

Rapid Embodiment Adaptation for Quadrupedal Locomotion

Dichen Li¹, Bo Ai^{1,2}, Nico Bohlinger³, Jan Peters^{3,4,6}, Hao Su⁵, Henrik I. Christensen¹

¹UC San Diego ²Stanford University ³TU Darmstadt ⁴hessian.AI ⁵Sudo AI GmbH

⁶German Research Center for AI (DFKI) & Robotics Institute Germany

<https://embodiment-adaptation.github.io/>

Abstract—Humans readily adapt their movements as their bodies change through aging, injury, or load carrying, but learning-based robot policies often break when hardware properties shift. We introduce an online embodiment adaptation framework for quadrupedal locomotion that infers embodiment parameters from short interaction histories and conditions control on the inferred hardware state. Our method pairs a generalist policy trained under embodiment randomization with a lightweight adaptation module that identifies physical changes within half a second. We evaluate two representative forms of embodiment variation: joint-range constraints and trunk-mass changes, corresponding to joint-level kinematic degradation and body-level dynamic variation. In simulation, the module accurately estimates these changes and enables closed-loop control that substantially outperforms policies conditioned directly on interaction history. On a real Unitree Go2 robot, our system maintains stable locomotion under severe instances of the evaluated changes, including a fully locked leg and a 5 kg payload, where non-adaptive methods fail. These results demonstrate the practicality of explicit online embodiment identification for rapid adaptation to joint-limit and payload-mass changes, and provide a step toward handling broader forms of uncertain, degraded, or changing robot hardware.

I. INTRODUCTION

Cross-embodiment learning aims to build generalist policies that can operate across diverse robot platforms. Recent progress has demonstrated shared controllers across morphologies in locomotion [3, 4, 6, 16] and robot manipulation [12, 22, 34, 24]. However, most prior work assumes a known and static embodiment at deployment time, including information on joint limits, link masses, inertial properties, and the kinematic structure. In contrast, humans continually recalibrate to embodiment changes such as injury, aging, fatigue, load carrying, and tool use [20, 35, 17]. This adaptivity suggests that the robot embodiment should be treated not as a static input, but as a variable that is inferred online for control. This raises a fundamental question: how can a cross-embodiment policy identify and adapt to unknown or changing embodiment parameters during deployment?

System identification provides a principled framework for estimating physical parameters. However, traditional methods that are formulated as an optimization problem require collecting calibration datasets in the target domain. In robot learning, prior work has focused primarily on estimating object properties [2] and environment parameters [15, 37]; the online identification of the robot’s internal embodiment, such

as joint range constraints, remains underexplored. Recent learning-based methods address this by mapping interaction histories directly to actions end-to-end [16], but their implicit adaptation requires long histories (e.g., several seconds of observations), incurring substantial training overhead and slow deployment-time adaptation.

To enable efficient training and rapid adaptation, we decompose control into two separately trained modules (Figure 1): (i) an embodiment adaptation module and (ii) a cross-embodiment policy. The adaptation module infers embodiment parameters online from short-horizon action-state histories, estimating quantities such as joint-limit scaling and trunk mass offset. These parameters represent two complementary forms of embodiment variation: joint-level kinematic constraints and robot-level dynamic changes. We study these variations on a fixed nominal morphology as a focused setting for online embodiment identification. The cross-embodiment policy, trained across randomized variations of these parameters, is explicitly conditioned on the inferred embodiment and generates control commands aligned with the current hardware configuration. This two-stage training provides direct supervision for each module, enabling efficient training and rapid parameter identification at inference compared with implicit end-to-end alternatives.

We validate the framework in both simulation and the real world. In simulation, the adaptation module identifies joint-limit scale and trunk mass offset within only 0.4 seconds of interaction, enabling closed-loop control that substantially outperforms the end-to-end implicit adaptation baseline across the evaluated settings. In the real world, the Unitree Go2 maintains stable locomotion under severe joint-range constraints and payload changes, including a heavily constrained front-right leg and a 5.0 kg payload, where policies without online adaptation fail. These results demonstrate rapid and robust adaptation to the evaluated joint-limit and mass variations, providing a step toward cross-embodiment policies that can handle broader forms of unknown or changing robot parameters.

II. RELATED WORK

A. Cross-Embodiment Learning

Cross-embodiment learning seeks to build generalist policies that learn from and generalize across robots of varying morphologies. Progress has been visible in manipulation, where controllers can operate across multi-fingered hands [12, 22, 34] or broader form factors including robot

(a) Cross-Embodiment Policy Learning

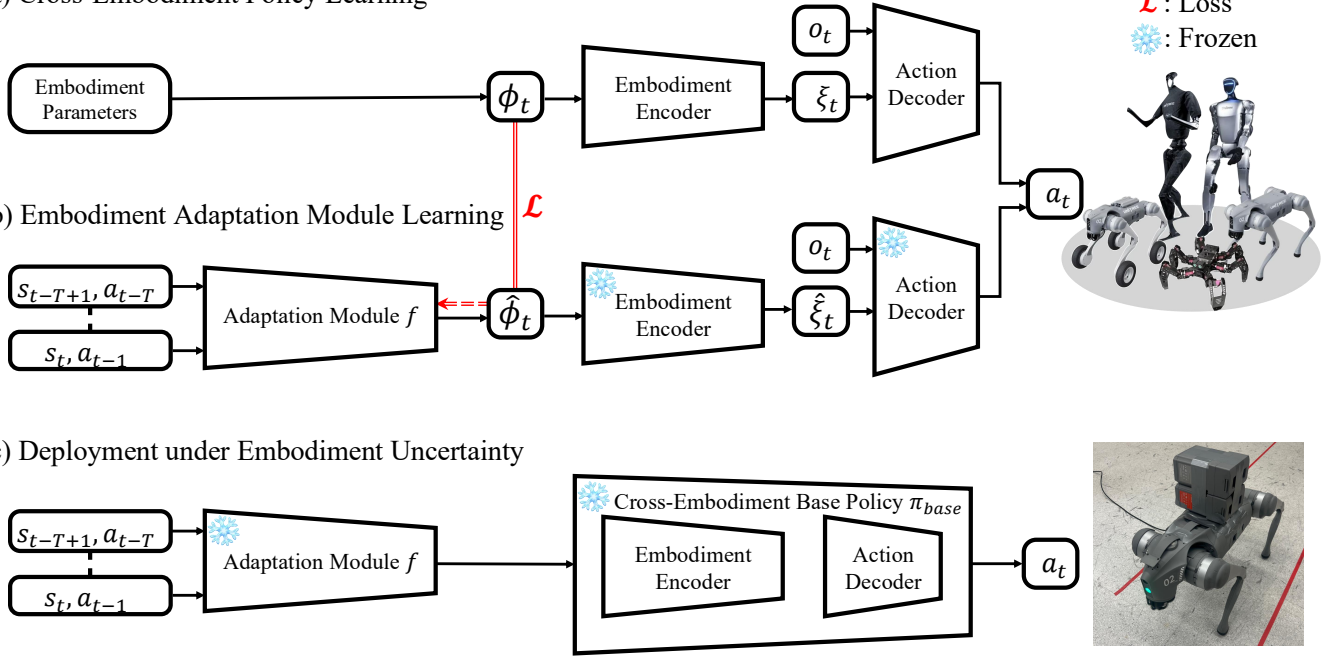


Fig. 1: **Pipeline of the proposed online embodiment adaptation framework.** The cross-embodiment base policy π_{base} is trained to handle locomotion tasks across different embodiments with the ground-truth embodiment description. The adaptation module f is trained to infer embodiment parameters from observation history for the base policy to condition on. The above adaptation method (Explicit Representation) infers physical embodiment parameters $\hat{\phi}_t$. An alternative method (Latent Representation) for inferring latent embodiment representations $\hat{\xi}_t$, not shown here, is described in III-C.

arms, grippers, and mobile manipulators [24]. In legged locomotion, most policies remain specific to a single platform [26, 38, 8, 5, 30, 7], motivating a line of work on universal locomotion controllers.

A central challenge is that different robots have variable observation and action spaces, varying joint counts, kinematic topologies, and actuator properties. Early approaches captured morphological structure with Graph Neural Networks (GNNs), which rely on morphology-specific graph designs and struggle to generalize across distinct morphology families [33, 13]. Transformer-based methods improved on this by globally aggregating embodiment information, though they often depend on manually crafted tokens or fixed joint-type vocabularies that limit transfer to unseen morphologies [11, 29, 23]. The Unified Robot Morphology Architecture (URMA) [6] took a further step by explicitly conditioning the policy on per-joint and general robot parameters; subsequent work has scaled this paradigm with stronger embodiment randomization [4] and shown that increasing training-embodiment diversity improves generalization [3].

These methods, however, assume that an accurate embodiment description is available at deployment, an assumption that breaks under wear-and-tear, changing payloads, or active modifications to the robot’s body [21]. Our work targets this gap by enabling an embodiment-conditioned controller to adapt online when robot-side parameters are unknown or changing.

B. Online Embodiment Identification

Online system identification estimates physical context from interaction during deployment. In robot learning, prior work has inferred object properties from interaction histories [1, 2, 36, 25], as well as environmental and dynamic factors such as terrain geometry, friction, payload, and actuator properties [15, 14, 18, 9, 31]. RMA-style methods [15, 14], in particular, train policies with privileged physical context and learn adaptation modules that recover this context from recent interaction history. Our framework follows this paradigm but applies it to a structured embodiment description that includes both body-level dynamics, such as payload, and per-joint kinematic properties, such as joint ranges and locking. While adaptation to dynamic variation has been studied, online identification of robot-side kinematic changes remains comparatively underexplored. Related implicit approaches instead map interaction histories directly to actions, allowing task-relevant embodiment information to emerge without explicit parameter estimation [16]. However, they typically require histories spanning several seconds. We use direct supervision to infer an explicit or latent embodiment representation from a short interaction history and condition a cross-embodiment policy on the estimate, enabling rapid deployment-time adaptation.

III. METHOD

A. Overview

We consider cross-embodiment learning for robotics, where control policies must generalize across diverse and

potentially changing embodiments. We treat the embodiment parameters ϕ_t , such as joint limits, torque limits, and mass properties, as hidden context variables that affect the robot dynamics and the optimal control strategy.

At time t , the robot receives an observation $o_t \in \mathcal{O}$, consisting of proprioceptive state and command inputs, and executes an action $a_t \in \mathcal{A}$. The system then transitions according to dynamics that depend on both the physical state and the current embodiment parameters ϕ_t . Since ϕ_t is not directly observed at deployment, the policy cannot condition on the true embodiment parameters and must infer them from interaction.

Let

$$h_t = \{(o_{t-T+1}, a_{t-T}), \dots, (o_t, a_{t-1})\}$$

denote a short sliding window of past observations and actions. We introduce an adaptation module f that estimates the current embodiment context from this history,

$$\hat{\phi}_t = f(h_t),$$

and a cross-embodiment policy π that predicts actions conditioned on the current observation and the inferred embodiment:

$$a_t \sim \pi(\cdot \mid o_t, \hat{\phi}_t).$$

This forms a closed-loop adaptation process: as the robot interacts with the environment, the history h_t is updated, the embodiment estimate $\hat{\phi}_t$ is refined online, and the policy adjusts its actions accordingly.

Our objective is to maximize the expected return under the true embodiment:

$$\max_{\pi, f} \mathbb{E} \left[\sum_{t=0}^H \gamma^t R(s_t, a_t; \phi_t) \right],$$

where the reward and transition dynamics depend on the unobserved embodiment parameters ϕ_t . In practice, we train π with reinforcement learning using oracle embodiment information in simulation, and train f via supervised learning to recover the embodiment context from interaction histories.

Next, we introduce the cross-embodiment policy learning (Section III-B), and the adaptation module (Section III-C).

B. Cross-Embodiment Policy

We use URMA [6] as the architecture for our cross-embodiment base policy, which we denote as $\pi_{\text{base}}(a_t \mid o_t, \phi)$. URMA consists of a latent encoder E_{lat} and an action decoder D_{act} . Similar to [4], we train π_{base} with deep reinforcement learning with diverse online embodiment randomization, including joint limits and mass distributions of the robot. Once trained, the parameters of the URMA policy are frozen for all subsequent online adaptation experiments.

C. Embodiment Adaptation Module

The goal of the module is to estimate the embodiment parameters from interaction history h_t . Following prior morphology-conditioned policy design [6, 3], the parameters include joint-level (e.g., joint limit) and body-level information (e.g., body mass).

1) *Embodiment representation*: The adaptation module can predict either physical embodiment parameters ϕ_t or learned latent representations ξ_t that emerge from cross-embodiment policy training. We investigate both representations in this work.

Explicit representation. f_{exp} predicts interpretable physical parameters consisting of per-joint parameters $\hat{d}_t$ and a global trunk mass term $\hat{m}_t$:

$$(\hat{d}_t, \hat{m}_t) = f_{\text{exp}}(h_t), \quad \hat{d}_{t,j} = (\hat{q}_{t,j}^0, \hat{\tau}_{t,j}^{\max}, \hat{q}_{t,j}^{\min}, \hat{q}_{t,j}^{\max}),$$

where $\hat{q}_{t,j}^0$ is the nominal joint position, $\hat{\tau}_{t,j}^{\max}$ is the torque limit, and $\hat{q}_{t,j}^{\min}, \hat{q}_{t,j}^{\max}$ are the joint range endpoints. We construct the full explicit embodiment description $\hat{\phi}_t$ by combining the predicted adaptable parameters with fixed structural descriptors d^{fix} , for all joints J :

$$\hat{\phi}_{t,j} = (d_j^{\text{fix}}, \hat{d}_{t,j}), \quad \hat{\phi}_t = \left[\{\hat{\phi}_{t,j}\}_{j=1}^J, \hat{m}_t \right],$$

The control action is then produced by the frozen base policy: $a_t \sim \pi_{\text{base}}(\cdot \mid o_t, \hat{\phi}_t)$.

Latent representation. f_{lat} predicts latent embodiment embeddings $\hat{z}_t$ of size d_z , where d_z is a hyperparameter:

$$(\hat{z}_t, \hat{m}_t) = f_{\text{lat}}(h_t), \quad \hat{z}_t \in \mathbb{R}^{J \times d_z},$$

with $z_t = E_{\text{lat}}(d_t)$ denoting the latent embedding of ground-truth explicit adaptable parameters d_t . This is lower-dimensional and potentially more structured [3], since the latent representation is optimized to capture the most useful information for control. We construct the full latent embodiment representation $\hat{\xi}_t$:

$$\hat{\xi}_t = [d^{\text{fix}}, \hat{z}_t, \hat{m}_t].$$

Given the latent representation $\hat{\xi}_t$, the latent encoder E_{lat} of the base policy is skipped, and $\hat{\xi}_t$ is directly input to the frozen action decoder D_{act} of the base policy to produce the action: $a_t \sim D_{\text{act}}(\cdot \mid o_t, \hat{\xi}_t)$.

2) *Network architecture*: The adaptation module predicts explicit or latent embodiment representations using the same model architecture. It first uses a Long Short-Term Memory (LSTM) [27] encoder to capture temporal dependencies from the interaction history. The encoded features are then processed by a Transformer backbone that performs attention across the joint dimension, enabling joint-wise aggregation of information. The network predicts trunk mass $\hat{m}_t$ and joint-level information $\hat{d}_t$ or $\hat{z}_t$ separately.

3) *Training*: We train the adaptation module f on offline trajectories collected in simulation under randomized embodiment variations. For each trajectory, we record the ground-truth adaptable embodiment parameters, consisting of per-joint parameters d_t and trunk mass m_t . Both variants minimize the same supervised learning objective,

$$\min_f \mathbb{E}_t \left[\|f(h_t) - \langle \rho_t, m_t \rangle\|^2 \right], \quad (1)$$

where $f = f_{\text{exp}}$ and $\rho_t = \hat{\phi}$ for explicit embodiment representation, and $f = f_{\text{lat}}$ and $\rho_t = \hat{\xi}_t$ for latent embodiment representation.

IV. EXPERIMENTAL SETUP

A. Task Setup

We evaluate the proposed framework on the standard command-velocity-tracking locomotion task [6, 4, 3, 15],

TABLE I: Reward terms in RL policy training.

Reward Term	Coefficient
Tracking XY Velocity Command	$2.0 \cdot \Delta t$
Tracking Yaw Velocity Command	$1.0 \cdot \Delta t$
Pitch Roll Position	$-5.0 \cdot \Delta t$
Base Height	$-30.0 \cdot \Delta t$
Base Z Velocity	$-2.0 \cdot \Delta t$
Air Time	$0.2 \cdot \Delta t$
Stand Still	$-5.0 \cdot \Delta t$
Joint Limit Threshold	$-200.0 \cdot \Delta t$
Joint Torque Threshold	$-200.0 \cdot \Delta t$
Joint Velocity Threshold	$-10.0 \cdot \Delta t$

TABLE II: Environment and embodiment randomization scheme in policy training.

Randomization Term	Range
Joint Limit Scale	[0, 0.5]
Trunk Mass Offset	[-3.0 kg, +7.0 kg]
Motor Strength	[50%, 150%]
PD Control Gains	[50%, 150%]
Ground Friction	[0.05, 2.0]
External Perturbation	± 1.0 m/s
Gravity Perturbation	± 1.0 m/s ²
Initial Orientation	Roll/Pitch: $\pm 11.25^\circ$; Yaw: $\pm 180^\circ$

testing its robustness to changes in embodiment parameters during an episode. In simulation, we sample longitudinal, lateral, and yaw-velocity commands, covering multidirectional locomotion. For the real-world demonstration, we use a fixed forward-velocity command on flat ground to enable controlled and clean comparisons.

We focus on two types of embodiment changes: joint-limit variations and trunk-mass variations. Joint-limit variations simulate actuator degradation or partial joint locking by scaling or restricting the range of motion. Trunk-mass offsets simulate payload additions, resulting in changes in body dynamics. These modifications represent joint-level kinematic constraints and body-level dynamic changes, respectively. We refer to them as joint-limit scaling, which multiplicatively scales the original joint range, and trunk-mass offset, which additively modifies the trunk mass.

Hardware setup. We use the Unitree Go2 quadruped as the robot platform. The robot’s nominal mass is 15.7 kg, including the battery, and it has 12 actuated degrees of freedom, with three joints per leg (hip, thigh, and knee). During deployment, we impose joint constraints by applying a high control gain outside a prescribed joint range and introduce mass changes by manually adding payloads..

B. Cross-Embodiment Policy Learning

We train the locomotion policy with Proximal Policy Optimization (PPO) [28] in the IsaacLab simulator [19]. Following [6, 3], the policy observes the embodiment description: joint description (axis position and orientation, torque limits, velocity limits, PD gains, nominal position, and range limits), body description (trunk mass only), joint state (position and velocity), and body state (trunk angular velocity, command velocity, and projected gravity). The policy predicts joint target positions relative to a nominal standing position, which

TABLE III: Error rate of embodiment parameter prediction on offline validation set.

Model	RMSE
MLP	2.53×10^{-2}
GRU [10]	2.28×10^{-2}
LSTM [27]	2.02×10^{-2}
Transformer [32]	2.39×10^{-2}

are tracked by a PD controller with $K_p = 20$ and $K_d = 0.5$ [26]. Table I summarizes key reward hyperparameters.

Environment and embodiment randomization. We use a performance-based curriculum that gradually expands the randomization ranges during training [4, 3]. This stabilizes learning by exposing the policy to increasingly challenging environments and embodiment variations over time. The final ranges are listed in Table II.

C. Adaptation Module Training

Model implementation details. The adaptation module embeds per-joint histories (8-dim) into a feature representation of dimension $F = 128$. A per-joint LSTM encodes temporal dependencies, followed by a joint-level Transformer with self-attention for cross-joint interactions [3]. The prediction is generated from the decoder at the last layer, which consists of two MLP heads: a per-joint head for joint-level description and a global head for body-level description.

Data collection. We generate multiple modified embodiments by sampling discrete joint limit scale and trunk mass offset within predefined ranges. We then collect trajectories for each embodiment. In total, we generate 36 distinct embodiment configurations evenly spanning the randomization ranges and collect 4096 trajectories with 1000 timesteps each, totaling 147 million timesteps. We split the dataset into training and validation sets at a ratio of 8:2.

Training. The adaptation module f (f_{exp} or f_{lat}) is trained separately via supervised learning on the offline trajectories, minimizing Eqn 1.

V. EXPERIMENTAL RESULTS

We evaluate the proposed online embodiment adaptation framework through the following questions:

- Q1.** Does the adaptation module accurately identify embodiment parameters?
- Q2.** Does online adaptation improve robustness under embodiment uncertainty in simulation?
- Q3.** Does the proposed approach lead to improved performance in the real world?

A. Offline Evaluation of Embodiment Identification

We first evaluate whether the adaptation module can accurately identify embodiment parameters from interactions. We evaluate different design choices by measuring the prediction error on offline datasets.

Architectural comparison. We compare different architectures for aggregating history actions and observations for embodiment parameter prediction (Table III). LSTM

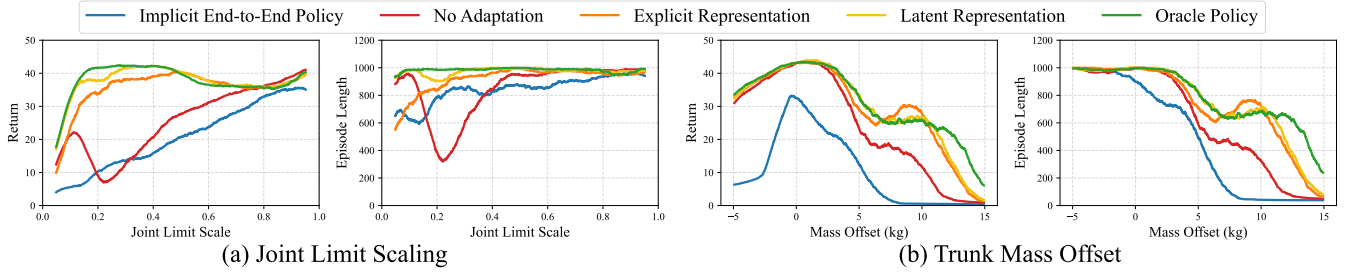


Fig. 2: Mean episode return and episode length versus joint limit scaling and trunk mass offset in simulation. Raw values; max 1000 steps = 20 s. We evaluate under (a) varying joint limit scales and (b) varying trunk mass offset.

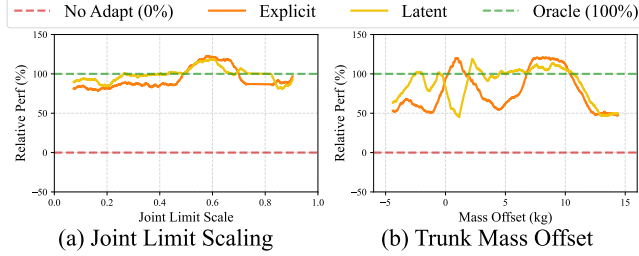


Fig. 3: Relative performance score S_{rel} of Explicit and Latent Representations normalized between No Adaptation (0%) and Oracle Policy (100%). We evaluate under (a) varying joint limit scales and (b) varying trunk mass offset.

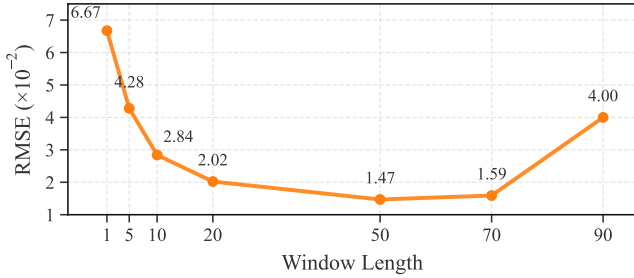


Fig. 4: Identification RMSE versus history window length on the validation set.

achieves the best performance and is used for the rest of the paper.

History length. Fig. 4 plots identification loss versus window length. Increasing the window improves accuracy from short to moderate horizons, peaking around 50 timesteps. Beyond that, further increases yield diminishing returns or degradation. We select 20 timesteps as a practical trade-off between accuracy and efficiency.

Qualitative results. We evaluate under abrupt embodiment changes, where parameters are modified instantaneously during an episode. Our adaptation module (LSTM encoder, 20-timestep window) predicts updated parameters at each timestep. Fig. 5 visualizes predictions when joint limits are modified, or payload is added during an episode. Predicted parameters converge to ground truth within 0.4 s, demonstrating rapid and accurate online adaptation.

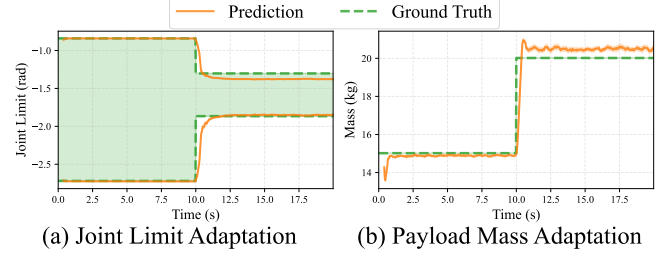


Fig. 5: Qualitative results on embodiment parameter prediction under 2 abrupt changes: (a) joint limit change; (b) payload mass change. Our adaptation module is able to identify changing parameters within 0.4 seconds from interaction history.

B. Closed-Loop Evaluation in Simulation

We evaluate the robustness of locomotion under controlled embodiment changes. Joint limits are scaled relative to the original per-joint limits. Trunk mass is swept as a continuous payload offset beyond the discrete masses used during training, testing, interpolation, and mild extrapolation.

We compare five models: (i) **Implicit End-to-End Policy**: a policy that maps interaction history h_t to actions without estimating embodiment parameters [16], (ii) **No Adaptation**: cross-embodiment base policy with the fixed embodiment description of a canonical Go2 robot, (iii) **Explicit Representation**: the base policy conditioned on explicitly predicted embodiment parameters $\hat{\phi}_t$, (iv) **Latent Representation**: the base policy conditioned on predicted latent embodiment representations $\hat{\xi}_t$, and (v) **Oracle Policy**: the base policy conditioned on the ground-truth embodiment description.

Fig. 2 shows that the No Adaptation degrades rapidly as joint limits and mass offset deviate from the nominal values (1.0 and 0.0 respectively). The Oracle Policy provides an upper bound. Adaptation using both explicit and latent embodiment representations improves robustness across a wide range of embodiment variations, approaching the Oracle Policy. The Implicit End-to-End Policy generally performs worse than both proposed adaptation variants across non-nominal embodiment settings, highlighting the effectiveness of explicitly estimating embodiment parameters.

We normalize performance relative to the (ii) No Adaptation and (v) Oracle Policy as lower and upper bounds.

$$S_{rel} = \frac{R - R_{blind}}{R_{oracle} - R_{blind}} \times 100\%, \quad (2)$$

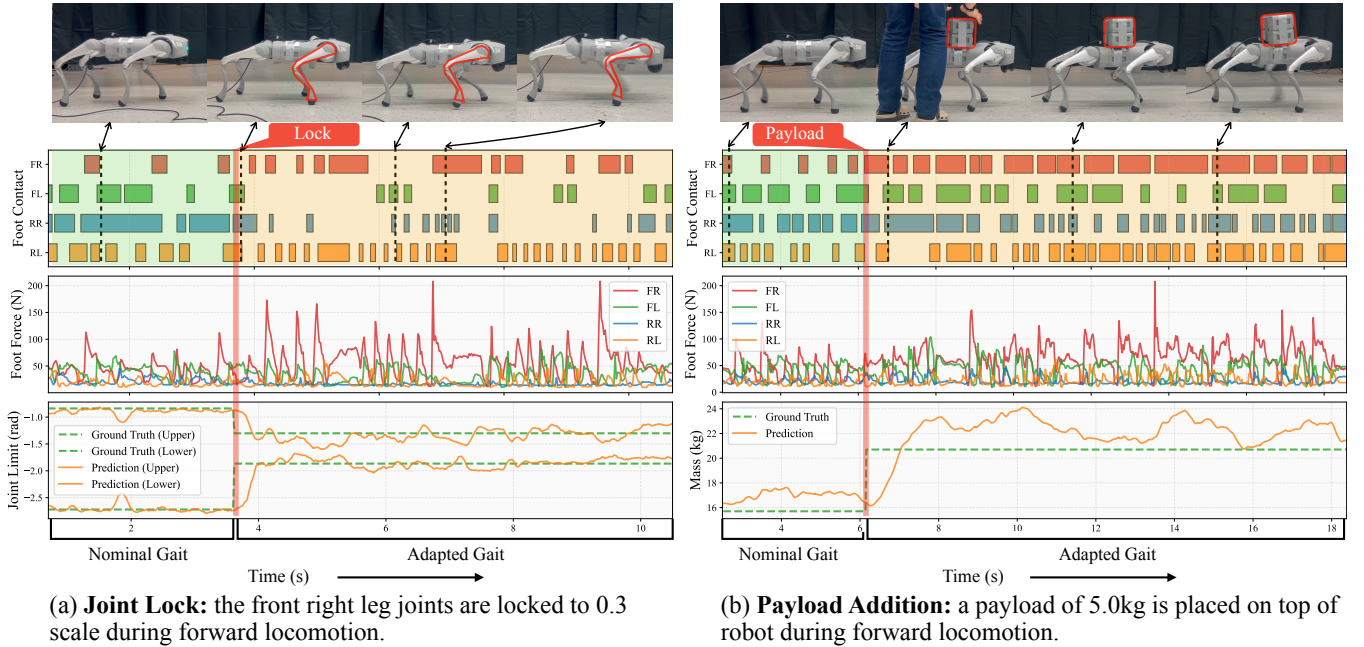


Fig. 6: **Qualitative results for online adaptation under embodiment changes during deployment.** Our adaptation module identifies embodiment parameters within half a second, and the policy conditioned on the inferred parameters adjusts the gait to account for the changed embodiment. (a) When the front-right leg is locked, the robot switches to an asymmetric gait to maintain balance. (b) When the payload is added, the robot shifts to a more load-bearing gait, with longer stance phases and increased foot forces to support the additional mass while maintaining forward locomotion.

TABLE IV: **Quantitative metrics over real-world trials under static embodiment uncertainty.** Each condition was evaluated for 10 independent trials. All metrics are normalized within each column (higher is better). Success rate, TTF (Time to Failure), and distance are scaled relative to the best observed value; heading deviation is inverted via cosine so that smaller errors yield larger normalized values. A score of 1.0 corresponds to a success rate of 80%, TTF of 13 s, distance of 2.4 m (limited by manual termination), and heading deviation of 3° .

Condition	Method	Success (norm.)	TTF (norm.)	Distance (norm.)	Deviation (norm.)
FR Joint Lock 0.1	No Adaptation	0.000	0.154	0.125	0.174
	Ours	0.375	0.308	0.417	0.259
FR Joint Lock 0.3	No Adaptation	0.250	0.385	0.417	0.643
	Ours	1.000	1.000	1.000	0.866
Payload Addition 2.5kg	No Adaptation	0.375	0.462	0.542	0.998
	Ours	0.875	0.923	0.917	0.999
Payload Addition 5.0kg	No Adaptation	0.000	0.308	0.333	0.985
	Ours	0.625	0.769	0.750	0.993

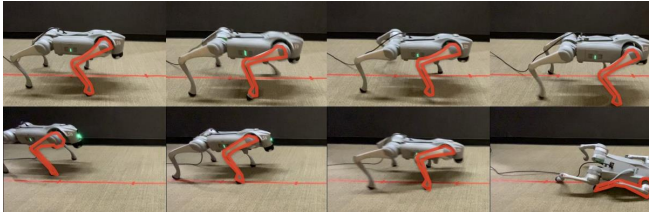
where R denotes the average episode return of a method under the same embodiment setting. By construction, $S_{\text{rel}}=0\%$ and $S_{\text{rel}}=100\%$ match the No Adaptation (canonical Go2) and Oracle Policy (ground-truth), respectively. This normalization quantifies how much of the “oracle gap” is closed and avoids over-interpreting absolute returns that vary across settings. Fig. 3 reports S_{rel} across the joint-limit and trunk-mass sweeps. Both adaptation methods achieve consistently high S_{rel} , indicating robustness under embodiment uncertainty.

C. Real-World Deployment

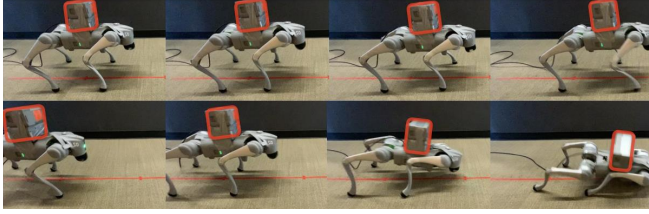
We validate the explicit embodiment representation in the real world. Given that the implicit end-to-end baseline underperforms both proposed adaptation variants across the

simulation sweeps, we focus the hardware evaluation on our framework. Between the explicit and latent variants, we select the explicit representation because its physical parameter estimates are interpretable and directly inspectable during real-world trials. We demonstrate our method under two conditions: scaling the three front-right leg joint limits and adding payload to the trunk. Payload is added by placing one or two battery packs (2.5 kg each) on the trunk. A trial fails if the robot falls or any joint velocity exceeds 15 rad/s.

Static embodiment. We first evaluate robustness when embodiment uncertainty is introduced at episode start. Tab. IV reports metrics averaged over 10 trials; Fig. 7 shows a representative gait comparison. Explicit Representation



(a) Joint Locking



(b) Payload Addition

Fig. 7: Gait comparison between our online adaptation (Explicit Representation) method and the No Adaptation baseline under embodiment changes. We evaluate two embodiment changes: (a) restricting the front-right leg joint limits to 0.3 of their nominal range and (b) adding a 5.0 kg payload to the trunk. The top row in each subfigure shows our method, while the bottom row shows the No Adaptation baseline. In both cases, online adaptation enables stable locomotion, whereas the baseline fails catastrophically.

substantially improves robustness over the No Adaptation across all conditions. In challenging settings (joint limits locked to 0.1, 5.0 kg payload), the policy without adaptation fails early while ours maintains stable locomotion longer and travels farther with less drift. Despite occasional failures in extreme settings, these results show that online adaptation compensates for substantial embodiment uncertainty.

Dynamic embodiment changes. We evaluate abrupt changes applied mid-episode: (i) sudden reduction of front-right leg joint ranges (scaling to 0.3) and (ii) instantaneous 5.0 kg payload addition during locomotion. Fig. 6 shows representative trials. As shown previously in Fig. 7, the base policy typically destabilizes shortly after the change and falls. In contrast, our method updates the parameters. Predictions converge within approximately 20 timesteps (≈ 0.4 s at 50 Hz), and the policy transitions to a gait adapted to the new constraints. For joint limits, the robot adopts an asymmetric gait: the FR leg acts as a crutch in continuous contact while the FL leg swings for forward motion, and stance/swing is redistributed across the remaining legs to keep balance. For payload, the policy lowers the trunk and increases ground contact duration of all four legs to support the load, while maintaining a symmetric gait.

VI. DISCUSSION

We presented an online embodiment adaptation framework for quadrupedal locomotion. The framework infers an explicit embodiment description from short interaction histories and conditions a cross-embodiment policy on the inferred

parameters. On the real-world Unitree Go2, adaptation converges within half a second, sustaining stable locomotion under a fully locked leg and a 5 kg payload where non-adaptive methods fail.

Beyond these results, the framework relaxes a common assumption in embodiment-conditioned policy deployment: that embodiment information is known and static. Our experiments focus on joint-range and trunk-mass changes within a single robot platform, providing a foundation for extending online identification to broader variations in morphology and actuation.

REFERENCES

- [1] Bo Ai et al. “A Review of Learning-Based Dynamics Models for Robotic Manipulation”. In: *Science Robotics* (2025). DOI: 10.1126/scirobotics.adt1497.
- [2] Bo Ai et al. “RoboPack: Learning Tactile-Informed Dynamics Models for Dense Packing”. In: *Robotics: Science and Systems (RSS)* (2024).
- [3] Bo Ai et al. “Towards Embodiment Scaling Laws in Robot Locomotion”. In: *Conference on Robot Learning (CoRL)* (2025).
- [4] Nico Bohlinger and Jan Peters. “Multi-Embodiment Locomotion at Scale with extreme Embodiment Randomization”. In: *arXiv preprint arXiv:2509.02815* (2025).
- [5] Nico Bohlinger et al. “Gait in Eight: Efficient On-Robot Learning for Omnidirectional Quadruped Locomotion”. In: *International Conference on Intelligent Robots and Systems* (2025).
- [6] Nico Bohlinger et al. “One Policy to Run Them All: an End-to-end Learning Approach to Multi-Embodiment Locomotion”. In: *Conference on Robot Learning* (2024).
- [7] Xuxin Cheng et al. “Expressive whole-body control for humanoid robots”. In: *arXiv preprint arXiv:2402.16796* (2024).
- [8] Xuxin Cheng et al. “Extreme Parkour with Legged Robots”. In: *RoboLetics: Workshop on robot learning in athletics @ CoRL*. 2023.
- [9] Suyoung Choi et al. “Learning quadrupedal locomotion on deformable terrain”. In: *Science Robotics* 8.74 (2023), eade2256.
- [10] Junyoung Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. In: *arXiv preprint arXiv:1412.3555* (2014).
- [11] Agrim Gupta et al. “Metamorph: learning universal controllers with transformers”. In: *International Conference on Learning Representations*. ICLR. 2022.
- [12] Zihao He et al. “Scaling Cross-Embodiment World Models for Dexterous Manipulation”. In: *arXiv preprint arXiv:2511.01177* (2025).
- [13] Wenlong Huang, Igor Mordatch, and Deepak Pathak. “One policy to control them all: Shared modular policies for agent-agnostic control”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4455–4464.
- [14] Ashish Kumar et al. “Adapting rapid motor adaptation for bipedal robots”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 1161–1168.
- [15] Ashish Kumar et al. “Rma: Rapid motor adaptation for legged robots”. In: 2021.
- [16] Min Liu, Deepak Pathak, and Ananye Agarwal. “LocoFormer: Generalist Locomotion via Long-context Adaptation”. In: *Conference on Robot Learning*. PMLR. 2025, pp. 532–546.

- [17] Angelo Maravita and Atsushi Iriki. “Tools for the body (schema)”. In: *Trends in Cognitive Sciences* 8.2 (2004), pp. 79–86. ISSN: 1364-6613. DOI: <https://doi.org/10.1016/j.tics.2003.12.008>. URL: <https://www.sciencedirect.com/science/article/pii/S1364661303003450>.
- [18] Takahiro Miki et al. “Learning robust perceptive locomotion for quadrupedal robots in the wild”. In: *Science robotics* 7.62 (2022), eabk2822.
- [19] Mayank Mittal et al. “Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning”. In: *arXiv preprint arXiv:2511.04831* (2025).
- [20] R Nezafat, R Shadmehr, and H H Holcomb. “Long-term adaptation to dynamics of reaching movements: a PET study”. en. In: *Experimental Brain Research* 140.1 (Sept. 2001), pp. 66–76.
- [21] Tønnes F Nygaard et al. “Real-world embodied AI through a morphologically adaptive quadruped robot”. In: *Nature Machine Intelligence* 3.5 (2021), pp. 410–419.
- [22] Austin Patel and Shuran Song. “GET-Zero: Graph Embodiment Transformer for Zero-shot Embodiment Generalization”. In: *2025 IEEE International Conference on Robotics and Automation (ICRA)*. 2025.
- [23] Austin Patel and Shuran Song. “Get-zero: Graph embodiment transformer for zero-shot embodiment generalization”. In: *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2025, pp. 14262–14269.
- [24] Physical Intelligence et al. “ $\pi_0.7$: A Steerable Generalist Robotic Foundation Model with Emergent Capabilities”. In: *arXiv preprint arXiv:2604.15483* (2026).
- [25] Haozhi Qi et al. “In-Hand Object Rotation via Rapid Motor Adaptation”. In: *Conference on Robot Learning (CoRL)*. 2022.
- [26] Nikita Rudin et al. “Learning to walk in minutes using massively parallel deep reinforcement learning”. In: *Conference on robot learning*. 2022.
- [27] Jürgen Schmidhuber, Sepp Hochreiter, et al. “Long short-term memory”. In: *Neural Comput* 9.8 (1997), pp. 1735–1780.
- [28] John Schulman et al. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).
- [29] Carmelo Sferrazza et al. “Body Transformer: Leveraging Robot Embodiment for Policy Learning”. In: *Conference on Robot Learning*. PMLR. 2025, pp. 3407–3424.
- [30] Maximilian Stasica et al. “Bridge the Gap: Enhancing Quadruped Locomotion with Vertical Ground Perturbations”. In: *International Conference on Intelligent Robots and Systems* (2025).
- [31] Giorgio Valsecchi et al. “Accurate power consumption estimation method makes walking robots energy efficient and

- quiet”. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 13282–13288.
- [32] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).
 - [33] Tingwu Wang et al. “Nervenet: Learning structured policy with graph neural networks”. In: *International conference on learning representations*. 2018.
 - [34] Zhenyu Wei et al. “ $\mathcal{D}(\mathcal{R}, \mathcal{O})$ Grasp: A Unified Representation of Robot and Object Interaction for Cross-Embodiment Dexterous Grasping”. In: *2025 IEEE International Conference on Robotics and Automation (ICRA)*. 2025, pp. 4982–4988. DOI: 10.1109/ICRA55743.2025.11127754.
 - [35] DA Winter. “Biomechanics and Motor Control of Human Movement”. In: ed. by DA Winter et al. 2nd. New York: John Wiley & Sons, 1990. Chap. 10, pp. 250–280.
 - [36] Zhenjia Xu et al. “DensePhysNet: Learning Dense Physical Object Representations Via Multi-Step Dynamic Interactions”. In: *Robotics: Science and Systems*. 2019.
 - [37] Dingqi Zhang et al. “A Learning-Based Quadcopter Controller With Extreme Adaptation”. In: *IEEE Transactions on Robotics* 41 (2025), pp. 3948–3964. DOI: 10.1109/TRO.2025.3577037.
 - [38] Ziwen Zhuang et al. “Robot Parkour Learning”. In: *Conference on robot learning*. 2023.