

# Track Any Motions under Any Disturbances

Zhikai Zhang<sup>\*1,3</sup> Jun Guo<sup>\*1,3</sup> Chao Chen<sup>2,3</sup> Jilong Wang<sup>2,3</sup> Chenghui Lin<sup>3</sup> Yunrui Lian<sup>1,3</sup>  
 Han Xue<sup>1,3</sup> Zhenrong Wang<sup>3</sup> Maoqi Liu<sup>3</sup> Jiangran Lyu<sup>2,3</sup> Huaping Liu<sup>1</sup> He Wang<sup>2,3</sup> Li Yi<sup>†1,4</sup>



Fig. 1: (a) The humanoid tracks *diverse, highly dynamic, and contact-rich* motions using a single policy. (b) The humanoid tracks motions on different *terrains*. (c) The humanoid tracks motions against different *external forces*, including pulling forces from the rope and pushing forces from the human feet. (d) The humanoid tracks motions against *physical property changes* (the payload on the back). (e) The humanoid tracks motions against *all dynamics disturbances*, including *terrains, external forces, and physical property changes* simultaneously.

**Abstract**—A foundational humanoid motion tracker is expected to be able to track diverse, highly dynamic, and contact-rich motions. More importantly, it needs to operate stably in real-world scenarios against various dynamics disturbances, including terrains, external forces, and physical property changes for general practical use. To achieve this goal, we propose Any2Track (Track Any motions under Any disturbances), a two-stage RL framework to track various motions under multiple disturbances in the real world. Any2Track reformulates dynamics adaptability as an additional capability on top of basic action execution and consists of two key components: AnyTracker and AnyAdapter. AnyTracker is a general motion tracker with a series of careful designs to

track various motions within a single policy. AnyAdapter is a history-informed adaptation module that endows the tracker with online dynamics adaptability to overcome the sim2real gap and multiple real-world disturbances. We deploy Any2Track on Unitree G1 hardware and achieve a successful sim2real transfer in a zero-shot manner. Any2Track performs exceptionally well in tracking various motions under multiple real-world disturbances, as shown in Figure 1. For real-world demos, please refer to <https://zzk273.github.io/Any2Track/>.

## I. INTRODUCTION

Humanoid motion tracking [1]–[6] aims to reproduce human motion sequences on a humanoid platform, enabling more expressive and anthropomorphic motion behavior compared with pure RL-based controllers [7]–[14]. A foundational humanoid motion tracker should be able to track

<sup>\*</sup>Equal Contributions, <sup>†</sup>Corresponding Author  
<sup>1</sup>Tsinghua University, <sup>2</sup>Peking University, <sup>3</sup>Galbot, <sup>4</sup>Shanghai Qi Zhi Institute  
 Paper website: <https://zzk273.github.io/Any2Track/>

diverse, highly dynamic, and contact-rich motions to capture general human motion knowledge. In addition, it needs to operate stably in real-world scenarios against various disturbances, including terrains, external forces, and physical property changes (e.g., payload, friction) for general practical use. However, existing humanoid motion trackers fail to have general motion tracking capability and dynamics adaptability simultaneously. They exhibit varying levels of limitations in motion tracking capabilities and resilience to different real-world disturbances. The comparison of existing works and our method is shown in Table I.

The key challenges to construct a fundamental humanoid motion tracker can be summarized in two points: 1) how to construct a unified general motion tracker to efficiently and with high quality learn diverse motion control strategies; 2) how to endow the controller with online dynamics adaptability to overcome the sim2real dynamics gap and various real-world disturbances.

Towards the goal of constructing a foundational humanoid motion tracker, we propose a two-stage RL framework called **Any2Track** (**Track Any** motions under **Any** disturbances) in this work. Any2Track reformulates dynamics adaptability as an additional capability on top of basic action execution and consists of two key components: AnyTracker and AnyAdapter.

In the first stage, we construct **AnyTracker**, a general motion tracker to track diverse, highly dynamic, and contact-rich motions. We found that the bottleneck in training a general motion tracker lies in the complex action spaces brought by high degrees of freedom and motion diversity. Thus, we propose a series of careful designs, including canonicalized action spaces and a specialist-to-generalist strategy, to alleviate the optimization difficulties brought by the complex action spaces. AnyTracker is trained as a base policy without any dynamics randomization to avoid tracking performance degradation.

In the second stage, we introduce dynamics variance and further propose a history-informed adaptation module on top of AnyTracker, which is called **AnyAdapter**. AnyAdapter endows the base tracker with online dynamics adaptability for different kinds of disturbances, including terrains, external forces, and physical property changes, without compromising its foundational expressive motion tracking capability. AnyAdapter utilizes dynamics-aware world model prediction as a proxy task to extract dynamics features as neural embeddings from the history buffer, which can serve as an informative representation to identify environment dynamics. Rather than directly finetuning the original parameters of the base tracker for dynamics adaptability, which may harm the tracking skills already acquired, we freeze the base tracker and introduce an adapter [15] architecture. The adapter is trained to adaptively adjust the base policy’s actions to accommodate different environments, taking dynamics embeddings as input.

We deploy Any2Track on Unitree G1 hardware and achieve a successful sim2real transfer in a zero-shot manner. Our method shows impressive results in tracking diverse,

highly dynamic, and contact-rich motions when faced with real-world disturbances from different sources, including terrains, external forces, and physical property changes. Extensive experiments are conducted both in the simulation and in the real world to validate the effectiveness of our major designs.

Our contributions are fourfold:

- **Any2Track**: A novel two-stage RL framework for robust humanoid motion tracking that excels under significant real-world disturbances.
- **AnyTracker**: A general motion tracker with a series of careful designs to alleviate the optimization difficulties brought by complex action spaces and track diverse, highly dynamic, and contact-rich motions.
- **AnyAdapter**: A history-informed module that provides online adaptation to dynamic shifts and external perturbations, including terrains, external forces, and physical property changes.
- **State-of-the-art Performance**: We demonstrate superior real-world tracking under disturbances on the Unitree G1 hardware, validated by extensive experiments in both simulation and reality.

## II. RELATED WORKS

### A. Motion Tracking

Motion tracking aims to reproduce human motions to learn anthropomorphic whole-body controllers, which has been extensively studied in character animation [19]–[26]. [19] shows progressive results in constructing a general motion tracking strategy with dynamic, versatile, and perpetual tracking capability.

Recently, robotics researchers have started to explore the reproduction of human motions on real-world humanoid hardware platforms [1]–[6], to enable more expressive motion behavior compared with pure RL-based controllers [7]–[10]. However, compared with character animation, motion tracking on humanoid robots still lags behind significantly due to the cross-embodiment gap, actuator limits, and sim-to-real challenges. HumanPlus [1] and OmniH2O [2] mainly focus on quasi-static loco-manipulation tasks. ASAP [6] achieves highly dynamic motion tracking. However, it overfits to motion clips of only a few seconds. GMT [5] constructs a general motion tracker with multiple careful designs in motion distribution sampling and network architecture. Although this method can track a considerable portion of data from LAFAN1 [16] and AMASS [17] datasets, it shows limited capability when dealing with highly dynamic and contact-rich motions. Moreover, GMT does not show adaptability to external disturbances, which limits its practicality in real-world scenarios.

Compared with existing works, Any2Track demonstrates unprecedented levels of motion diversity, high dynamism, and contact complexity within a unified tracking policy. More distinctively, our method endows the motion tracker with online dynamics adaptability to overcome the sim-to-real gap and various real-world disturbances, marking a significant step toward general real-world use.

TABLE I: **Overall comparison of different motion tracking methods.** We compare Any2Track with existing methods according to their performance in publicly available demos and their proposed methods. Any2Track demonstrates unprecedented levels of motion diversity, high dynamism, and contact complexity. More distinctively, our method adapts to multiple real-world disturbances for general practical use. It is worth noting that, although both Any2Track and GMT [5] use the combination of LAFAN1 [16] and AMASS [17] as the training dataset, GMT removed a significant portion of highly dynamic or contact-rich motions. In contrast, we retained these motions.

Method	Track Any Motions			Adapt to Any Disturbances		
	Diversity	Highly Dynamic	Contact-rich	Terrain	External Force	Physical Property Change
Exbody2 [3]	CMU dataset [18]	✗	✗	✗	✗	✗
ASAP [6]	One motion clip	✓	✗	✗	✗	✗
GMT [5]	LAFAN1 + AMASS*	✓	✗	✗	✗	✗
<b>Ours</b>	<b>LAFAN1 + AMASS</b>	✓	✓	✓	✓	✓

### B. Online Dynamics Adaptation for Legged Robots

A key challenge in constructing a fundamental humanoid motion tracker is to overcome the sim2real dynamics gap and various real-world disturbances for general practical use. Existing works [1], [3], [5] often rely on naïve domain randomization in the hope that the policy can perform robustly under different environment dynamics. However, due to the unawareness of the environment dynamics, the policy cannot adaptively adjust its actions, causing it to behave conservatively when faced with large dynamic variance.

To cope with more complex real-world environments for legged robots, online dynamics adaptation methods [9], [27]–[29] are needed. Existing works [9], [27], [28] utilize robot-environment interaction history to adaptively estimate real-world dynamics and adjust robot behavior accordingly. Rapid Motor Adaptation (RMA) [28] learns to encode recent state histories into a latent representation of environment dynamics by distilling knowledge from a privileged teacher encoder. Li *et al.* [27] designs a dual-history architecture to overcome the sim2real dynamics gap with history information over varying horizons. DWL [9] utilizes a denoising world model to predict environment configuration and privileged robot states simultaneously from history information.

Compared with existing works, we take a further step in online dynamics adaptation. To learn better dynamics representations from the history buffer, we introduce dynamics-aware world model prediction as a proxy task. The learned representations are more informative in dynamics. To decouple the acquisition of motion tracking capability and dynamics adaptability to reduce the difficulty of joint optimization, we introduce an adapter architecture and a finetuning stage on top of a base tracker to accommodate different environment dynamics.

## III. TRACK ANY MOTIONS UNDER ANY DISTURBANCE

We propose Any2Track in this work, a two-stage RL framework to build a foundational humanoid motion tracker. The design philosophy of Any2Track is to reformulate dynamics adaptability as an additional capability on top of basic action execution and decouple its learning process. Such a design can demonstrate exceptional robustness to environmental disturbances while preserving its motion expressiveness. Concretely, we first construct AnyTracker, a general

motion tracker with a series of careful designs to alleviate the optimization difficulties brought by complex action spaces and track various motions, as shown in Section III-A. At this stage, AnyTracker is trained as a base policy without any dynamics randomization to avoid tracking performance degradation. We then propose AnyAdapter, a history-informed adaptation module to overcome the sim2real dynamics gap and various real-world disturbances, as shown in Section III-B. At this stage, we introduce dynamics domain randomization and fine-tune the motion tracker for dynamics adaptability. The pipeline is shown in Figure 2. We use PPO [30] as our reinforcement learning framework and MuJoCo [31], [32] for simulation. We train our policy in parallel on 8 GPUs.

### A. AnyTracker: Track Any Motions

Our tracker can be seen as an RL policy  $\pi : \mathcal{G} \times \mathcal{S} \mapsto \mathcal{A}$ , which maps humanoid proprioception state and tracking goals to low-level robot actions. At each timestep  $t$ , the inputs of policy  $\pi$  consist of current state  $s_t$  and current tracking goals  $g_t$ .  $s_t$  includes angular velocity, projected gravity, per-joint position, per-joint velocity, and last-frame action.  $g_t$  depicts next-frame humanoid target motion, including target per-joint position, target per-joint velocity, and target rigid body information in the local frame. The policy  $\pi$  needs to output per-joint action  $a_t$ , which is further fed into a PD controller to compute actuator torques. The task goal of motion tracking is that at timestep  $t + 1$ , the robot’s state  $s_{t+1}$  should be as close as possible to the target motion depicted by  $g_t$  while maintaining balance and safety. The reward terms we used are shown in Table II.

To achieve general motion tracking, we use the combination of AMASS [17] and LAFAN1 [16] motion datasets as our training data. We filter out motions infeasible for tracking following PHC [19], primarily those involving interactions with an unavailable scene (e.g., climbing stairs). However, unlike GMT [5], we retain highly dynamic and contact-rich motions. We found that with proper reward and policy observation design, training a motion tracker on a few highly dynamic or contact-rich motion trajectories is no longer difficult. The bottleneck in training a general motion tracker lies in the complex action spaces brought by the humanoid’s high degrees of freedom and the motion diversity. For different joints and different motion categories, the

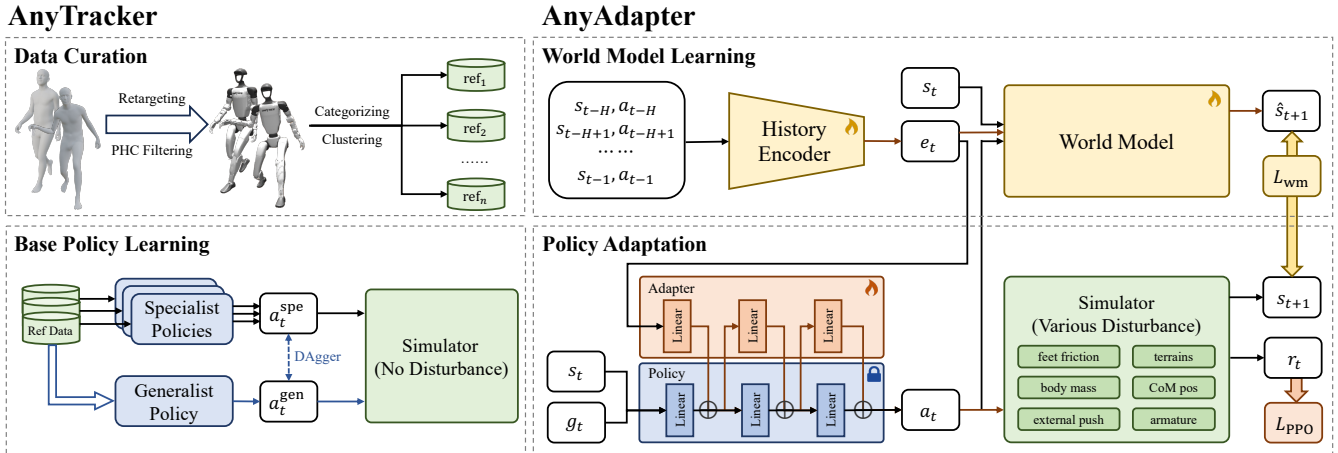


Fig. 2: **Overview of our method.** Any2Track consists of two key components: AnyTracker and AnyAdapter. **AnyTracker** is a general motion tracker with a series of careful designs. **AnyAdapter** is a history-informed adaptation module on top of AnyTracker. AnyAdapter endows the base tracker with online dynamics adaptability without compromising its fundamental expressive motion tracking capability.

TABLE II: Reward terms used in AnyTracker.

Term	Weight	Term	Weight
Task			
Upper-body position	1.0	Lower-body position	0.5
Torso roll/pitch	1.0	Body rotation	0.5
Body linear velocity	0.5	Body angular velocity	0.5
DoF position	0.75	DoF velocity	0.5
Root linear velocity	1.0	Root angular velocity	1.0
Root height	1.0	Feet height	1.0
Regularization			
Action rate	-0.5	DoF velocity rate	$-1 \times 10^{-6}$
Torques	$-2 \times 10^{-5}$		
Penalty			
DoF position limits	-10.0	DoF velocity limits	-5.0
Self collision	-10.0	Termination	-200.0

action distributions vary significantly, making it challenging to learn all distributions well in a single RL optimization process. Therefore, we propose novel designs to alleviate this difficulty in terms of degrees of freedom and motion diversity, respectively.

### 1) Canonicalized Action Spaces

The huge diversity of each joint’s action distribution makes it difficult for the policy to learn a general and high-quality control strategy. Thus, we propose to design a canonicalized motion tracking action space so that the policy only needs to predict a compact multi-joint action distribution. Concretely, we first utilize the  $\tanh$  function to map the policy prediction  $\pi(a_t|s_t)$  to the range  $[-1, 1]$ . However, the action range varies for different joints, and using a unified policy prediction range is not suitable for all joints and can severely hinder the tracker’s performance. Therefore, we use empirically designed hyperparameters  $\alpha \in \mathbb{R}^{\text{num.joints}}$  to rescale each joint’s action scale. Instead of directly predicting the PD targets, we predict residual PD offsets relative to the

reference motion. The final PD targets can be written as:

$$q_d = \tilde{q}_{t+1} + \alpha \tanh(\pi(a_t|s_t)), \quad (1)$$

where  $\tilde{q}_{t+1}$  is the target next-frame joint position.

### 2) Motion Clustering and Specialist-to-Generalist

Different categories of motions exhibit different motion patterns and action distributions, posing challenges for general motion tracker training. Therefore, we propose to cluster motions according to categories and train a specialist for each motion cluster. By doing so, each specialist only needs to handle a set of motions with similar action distributions, which greatly reduces the training difficulty and improves the final results. For LAFAN1 [16], as the motion categories are already provided, we train a specialist policy for each motion category. For AMASS [17], HumanML3D [33] provides motion categories labels (the “VERB” term in the annotation). AMASS contains a large number of motion categories, and thus training a specialist for each category would be inefficient. Therefore, we feed the motion category labels into CLIP [34] to compute text feature embeddings. Then we cluster motions based on these embeddings using the K-means [35] method. We split AMASS into 6 subsets and train corresponding specialists. At last, we adopt the specialist-to-generalist paradigm [8] and use DAgger [36] to distill all specialists into a generalist policy.

### B. AnyAdapter: Adapt to Any Disturbances

Based on the AnyTracker acquired previously, we introduce environment dynamics variance in Table III at this stage and propose AnyAdapter, a history-informed adaptation module for different kinds of disturbances.

We argue that there are two key components for online dynamics adaptability: 1) a good dynamics encoder that estimates adequate environment dynamics features from robot-environment interaction history buffer; 2) an effective training strategy that helps the control policy to properly leverage these dynamics features to adaptively adjust its behavior

TABLE III: **Dynamics variance used in Any2Track.** We add three types of randomization, including terrains, external forces, and physical property changes.

Term	Value
Terrains	
Floor friction	$\mathcal{U}(0.3, 2.0)$
Max terrain height	0.3
Noise scale	$\mathcal{U}(10.0, 16.0)$
Noise octaves	$\mathcal{U}(5.0, 8.0)$
Noise persistence	$\mathcal{U}(0.3, 0.5)$
Noise lacunarity	$\mathcal{U}(2.0, 4.0)$
External Forces	
Interval range	$\mathcal{U}(5.0, 10.0)$
Velocity magnitude range	$\mathcal{U}(0.1, 1.0)$
Physical Property Changes	
DoF friction scaling	$\mathcal{U}(0.5, 2.0)$
Armature scaling	$\mathcal{U}(1.0, 1.05)$
Torso CoM position change	$\mathcal{U}(-0.15, 0.15)$
Torso mass change	$\mathcal{U}(-3.0, 6.0)$
Default DoF position jittering	$\mathcal{U}(-0.05, 0.05)$

without sacrificing motion expressiveness. Thus, AnyAdapter utilizes dynamics-aware world model prediction as a proxy task to extract dynamics features into neural embeddings from the history buffer, which can serve as an informative representation to tell different dynamics. To facilitate the motion tracker’s learning of dynamics adaptability, AnyAdapter refrains from coupling the basic motion-tracking capability with dynamics adaptability in a single network and instead introduces an additional adapter architecture.

#### 1) Learn Informative Dynamics Embeddings

Given the history of robot–environment interactions, the motion tracker is expected to leverage this information to estimate the current environment dynamics and adapt its behavior accordingly. However, we found that the interaction history often contains substantial irrelevant information and noise, making direct utilization challenging for model learning. To address this, it is necessary to introduce proxy tasks for representation learning, enabling the extraction of informative feature embeddings that capture rich dynamics knowledge. In this work, we propose dynamics-aware world model prediction, which contains a history encoder  $\phi$  and a world model  $\omega$ . At each timestep  $t$ , the history encoder extracts dynamics feature embeddings from history information:

$$e_t = \phi(h_t), \quad (2)$$

where  $h_t = \{s_{t-H}, a_{t-H}, \dots, s_{t-1}, a_{t-1}\}$  and  $H = 79$  is the history window size. The world model needs to predict the next-frame robot state autoregressively:

$$\hat{s}_{t+i+1} = \omega(\hat{s}_{t+i}, a_{t+i}, e_{t+i}), \quad (3)$$

where  $i \in \{0, 1, \dots, N-1\}$  and  $N = 20$  is the autoregressive prediction window size. Since we introduce environment dynamics variance at this stage, different environment configurations can significantly affect the forward dynamics. The history encoder needs to provide sufficiently informative dynamics features to identify environments, so that the

world model can correctly predict the next-frame robot state taking the features as input (**dynamics-aware** world model prediction).

To train the history encoder  $\phi$  and world model  $\omega$ , we first sample a window of  $H + 1 + N = 100$  state-action pairs from the data buffer. For initialization, the history encoder  $\phi$  takes the first  $H$  data pairs as history input and computes initial dynamics feature embeddings  $e_t$ . Then the world model takes the  $(H + 1)$ -th state  $s_t$  as the initial state  $\hat{s}_t$ , and autoregressively predicts the next  $N$  states. The loss function can be written as:

$$\mathcal{L}_{\text{wm}} = \sum_{i=1}^N \|s_{t+i} - \hat{s}_{t+i}\|_1. \quad (4)$$

$\mathcal{L}_{\text{wm}}$  updates both the history encoder  $\phi$  and the world model  $\omega$  simultaneously through backpropagation.

#### 2) Dynamics Adaptability Injection

Existing works [27], [28] often use a single network to handle both basic action execution and dynamics adaptability. Such coupling will increase the difficulty of network learning. When the dynamics vary excessively, basic action execution capability can be severely affected and tends to become over-conservative, which is highly detrimental to motion tracking tasks that demand expressiveness and anthropomorphic quality.

In this work, we reformulate dynamics adaptability as an additional capability on top of basic action execution. To avoid compromising the already acquired motion tracking capability. We freeze the network parameters of Any2Track and introduce an adapter [15] architecture for fine-tuning. The adapter  $\xi$  consists of  $M$  layers with zero-initialized weights  $\xi = \{\xi_1, \dots, \xi_M\}$ , where  $M$  is the number of layers of AnyTracker network. At the beginning, the zero-initialized adapter does not affect the base tracker’s output. As fine-tuning progresses, the adapter injects dynamics adaptability into the base model through layer-wise feature fusion as shown in Figure 2. Such a training paradigm avoids degrading the already acquired motion tracking performance, ultimately yielding both rich motion expressiveness and strong dynamics adaptability.

We use the same rewards in Table II to finetune the adapter, but within an environment with dynamics variance. In each iteration, we first update the data replay buffer, and then the adapter and the history encoder are alternately updated.

## IV. EXPERIMENTS

In this section, we provide extensive experimental results in both the MuJoCo [31] simulator and the real-world deployment. We choose the 29-DoF Unitree G1 humanoid robot for all our experiments. Here, we aim at addressing the following three questions:

- **Q1:** Can AnyTracker improve the performance of general motion tracking compared to existing methods?
- **Q2:** Can AnyAdapter outperform other baseline methods in online dynamics adaptation capability against disturbances from different sources?

- **Q3:** How does Any2Track perform in various real-world scenarios?

### A. Results of Tracking Any Motions

To address **Q1** (*Can AnyTracker improve the performance of general motion tracking compared to existing methods?*), we compare AnyTracker with baseline methods and evaluate each of our major designs in this part.

#### 1) Experiment Setting

We use our curated motion datasets, as mentioned in Section III-A, to train all the methods in simulation. We evaluate on the AMASS test set and LAFAN1 following [5]. In this experiment, we do not introduce any dynamics variance, only compare the quality of motion tracking.

#### 2) Experiment Metrics

We use the following metrics:

- **Success Rate (SR, %)** records the percentage of successful tracking trials. For motion tracking of a certain trajectory, the tracking is defined as unsuccessful when the averaged joint position error or the root height error is greater than 0.2m.
- **Mean Per Joint Position Error (MPJPE, mm)** is the average position error of all links.
- **Mean Per Joint Velocity Error (MPJVE, mm/frame)** is the average velocity error of all links.

#### 3) Baselines

Since different methods utilize different simulators, motion datasets, and even different robots, directly comparing each method is unfair. Thus, we choose existing general motion trackers with open-source training code as our baseline methods and reproduce them in MuJoCo using our curated motion datasets. In this experiment, we re-implement OmniH2O and ExBody2.

We also ablate our major designs in AnyTracker, including canonicalized action spaces (**Ours w/o CAS**) and specialist-to-generalist distillation (**Ours w/o distillation**), to validate their effectiveness.

#### 4) Experiment Results

The results are shown in Table IV. AnyTracker outperforms baseline methods in motion tracking quality because of a series of careful designs. The effectiveness of canonicalized action spaces and specialist-to-generalist distillation is also validated. These two designs help alleviate the optimization difficulty brought by complex action spaces and improve the final quantitative results.

### B. Results of Adapting to Any Disturbances

To address **Q2** (*Can AnyAdapter outperform other baseline methods in online dynamics adaptation capability against disturbances from different sources?*), we compare AnyAdapter with baseline methods and evaluate each of our major designs in this part.

#### 1) Experiment Setting

In this experiment, we introduce dynamics variance in the training process of all methods. Training a general motion tracker across LAFAN1 and AMASS under environment disturbances is very challenging, and the specialist-to-generalist

TABLE IV: **Comparison with baseline motion tracking methods.** We use our curated motion datasets to evaluate existing general motion tracking methods and our major designs. **Bold** numbers indicate the best performance.

Method	SR $\uparrow$	MPJPE $\downarrow$	MPJVE $\downarrow$
OmniH2O	75.64	36.12	12.24
Exbody2	79.68	34.70	13.69
Ours w/o CAS	84.92	30.15	9.87
Ours w/o distillation	83.32	31.29	9.61
<b>Ours</b>	<b>89.23</b>	<b>27.96</b>	<b>6.43</b>

distillation process is quite cumbersome. Therefore, all experiments are conducted solely on the LAFAN1 dataset and remove the distillation process. We compare them in the following environments to evaluate their motion tracking quality under disturbances:

- **Terrains:** We use Perlin noise to generate smooth and natural terrains, where we evaluate the terrain adaptability of all methods.
- **External Forces:** We randomly apply a force of random magnitude and direction to the robot’s torso during motion tracking.
- **Physical Property Changes:** We randomly change the torso mass, torso CoM position, and joint friction of the robot.

Besides, we also compare them in an environment without any disturbances to evaluate their basic motion tracking capability.

#### 2) Experiment Metrics

We adopt the same metrics in Section IV-A.2.

#### 3) Baselines

We choose the following methods as baselines:

- **Vanilla PPO:** We directly train the policy with the vanilla PPO algorithm as we described in AnyTracker, with dynamics variance introduced during training. The policy cannot observe the history or the disturbance.
- **Dual History** [27]: The policy can receive both long-term and short-term history as observations. The long-term history is encoded by a convolutional network and concatenated with the short-term history.
- **RMA** [28]: It utilizes a two-stage framework to adapt for different environments. In stage 1, the privileged environment factor is encoded by an MLP to form the environment embedding, and the embedding is fed into the policy network and trained with PPO. In stage 2, the policy network is fixed, and a history encoder is trained to regress the environment embedding. We report the performance of RMA in both stages.
- **DWL** [9]: An encoder-decoder structure is applied to predict privileged information, including privileged robot states and environment dynamics configurations, from the observation history. The decoder is discarded during inference, and the history embedding from the encoder is fed into the policy network.

A brief overview of these methods is depicted in Figure 3. Moreover, we conduct ablation experiments to evaluate the effectiveness of policy adaptation and world model learn-

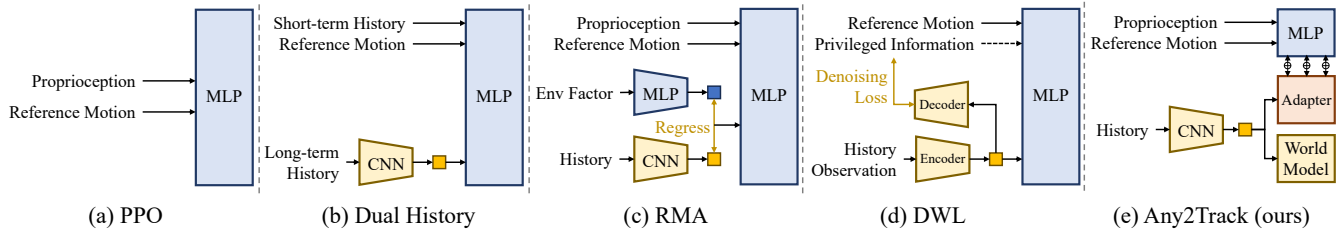


Fig. 3: **Overview of methods evaluated in online dynamics adaptation experiments.** All algorithms are trained with asymmetric PPO, and the critics are omitted in the figure.

TABLE V: **Simulation performance of compared methods on online dynamics adaptation.** Bold numbers indicate the best performance.

Method	w/o Disturbance			Terrains			External Forces			Physical Property Changes		
	SR $\uparrow$	MPJPE $\downarrow$	MPJVE $\downarrow$	SR $\uparrow$	MPJPE $\downarrow$	MPJVE $\downarrow$	SR $\uparrow$	MPJPE $\downarrow$	MPJVE $\downarrow$	SR $\uparrow$	MPJPE $\downarrow$	MPJVE $\downarrow$
PPO	87.4	28.30	7.65	73.0	31.01	8.99	50.0	37.60	12.06	73.6	34.14	8.64
RMA (Stage 1)	88.8	18.58	6.54	75.8	23.99	8.45	52.2	30.22	11.18	77.8	29.54	8.29
RMA (Stage 2)	88.4	19.68	6.76	75.8	24.60	8.56	51.8	30.63	11.33	77.8	29.52	8.35
DWL	87.4	26.04	7.83	78.6	29.27	9.31	42.6	37.24	11.92	70.6	35.11	9.61
Dual History	87.4	22.62	6.92	79.6	26.38	8.58	43.6	34.29	11.62	74.8	32.06	8.76
Ours w/o Adapter	89.6	19.12	6.42	82.4	22.36	7.90	57.6	<b>28.91</b>	11.16	<b>80.6</b>	29.08	8.61
Ours w/o World Model	88.8	21.42	6.56	82.6	26.32	8.21	49.8	36.43	11.40	70.4	32.09	8.52
<b>Any2Track (ours)</b>	<b>89.8</b>	<b>16.46</b>	<b>6.04</b>	<b>83.2</b>	<b>20.68</b>	<b>7.82</b>	<b>59.0</b>	28.97	<b>10.81</b>	<b>80.6</b>	<b>27.75</b>	<b>8.15</b>

ing in the AnyAdapter. **Ours w/o Adapter** train the base policy from scratch with history encoder and world model, without the finetuning stage and adapter architecture. **Ours w/o World Model** train the adapter and history encoder simultaneously via the PPO loss without introducing the world model to provide informative dynamics embeddings.

#### 4) Experiment Results

The results are exhibited in Table V. We observe that Any2Track surpasses all baseline methods under all disturbances. When there is no disturbance, our method achieves the highest success rate and lowest tracking error. When dynamic disturbances are introduced into the test environment, our method shows less performance drop on motion tracking compared with the baseline methods, which indicates our better adaptability. Notably, our method even outperforms the RMA (Stage 1), which takes privileged environment factors as input. This is mainly because Any2Track provides better dynamics features and an adaptable learning strategy.

As for ablation studies, we observe that the world model plays an important role in our method for providing informative dynamics embeddings. The tracking performance without the world model is even worse than vanilla PPO when external push or physical property change is applied. Our method without the adapter also suffers a clear performance drop, which validates the effectiveness of decoupling the learning of basic action execution and adaptability.

### C. Real-World Evaluations

To address **Q3** (*How does Any2Track perform in various real-world scenarios?*), we deploy our policy on the real Unitree G1 robot and report the results in this part.

#### 1) Experiment Setting

We quantify the performance of Any2Track on real-world robots. As we show in Figure 4, we conduct real-world experiments under various disturbance conditions, including (a) complex terrain: we use wooden boards, cardboard, foam,

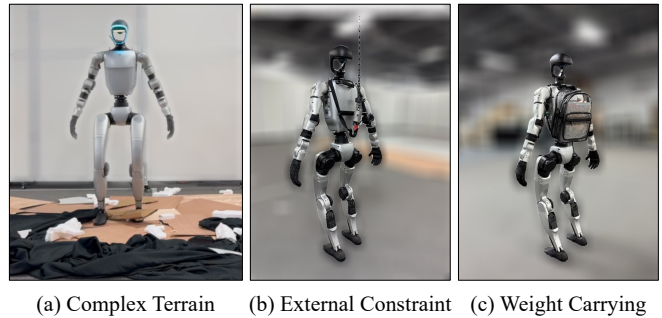


Fig. 4: **Different environment disturbance settings in the real-world experiment.**

TABLE VI: **Real-world performance of compared methods.** Methods are evaluated on identical trajectories and settings.

Method	PPO		Any2Track	
	MPJPE $\downarrow$	MPJVE $\downarrow$	MPJPE $\downarrow$	MPJVE $\downarrow$
w/o Disturbance	29.38	14.26	17.38 (-12.00)	11.56 (-2.70)
Complex Terrain	37.21	15.75	18.34 (-18.87)	11.87 (-3.88)
External Constraint	39.84	16.01	19.17 (-20.67)	13.04 (-2.97)
Weight Carrying	37.52	16.91	23.24 (-14.28)	12.69 (-4.22)

and fabric as terrains; (b) external constraint: the robot’s back was connected to a hoist via a fixed-length rope as external forces; and (c) weight carrying: we add a 5 kg payload on the robot’s back as physical property changes in robot mass. All results are averaged over 5 different motion trajectories.

#### 2) Experiment Metrics

We report MPJPE and MPJVE (mentioned in Section IV-A.2) in this experiment. We collect joint position/velocity information from the motor sensors and calculate the metrics using forward kinematics.

#### 3) Baselines

We choose vanilla PPO with domain randomization as our baseline method.

#### 4) Experiment Results

Table VI depicts the real-world deployment results. The performance of Any2Track beats PPO with domain randomization in all test environments. We found that the advantage of Any2Track increases as the environment disturbances are introduced.

### V. CONCLUSIONS

We propose Any2Track, a novel two-stage motion tracking framework for humanoid robots. Any2Track can track diverse, highly dynamic, and contact-rich motions under multiple real-world disturbances, including terrains, external forces, and physical property changes. Our work paves the way for a foundational humanoid motion tracker in open-world environments with general practical use. For future research, Any2Track can serve as a robust base motion tracking model for different downstream tasks, including whole-body tele-operation, humanoid skill learning, humanoid VLA model, and so on.

### REFERENCES

- [1] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "Humanplus: Humanoid shadowing and imitation from humans," *arXiv preprint arXiv:2406.10454*, 2024.
- [2] 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.
- [3] 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.
- [4] Y. Ze, Z. Chen, J. P. AraÅsjo, Z.-a. Cao, X. B. Peng, J. Wu, and C. K. Liu, "Twist: Teleoperated whole-body imitation system," *arXiv preprint arXiv:2505.02833*, 2025.
- [5] Z. Chen, M. Ji, X. Cheng, X. Peng, X. B. Peng, and X. Wang, "Gmt: General motion tracking for humanoid whole-body control," *arXiv preprint arXiv:2506.14770*, 2025.
- [6] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbab, C. Pan, *et al.*, "Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills," *arXiv preprint arXiv:2502.01143*, 2025.
- [7] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang, "Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit," *arXiv preprint arXiv:2502.13013*, 2025.
- [8] Z. Zhang, C. Chen, H. Xue, J. Wang, S. Liang, Y. Liu, Z. Zhang, H. Wang, and L. Yi, "Unleashing humanoid reaching potential via real-world-ready skill space," *arXiv preprint arXiv:2505.10918*, 2025.
- [9] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen, "Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning," *arXiv preprint arXiv:2408.14472*, 2024.
- [10] Y. Xue, W. Dong, M. Liu, W. Zhang, and J. Pang, "A unified and general humanoid whole-body controller for versatile locomotion," *arXiv preprint arXiv:2502.03206*, 2025.
- [11] C. Sferazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel, "Humanoid-bench: Simulated humanoid benchmark for whole-body locomotion and manipulation," *arXiv preprint arXiv:2403.10506*, 2024.
- [12] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi9579, 2024.
- [13] Y. Li, Y. Zhang, W. Xiao, C. Pan, H. Weng, G. He, T. He, and G. Shi, "Learning gentle humanoid locomotion and end-effector stabilization control," *arXiv preprint arXiv:2505.24198*, 2025.
- [14] Y. Zhang, Y. Yuan, P. Gurunath, T. He, S. Omidshafiei, A.-a. Aghamohammadi, M. Vazquez-Chanlatte, L. Pedersen, and G. Shi, "Falcon: Learning force-adaptive humanoid loco-manipulation," *arXiv preprint arXiv:2505.06776*, 2025.
- [15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models," *ICLR*, vol. 1, no. 2, p. 3, 2022.
- [16] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 60–1, 2020.
- [17] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5442–5451.
- [18] Carnegie Mellon University, "Carnegie-Mellon mocap database," <http://mocap.cs.cmu.edu/>, Pittsburgh, PA, USA, Mar 2007, [Online].
- [19] Z. Luo, J. Cao, K. Kitani, W. Xu, *et al.*, "Perpetual humanoid control for real-time simulated avatars," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 895–10 904.
- [20] 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 (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.
- [21] Z. Zhang, Y. Li, H. Huang, M. Lin, and L. Yi, "Freemotion: Mocap-free human motion synthesis with multimodal large language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 403–421.
- [22] C. Tessler, Y. Guo, O. Nabati, G. Chechik, and X. B. Peng, "Masked-mimic: Unified physics-based character control through masked motion inpainting," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–21, 2024.
- [23] Z. Luo, J. Cao, J. Merel, A. Winkler, J. Huang, K. Kitani, and W. Xu, "Universal humanoid motion representations for physics-based control," *arXiv preprint arXiv:2310.04582*, 2023.
- [24] Z. Luo, J. Cao, S. Christen, A. Winkler, K. Kitani, and W. Xu, "Grasping diverse objects with simulated humanoids," *arXiv preprint arXiv:2407.11385*, 2024.
- [25] Y. Li, M. Lin, Z. Lin, Y. Deng, Y. Cao, and L. Yi, "Learning physics-based full-body human reaching and grasping from brief walking references," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 27 673–27 682.
- [26] Y. Liu, B. Yang, L. Zhong, H. Wang, and L. Yi, "Mimicking-bench: A benchmark for generalizable humanoid-scene interaction learning via human mimicking," *arXiv preprint arXiv:2412.17730*, 2024.
- [27] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 840–888, 2025.
- [28] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.
- [29] J. Lyu, Z. Li, X. Shi, C. Xu, Y. Wang, and H. Wang, "Dywa: Dynamics-adaptive world action model for generalizable non-prehensile manipulation," *arXiv preprint arXiv:2503.16806*, 2025.
- [30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [31] 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*. IEEE, 2012, pp. 5026–5033.
- [32] K. Zakka, B. Tabanpour, Q. Liao, M. Haiderbhai, S. Holt, J. Y. Luo, A. Allshire, E. Frey, K. Sreenath, L. A. Kahrs, *et al.*, "Mujoco playground," *arXiv preprint arXiv:2502.08844*, 2025.
- [33] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. Cheng, "Generating diverse and natural 3d human motions from text," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5152–5161.
- [34] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmlR, 2021, pp. 8748–8763.
- [35] J. B. McQueen, "Some methods of classification and analysis of multivariate observations," in *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*, 1967, pp. 281–297.
- [36] 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*, 2011, pp. 627–635.