MorFiC: Fixing Value Miscalibration for Zero-Shot Quadruped Transfer

Prakhar Mishra[1], **Amir Hossain Raj**[2], **Xuesu Xiao**[2], **and Dinesh Manocha**[1]
[1]University of Maryland, College Park, MD, USA
[2]George Mason University, Fairfax, VA, USA

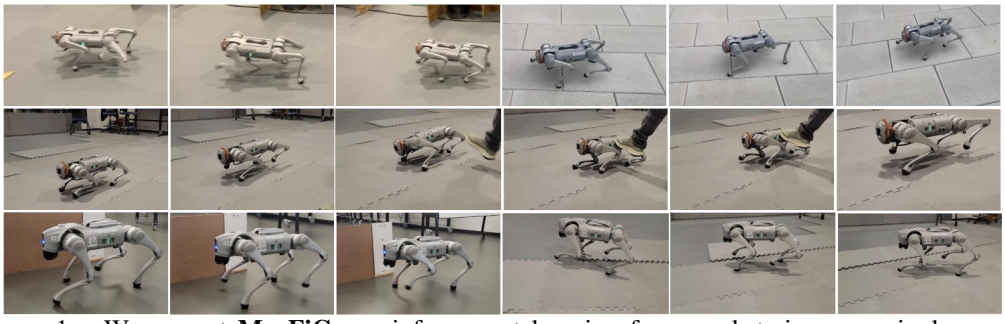

Figure 1: We present **MorFiC**, a reinforcement learning framework trains on a single source quadruped and transfers zero-shot to multiple robots by improving the *value function* via multiplicative conditioning. Real-world deployment on a Unitree Go1: MorFiC runs stably without task-specific fine-tuning. *Top two rows:* Go1 indoor/outdoor ($\approx$ 1.4–1.7 m/s). *Second row:* push-disturbance recovery on Go1. *Bottom row:* Zero-shot Go2 ($\approx$ 1.1–1.3 m/s) same policy weights.

Abstract: Generalizing learned locomotion policies across quadrupedal robots with different morphologies remains a challenge. Policies trained on a single robot often fail when deployed on embodiments with different mass distributions, kinematics, joint limits, or actuation constraints, forcing per-robot retraining. Prior works have approached this primarily either by *scaling* via training across real or generated embodiments or using *large* architectures producing transferable policies which are both storage and compute heavy. We argue that both of these approaches circumvent a key failure mode in actor-critic learning: a shared value function tends to average incompatible value targets across embodiments, yielding miscalibrated advantages and fixing that helps transfer and reduce the compute cost. We present MorFiC, a reinforcement learning approach for zero-shot cross-morphology locomotion which fixes this issue using multiplicative critic conditioning on a morphology latent. Trained with a single source robot with morphology randomization in simulation, MorFiC achieves zero-shot transfer to total seven robots and reaching forward velocity competitive or surpassing scaling-based and morphology-conditioned PPO baselines for examples 1.98 m/s on AlienGo, where as additive PPO baselines remain below 0.65 m/s. MorFiC achieves this using single training robot within 2 hours of training time compared to tens or hundreds of hours required for scaling baselines training on multiple embodiments. We diagnose critic calibration via three metric: Explained variance, advantage sign-flip rate and policy gradient cosine, confirming our critic conditioning as determinant for policy update direction. Finally, we demonstrate zero-shot deployment on Unitree Go1 and Go2 robots without fine-tuning.

Keywords: Cross-morphology generalization, Value Learning

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

1 Introduction

Reinforcement learning has allowed quadrupedal robots to achieve agile and robust locomotion across a wide range of speeds, terrains, and disturbances. Despite this progress, most of the learned locomotion policies remain tightly coupled to a single robot embodiment. Even within the same family of quadrupeds, policies trained on one platform often fail when deployed on another, substantially limiting policy reuse and increasing the deployment cost [1, 2, 3, 4].

Generalizing locomotion policies across robot morphologies is challenging because relatively small changes in physical structure, such as mass distribution, joint limits, or actuation capacity, can significantly alter the underlying dynamics [5]. Policies trained on a single robot tend to exploit morphology-specific regularities, which leads to brittle behavior when transferred to robots with different physical properties. Although domain randomization improves robustness to modeling errors and facilitates sim-to-real transfer [6, 7, 8], it does not explicitly address systematic shifts in dynamics caused by morphology variation. As a result, policies often degrade sharply when evaluated in robots whose morphology lies near or beyond the boundary of the training distribution.

A natural response to this limitation is to condition policies on explicit morphology descriptors, *i.e.,* a vector of robot physical and control parameters (e.g., masses, joint limits, torque limits). The field has taken two main approaches to address this: (i) Scaling methods to train across multiple embodiments (real or generated) or (ii) using large architecture models and morphology randomization [9, 10, 11, 12, 13, 14, 15]. Training these approaches to achieve cross-morphology transfer comes with huge compute and storage cost and even then these morphology-agnostic policies might still generalize within in-family range of embodiments. However, in practice, reliable zero-shot transfer remains difficult, particularly for robots that differ substantially from those seen during training [13, 9, 12, 10, 15].

We argue that this limitation arises not primarily from insufficient policy expressiveness but from miscalibrated value estimation (i.e., the model's predicted long-term return for a given state, action and morphology) across morphologies and in both of the current approach, a shared failure mode has been underexplored. We argue that the value function, not only the policy, should also generalize, in standard actor-critic methods such as PPO [16], a single critic is trained to approximate expected returns in all training environments. When multiple morphologies are present, the same state can correspond to fundamentally different future outcomes depending on the robot's physical properties. A shared critic is therefore forced to average incompatible value targets, inducing morphology-dependent bias in the value function. This bias propagates directly to the advantage estimates used for policy updates, destabilizing learning and disproportionately harming performance on edge-case morphologies [17].

We propose MorFiC which addresses this critic miscalibration via multiplicative conditioning using morphology latent. So our key findings are: (i) A diagnosis: We identify critic miscalibration as the poor cross morphology transfer, present in both scaling and large architecture based approaches. (ii) A method: MorFiC a multiplicative critic conditioning trained from single source robot and transfers to seven total quadrupeds using roughly two hours of GPU training significantly less that the current methods taking (40-300 hours) (iii) Evidence: critic diagnostic for the PPO and external baselines, a controlled actor only experiment with Monte carlo return, show why additive conditioning fails and zero-shot transfer on Go1 and Go2 robots.

2 Related Work

Learning-Based Quadrupedal Locomotion: Reinforcement learning has enabled quadrupedal robots to achieve agile and robust locomotion, including high-speed running and disturbance recovery, both in simulation and on hardware [3, 2, 18]. These successes are largely driven by large-scale simulation and careful reward design, but typically assume a fixed robot embodiment during training. As a result, the learned policies are highly robot-specific and often fail when deployed on robots

with different morphologies. In contrast, MorFiC explicitly targets cross-morphology generalization by addressing value-function errors that arise when training across varying robot dynamics.

Morphology-Conditioned and Multi-Robot Policies: Several recent works explore generalizing locomotion policies across robot morphologies by conditioning on physical parameters or learned embodiment representations. URMA [19], Embodiment Scale [12], GenLoco [9], Body Transformer [10], and ManyQuadrupeds [11] demonstrate that a single policy can control multiple quadrupedal robots when provided with morphology information. However, these approaches rely on shared critics and assume that value functions generalize across morphologies.

Value Learning, & Training Stability : Value-function accuracy is critical for stable actor-critic learning, yet its role in cross-morphology generalization has received limited attention in locomotion research. Previous work has shown that value misestimation can destabilize deep reinforcement learning [17], and recent studies highlight interference issues in shared critics in multiple tasks or environments [20]. MorFiC addresses this issue directly by identifying morphology-induced value bias and mitigating it through morphology-aware critic modulation, leading to more stable advantage estimates and improved policy optimization.

3 Problem Setup & Motivation

We consider quadrupedal locomotion under morphology variation as a parameterized control problem. Each robot embodiment is identified by a vector of morphology parameters $m \in \mathcal{M}$ encoding physical and control properties (for example, mass distribution, joint limits, actuator limits). Training is performed on a morphology distribution $p_{\text{train}}(m)$, while evaluation is performed on a holdout morphology $m^\star$ without additional training, fine-tuning, or online adaptation.

Formulation: Let locomotion across embodiments be modeled as a parameterized Markov Decision Process: $\mathcal{M}(m) = \langle \mathcal{S}, \mathcal{A}, P_m, R_m, \gamma \rangle$, with state space $\mathcal{S}$, action space $\mathcal{A}$, morphology-dependent transition function $P_m : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and reward function $R_m : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and discount factor $\gamma \in (0, 1)$. This formulation is closely related to contextual and hidden-parameter MDPs that model families of environments with shared structure but varying dynamics [21].

Advantage Function: Policies are trained using advantage-based actor–critic methods, via PPO optimization [16]. So, the advantage estimate: $\hat{A}_t = \hat{G}_t - V_\phi(s_t)$ becomes important, where $\hat{G}_t$ denote an empirical return estimate, V_ϕ is learned value function. In practice, generalized advantage estimation is widely used to compute $\hat{A}_t$ with a favorable bias–variance tradeoff [22]. Because the advantage weights the policy gradient, the systematic value error directly alters the direction and magnitude of policy updates [16].

Critic Miscalibration: In multi-morph legged locomotion, the true value is morphology-dependent, $V^\pi(s_t, m)$ [9], but a shared critic with additive (concatenation) or scaling-based conditioning tends to learn the average value $\bar{V}_\phi(s_t) \approx \mathbb{E}_{m \sim p_{\text{train}}}[V^\pi(s_t, m)]$. Since identical (s_t, a_t) inputs can have different returns under different morphologies, a shared critic averages these incompatible targets, yielding morphology-dependent bias $b_m(s_t) = \bar{V}_\phi(s_t) - V^\pi(s_t, m)$ [23, 11]. This bias propagates to the advantage: $\hat{A}_t(m) \approx A^\pi_m(s_t, a_t) - b_m(s_t)$ which drives the PPO update, this produces inconsistent gradients across morphologies and unstable zero-shot transfer on OOD robots. This motivates us to fix the critic to represent morphology-specific returns within a single shared network.

4 MorFiC: Morphology Conditioned Critic Modulation

MorFiC is a morphology-aware actor–critic framework for zero-shot cross-morphology locomotion. The starting point is the failure mode in Section 3 critic miscalibration: MorFiC fixes this by conditioning the critic. This conditioning mechanism is derived in the rest of this section by (i) explicitly sampling morphology parameters and embedding, and (ii) addressing Additive Vs. Multiplicative principle (iii) critic Modulation Variants (iv) morphology-aware advantage.

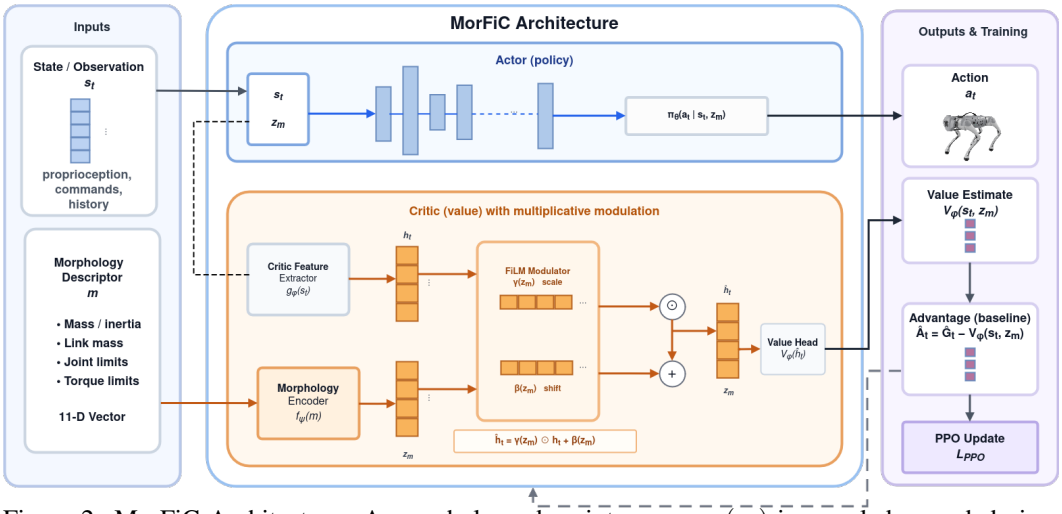

Figure 2: MorFiC Architecture. A morphology descriptor $m \sim p(m)$ is sampled, encoded via a morphology latent $z = f(m)$; and used to modulate the crtic via multiplicative conditioning; actor is also conditioned via same z.

Design Requirement : A morphology-aware critic must satisfy two criteria: (i) representational sufficiency, so the critic must be able to represent distinct morphology landscape within the shared network (ii) PPO stability, the critic conditioning should not introduce noisy, or unstable value estimates during optimization, which destabilizes the advantage updates $\hat{A}_t(m)$. In the rest of this section, we develop a morphology-conditioned critic modulation method which satisfies these two requirements while maintaining the efficiency of shared actor-critic network.

Morphology Randomization & Latent Encoder : During training, each of the parallel environments samples a randomized morphology vector $m \sim p_{\text{train}}(m)$ during reset, selected from a defined range (Appendix). The sampled normalized vector is applied to these environments in the simulator and kept fixed during the episode, so each robot acts as a single consistent embodiment. MorFiC maps m to a learned latent representation $z_m = f_\psi(m)$, $z \in \mathbb{R}^{d_z}$, using a lightweight MLP. The same z is provided to both the actor and the critic.

Additive Vs. Multiplicative Principle: To satisfy the design requirements, we need a critic to represent a morphology-indexed value functions. This gives us three possible choices: concatenating the morphology/other relevant input the critic, per-morphology heads and multiplicative conditioning. Per-morphology head can solve the value miscalibration challenge since it does not define critic for unseen morphology, it violates the zero-shot transfer and generalization argument. In concatenation, we simply append the z_m vector to the input but this signal gets suppressed due to the large observation size. As Jayakumar et. al [24] contrast this additive limitation with multiplicative conditioning, for example $f(x, z) = W[x; z] + b$ where $[x; z]$ represents concatenation of x & z, which can be further decomposed to (1), while the multiplicative interactions would have form: $f(x, z) = z^\top W x + z^\top U + V x + b$, represented by (2), (3).

$$\begin{aligned} f_{\text{cat}}(x, z) &= W_x x + W_z z + b, \\ V_\phi^{\text{cat}}(s_t, z_m) &= w_h^\top h_t + w_z^\top z_m + c, \end{aligned} \tag{1}$$

$$V_\phi^{\text{mult}}(s_t, z_m) = z_m^\top W h_t + z_m^\top U + B h_t + c \tag{2}$$

$$V_\phi^{\text{mult}}(s_t, z_m) = A_m h_t + c_m, \qquad A_m = z_m^\top W + B, \qquad c_m = z_m^\top U + c. \tag{3}$$

As its clear from the equation (1), there is no h_t and z_m interaction in additive conditioning, while multiplicative conditioning has $z_m^\top W h_t$, which is strictly missing from concatenation. So, Criterion (i), therefore, eliminates the additive conditioning, hence the critic must use multiplicative conditioning to represent the diverse value maps.

Modulation Variants: We have considered three modulation variants: FiLM, AdaLN and Hypernetwork conditioning [25, 26, 27]. These are the different approximation for the multiplicative principles rather than different conditioning. As Jayakumar et. al describe gating like FiLM, AdaLN style modulation and hypernetworks as the variation of multiplicative interactions [24].

A FiLM network modulates the critic features through morphology-conditioning via affine transformations and the h_t from (2) becomes:

$$h_t^{\text{FiLM}} = \gamma(z_m) \odot g_\phi(s_t) + \beta(z_m). \tag{4}$$

While AdaLN applies those morphology dependent affine transformation after normalizing the critic features and the h_t becomes:

$$h_t^{\text{AdaLN}} = \gamma(z_m) \odot \text{LN}(g_\phi(s_t)) + \beta(z_m). \tag{5}$$

So, both FiLM and AdaLN apply morphology dependent affine transformation, while AdaLN additionally controls feature via applying normalization. However, Hypernetworks directly generates the morphology-conditioned critic parameters using (6):

$$V_\phi^{\text{hyper}}(s_t, z_m) = A_m h_t + c_m, \quad (A_m, c_m) = H_\eta(z_m). \tag{6}$$

This allows the morphology to modulate the critic head itself allowing cross-feature mixing rather than just passive concatenation. Thus richer conditioning can improve morphology dependent generalization but might introduce higher variance value fitting. All the three variants satisfy the criterion (i) and the criterion (ii) PPO stability, is empirical and ablation section compare these. Since FiLM gives the best overall trade-off and we adopt it as the default instantiation of **MorFiC**: **Mor**phology-aware **Fi**LM **C**ritic.

Morphology-Aware Advantage Estimation: MorFiC uses standard PPO [16] without changing the sampling procedure, policy loss, or update rule. The only intervention is the value baseline used for advantage estimation $\hat{A}_t = \hat{G}_t - V_\phi(s_t, z_m)$, where z_m is the morphology latent shared by the actor and critic. This replaces the morphology-averaged baseline $\bar{V}_\phi(s_t)$ with a morphology-aware critic, directly reducing the bias term $b_m(s_t)$ identified in Section 3.

5 Experimental Setup

Tasks and Robots : We evaluate MorFiC on linear and angular command conditioned locomotion where policy tracks commanded forward/lateral linear and angular velocities. Commands are sampled via a curriculum at the start of the episode and gradually expanded during training and this mechanism is constant across all methods. And, An episode ends if the robot falls, collapses, or reaches time limit. We used Unitree Go2 robot as the base robot template with morphology randomization and evaluation is conducted on the source robot and the held-out robot morphologies like Go1, A1, Aliengo, MIT Mini Cheetah, ANYMal C robots.

Evaluation Protocol : Each parallel environment samples a morphology descriptor $m \sim p_{\text{train}}(m)$ at the episode reset, applies the corresponding physical parameters to the source robot in the simulator, and keeps the descriptor same during the episode. Training is performed on a single source Go2 robot template with morphology randomization applied to that template.

Internal PPO baselines : To isolate the effect of morphology conditioning, We have compared total six variants of PPO under identical training conditions, including MorFiC variants. These variants only differ in how the z_m conditions the actor–critic architecture.

- **PPO:** No morphology conditioning, neither actor nor critic receives z_m.
- **Actor only:** Actor receives z_m; critic is unconditioned.
- **Actor + Critic** z_m is concatenated to both actor and critic (additive conditioning).
- **MorFiC-F:** Actor gets z_m, and critic is modulated via FiLM.
- **MorFiC-A:** Actor gets z_m, and critic is modulated via AdaLN.
- **MorFiC-H:** Actor gets z_m, and critic head is generated using a Hypernetwork.

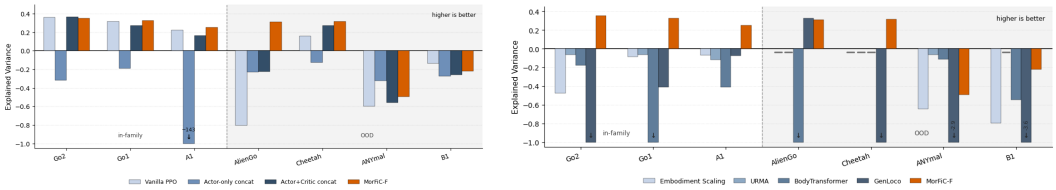

Figure 3: Critic calibration under morphology shift. Left: EV for internal PPO variants. Right: external baselines. Higher EV or closer to 1 is better.

External baselines : In addition to the PPO variants, we compare MorFiC with four cross-morphology locomotion generalization methods namely: URMA [19], GenLoco [9], BodyTransformer [10], and Embodiment Scaling Laws [12]. These situate MorFiC relative to two dominant approaches: (i) Scaling the number of embodiments, or (ii) training on large models and neither of these approaches address value-function calibration. For URMA, GenLoco, Embodiment Scaling Laws, we used the publicly available trained weights, but for BodyTransformer we trained their original available weights to 400M timestep to match MorFiC budget (see appendix).

Metrics: We utilized the following key metrics to measure the performance, survival rate and speed of MorFiC trained policy.

- **Forward speed $\bar{v}$ (m/s):** We measured the forward velocity without tipping or falling on a given command velocity v_{cmd}.
- **Survival Rate S:** For simulation, we define it as the fraction of environment complete the given timestep. And in real-world setting, it's percentage of run, which were successful without any freezing or gait failure. Also, survival rate shall be better interpreted along with the speed because in some case the policy might have perfect survival rate without any feasible locomotion.
- **Explained Variance (EV) :** EV measures how well does the critic predict returns and is give by: $\mathrm{EV} = 1 - \frac{\mathrm{Var}(G_t - V_\phi(s_t))}{\mathrm{Var}(G_t)}$.

6 Results

Zero-shot cross morphology transfer: Table 2 compares MorFiC against three PPO variants for zero-shot transfer from Go2 training morphology to target robots. MorFiC achieves highest forward velocity on every target robot and highest gains come on OOD robots: for example on Aliengo it reaches 1.98 m/s while rest of the additive PPO variants remain below 0.64 m/s. B1 is the exception: all variant fail to have a proper locomotion gait.

Critic Diagnostic: We evaluated the critic miscalibration via explained variance (Fig. 3). Additive conditioning fails to calibrate as morphology shifts towards the heavier and out of distribution robots (Fig. 3). Multiplicative conditioning maintains calibration better and succeed on OOD robots like Cheetah and Aliengo. The Actor-only was most miscalibrated, indicating leaving critic unconditioned makes the overall return noisy and not good for transfer.

Across external baselines, Most critic remain miscalibrated even when the transfer succeed, partly because successful targets fall within their in-family training distribution. So, these large model or embodiment scaling method do not solve the value miscalibration but rather circumvent via scaling,

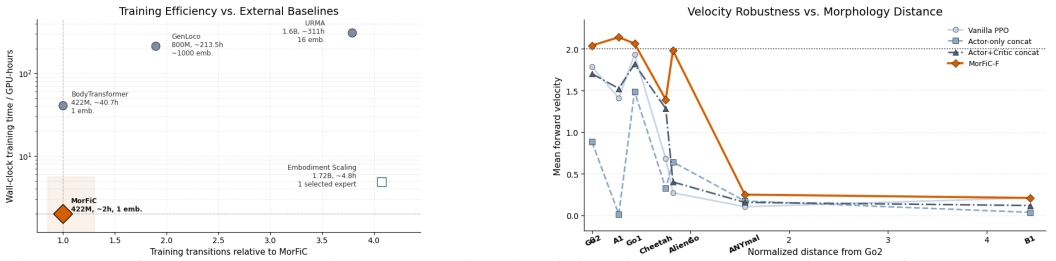

Figure 4: Left: Compute cost (in hours) and total training timesteps used. Right: Performance w.r.t. normalized morphology distance. MorFiC performs better than both external and PPO baselines.

Table 2: **Zero-shot cross-morphology transfer.** Training is on Go2. Each cell reports $\bar{v}_x \pm \sigma/S$, where $\bar{v}_x$ is forward speed in m/s and S is survival rate. Bold denotes the best speed and survival rate. We run all the PPO variants with $v_x^{cmd} = 2.0$ m/s for all 7 robots. All the experiments run for 500 time-steps on total 64 environments.

Target	**PPO** $\bar{v}_x \pm \sigma/S$	**Actor-only** $\bar{v}_x \pm \sigma/S$	**Actor+Critic** $\bar{v}_x \pm \sigma/S$	**MorFiC-F** $\bar{v}_x \pm \sigma/S$
Go2	1.79±0.31/0.984	0.88±0.23/0.265	1.70±0.30/0.891	**2.04±0.37/0.984**
Go1	1.93±0.27/**0.906**	1.49±0.36/0.594	1.82±0.32/0.859	**2.06±0.46**/0.547
A1	1.41±0.42/0.406	0.01±0.06/0.000	1.52±0.27/0.422	**2.14±0.36/0.516**
AlienGo	0.27±0.07/0.000	0.64±0.26/**1.000**	0.40±0.18/0.156	**1.98±0.51/1.000**
B1	0.21±0.06/**1.000**	0.04±0.07/**1.000**	0.12±0.07/**1.000**	**0.22±0.08/1.000**
Cheetah	0.68±0.51/0.282	0.32±0.49/**1.000**	1.28±0.58/0.382	**1.39±0.69**/0.422
ANYmal-C	0.11±0.04/0.135	0.17±0.08/0.469	0.16±0.10/0.216	**0.25±0.21/0.969**

Table 3: **External baselines vs. MorFiC.** $\bar{v}_x \pm \sigma/S$ (m/s; survival). – = no forward motion. $v_x^{\text{cmd}} = 2.0$ m/s (GenLoco: no v_x^{cmd} input). Bold = best.

Target	**URMA** $\bar{v}_x \pm \sigma/S$	**BodyTrans.** $\bar{v}_x \pm \sigma/S$	**GenLoco** $\bar{v}_x \pm \sigma/S$	**Embod.** $\bar{v}_x \pm \sigma/S$	**MorFiC-F** $\bar{v}_x \pm \sigma/S$
Go2	1.84±0.26/**1.000**	0.30±0.21/0.656	0.59±0.05/0.922	1.23±0.06/0.906	**2.04±0.37**/0.984
Go1	1.93±0.28/**1.000**	0.21±0.13/**1.000**	0.98±0.01/**1.000**	1.94±0.09/0.953	**2.06±0.46**/0.547
A1	1.79±0.25/**1.000**	0.39±0.15/0.421	0.92±0.01/**1.000**	1.86±0.08/0.906	**2.14±0.36**/0.516
AlienGo	–/–	0.30±0.28/0.234	0.96±0.03/**1.000**	0.41±0.08/0.000	**1.98±0.51/1.000**
B1	–/–	0.04±0.11/0.796	0.00/0.000	**2.86±0.11**/0.969	0.21±0.08/**1.000**
Cheetah	–/–	–/**1.000**	0.73±0.06/0.969	0.06±0.09/0.000	**1.39±0.69**/0.422
ANYmal-C	**1.80±0.28/1.000**	0.01±0.06/0.875	0.71±0.01/**1.000**	0.44±0.15/0.656	0.25±0.21/0.969

while MorFiC achieves competitive OOD results only by single source robot training and fixing the critic.

Causality analysis. To fully isolate the critic's effect, we freeze the MorFiC's actor policy and collected roll-out data on the target OOD Morphologies. We then varied the four PPO variant's critic to calculate the advantage sign flip rate against Monte-Carlo returns and gradient cosine values induced by the PPO variants' respective critics and report in Table 1. MorFiC obtains the lowest flip rate and a high cosine value, sustaining the earlier argument that critic modulation is important for policy update during morphology shift.

Table 1: **Fixed-rollout diagnostic.** Sign Flip against wrong sign; Cos.: gradient cosine vs. MC.

	AlienGo		Cheetah	
Src.	Flip	Cos.	Flip	Cos.
MC	0.000	1.000	0.000	1.000
MorFiC	**0.255**	**0.366**	**0.209**	**0.679**
Concat	0.329	0.004	0.236	0.575
Act-only	0.318	0.342	0.265	0.644
PPO	0.322	-0.113	0.253	0.264

Comparison to other Baselines. Table 3 compares MorFiC with the four current state of art baselines: URMA [19], GenLoco [9], BodyTransformer [10], and Embodiment Scaling Laws [12]. MorFiC achieves the highest forward velocity on five of seven target robots. URMA despite strong on in-family distribution, still fails on AlienGo, B1 , Cheetah (with no valid forward motion). Embodiment Scaling also barely sustains motion on AlienGo, Cheetah but has surprising 2.86 m/s on B1 robot. GenLoco remains consistent across six robots ($\leq$ 0.98 m/s everywhere except on B1). While BodyTransformer remains weak on nearly all target robots.

External baselines are computationally and storage expensive (Fig. 4), for example URMA roughly takes 311 hours, BodyTransfomrmer 40 hours, GenLoco 213 hours, while single expert policy of Embodiment Scale take 4.7 hours (1.7B) but training the full generalization would exceed more than 60 days and roughly 5TB to store those checkpoints. In contrast, we can train MorFiC in less than 2 hours with very little storage requirement.

Table 4: **Ablation Table.** Each cell reports $\bar{v}_x \pm \sigma/S$, where $\bar{v}_x$ is forward speed in m/s and S is survival rate. Alive time is omitted. Bold denotes the best speed per target robot.

Target	**MorFiC-F** $\bar{v}_x \pm \sigma/S$	**MorFiC-F(Scaled)** $\bar{v}_x \pm \sigma/S$	**MorFiC-H** $\bar{v}_x \pm \sigma/S$	**MorFiC-A** $\bar{v}_x \pm \sigma/S$
Go2	$2.04 \pm 0.37/0.984$	$2.03 \pm 0.26/0.875$	$\mathbf{2.10 \pm 0.28}/0.953$	$2.03 \pm 0.19/\mathbf{1.000}$
Go1	$\mathbf{2.06 \pm 0.46}/0.647$	$1.98 \pm 0.34/0.823$	$1.99 \pm 0.31/0.766$	$1.99 \pm 0.35/\mathbf{0.853}$
A1	$\mathbf{2.14 \pm 0.36}/0.516$	$2.04 \pm 0.24/0.531$	$2.02 \pm 0.26/\mathbf{0.766}$	$1.76 \pm 0.26/0.609$
AlienGo	$\mathbf{1.98 \pm 0.51/1.000}$	$1.78 \pm 0.45/\mathbf{1.000}$	$1.75 \pm 0.46/\mathbf{1.000}$	$1.79 \pm 0.52/\mathbf{1.000}$
B1	$\mathbf{0.21 \pm 0.08/1.000}$	$0.18 \pm 0.08/\mathbf{1.000}$	$0.20 \pm 0.03/\mathbf{1.000}$	$0.18 \pm 0.06/\mathbf{1.000}$
Cheetah	$1.39 \pm 0.69/0.422$	$\mathbf{1.97 \pm 0.62}/0.359$	$1.61 \pm 0.45/0.344$	$1.57 \pm 0.56/\mathbf{0.703}$
ANYmal	$0.25 \pm 0.21/0.969$	$0.24 \pm 0.06/\mathbf{1.000}$	$\mathbf{0.46 \pm 0.19/1.000}$	$0.39 \pm 0.28/\mathbf{1.000}$

7 Ablations

MorFiC instantiates multiplicative conditioning with FiLM critic but to test whether the transfer affected by the specific choice or the broader multiplicative-conditioning principle. We evaluated three other multiplicative variants and report them in Table 4. Our results show that improvements are not FiLM specific, all other variants like hypernetworks, AdaLN and Scaled FiLM achieve similar transfer performance across held-out morphologies. FiLM leads on Go1, A1, AlienGo, the hypernetwork on ANYMal and Go2 and AdaLN was very high on survival rate and stability. We adopt FiLM as the default instantiation since it provided higher velocity on four robots and stable PPO updates, satisfying our design requirements in section III.

8 Real-Robot Results

Zero-shot Hardware Deployment. We deploy MorFiC policy on two hardware platforms namely Go1 and Go2 using an onboard LCM stack for state estimation and actions. We deploy the final policy on robots without any finetuning, MorFiC reaches a nearly ≈ 1.4–1.7 m/s on Go1 robot over 10 trials with a survival rate of over 80% while on Go2 robots it reaches roughly ≈ 1.1–1.3 m/s. We deployed the same go2 policy weights which we used for zero-shot evaluation in sim-transfer.

Failure Modes & Analysis : We have observed two categories of failures, System-level (generic) failure and Robot specific failures (Go2). The generic failures are mainly due to i) Latency and state-estimator drift, ii) Contact mismatch during runs (either foot slips or uneven gait). For robot specific failures, Go2 exhibits a reduced speed and also a higher failure rate under our current deployment pipeline.

9 Limitations

MorFiC relies on morphology randomization during training and we found transfer depends heavily on the normalized distance, so Go2 trained robot transfer well to nearby quadrupeds well (Go1, A1, Cheetah) but fails on distant targets like B1 and Anymal. Since this distance in our case was heavily influenced by base mass and joint limits, even increasing those limits during randomization made policy conservative (Appendix). Training robot choice anchors the transfer direction, Go2 trained policy transfer well to mid-size robots but fails on heavier one; a heavier training robot would likely have inverted pattern. A natural extension to that is combine MorFiC to limited multi-source training rather than large set of robots in scaling methods. Finally we evaluate MorFic transfer on the quadrupeds locomotion tasks; extending the framework to bipeds and humanoids is left to future work.

10 Conclusion

We have studied the zero-shot cross morphology problem and found that transfer failure is strongly driven by the critic miscalibration, leads to biased advantage, and ultimately leads to poor PPO

updates. MorFiC addresses this by conditioning the critic function via multiplicative conditioning. Based on our results, MorFic improves survival rate, speed, and OOD generalization better than any PPO variant or external baseline while remaining highly competitive in family robot transfer and transfers zero-shot to 2 real robots (Go1 & Go2).

Author Contributions

Prakhar Mishra: Conceptualization, methodology, implementation, experimental design, and analysis.

Amir Hossain Raj: Experimental implementation, robot deployment, evaluation, and discussions.

Xuesu Xiao and Dinesh Manocha: Supervision, technical guidance, and manuscript review.

References

[1] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), 2019.

[2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47), 2020.

[3] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2022.

[4] G. Pan, Q. Ben, Z. Yuan, G. Jiang, Y. Ji, S. Li, J. Pang, H. Liu, and H. Xu. Roboduet: Learning a cooperative policy for whole-body legged loco-manipulation. *IEEE Robotics and Automation Letters*, 2025. doi:10.1109/LRA.2025.3564564.

[5] S. Yang, Z. Fu, Z. Cao, G. Junde, P. Wensing, W. Zhang, and H. Chen. Multi-loco: Unifying multi-embodiment legged locomotion via reinforcement learning augmented diffusion. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2025.

[6] J. Tan, T. Zhang, E. Coumans, S. Ha, J. Geijtenbeek, and C. K. Liu. Sim-to-real: Learning agile locomotion for quadruped robots. In *Robotics: Science and Systems (RSS)*, 2018.

[7] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[8] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[9] G. Feng, H. Zhang, Z. Li, X. B. Peng, A. Kanazawa, and S. Levine. Genloco: Generalized locomotion controllers for quadrupedal robots. In *Conference on Robot Learning (CoRL)*, 2023.

[10] C. Sferrazza and L. Righetti. Learning generalizable locomotion for quadrupedal robots with body transformer. In *Robotics: Science and Systems (RSS)*, 2023.

[11] S. Shafiee, G. Bellegarda, and A. Ijspeert. Manyquadrupeds: Learning a single locomotion policy for diverse quadruped robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[12] B. Ai, L. Dai, N. Bohlinger, D. Li, T. Mu, Z. Wu, K. Fay, H. I. Christensen, J. Peters, and H. Su. Towards embodiment scaling laws in robot locomotion. *arXiv preprint arXiv:2505.05753*, 2025.

[13] N. Bohlinger, G. Czechmanowski, J. Peters, and D. Tateo. One policy to run them all: An end-to-end learning approach to multi-embodiment locomotion. In *Conference on Robot Learning (CoRL)*, 2024.

[14] M. Shafiee, G. Bellegarda, and A. Ijspeert. Manyquadrupeds: Learning a single locomotion policy for diverse quadruped robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3471–3477. IEEE, 2024.

[15] M. Liu, D. Pathak, and A. Agarwal. Locoformer: Generalist locomotion via long-context adaptation. *arXiv preprint arXiv:2509.23745*, 2025.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. In *International Conference on Learning Representations (ICLR)*, 2017.

[17] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, 2018.

[18] G. B. Margolis, G.-Z. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research*, 43(4):572–587, 2024.

[19] N. Bohlinger, G. Czechmanowski, M. Krupka, P. Kicki, K. Walas, J. Peters, and D. Tateo. One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion. *arXiv preprint arXiv:2409.06366*, 2024.

[20] S. Liu, D. Xu, and L. Pinto. Understanding and mitigating interference in multi-task reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2023.

[21] F. Doshi-Velez and G. Konidaris. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[22] J. Schulman et al. High-dimensional continuous control using generalized advantage estimation. In *ICLR*, 2016.

[23] T. Yu et al. Gradient surgery for multi-task learning. In *NeurIPS*, 2020.

[24] S. M. Jayakumar, W. M. Czarnecki, J. Menick, J. Schwarz, J. Rae, S. Osindero, Y. W. Teh, T. Harley, and R. Pascanu. Multiplicative interactions and where to find them. In *International conference on learning representations*, 2020.

[25] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[26] D. Ha, A. Dai, and Q. V. Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[27] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[28] G. B. Margolis and P. Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.

[29] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Nguyen, M. Macklin, G. Lukamowicz, Y. State, J. Liang, J. Zhu, et al. Isaac gym: High performance GPU-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

A Evaluation Metrics and Diagnostic Definitions

Normalized morphology distance. To analyze how transfer performance degrades with morphology shift, we have defined the normalized distance $d \in [0, 1]$ between target robot and training robot based on the 11-D morphology descriptor selected from Table I. We compute.

$$d(m_{\text{target}}, m_{\text{train}}) = \frac{1}{\sqrt{11}} \left\| \tilde{m}_{\text{target}} - \tilde{m}_{\text{train}} \right\|_2 . \tag{7}$$

This distance is used for transfer loss analysis and is not provided to the policy or critic during training (refer to Fig. 5).

Normalized Value Bias (NVB). NVB measures whether the critic systematically overestimates or underestimates returns, where close to 0 equals low systematic bias and higher value indicate higher bias and is given by:

$$\text{NVB}_\phi = \frac{\mathbb{E}_{(t,i)\in\mathcal{I}} \left[V_\phi(s_{t,i}, z_{m_i}) - \hat{G}_{t,i} \right]}{\sigma_G + \epsilon}. \tag{8}$$

Normalized Value Error (NVE). NVE measures the total critic prediction error, normalized by the return scale, lower value means better calibrated critic and is given by:

$$\text{NVE}_\phi = \frac{\sqrt{\mathbb{E}_{(t,i)\in\mathcal{I}} \left[\left(V_\phi(s_{t,i}, z_{m_i}) - \hat{G}_{t,i} \right)^2 \right]}}{\sigma_G + \epsilon}.. \tag{9}$$

Advantage Sign Agreement (ASA). ASA measures whether the critic-induced advantage has the same sign as the Monte-Carlo reference advantage, higher ASA is better and is given by:

$$\hat{A}^{\text{MC}}_{t,i} = \hat{G}_{t,i} - \frac{1}{|\mathcal{I}_\epsilon|} \sum_{(t',i')\in\mathcal{I}_\epsilon} \hat{G}_{t',i'}. \tag{10}$$

$$\hat{A}^{\phi}_{t,i} = \hat{G}_{t,i} - V_\phi(s_{t,i}, z_m). \tag{11}$$

$$\text{ASA}_\phi = \frac{1}{|\mathcal{I}_\epsilon|} \sum_{(t,i)\in\mathcal{I}_\epsilon} \mathbf{1} \left[\text{sign}\left(\hat{A}^{\phi}_{t,i} \right) = \text{sign}\left(\hat{A}^{\text{MC}}_{t,i} \right) \right]. \tag{12}$$

Sign Flip Rate. Sign flip rate measures how often the critic-induced advantage has the opposite sign from the MC reference advantage and lower rate is better and is given by:

$$\text{Flip}_\phi = \frac{1}{|\mathcal{I}_\epsilon|} \sum_{(t,i)\in\mathcal{I}_\epsilon} \mathbf{1} \left[\text{sign}\left(\hat{A}^{\phi}_{t,i} \right) \neq \text{sign}\left(\hat{A}^{\text{MC}}_{t,i} \right) \right]. \tag{13}$$

$$\text{Flip}_\phi = 1 - \text{ASA}_\phi. \tag{14}$$

Gradient Cosine. Gradient cosine measures whether the PPO actor update direction produced by the critic-induced advantage matches the actor update direction produced by the MC reference advantage, where close to 1 means critic gives almost the same actor update as MC return, so a value higher than 0 and closer to 1 is better and is given by:

$$g(\hat{A}^{\phi}) = \nabla_\theta \mathcal{L}_{\text{actor}} \left(\theta; \hat{A}^{\phi} \right), \qquad g(\hat{A}^{\text{MC}}) = \nabla_\theta \mathcal{L}_{\text{actor}} \left(\theta; \hat{A}^{\text{MC}} \right). \tag{15}$$

Table 5: **Morphology randomization parameters.** Each row reports one dimension of the 11-D morphology vector used by MorFiC. Values outside the training support are shown in bold with ↑ / ↓.

Parameter	Train min	Train max	Go2	Go1	A1	Cheetah	AlienGo	ANYmal	B1
Thigh mass	0.634	1.152	1.152	0.899	1.013	0.634	0.639	0.740	**4.300**↑
Calf mass	0.064	0.166	0.154	0.158	0.166	0.064	**0.207**↑	**0.337**↑	**1.050**↑
Hip mass	0.510	0.696	0.678	0.510	0.696	0.540	**1.000**↑	**0.740**↑	**2.150**↑
Foot mass	0.000	0.060	0.040	0.060	0.060	0.060	0.060	**0.250**↑	0.050
Base mass	3.300	6.921	6.921	4.800	6.000	3.300	**11.644**↑	**18.210**↑	**29.450**↑
Hip limit min	-1.100	-0.800	-1.047	-0.803	-0.803	**-1.600**↓	**-1.222**↓	**-0.720**↑	**-0.750**↑
Hip limit max	0.800	1.100	1.047	0.803	0.803	**1.600**↑	**1.222**↑	**0.720**↓	**0.750**↓
Thigh limit min	-1.650	-1.050	-1.571	**-1.047**↑	**-1.047**↑	**-2.600**↓	**-3.142**↓	-1.650	**-1.000**↑
Thigh limit max	3.490	4.050	3.491	**4.189**↑	**4.189**↑	**2.600**↓	**3.142**↓	4.050	3.500
Calf limit min	-2.720	-2.600	**-2.723**↓	-2.697	-2.697	-2.600	**-2.775**↓	-2.720	-2.600
Calf limit max	-0.950	-0.780	-0.838	-0.916	-0.916	-0.800	**-0.646**↑	-0.780	**-0.600**↑

$$\mathrm{CosGrad}_\phi = \frac{g(\hat{A}^\phi)^\top g(\hat{A}^{\mathrm{MC}})}{\left\|g(\hat{A}^\phi)\right\|_2 \left\|g(\hat{A}^{\mathrm{MC}})\right\|_2}. \tag{16}$$

Validity of survival-based metrics. Survival rate alone is not sufficient to indicate successful locomotion under cross-morphology transfer. In several in & out-of-distribution cases, a policy can avoid early termination while producing negligible forward motion, unstable stepping, collapsed postures, or physically invalid behaviors such as dragging or non-propulsive oscillations. And in some cases the policy do have valid motions while also producing behaviors like upside down leg movement, forward falling, disabled rear legs etc in some fraction of robot while the rate remains 1.00 or 100%. We therefore interpret survival jointly with achieved forward velocity, tracking ratio, and rollout validity. In the result tables, high survival with near-zero forward velocity is treated as a failure to transfer rather than successful locomotion.

This distinction is especially important for distant morphologies such as B1 and ANYmal-C, where policies may remain non-terminated but fail to generate a valid gait. High survival with near-zero forward velocity is not considered successful transfer; survival is interpreted jointly with achieved speed and rollout validity.

B Morphology Descriptor and Randomization

MorFiC uses an 11-D morphology vector containing link masses and joint-limit parameters. During training, each environment samples a morphology vector from the support in Table 5; this vector is applied in simulation and kept fixed for the episode. The same vector is encoded into the latent z_m used by the actor and critic. Table 5 also reports the descriptors of all evaluation robots, with out-of-support values marked by ↑ or ↓. Go1 and A1 remain largely near the training support, while AlienGo, ANYmal-C, and especially B1 introduce larger shifts in base mass, limb masses, and joint limits.

B.1 Morphology Randomization

The randomization range was chosen to cover moderate morphology variation around the Go2 source template while preserving stable PPO training. This creates a controlled zero-shot setting: MorFiC is not trained on the target robot models, but observes morphology variations that partially cover nearby quadrupeds. B1 is the most distant target, with substantially larger link and base masses than the training support.

B.2 Effect of Expanding Morphology Randomization

We also tested whether widening the morphology range improves transfer to distant robots such as B1. Expanding the support to include much heavier morphologies made the policy more conserva-

Table 6: **Effect of expanded morphology ranges and FiLM conditioning.** Each entry reports $v_x^{\max} \pm \sigma/S$, where S denotes survival rate. Bold indicates the higher value for each metric independently; ties are bolded for both methods.

Target	**Expanded range** $v_x^{\max} \pm \sigma/S$	**FiLM** $v_x^{\max} \pm \sigma/S$
Go2	$1.9013 \pm 0.3510/0.8531$	$\mathbf{2.0415 \pm 0.3670/0.9844}$
Go1	$1.9208 \pm 0.3087/\mathbf{0.7656}$	$\mathbf{2.0649 \pm 0.4552}/0.5469$
A1	$1.8241 \pm 0.2322/\mathbf{0.7596}$	$\mathbf{2.1404 \pm 0.3608}/0.5156$
AlienGo	$1.5499 \pm 0.5078/\mathbf{1.0000}$	$\mathbf{1.9812 \pm 0.5146/1.0000}$
B1	$0.2001 \pm 0.0215/\mathbf{1.0000}$	$\mathbf{0.2109 \pm 0.0768/1.0000}$
Cheetah	$1.2362 \pm 0.4805/0.3348$	$\mathbf{1.3922 \pm 0.6902/0.4219}$
ANYmal-C	$\mathbf{0.3079 \pm 0.1803/1.0000}$	$0.2505 \pm 0.2147/0.9688$

Table 7: Observation streams used for policy and value learning.

Stream	Signals	Dim
o_t	$[\, v,\ \omega,\ g_{\text{proj}},\ c,\ q - q_0,\ \dot{q},\ a_{t-1} \,]$	48
h_t	$\{o_{t-k}\}_{k=1}^{15}$	15×48
p_t	$[\mu_{\text{fric}},\ e,\ m_{\text{base}},\ \Delta c,\ \mu_{\text{joint}}]$	18
z_m	morphology encoder output	$d = 11$

tive, reducing forward tracking without resolving the B1 failure mode (see Table 6). This suggests that distant morphology transfer is not solved by naively widening randomization alone, and may require structured morphology curricula or limited multi-source training.

C Policy Architecture, Observation Streams, and PPO Hyperparameters

C.1 Observation Streams

Table 7 summarizes the observation streams used by MorFiC. The policy receives a history of proprioceptive observations, including base velocity, angular velocity, projected gravity, command, joint positions, joint velocities, and previous actions. The privileged stream contains simulator-side physical parameters used during training, while the morphology stream contains the 11-D descriptor encoded into z_m. All methods use the same observation history and privileged information; they differ only in how morphology information is injected into the actor–critic architecture.

C.2 Actor-Critic and Morphology Encoder Architecture

Table 8 & 9 reports the architecture used for MorFiC. We use separate actor and critic MLP trunks with ELU activations. The morphology vector is encoded using a lightweight MLP and used to condition the critic through residual scaled FiLM modulation at each hidden layer. The final FiLM layer is initialized to zero, so training starts close to an unconditioned critic and gradually learns morphology-dependent modulation.

C.3 PPO Hyperparameters

Table 10 lists the PPO hyperparameters used across all baseline and MorFiC experiments. All variants share the same rollout setup, learning rate schedule, clipping parameters, entropy coefficient, value loss coefficient, and optimization settings. Thus, performance differences are attributable to the morphology-conditioning mechanism rather than changes in PPO training.

Table 8: Policy, critic, and morphology-conditioning architecture used in MorFiC. The PPO optimizer and rollout hyperparameters are shared across all methods and reported separately in Table 10.

Module	Specification
Actor MLP	$[\text{obs history}, \hat{z}_{\text{priv}}, z_m] \to [512, 256, 128] \to a$, ELU
Critic trunk	$[\text{obs history}, z_{\text{priv}}] \to [512, 256, 128] \to V$, ELU
Critic trunk (FiLM scaled)	$[\text{obs history}, z_{\text{priv}}] \to [1024, 512, 256] \to V$, ELU
Initial action std.	$\sigma_0 = 1.0$
Adaptation module	obs history $\to [256, 128] \to \hat{z}_{\text{priv}}$, ELU
Student latent dropout	$p = 0.2$ during training
Morphology encoder	$d_m \to 128 \to 64$, ELU
Main critic conditioning	Residual scaled FiLM at each critic hidden layer
FiLM generator	$64 \to 128 \to 2H$, ELU
FiLM update	$h \leftarrow h(1 + \gamma) + \beta$
FiLM scale	0.1
FiLM initialization	Last FiLM layer weights and biases initialized to zero
Decoder	Disabled

Table 9: Morphology-conditioning variants evaluated in the ablation study. All variants use the same PPO settings and base actor–critic widths as Table 8.

Variant	Critic conditioning	Key detail
Vanilla PPO	None	Critic does not receive morphology information
Concat-z	Direct concatenation	z_m is appended to the critic input
FiLM	Feature-wise modulation	$h \leftarrow h(1 + \gamma) + \beta$
Scaled FiLM	Residual FiLM with scale	FiLM outputs multiplied by 0.1
AdaLN	LayerNorm + FiLM	$h \leftarrow \text{LN}(h)$ before FiLM modulation
HyperNet	Low-rank residual adapter	$h \leftarrow Wh + B_z A_z h$, rank 16, scale 0.1

D Reward Terms and Weights

We shape our MorFiC rewards as a weighted sum of (i) command tracking terms for linear and angular velocities using exponential, (ii) stability rewards that encourage upright posture and dynamic balance via a center-of-pressure (CoP) proximity reward, (iii) gait/contact shaping terms that match desired contact schedules by penalizing unwanted swing forces and stance-phase foot velocities and a detailed definition and weights are given in Table V.

D.1 Reward structure

Our reward structure is divided into three main categories: Task/Tracking, Penalties and stability shaping rewards, and the total reward sum is given by (17).

$$r_t \;=\; \sum_{i \in T} w_i \, r_t^{(i)} \;-\; \sum_{j \in P} w_j \, p_t^{(j)} \;+\; \sum_{k \in S} w_k \, s_t^{(k)}. \tag{17}$$

Task/Tracking rewards encourage the robot to complete the commanded locomotion objective. Specifically, we tracked the commanded linear velocity and the angular velocity using exponential rewards. (Table 11)

Penalties are assigned as a way to discourage unsafe movement, which could lead to collisions, bad gait, or physically collapse(like jerks, jerky motions, base collision, joint limit violation, etc). For example, we penalize body collisions

$$p_{\text{col}} \;=\; \sum_{b \in B} 1.(\|\mathbf{f}_b\| > f_{\min}) \,, \tag{18}$$

joint limit violations and is given by;

$$p_{\text{lim}} \;=\; \sum_i \Big(\max(0,\; q_i - q_i^{\max}) + \max\big(0,\; q_i^{\min} - q_i\big) \Big). \tag{19}$$

Table 10: PPO hyperparameters used in all baseline and MorFiC experiments.

PPO setting	Value
Discount γ	0.99
GAE λ	0.95
Clip ϵ	0.2
Entropy coeff.	0.01
Value loss coeff.	1.0
Clipped value loss	True
Learning rate	1×10^{-3}
Schedule	adaptive (KL-targeted)
Target KL	0.01
Epochs / update	5
Mini-batches / update	4
Grad norm clip	1.0

These help the robot in not deviating from the desired locomotion gait.

Stability Shaping. Motivated by our repeated failure on unseen morphologies, we introduce 3 lightweight stability rewards namely upright stability, CoP stability, and feet-contact stability to reduce catastrophic tipping and improve recovery.

1. **Upright stability.** Encourages recovery from large pitch/tilt by penalizing deviation from projected gravity.

$$s_{\text{upright}} = \exp\left(-\frac{\|\mathbf{g}_{xy}\|^2}{\sin^2(\theta_0)}\right). \tag{20}$$

2. **Center-of-pressure (CoP) stability.** Promotes balanced load distribution across the support polygon.

$$s_{\text{CoP}} = \exp\left(-\frac{\|\text{CoP} - \text{center}\|}{\tau}\right) \cdot 1(n_{\text{contact}} \geq 2). \tag{21}$$

3. **Feet-contact / stance-width stability.** Penalize overly narrow instances during unstable locomotion (caused by large tilt or rapid movement when unstable). We gate this using eqns. 22, 23.

$$\text{gate} = 1\Big(\|\mathbf{g}_{xy}\|^2 > \eta \ \vee \ \|\mathbf{v}_{xy}\| > v_{\text{th}}\Big), \tag{22}$$

and define

$$s_{\text{width}} = \text{gate} \cdot \exp\left(-\frac{\max(0,\, w_{\min} - w)}{\sigma_w}\right) + (1 - \text{gate}). \tag{23}$$

In addition to these components, we include a small set of rewards adapted from earlier work [28]. **All baselines were trained with the same reward**, so as to clearly isolate the performance difference caused by the poor approximation of the value function.

E Training Details

E.1 Random Seeds and Reporting

Unless stated otherwise, each internal PPO and MorFiC variant is trained with $n = 3$ independent random seeds. For simulation evaluation, each checkpoint is evaluated using the same target robots, command settings, rollout horizon, and number of parallel environments. Reported values are averaged across evaluation rollouts; standard deviations are reported over rollout-level measurements. For external baselines, we use the available released checkpoints or reproduced checkpoints as described in the corresponding baseline setup.

Table 11: **Reward terms** used in our MorFiC locomotion objective. w_k are set in the reward configuration; fixed scalings appearing in code (e.g., the factor 4 in r_{lin}) are shown explicitly.

Term	Definition	Weight
r_{lin}	$4\exp\left(-\frac{\|\mathbf{v}_{xy}^{cmd}-\mathbf{v}_{xy}\|^2}{\sigma_{\text{lin}}}\right)$	w_{lin}
r_{yaw}	$\exp\left(-\frac{(\omega_z^{cmd}-\omega_z)^2}{\sigma_{\text{yaw}}}\right)$	w_{yaw}
p_{v_z}	$(v_z)^2$	w_{v_z}
$p_{\omega_{xy}}$	$\|\boldsymbol{\omega}_{xy}\|^2$	w_{ω}
p_{ori}	$\|\hat{\mathbf{g}}_{xy}\|^2$	w_{ori}
p_{τ}	$\|\boldsymbol{\tau}\|^2$	w_{τ}
$p_{\ddot{q}}$	$\left\|\frac{\dot{q}_{t-1}-\dot{q}_t}{\Delta t}\right\|^2$	$w_{\ddot{q}}$
$p_{\Delta a}$	$\|a_{t-1}-a_t\|^2$	$w_{\Delta a}$
p_{coll}	$\sum_{b\in\mathcal{B}}\mathbb{I}(\|\mathbf{f}_b\|>0.1)$	w_{coll}
p_{jl}	$\sum_i\left[(q_i-q_i^{\min})_- + (q_i-q_i^{\max})_+\right]$	w_{jl}
r_{upright}	$\exp\left(-\frac{\|\hat{\mathbf{g}}_{xy}\|^2}{\sin^2(\theta_0)}\right)$, θ_0=0.3	w_{upright}
r_{cop}	$\mathbb{I}(n_c\geq 2)\exp\left(-\frac{\|\text{CoP}-\text{center}\|}{\tau}\right)$, τ=0.10	w_{cop}
r_{width}	*(yaw-frame)* $\exp\left(-\frac{(w_{\min}-w)_+}{\sigma_w}\right)$ (gated)	w_{width}
r_{gaitF}	$-\frac{1}{4}\sum_i(1-d_i)\left(1-\exp(-f_i^2/\sigma_F)\right)$	w_{gaitF}
r_{gaitV}	$-\frac{1}{4}\sum_i d_i\left(1-\exp(-\|\dot{\mathbf{x}}_i\|^2/\sigma_V)\right)$	w_{gaitV}
p_{slip}	$\sum_i\mathbb{I}(c_i)\|\dot{\mathbf{x}}_{i,xy}\|^2$	w_{slip}
p_{impact}	$\sum_i\mathbb{I}(c_i)\,\text{clip}(\dot{z}_{i,t-1},-100,0)^2$	w_{impact}
p_f	$\sum_i\left(\|\mathbf{f}_i\|-f_{\max}\right)_+$	w_f
p_q	$\|q-q^0\|^2$	w_q
$p_{\dot{q}}$	$\|\dot{q}\|^2$	$w_{\dot{q}}$
r_{jump}	$-(z-(h^{cmd}+h_0))^2$	w_{jump}
r_{raibert}	$\|\mathbf{x}_{foot}^{des}-\mathbf{x}_{foot}\|^2$ (body yaw frame)	w_{raibert}

E.2 Simulation and Training Setup

All policies are trained in IsaacGym Simulator [29] with 4096 parallel environments with a time-step of dt = 0.005s; episode lasts roughly 20s. A command curriculum and domain randomization (joint friction, motor delays, sensor noise) are applied identically across all baselines. We used the same PPO hyperparameters, reward functions, command curriculum, morphology ranges, and network sizes across all baselines variants (See Appendix). And finally, each policy is trained a total of 400 million time steps utilizing a Nvidia RTX 4090 laptop-based GPU, requiring roughly 2 hours of wall clock time. We have adapted our code mainly from open-source repositories [28, 3].

E.3 Domain Randomization

We apply domain randomization during training to improve robustness and sim-to-real transfer. Randomization is applied periodically with the interval 10 s (rigid-body properties may be randomized after initialization). We randomize the properties of contact with the ground by sampling the friction coefficient uniformly from $[0.5,\ 1.25]$. We inject external disturbances by pushing the robot at fixed intervals of 15 s with a maximum planar push velocity of 1.0 m/s. To model actuation/communication delay, we randomize action latency using a history buffer of up to 6 timesteps (randomized lag timesteps enabled).

E.4 Evaluation Fairness

In order to maintain the fairness of the evaluation, we kept all the necessary parameters like morphological and domain randomization parameters for the training robot, including all the PPO hyperparameters, command curriculum ranges and reward functions, latent dimensions the same across all the variants.

Table 12: **External baselines vs. MorFiC at higher commanded speed.** Each entry reports $v_x^{\max} \pm \sigma/S$ (m/s; survival). All methods are evaluated at $v_x^{\mathrm{cmd}} = 2.5$ m/s, except GenLoco, which does not take a commanded forward velocity as input. Bold indicates the best value for each metric independently.

Target	URMA $v_x^{\max} \pm \sigma/S$	BodyTrans. $v_x^{\max} \pm \sigma/S$	GenLoco $v_x^{\max} \pm \sigma/S$	Embod. $v_x^{\max} \pm \sigma/S$	MorFiC-F $v_x^{\max} \pm \sigma/S$
Go2	2.389±0.405/**1.000**	0.313±0.232/0.625	0.592±0.047/0.922	1.072±0.092/0.859	**2.405**±**0.405**/0.859
Go1	**2.514**±**0.430**/**1.000**	0.206±0.151/**1.000**	0.985±0.007/**1.000**	2.343±0.131/0.797	1.972±0.442/0.516
A1	**2.230**±**0.336**/**1.000**	0.374±0.183/0.515	0.919±0.011/**1.000**	2.143±0.116/0.719	1.595±0.467/0.156
AlienGo	−/0.000	0.358±0.315/0.203	**0.964**±**0.032**/**1.000**	0.452±0.085/0.000	0.832±0.542/**1.000**
B1	−/0.000	0.080±0.140/0.640	0.00/0.000	**3.062**±**0.214**/**0.797**	0.121±0.060/0.000
Cheetah	−/0.000	−/**1.000**	0.729±0.061/0.969	−0.118±0.091/0.000	**1.128**±**0.986**/0.234
ANYmal-C	**2.159**±**0.354**/**1.000**	0.014±0.081/0.843	0.708±0.005/**1.000**	0.359±0.150/0.453	0.303±0.422/0.984

Table 13: **Extended critic calibration diagnostics across PPO variants.** Each entry reports |NVB|/NVE/ASA, where NVB is the normalized value bias, NVE is the normalized value error, and ASA is the advantage-sign agreement. Lower |NVB| and NVE are better, while higher ASA is better. Best value for each metric within a target row is bolded.

Target	Vanilla PPO |NVB|/NVE/ASA	Concat-z Actor only	Concat-z Actor+Critic	FiLM |NVB|/NVE/ASA
Go2	1.944/1.981/0.610	**0.177**/1.168/0.689	0.535/**0.960**/0.670	1.768/1.943/**0.771**
Go1	1.591/1.817/0.612	**0.068**/**1.096**/0.662	1.623/1.834/0.674	1.375/1.653/**0.681**
A1	2.027/2.203/0.603	2.033/16.463/**0.754**	**0.199**/**0.936**/0.701	1.482/1.813/0.675
AlienGo	0.070/1.454/0.612	0.468/1.209/0.569	0.153/1.117/0.607	**0.048**/**0.835**/**0.706**
B1	0.182/1.231/0.618	0.358/1.187/0.570	**0.040**/1.134/**0.624**	0.645/**1.279**/0.618
Cheetah	0.362/1.869/0.428	**0.220**/1.086/0.638	1.085/1.344/0.730	0.464/**0.947**/**0.768**
ANYmal-C	**0.091**/1.229/0.514	0.411/**1.227**/0.528	0.385/1.307/**0.620**	0.120/1.229/0.578

F Additional Transfer Results

F.1 Additional Results MorFiC Vs. external baselines

Table 12 reports an additional evaluation at a higher commanded forward speed, $v_x^{cmd} = 2.5$ m/s. This setting tests whether the transfer trends from the main 2.0 m/s evaluation persist under a more aggressive command. MorFiC-F remains competitive on the source and nearby morphologies, while performance degrades on more distant targets such as B1 and ANYmal-C. As in the main results, survival is interpreted jointly with achieved forward velocity, since high survival with negligible motion does not indicate successful transfer.

G Extended Critic Calibration Diagnostics

The main paper reports explained variance as the primary critic-calibration metric. Here, we provide additional diagnostics that capture complementary aspects of value-function quality. Specifically, we report |NVB|, NVE, and ASA for both internal PPO variants and external baselines. Lower |NVB| indicates smaller systematic value bias, lower NVE indicates smaller value prediction error, and higher ASA indicates better agreement between critic-induced and Monte-Carlo advantage signs.

Table 13 compares the internal PPO variants. The results show that additive morphology conditioning does not consistently resolve critic error across morphologies. FiLM improves the strongest OOD cases such as AlienGo and Cheetah, where it achieves lower NVE and higher ASA than the vanilla and concatenation-based variants. This supports the claim that multiplicative critic conditioning improves the quality of the value baseline under morphology shift.

Table 14 reports the same diagnostics for external baselines. Although some external methods achieve reasonable locomotion performance on certain robots, their value predictions often remain poorly calibrated, with large |NVB| and NVE. MorFiC-F generally gives substantially lower value

Table 14: **Extended critic calibration diagnostics for external baselines.** Each entry reports |NVB|/NVE/ASA, where NVB is normalized value bias, NVE is normalized value error, and ASA is advantage-sign agreement. Lower |NVB| and NVE are better, while higher ASA is better. Best value for each metric within a target row is bolded.

Target	Embod. Scale |NVB|/NVE/ASA	URMA |NVB|/NVE/ASA	BodyTrans. |NVB|/NVE/ASA	GenLoco |NVB|/NVE/ASA	MorFiC-F |NVB|/NVE/ASA
Go2	3.437/3.646/0.657	3.005/3.178/**0.702**	3.319/3.492/0.652	5.458/5.674/0.561	**1.768**/**1.943**/0.671
Go1	2.569/2.772/0.607	3.170/3.333/0.692	4.771/4.983/**0.724**	3.402/3.594/0.623	**1.375**/**1.653**/0.681
A1	2.368/2.585/0.599	3.390/3.551/**0.686**	3.853/4.032/0.686	2.956/3.128/0.612	**1.482**/**1.813**/0.675
AlienGo	–	–	3.104/3.476/0.661	1.170/1.522/**0.709**	**0.098**/**0.835**/0.706
B1	3.330/3.590/0.657	–	3.571/3.781/**0.694**	7.428/7.729/0.667	**0.645**/**1.279**/0.618
Cheetah	–	–	–	5.294/5.511/0.594	**0.464**/**0.947**/**0.768**
ANYmal-C	3.589/3.812/0.667	2.538/2.740/0.728	3.368/3.529/0.674	1.107/2.324/**0.733**	**0.120**/**1.229**/0.578

Table 15: **Fixed-rollout critic diagnostic across target morphologies.** Each entry reports Flip/Cos., where Flip is the sign-flip rate against the MC reference advantage (lower is better) and Cos. is the actor-gradient cosine with the MC-reference gradient (higher is better). Bold indicates the best learned method for each metric independently; the MC row is the reference and is not considered for bolding.

Src.	Go2		Go1		A1		AlienGo		B1		Cheetah		ANYmal-C	
	Flip	Cos.	Flip	Cos.	Flip	Cos.	Flip	Cos.	Flip	Cos.	Flip	Cos.	Flip	Cos.
MC	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
MorFiC	**0.227**	0.880	0.187	**0.972**	**0.239**	**0.785**	**0.255**	**0.366**	0.272	-0.182	**0.209**	**0.679**	**0.402**	-0.149
Concat	0.307	0.720	**0.151**	0.830	0.344	0.427	0.329	0.004	0.269	0.106	0.236	0.575	0.479	-0.419
Act-only	0.524	**0.890**	0.502	0.924	0.511	0.717	0.318	0.342	0.565	**0.647**	0.265	0.644	0.470	**-0.044**
Vanilla PPO	0.235	0.865	0.229	0.684	0.260	0.653	0.322	-0.113	**0.176**	0.601	0.253	0.264	0.461	-0.074

error, especially on OOD targets such as AlienGo, B1, Cheetah, and ANYmal-C. ASA is more mixed, so we interpret it jointly with |NVB| and NVE rather than as a standalone measure.

H Fixed-Rollout Critic Causality Analysis

To test whether morphology conditioning improves the actor through better critic estimates, we run a fixed-rollout diagnostic. For each trained policy, we freeze the sampled trajectories and compare the critic-induced advantage signs against an MC-return reference. We report the sign-flip rate, where lower values indicate fewer wrong-sign policy updates, and the gradient cosine against the MC-reference actor gradient, where higher values indicate better update alignment.

MorFiC generally yields lower flip rates and higher gradient alignment on near-support targets such as Go2, A1, AlienGo, and Cheetah, suggesting that FiLM conditioning improves the critic signal used by PPO. However, on distant morphologies such as B1 and ANYmal-C, the cosine can become negative, exposing a failure mode where the learned critic induces updates that are misaligned with the MC reference. This supports our diagnosis that cross-embodiment failure is not only an actor-transfer problem, but also a critic-miscalibration problem. (see Table 15)

I Training Compute Methodology

We report transition-normalized GPU-hours because the baselines use different training budgets. For each method, we profile the released training code on our local hardware for 100 iterations, measure the average wall-clock time per iteration, and compute the transitions per iteration from the method configuration. We then extrapolate total GPU-hours to the reported transition budget:

$$\text{GPU-hours} = \frac{T}{N_{\text{env}} H} \cdot \frac{\tau_{\text{iter}}}{3600},$$

where T is the total transition budget, N_{env} is the number of parallel environments, H is the rollout horizon, and τ_{iter} is the measured time per iteration. We also report GPU-hours normalized to

Table 16: **Training compute comparison.** We report total transitions, approximate PPO iterations, wall-clock seconds per iteration, total single-GPU-equivalent training hours, and GPU-hours normalized to a 400M-transition budget. Lower GPU-hours@400M indicates lower compute cost per transition. Expert-pool methods are separated because they train many specialized policies rather than one generalist policy.

Method	**Transitions**	**Iters.**	**Sec/iter**	**GPU-h**	**GPU-h @400M**	**Notes**
MorFiC	400M	4,000	**1.98–2.70**	**2.2–3.0**	**2.2–3.0**	Single generalist policy; local run
BodyTransformer	400M	8,000	17.25	38.3	39.0	Reproduced `model_8000`
URMA	1.600B	3,064	366–392	311–333	77.8–83.3	Original training budget
GenLoco	200M	1,526	125.8	53.3	106.6	200M setting
GenLoco	800M	6,104	125.8	213.5	106.8	800M setting
Expert-pool / selected-expert methods						
Embod. Scaling, Gendog8 expert	1.720B	17,499	0.94	4.5	1.05	One selected released expert
Embod. Scaling, full Gendog pool	570.6B	$\sim$5.80M	0.94	1507	1.06	332 experts; $\sim$5 TB storage

400M transitions for fair per-transition comparison. For BodyTransformer, we reproduce training to approximately the MorFiC budget; for larger-budget methods, we extrapolate using their reported transition counts.

J Real-Robot Rollout Diagnostics

To examine whether the learned critic remains meaningful during real deployment, we record reward and value trajectories from Go1 hardware rollouts. Figure 5 shows that MorFiC produces smoother value estimates that follow the trend of the discounted return despite noisy instantaneous rewards. In contrast, the non-MorFiC critic exhibits higher variance and less stable predictions. This suggests that morphology-conditioned value estimation can improve critic stability beyond simulation and remains useful under real-world rollout noise.

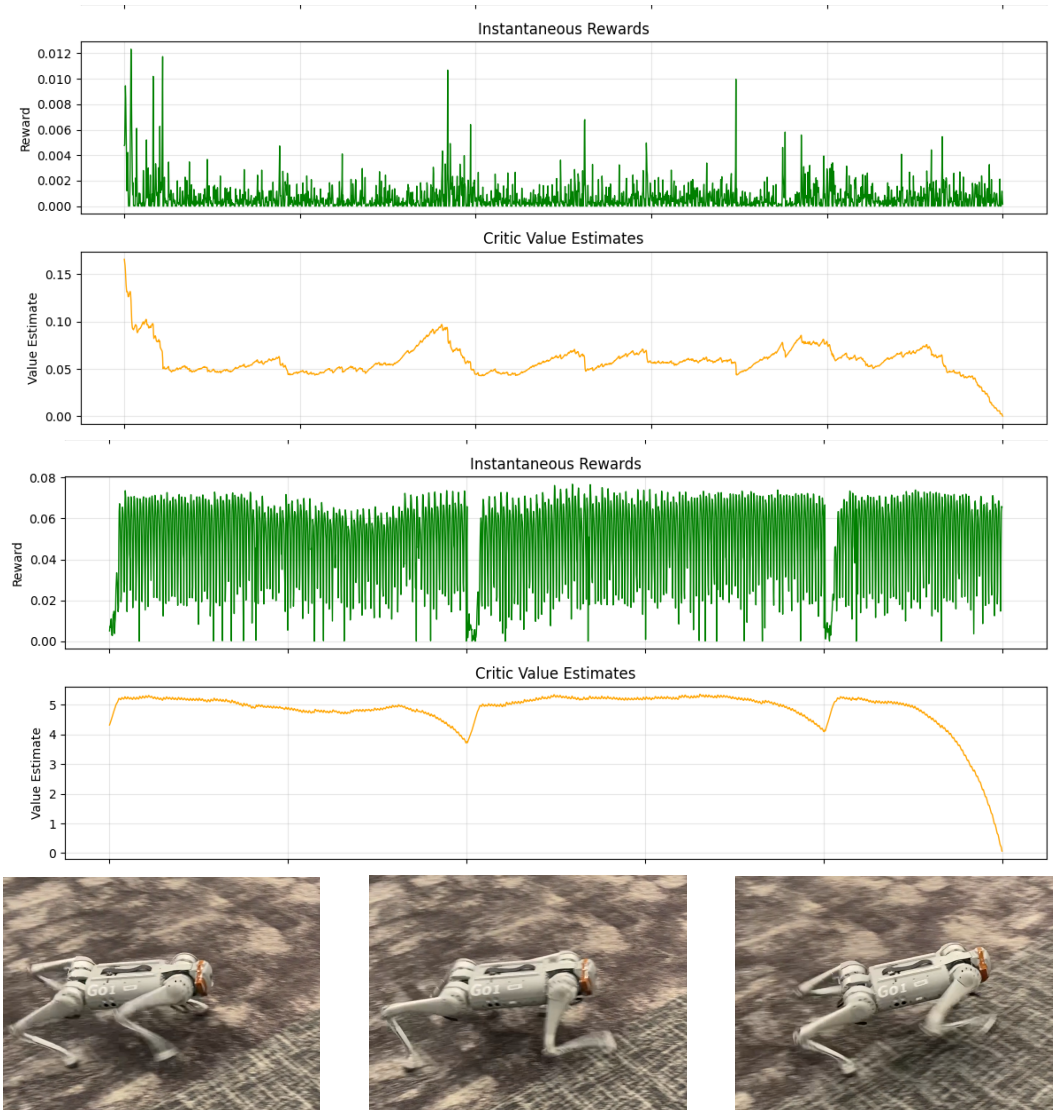

Figure 5: Above figure shows representative reward and value trajectories during deployment on the Go1 robot. The MorFiC produces smooth value estimates (orange) (second row) that track the expected discounted returns from noisy instantaneous rewards (green). While non-MorFic value estimates are noisy (first row), MorFiC's morphology-conditioned critic maintains stable value predictions even on unseen morphologies.