theta(\mathbf{A}_t + \epsilon^{\mathbf{k}}; \mathbf{G}_t, \mathbf{S}_t, \mathbf{k})), \quad \mathbf{k} \in \mathbb{Z}^{T_a}. \tag{5}$$

This formulation enables us to implement different action sequence noise schedules at inference time by setting $\mathbf{k} = [k_0, \ldots, k_{T_a-1}]^{\mathrm{T}}$ during training, as shown in Fig. 2. To enable the recursive sampling described in Section 3.2.1, we apply linearly increasing noise to the steps of the action sequence during training, consistent with Eq. (3).

3.3 Data Collection & Training Recipe

Policy Rollouts. To train **HDC** purely from offline data, we utilize an RL-based Oracle motion tracking policy [13], which leverages extensive privileged information during training, to collect state-action-goal pairs $[\mathbf{S}_t, \mathbf{A}_t, \mathbf{G}_t]$ in parallel during closed-loop rollouts of the robot's dynamics in simulation. Importantly, these state-action-goal pairs are source-agnostic and can be gathered from

Table 1: Comparison of **HDC** and baseline methods on motion tracking. Domain randomization introduces variability in simulation parameters, while observation noise perturbs sensor measurements. The full 22-keypoint configuration tracks every joint of the humanoid, whereas the 3-keypoint configuration tracks only the head and both hands.

Method	Keypoints	Sim2Real	Succ (%) ↑	$E_{g\text{-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{acc} ↓	E_{vel} ↓
Clean Environment							
OmniH2O Oracle Policy	22	✗	91.74	129.89	82.51	3.17	6.47
OmniH2O Student Policy	3	✓	90.27	155.07	90.40	3.37	7.05
HDC	3	✓	92.59	154.06	92.35	4.95	7.43
HDC-full	22	✓	**93.64**	**135.13**	**82.06**	**3.23**	**6.46**
Domain Randomization							
OmniH2O Oracle Policy	22	✗	88.67	139.50	84.28	3.50	7.01
OmniH2O Student Policy	3	✓	82.05	176.48	93.97	**3.77**	7.72
HDC	3	✓	88.88	169.93	95.01	5.88	8.39
HDC-full	22	✓	**89.83**	**149.11**	**85.26**	4.07	**7.35**
Domain Randomization + Observation Noise							
OmniH2O Oracle Policy	22	✗	88.75	138.94	84.12	3.47	6.99
OmniH2O Student Policy	3	✓	70.89	227.28	104.83	7.12	11.52
HDC	3	✓	79.10	205.53	103.78	7.44	11.37
HDC-full	22	✓	**86.79**	**170.25**	**89.39**	**5.96**	**6.48**

oracle policies operating across diverse environments. This flexibility not only enables the integration of heterogeneous expert data, including policies derived from both reinforcement learning and model-based controllers, but also offers the potential to extend data collection to the real world.

Training Recipe. To fully leverage the model capacity of **HDC**, constructing a diverse and robust offline dataset is crucial. During data collection, we inject action noise into Oracle rollouts to expose the system to a broader range of states [46], and perform rollouts under randomized dynamics to naturally capture variations across environments. Specifically, we adopt a two-stage training recipe. In the pre-training phase, we collect a large-scale dataset using stronger action noise and wider domain randomization to encourage robustness and recovery in diverse and perturbed conditions. We then apply post-training on a smaller dataset, where observation noise is injected into the state-goal pairs to enhance **HDC**'s resilience to noisy perception, which is critical for real-world deployment. Experimental results in Section 4.2 confirm that this data and training strategy enables **HDC** to learn precise, high-quality, and robust whole-body control purely from offline data.

4 Experiment Results

In this section, we present extensive experimental results in both simulated and real-world environments to address the following questions:

- **Q1**: Can **HDC** learn highly dynamic, self-stabilizing, and precise whole-body control capabilities purely from large-scale offline data, without requiring online interaction with a simulator?
- **Q2**: Does our proposed data collection pipeline and training recipe effectively enhance **HDC**'s control performance?
- **Q3**: Does **HDC** outperform vanilla diffusion policies and discriminative MLP-based controllers in humanoid whole-body control?
- **Q4**: Can **HDC** achieve zero-shot generalization to both unseen motions and diverse environments—including the real world—when trained purely offline?

Table 2: Performance and ablation analyses of **HDC** across different aspects.

(a) Training data and recipe ablation.

Training Setup	Succ (%) ↑		$E_{g\text{-mpjpe}}$ (mm) ↓	
	Clean Env.	Noisy Obs.	Clean Env.	Noisy Obs.
Pre-training				
Clean Data	0.00	0.00	N/A	N/A
Data with Action Noise	95.43	29.46	144.38	238.14
Post-training				
Clean Data	92.32	29.25	145.85	263.23
Data with Obs. Noise	97.30	90.87	132.30	172.46

(b) Model comparison in the offline training setting. We compare **HDC** against discriminative models (MLP) and generative diffusion-based models (DDPM, DDIM).

Method	Succ ↑	Succ (DR) ↑	$E_{g\text{-mpjpe}}$ ↓	$E_{g\text{-mpjpe}}$ (DR) ↓	Freq. (Hz)
MLP Predictor	0.00	0.00	N/A	N/A	1000
DDPM	96.88	78.83	140.83	176.04	15
DDIM	96.26	73.65	142.15	171.85	100
HDC	**98.54**	**95.31**	**128.24**	**137.66**	100

(c) Comparison of generalizability to unseen motions.

Tested Motion Set	Succ (%) ↑		In Dist.
	Clean Env.	DR.	
OmniH2O (Trained on $\mathcal{M}_{\text{all}}$)			
$\mathcal{M}_{\text{all}}$	90.27	82.05	✓
$\mathcal{M}_{\text{val}}$	82.01	55.29	✓
HDC (Trained on $\mathcal{M}_{\text{train}}$)			
$\mathcal{M}_{\text{all}}$	92.59	88.88	✓
$\mathcal{M}_{\text{val}}$	84.16	66.23	✗

(d) Generalizability of **HDC** to unseen long-duration motion sequences in various environments (including sim-to-sim and sim-to-real), reported with E_{mpjpe}.

Tested Motion	IssacGym	Genesis[50]	Real-world
Play the Drum	74.05	80.77	79.64
Wipe the Window	69.37	73.55	67.82
Play the Violin	63.84	67.10	67.67
Stand and Punch	48.51	55.01	55.95

Experiment Setup. To answer the above questions, we evaluate **HDC** on motion tracking tasks across multiple simulated environments and the real world, using motions retargeted from the AMASS dataset [47] with infeasible ones filtered out [11]. All evaluations are conducted on the Unitree-H1 humanoid, following the PHC [48]. We report the Success Rate (Succ), global MPJPE $E_{g\text{-mpjpe}}$, root-relative MPJPE E_{mpjpe} (in mm), joint acceleration error E_{acc} (mm/frame^2), and joint velocity error E_{vel} (mm/frame). Unless otherwise stated, all simulations are run in IsaacGym. As a baseline, we compare against OmniH2O [13], a recent online RL-based whole-body controller trained in IsaacGym. For fairness, we match both the state and action spaces between **HDC** and OmniH2O. Additional details on the environments, dataset construction, experimental settings, and baseline implementations are provided in the *supplementary material*.

4.1 Motion-Tracking Results

To answer **Q1**, we compare **HDC** with the representative online RL-based whole-body controller OmniH2O [13]. To further evaluate the robustness of **HDC**, we conduct evaluations under domain randomization[49] and with added observation noise. As shown in Tab. 1, **HDC** consistently outperforms the online RL-based baseline, demonstrating its capability to acquire high-dynamic control skills from offline data while maintaining robustness against dynamics uncertainty, observation noise, and compounding errors. Notably, **HDC** *surpasses the oracle policy in terms of success rate, highlighting its strong multi-skill learning ability*. We attribute this advantage to the model's high capacity and its ability to represent multi-modal action distributions, without being limited to a single behavior mode.

4.2 Training and Data Recipe for HDC

We further analyze the impact of training data and recipe design choices (Section 4.2) on the performance of **HDC**. During the pre-training phase, injecting random action noise when collecting data in the simulator significantly improves performance, as it enables broader state coverage [46] and helps **HDC** better handle compounding errors. In contrast, training solely on clean data results in severe compounding errors at test time. While the pre-trained model exhibits strong resilience to compounding errors and dynamics uncertainty, it remains vulnerable to noisy state estimation. To address this, we introduce observation noise during post-training, which, as shown in Tab. 2a, effectively enhances **HDC**'s robustness to state estimation noise. Based on the above results, we answer **Q2** and demonstrate that *an appropriate data and training recipe enhances* **HDC***'s ability to learn robust control from offline data*.

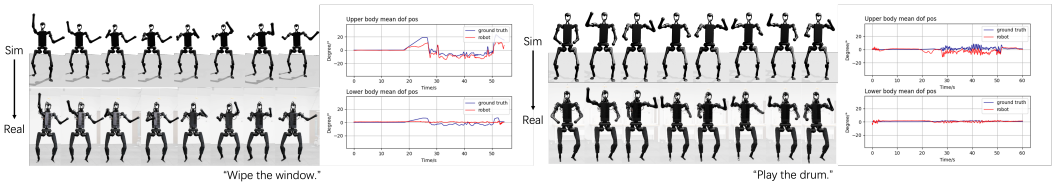

Figure 3: Visualization of **HDC**'s real-world deployment on long-horizon motions. In each block, **left**: from top to bottom are execution frames in simulation and the real world; **right**: mean joint position (DoF pos) trajectories for the upper and lower body over time.

4.3 Evaluating Model Design and Expressiveness for Whole-body Control

To address **Q3**, we investigate **HDC**'s capabilities from a model perspective under the offline learning setting. Specifically, we compare **HDC** against discriminative models implemented as MLPs that predict single-step actions based on the current observation and goal state. The results in Tab. 2b demonstrate that without access to online environment interaction, MLPs suffer from severe compounding errors, often failing to perform any motion. Additionally, we compare **HDC** with a standard diffusion model (DDPM [33]) and its accelerated version using DDIM [45], where the inference frequency is matched to that of **HDC**. Results in Tab. 2b demonstrates that **HDC** achieves better performance and we attribute this to two factors: 1) our partially noisy action generation allows the model to better capture the uncertainty of future actions during training; 2) the deterministic sampling in DDIM introduces a training-inference mismatch, which can degrade accuracy. In contrast, **HDC***'s inference strategy is designed to mitigate such discrepancy, thereby preserving both precision and robustness during deployment.*

4.4 Generalizability to Unseen Motions and Environments

To evaluate the motion-level generalization of **HDC**, we split the original OmniH2O motion dataset $\mathcal{M}_{\text{all}}$ into a training subset $\mathcal{M}_{\text{train}}$ and a held-out validation subset $\mathcal{M}_{\text{val}}$. We train **HDC** only using data collected based on motions from $\mathcal{M}_{\text{train}}$, and test it on unseen motions from $\mathcal{M}_{\text{val}}$. As shown in Tab. 2c, **HDC generalizes well to out-of-distribution motions, achieving higher success rates than OmniH2O even on motion sequences that were included in OmniH2O's training set**. We further deploy **HDC** in new simulation and real-world environments. To test its robustness under long-horizon control, we select several unseen motion sequences exceeding 50 seconds in length, including *Play the Drum*, *Wipe the Window*, *Play the Violin*, and *Stand and Punch*. We report the E_{mpjpe} across different environments in Tab. 2d, and visualize both the execution trajectories and mean joint positions for selected sequences in the real world, as shown in Fig. 3. The results demonstrate that **HDC** maintains high-quality control and generalizes across diverse environments. These results respond to **Q4** and suggest that **HDC** holds strong potential to serve as a general-purpose whole-body humanoid robot controller.

5 Conclusion

We have presented **HDC**, the first diffusion-based generative controller for real-time humanoid whole-body control. Trained on large-scale offline datasets, **HDC** leverages a scalable Diffusion Transformer architecture to model complex, multimodal action distributions and generate smooth, coherent motions over long horizons. Our experimental results demonstrate that expressive generative controller offer a scalable and effective paradigm for general-purpose humanoid control.

6 Limitations

While **HDC** demonstrates strong performance, several limitations remain. First, our current implementation primarily relies on simulator-collected data, without incorporating other valuable sources such as real-world trajectories or model-based rollouts. This limits our ability to fully exploit the generalization and expressiveness potential of large-scale generative models. Second, in real-world deployments, **HDC** can be sensitive to large sensor noise, especially severe drift in odometry-based localization. Unlike RL-based controllers trained with termination-aware penalties—which tend to act conservatively under high uncertainty—**HDC**, trained purely from expert demonstrations, may attempt overly precise control and fail under significant observation errors. These observations highlight the importance of improving robustness through more diverse and noise-aware offline datasets as a key direction for future work.

Acknowledgments

We thank Unitree Robotics for their help with the H1 robots. We thank Le Ma and Peiyuan Zhi for their assistance with the hardware codebase and setup. We also thank Yingshi Liang for her support with video production.

References

[1] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. High speed whole body dynamic motion experiment with real time master-slave humanoid robot system. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

[2] J. Ramos and S. Kim. Humanoid dynamic synchronization through whole-body bilateral feedback teleoperation. *Transactions on Robotics (T-RO)*, 34(4):953–965, 2018.

[3] F.-J. Montecillo-Puente, M. Narsipura Sreenivasa, and J.-P. Laumond. On real-time whole-body human to humanoid motion transfer. 2010.

[4] J. Siekmann, Y. Godse, A. Fern, and J. Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[5] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[6] H. Duan, B. Pandit, M. S. Gadde, B. J. van Marum, J. Dao, C. Kim, and A. Fern. Learning vision-based bipedal locomotion for challenging terrain. *arXiv preprint arXiv:2309.14594*, 2023.

[7] J. Dao, K. Green, H. Duan, A. Fern, and J. Hurst. Sim-to-real learning for bipedal locomotion under unsensed dynamic loads. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[8] I. Radosavovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik. Humanoid locomotion as next token prediction. *arXiv preprint arXiv:2402.19469*, 2024.

[9] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

[10] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *arXiv preprint arXiv:2401.16889*, 2024.

[11] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi. Learning human-to-humanoid real-time whole-body teleoperation. *arXiv preprint arXiv:2403.04436*, 2024.

[12] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

[13] 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. *arXiv preprint arXiv:2406.08858*, 2024.

[14] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

[15] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. In *Conference on Robot Learning (CoRL)*, 2024.

[16] Y. Xue, W. Dong, M. Liu, W. Zhang, and J. Pang. A unified and general humanoid whole-body controller for fine-grained locomotion. *arXiv preprint arXiv:2502.03206*, 2025.

[17] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[18] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *International Journal of Robotics Research (IJRR)*, page 02783649241273668, 2023.

[19] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[20] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[21] A. Sridhar, D. Shah, C. Glossop, and S. Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[22] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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

[24] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[25] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision (ICCV)*, 2023.

[26] S. H. Høeg, Y. Du, and O. Egeland. Streaming diffusion policy: Fast policy synthesis with variable noise diffusion models. *arXiv preprint arXiv:2406.04806*, 2024.

[27] B. Chen, D. Martí Monsó, Y. Du, M. Simchowitz, R. Tedrake, and V. Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

[28] Z. Zhuang, S. Yao, and H. Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024.

[29] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, C. Liu, G. Shi, X. Wang, L. Fan, and Y. Zhu. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024.

[30] I. Radosavovic, S. Kamat, T. Darrell, and J. Malik. Learning humanoid locomotion over challenging terrain. *arXiv preprint arXiv:2410.03654*, 2024.

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

[32] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011.

[33] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020.

[34] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. pmlr, 2015.

[35] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[36] S. Huang, Z. Wang, P. Li, B. Jia, T. Liu, Y. Zhu, W. Liang, and S.-C. Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[37] Z. Xian and N. Gkanatsios. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *Conference on Robot Learning (CoRL)*. Proceedings of Machine Learning Research, 2023.

[38] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[39] X. Huang, Y. Chi, R. Wang, Z. Li, X. B. Peng, S. Shao, B. Nikolic, and K. Sreenath. Diffuseloco: Real-time legged locomotion control with diffusion from offline datasets, 2024. URL https://arxiv.org/abs/2404.19264.

[40] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *International Conference on Computer Vision (ICCV)*, 2023.

[41] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[42] 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. *arXiv preprint arXiv:2412.14172*, 2024.

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

[44] G. E. Uhlenbeck and L. S. Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

[45] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[46] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on Robot Learning (CoRL)*. PMLR, 2017.

[47] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *International Conference on Computer Vision (ICCV)*, 2019.

[48] Z. Luo, J. Cao, K. Kitani, W. Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *International Conference on Computer Vision (ICCV)*, 2023.

[49] 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. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017.

[50] G. Authors. Genesis: A universal and generative physics engine for robotics and beyond, December 2024. URL https://github.com/Genesis-Embodied-AI/Genesis.

[51] W. Xu and F. Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.

Supplementary Materials

A State Space Defination

The state space composition of **HDC**(3-point) and **HDC**(full) policy is shown in Tab. 3 and Tab. 4. The motion goal is constituted by three distinct components: the positional discrepancy between the reference motion and the robot, the reference motion position and reference motion velocity.It is noteworthy that all three aforementioned components are characterized within the robot's local coordinate system.

Table 3: State composition in 3-point control setting

State term	Dimensions
DoF position	19
DoF velocity	19
Base(Torso) angular velocity	3
Base gravity	3
Motion goal	27
Total dim	71

Table 4: State composition in full-point control setting

State term	Dimensions
DoF position	19
DoF velocity	19
Base(Torso) angular velocity	3
Base gravity	3
Motion goal	207
Total dim	251

B HDC Implementation

B.1 Observation, Goal and Action Horizon

In our implementation, we use an observation and goal horizon of $T_o = 4$ and an action horizon of $T_a = 4$. This design enables a 4× acceleration during inference. HDC uses 15 denoising inference steps based on the x_0 parameterization.

B.2 HDC Training

During training, inspired by [26], we adopt two noise injection strategies: (1) Linearly Increasing Variance: The action sequence is treated as a whole, and linearly increasing noise is applied across time steps. (2) Independent Noise Levels: For each training sample, a step index $k_i \sim \mathcal{U}[1, \ldots, N]$ is drawn, and each step receives an independently sampled noise level from a uniform distribution. In practice, we apply the linear strategy to 80% of the training samples and the independent strategy to the remaining 20%. During training, we use the AdamW optimizer with a learning rate of 1.5×10^{-4}, $\beta = (0.95, 0.999)$, and a weight decay of 1.0×10^{-5}. Training is performed on two NVIDIA A100 GPUs for one day, using a batch size of 4096. The learning rate follows a cosine schedule with 1000 warm-up steps.

B.3 HDC Inference

The inference procedure of **HDC** is summarized in Algorithm 1. At the beginning of sampling from the **HDC**, the action buffer $\mathbf{A}_t$ must be initialized for each diffusion step k_i of action $i = 1, \ldots, n$.

One simple option is the **Zero** primer, where $\mathbf{A}_t$ is initialized as an all-zero tensor. Then, temporally increasing noise is added according to the diffusion step.

Algorithm 1: HDC Inference

Require: Buffer of future actions $\mathbf{A}_t$ at each timestep t, Per-action noise level $\mathbf{k}$, Denoiser $\epsilon_\theta(\mathbf{A}_t, \mathbf{O}_t, \mathbf{k})$, Observations $\mathbf{O}_t$, Diffusion Noise Levels N, Action horizon T_a

▷ Initialize buffer $\mathbf{A}_t$ and per action noise levels $\mathbf{k}$
$\mathbf{A}_t, \mathbf{k} \leftarrow$ Initialize buffer$(\mathbf{O}_t)$

▷ Execute **HDC** in environment.
while task not complete **do**
  ▷ Denoise actions
  for T_a denoising iterations **do**
    ▷ Run one denoising step
    $\mathbf{A}_t, \mathbf{k} \leftarrow \epsilon_\theta(\mathbf{A}_t, \mathbf{O}_t, \mathbf{k})$
  end for

  ▷ Execute first action in environment.
  env.execute$(\mathbf{A}_t[0])$
  ▷ Remove executed first action.
  $\mathbf{A}_t \leftarrow \mathbf{A}_t[1:], \mathbf{k} \leftarrow \mathbf{k}[1:]$

  ▷ Create new fully noised action.
  Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  ▷ Append noise action to buffer / noise levels
  $\mathbf{A}_t$.append$(\mathbf{z})$, $\mathbf{k}$.append$([N])$
  $t = t + 1$
end while

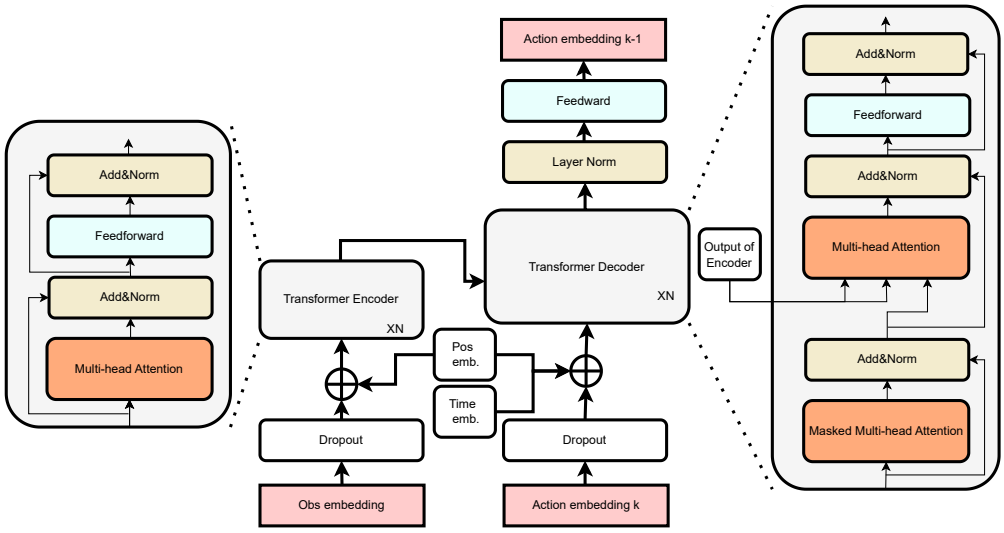

Figure 4: Architecture of the **HDC** backbone. The observation embeddings are processed by a Transformer encoder, and the resulting output is passed to the cross-attention layers of the decoder. The decoder performs one denoising step, refining the action embeddings from step k to $k-1$. Positional embeddings are added only to the observation inputs, while both positional and timestep embeddings are added to the action inputs.

B.4 Architecture

The **HDC** is based on a causal Transformer architecture designed for sequence modeling. The Transformer consists of 8 layers with 8 attention heads and an embedding dimension of 256. Dropout is applied with a rate of 0.01 on attention weights. Time-step embeddings and conditional inputs are incorporated via addition. The entire model contains approximately 8 million parameters. The detailed architecture of HDC is shown in Fig. 4.

B.5 Data Collection Details

We collect training data efficiently in parallel using an oracle policy in IsaacGym, and split the resulting trajectories into segments for training **HDC**. The pre-training dataset contains 200 million samples, while the post-training dataset contains 50 million samples. During data collection, we injected the same domain randomization and observation noise as used during Oracle policy training.

C Simulation Baselines and Ablations

C.1 Baselines

C.1.1 Oracle Policy

State composition: The Oracle policy[13] is a comprehensive control strategy that leverages privileged state information to govern the robot. The composition of the state space utilized by this policy is detailed in Tab. 5. Given the challenges associated with accurately obtaining the global velocity and global angular velocity in real word,the Oracle policy is only feasible in simulation environments.

Table 5: State composition of Oracle policy[13]

State term	Dimensions
Motion goal DoF position	66
Motion goal Dof rotation	138
Motion goal Dof velocity	69
Motion goal DoF angular velocity	69
DoF position difference	69
DoF rotation difference	138
DoF velocity difference	69
DoF angular velocity difference	69
Local DoF position	69
Local DoF rotation	138
Previous Action	19
Total dim	913

Training: The Oracle policy is trained on the PHC-filtered AMASS dataset utilizing PPO (Proximal Policy Optimization) and IsaacGym simulator. The model architecture is a standard MLP. The specific details regarding the training reward, domain randomization and noise scales are presented in Tab. 6, Tab. 7, Tab. 9

Table 6: Reward components and weights of Oracle policy: penalty rewards for preventing undesired behaviors for sim-to-real transfer, regularization to refine motion, and task reward to achieve successful whole-body tracking in real-time.[13]

Term	Expression	Weight
	Penalty	
Torque limits	$\mathbb{1}(\boldsymbol{\tau}_t \notin [\boldsymbol{\tau}_{\min}, \boldsymbol{\tau}_{\max}])$	-2
DoF position limits	$\mathbb{1}(\boldsymbol{d}_t \notin [\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max}])$	-125
DoF velocity limits	$\mathbb{1}(\dot{\boldsymbol{d}}_t \notin [\dot{\boldsymbol{q}}_{\min}, \dot{\boldsymbol{q}}_{\max}])$	-50
Termination	$\mathbb{1}_{\text{termination}}$	-250
	Regularization	
DoF acceleration	$\|\ddot{\boldsymbol{d}}_t\|_E 2$	-0.000011
DoF velocity	$\|\dot{\boldsymbol{d}}_t\|_2^2$	-0.004
Lower-body action rate	$\|\boldsymbol{a}_t^{\text{lower}} - \boldsymbol{a}_{t-1}^{\text{lower}}\|_2^2$	-3
Upper-body action rate	$\|\boldsymbol{a}_t^{\text{upper}} - \boldsymbol{a}_{t-1}^{\text{upper}}\|_2^2$	-0.625
Torque	$\|\boldsymbol{\tau}_t\|$	-0.0001
Feet air time	$T_{\text{air}} - 0.25$	1000
Max feet height for each step	$\max\{\boldsymbol{h}_{\text{max feet height for each step}} - 0.25, 0\}$	1000
Feet contact force	$\|F_{\text{feet}}\|_2^2$	-0.75
Stumble	$\mathbb{1}(F_{\text{feet}}^{xy} > 5 \times F_{\text{feet}}^{z})$	-0.00125
Slippage	$\|\boldsymbol{v}_t^{\text{feet}}\|_2^2 \times \mathbb{1}(F_{\text{feet}} \geq 1)$	-37.5
Feet orientation	$\|\boldsymbol{g}_z^{\text{feet}}\|$	-62.5
In the air	$\mathbb{1}(F_{\text{feet}}^{\text{left}}, F_{\text{feet}}^{\text{right}} < 1)$	-200
Orientation	$\|\boldsymbol{g}_z^{\text{root}}\|$	-200
	Task Reward	
DoF position	$\exp(-0.25\|\hat{\boldsymbol{d}}_{\boldsymbol{t}} - \boldsymbol{d}_t\|_2)$	32
DoF velocity	$\exp(-0.25\|\hat{\dot{\boldsymbol{d}}}_t - \dot{\boldsymbol{d}}_t\|_2^2)$	16
Body position	$\exp(-0.5\|\boldsymbol{p}_t - \hat{\boldsymbol{p}}_t\|_2^2)$	30
Body position VRpoints	$\exp(-0.5\|\boldsymbol{p}_t^{\text{real}} - \hat{\boldsymbol{p}}_t^{\text{real}}\|_2^2)$	50
Body rotation	$\exp(-0.1\|\boldsymbol{\theta}_t \ominus \hat{\boldsymbol{\theta}}_t\|)$	20
Body velocity	$\exp(-10.0\|\boldsymbol{v}_t - \hat{\boldsymbol{v}}_t\|_2)$	8
Body angular velocity	$\exp(-0.01\|\boldsymbol{\omega}_t - \hat{\boldsymbol{\omega}}_t\|_2)$	8

Table 7: Domain randomization parameters used for training the Oracle policy [13].

Parameter	**Range / Description**
	Dynamics Randomization
Friction coefficient	$\mathcal{U}(-0.6,\ 1.2)$
Base COM offset	$\mathcal{U}(-0.1,\ 0.1)\,\mathrm{m}$
Link mass scale	$\mathcal{U}(0.7,\ 1.3)\times$ relative to default
Proportional gain (P)	$\mathcal{U}(0.75,\ 1.25)\times$ default
Derivative gain (D)	$\mathcal{U}(0.75,\ 1.25)\times$ default
Torque RFI	$0.1\times$ torque limit $(\mathrm{N{\cdot}m})$
Control latency	$\mathcal{U}(0,\ 90)\,\mathrm{ms}$
Motion reference offset	$\mathcal{U}([-0.02,\ 0.02], [-0.02,\ 0.02], [-0.1,\ 0.1])\,\mathrm{cm}$
	External Perturbation
Lateral push	Every 5s, $v_{xy} = 1\,\mathrm{m/s}$
	Randomized Terrain
Terrain type	flat, rough, low obstacles

C.1.2 Student Policy

State composition: The state space of the student policy is constructed using information that is available during deployment. It adopts a 3-point control scheme and includes a 25-step history of joint positions and velocities (DoF), torso angular velocity, torso-projected gravity, and previous actions. The detailed composition of the student policy's state space is provided in Tab. 8.

Table 8: State composition of OmniH2O student policy[13]

State term	Dimensions
DoF position	19
DoF velocity	19
Base(Torso) angular velocity	3
Base gravity	3
Motion goal	27
Previous Action	19
Single step total dim	90
History state term	Dimensions
DoF position	19
DoF velocity	19
Base(Torso) angular velocity	3
Base gravity	3
Previous Action	19
History single step total dim	63
Total dim	1665 $(63 \times 25 + 90)$

Table 9: Noise scales used for training the Oracle policy [13].

Parameter	**Noise Scale**
Motion goal DoF position	0.01
Motion goal DoF rotation	0.01
Motion goal DoF velocity	0.01
Motion goal DoF angular velocity	0.01
DoF position difference	0.05
DoF rotation difference	0.01
DoF velocity difference	0.01
DoF angular velocity difference	0.01
Local DoF position	0.01
Local DoF rotation	0.01
Previous action	0.00

Training: The student policy is derived from Oracle policy through a distillation process. It utilizes the identical motion dataset and simulation environment as the Oracle policy. The architecture of the student policy is also a vanilla MLP. The distillation is implemented using the DAgger algorithm, which employs the action generated by Oracle policy to supervise the output of student policy. The loss function employed L2 norm, mathematically defined as $\|a_t^{privileged} - a_t\|_2^2$.

C.1.3 Vanilla DDPM and DDIM

In the experiments of Section 4.3, we evaluate a vanilla DDPM-based baseline implemented using our Transformer architecture. The model is trained with 25 denoising steps and adopts the ϵ-prediction parameterization. The noise schedule is with $\beta \in [1\text{e}{-4}, 0.02]$, implemented via the DDIM scheduler in the `diffusers` library, and for DDIM inference, 5 sampling steps are used.

The model is trained using a batch size of 4096, AdamW optimizer with learning rate 3×10^{-4}, $\beta = (0.95, 0.999)$, weight decay 1×10^{-6}, and cosine learning rate schedule with 1000 warm-up steps.

C.1.4 MLP Predictor

In the experiments of Section 4.3, we additionally implement a single-step MLP predictor as a baseline. This model receives the past $T_o = 4$ observations as input and predicts the next action in a autoregressive manner. The MLP consists of two hidden layers with 512 units each and a dropout rate of 0.1. The model is trained using the AdamW optimizer with a learning rate of 1.5×10^{-4}, $\beta = (0.95, 0.999)$, and a weight decay of 1.0×10^{-5}.

C.2 Domain Randomization and Observation Noise during Evaluation

The detailed domain randomization parameter and observation noise for the experiment results in Section 4 are shown in Tab. 10,Tab. 11.

Table 10: Domain randomization parameter for testing

Term	Value
Dynamics Randomizations	
Friction	$\mathcal{U}(-0.6, 1.2)$
Base Com offset	$\mathcal{U}(-0.1, 0.1)$m
Link mass	$\mathcal{U}(0.7, 1.3) \times$ default kg
P Gain	$\mathcal{U}(0.75, 1.25) \times$ default
D Gain	$\mathcal{U}(0.75, 1.25) \times$ default
Torque RFI	$0.1 \times$ torque limit N $\cdot$ m
Control delay	$\mathcal{U}(30, 90)$ms
Motion reference offset	$\mathcal{U}([-0.02, 0.02], [-0.02, 0.02], [-0.1, 0.1])$cm
External Perturbation	
Push robot	not activated
Randomized Terrain	
Terrain type	flat, rough, low obstacles

Table 11: Noise scales for testing

Term	Value
DoF position	0.01
DoF velocity	0.01
Base angular velocity	0.5
Base gravity	0.1
Motion goal difference	0.05
Motion goal reference	0.05
Motion goal velocity	0.01

C.3 Motion Tracking Experiment Setup

All comparisons in Section 4.1 are conducted on the AMASS dataset filtered using the PHC model, evaluating motion tracking performance.

C.4 Ablation Experiment Setup

The experiments in Section 4.2 and Section 4.3 are conducted by randomly sampling 500 motions from the AMASS dataset for efficient ablation analysis.

C.5 Generalization Analyses Experiment Setup

We evaluate the oracle policy on the full motion dataset $\mathcal{M}_{\text{all}}$ using motion tracking metrics, specifically computing the lower-body $E_{g\text{-mpjpe}}$ and E_{mpjpe} for each sequence. The average of these two metrics is used as a sorting criterion. We select the top 60% of motions with the lowest error to collect rollout data for training **HDC**. Notably, **HDC** demonstrates strong performance not only on these in-distribution motions, but also generalizes well to the remaining 40% of motions that it was not trained on.

D Real world Setup

To obtain observation in the real world, we need to obtain the robot proprioception and motion goal, as shown in Appendix A.

Robot proprioception: Proprioceptive signals are received via socket communication, including motor joint positions and velocities, as well as torso orientation and angular velocity obtained from the IMU mounted on the torso link of robot.

Motion goal: We use LiDAR-based SLAM [51] to obtain the global position and orientation of the robot's head. This head rotation is used to compute the directional offset between the SLAM coordinate frame and the torso-IMU frame, allowing us to transform the SLAM global position into the torso-IMU coordinate system. Given the head position, torso orientation (from the IMU), and joint positions (from DoF readings), the global position and rotation of each joint can be computed via forward kinematics and standard translation-rotation transformations. For the reference motion, we select several sequences from the AMASS dataset and prepend it with the robot's default pose. Interpolation is applied to smoothly transition from the default pose to the first frame of the motion. Reference velocities are computed as finite differences between adjacent frames.

When launching the policy, the observation is computed by the proprioception and motion goal and the policy use it to output action. This action is taken as PD target angles that are sent to robot's motors.

E Additional Results for Controlling Long-Horizon Motions in Real World

To further support the generalization capability of **HDC** across environments and long-horizon motion sequences, we provide additional results in this section. we visualize the mean joint position of humanoid robot over time for several unseen motions exceeding 20 seconds in duration. These include motions such as *Single arm punching*, *Double arm punching*, *Freestyle swimming*, and *Chopping with both arm*, *Sweeping a hoop*, *Walking fast*, and *Breaststroke swimming*. The results show that **HDC** successfully preserves stability, smoothness, and fidelity in physical conditions. These observations further confirm the robustness and general applicability of **HDC** as a whole-body controller, addressing **Q4** in the main paper.

Figure 5: Single-arm punch

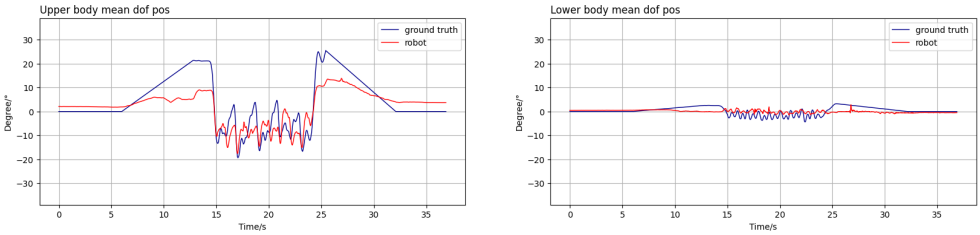

Figure 6: Double-arm punch

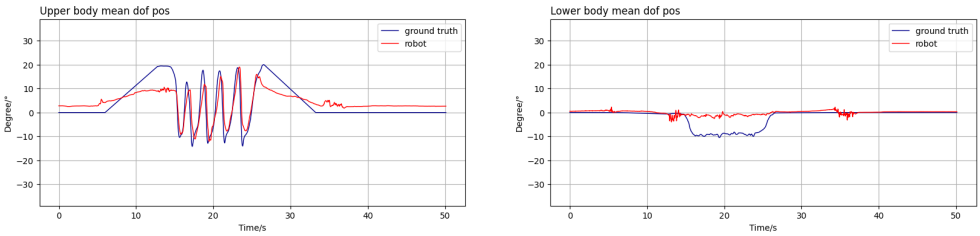

Figure 7: Freestyle swimming

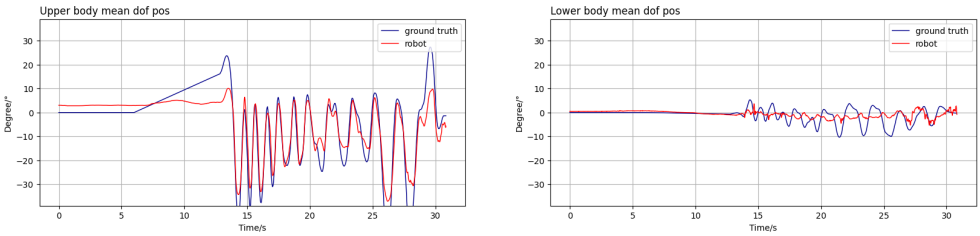

Figure 8: Chopping wood with both arms

Figure 9: Sweeping a hoop

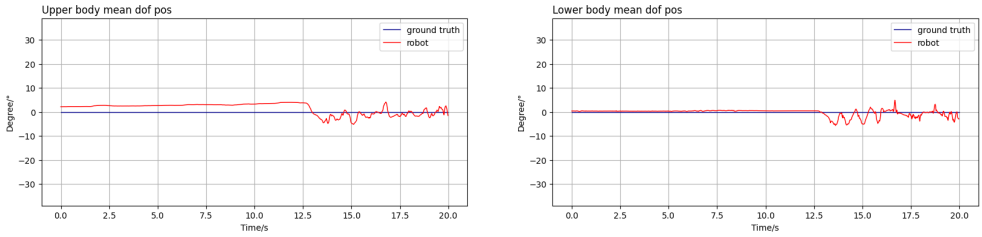

Figure 10: Fast walking

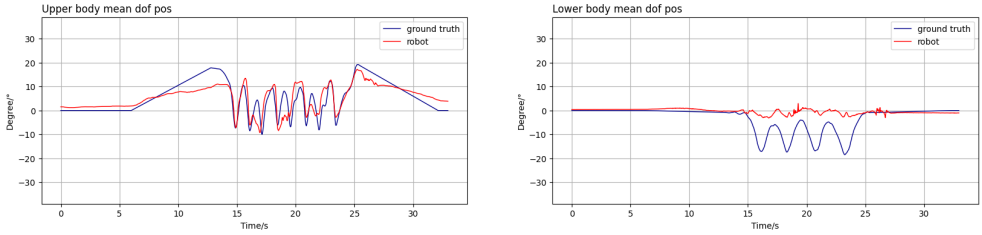

Figure 11: Breaststroke swimming