

Cross-Embodiment Transfer of Local Navigation Skills via Configuration Randomization

Sakura Ishikawa • Kenji Watanabe • Daichi Mori • Aiko Tanaka

Legged Systems Group, The University of Tokyo, Tokyo, Japan

PUBLISHED

August 12, 2026

REGISTERED

August 13, 2026

RECORD TYPE

Preprint

ABSTRACT

Local navigation skills learned for one quadruped transfer poorly to another, because leg length, payload, and sensor placement change what the robot perceives and which motions it can execute. This paper extends configuration randomization from low-level locomotion to the skill level. A single local navigation policy is trained in simulation with the AgniNav configuration descriptor, augmented with leg length and payload mass, as an explicit input, while the configuration parameters are sampled from the envelope spanned by a robot fleet. The descriptor modulates the policy through feature-wise affine conditioning and scales its velocity limits. Each new embodiment is adapted with a few hundred rollout episodes by a fine-tune that updates only the conditioning and output layers and is anchored to the pretrained policy by a KL penalty. In Isaac Lab, with a fleet of four quadruped morphologies (leg lengths 0.42 to 0.58 m, payloads 0 to 12 kg, sensor heights 0.35 to 0.55 m), 20 navigation scenarios, and 5 seeds, the method reaches 82.6 percent success on held-out morphologies outside the training support, compared with 73.1 percent for morphology-specific training and 77.8 percent for per-morphology fine-tuning from scratch, and reaches 90 percent of its final success after 420 fine-tune episodes on average, compared with 481 and 455. Ablations identify the configuration descriptor and payload randomization as the most consequential components. The gains come with a smaller clearance margin, and success drops to 74.1 percent when leg length is extrapolated by 22 percent.

INDEX TERMS cross-embodiment transfer, configuration randomization, legged robot navigation, local navigation, reinforcement learning, quadruped robots

KAGG NOTE Treating the robot configuration as an explicit, randomized input to a learned local navigation skill makes adaptation to a new quadruped body a short anchored fine-tune, but it transfers geometry and velocity limits more reliably than contact dynamics.

01 Introduction

Reinforcement learning in massively parallel simulation has made learned locomotion and local navigation practical on quadruped robots. Policies trained in simulation now traverse rough terrain, stairs, and cluttered spaces on physical hardware [1, 2, 3, 4], and recent work has extended the learned layer upward from velocity tracking to local navigation, where the policy chooses its own motion toward a goal while avoiding obstacles [5, 6, 7]. These navigation skills, however, are almost always trained for one body. A skill trained for a robot with short legs and no payload encodes, implicitly, which gaps that robot fits through, how fast it can turn, how far it needs to brake, and how its sensor sees obstacles from its own mounting height. When the same skill is deployed on a robot with longer legs, a heavier payload, or a raised sensor, each of these implicit assumptions can fail at once.

The obvious remedy, retraining per morphology, scales poorly for a fleet. Laboratories and operators increasingly maintain several variants of the same platform family: units with different leg kinematics, units carrying inspection payloads of different mass, and units with sensors remounted for a task. Each variant changes the feasible motion set of the navigation problem, so a morphology-specific skill must be re-derived whenever the fleet changes, and the data needed for it grows with the number of variants rather than with the difficulty of the task.

Two lines of prior work address parts of this problem. Configuration-driven planners such as AgniNav [8] make the robot description explicit: a single local planning stack operates on an abstract robot configuration (body geometry, sensors, and velocity limits) and therefore runs across embodiments without retraining. Domain and dynamics randomization [9, 10, 11], in turn, trains a learned controller over a distribution of physical parameters so that it is robust to the parameters it will meet at deployment, and morphology-randomized locomotion controllers have shown that one policy can walk on many quadruped bodies [12, 13]. What remains open is the combination at the *skill* level: a learned local navigation policy that, like a configuration-driven planner, is told what body it is operating, and that, like a randomized controller, has been trained across the full range of bodies it may be asked to operate, including bodies whose locomotion dynamics differ substantially because of mass and leg length.

This paper studies that combination. A single local navigation policy receives the AgniNav configuration descriptor [8], augmented with leg length and payload mass, as an explicit input, and is trained with configuration parameters randomized over the envelope spanned by a fleet of four quadruped morphologies. For each new embodiment, the policy is then fine-tuned with a few hundred rollout episodes. The evaluation targets the regime where the problem is hardest: held-out mor-

phologies that lie outside the training support in leg length and payload.

The contributions are as follows.

1. A formulation of cross-embodiment local navigation as learning a configuration-indexed family of policies, with explicit definitions of the configuration envelope, the held-out transfer set, and transfer metrics for success, clearance, and fine-tune sample efficiency (Section 3).
2. A configuration-conditioned navigation policy that consumes an extended AgniNav descriptor through feature-wise affine (FiLM) conditioning [14], issues velocity commands in a configuration-scaled action space derived from Froude similarity, and runs above a configuration-conditioned locomotion controller with a footprint-aware velocity governor (Section 4).
3. Skill-level configuration randomization over leg length, payload mass, and sensor height, combined with an anchored few-shot fine-tune that updates only the conditioning and output layers under a KL penalty to the pretrained policy (Section 4).
4. An evaluation in Isaac Lab on held-out morphologies outside the training support, with 20 navigation scenarios and 5 seeds, comparing against morphology-specific training, zero-shot configuration randomization, and per-morphology fine-tuning from scratch, together with an ablation over the descriptor and randomization axes and an analysis of recorded failure cases (Section 5).

02 Related Work

2.1 Learned locomotion and local navigation for legged robots

Hwangbo et al. [1] showed that actuator networks and simulation-trained policies could transfer agile locomotion to the ANYmal quadruped. Lee et al. [2] trained a blind proprioceptive controller with a teacher-student scheme that generalized to unstructured natural terrain, and Miki et al. [3] fused exteroceptive height maps with proprioception through an attention-like belief encoder so that the controller could discount unreliable perception. Rudin et al. [4] demonstrated that thousands of parallel environments on a GPU reduce locomotion training to minutes, which is the training regime adopted here through Isaac Lab. Above the locomotion level, Hoeller et al. [5] learned a compact state representation and a navigation policy for cluttered, dynamic environments; Rudin et al. [6] learned locomotion and local navigation end-to-end, so that a single policy reaches position goals over obstacles that a velocity-tracking interface could not express; and ANYmal Parkour [7] composed perception, a navigation module, and a library of locomotion skills into agile navigation. Robot Parkour Learning [15] and Extreme Parkour [16] further showed that learned policies can exploit the full capability of a specific body. All of these systems are trained for one robot. Their navigation competence is entangled with that robot's dimensions and dynamics, which is precisely what the present work aims to factor out. The hierarchical structure used here, a navigation policy issuing velocity commands to a learned locomotion controller, follows the modular designs of [5, 7] rather than the end-to-end design of [6], because a velocity interface lets the configuration-dependent velocity limits be imposed explicitly.

Truong et al. [17] are closest in spirit at the skill level. They trained navigation policies for legged robots in simulation with randomized dynamics and a learned robot embedding inferred from history, so that the policy could adapt to the dynamics of the deployed robot. That embedding is latent and inferred online; the present work instead conditions on an explicit, measured configuration descriptor that includes geometric quantities (footprint and sensor height) which a dynamics embedding does not capture, and it addresses explicit morphological change rather than dynamics perturbation of a fixed body.

2.2 Domain, dynamics, and morphology randomization

Tobin et al. [9] introduced domain randomization for visual sim-to-real transfer, and Peng et al. [10] randomized physical dynamics (masses, friction, damping, latency) so that a recurrent policy trained in simulation could control a real manipulator. Automatic domain randomization [11] expanded the randomization range progressively as performance improved, showing that the distribution itself can be a training curriculum. Yu et al. [18] trained a universal policy conditioned on explicit dynamics parameters together with an online system identification module, a design that anticipates the explicit conditioning used here; the difference is that the parameters of interest in this work (leg length, footprint, sensor height) are measured rather than identified, and they change the geometry of the task, not only its dynamics. RMA [19] distilled privileged environment parameters into an adaptation module estimated from proprioceptive history, enabling rapid online adaptation of a single quadruped to new terrains and payloads. Model-agnostic meta-learning [20] formalizes the alternative of training for fast fine-tuning directly; the anchored fine-tune here is deliberately simpler, relying on the configuration-conditioned prior rather than a meta-objective.

Morphology randomization for locomotion is the most direct precedent. GenLoco [12] trained a single locomotion controller across procedurally generated quadruped morphologies and transferred it zero-shot to several commercial robots. Bohlinger et al. [13] trained one policy to locomote across a wide set of embodiments, including quadrupeds, hexapods, and humanoids, with an architecture that handles varying joint counts. These works randomize morphology for the *locomotion* problem, where the body's geometry mainly affects contact dynamics. In local navigation, morphology additionally changes which free space is traversable and how obstacles are observed, so randomizing it without telling the policy which body it controls produces observation-action ambiguity, a failure mode measured directly in the ablation of Section 5.4.

2.3 Cross-embodiment navigation and generalist policies

A complementary line of work pools data from many robots. GNM [21] trained a goal-conditioned navigation policy on trajectories from six robots with a normalized action space and temporal context, and ViNT [22] scaled this to a Transformer foundation model that can be adapted to new tasks. NoMaD [23] unified goal-reaching and exploration with a diffusion policy trained on the same multi-robot data. RING [24] randomized body dimensions, rotation pivot, and camera placement in simulation to obtain an indoor navigation generalist, and NavFoM [25] trained a single navigation model across quadrupeds, drones, wheeled robots, and vehicles using camera-view and temporal identifiers. EA-Nav [26] conditioned visual navigation on embodiment geometry tokens and trained risk-aware

corrections on augmented high-risk data. Beyond navigation, Open X-Embodiment [27] assembled manipulation data from many robots, Chen et al. [28] conditioned manipulation policies on a hardware vector, shared modular policies [29] handled agent-agnostic control through message passing between limbs, and CrossFormer [30] trained one Transformer policy across manipulation, navigation, locomotion, and aviation. The present work shares with RING and EA-Nav the view that embodiment must be visible to the policy, but differs in three respects: it targets legged dynamics, in which payload mass and leg length change what velocities are achievable rather than only which spaces fit the body; it uses an explicit physical descriptor from a planning stack rather than learned identifiers; and it studies a deliberate few-shot fine-tune as part of the transfer protocol rather than zero-shot deployment alone.

2.4 Configuration-driven local planning

AgniNav [8] is the planning-side foundation of this work. It describes each robot by an abstract configuration consisting of body geometry, sensor placement, and velocity limits, with a shared collision envelope parameterized by collision-relevant height, front length, rear length, and half width. The collision-relevant height conditions a monocular image-to-scan prediction so that the perceived obstacle scan corresponds to the obstacles the body can actually hit, while the planar envelope parameters configure collision checking inside the local planner. Because every embodiment-specific quantity is routed through this configuration, a single planning stack runs on different robots by changing parameters, without retraining. The present work adopts the AgniNav descriptor as the interface between robot and skill, but makes two changes. First, the consumer of the descriptor is a learned policy rather than a sampling-based planner, so the skill can exploit learned priors about terrain and obstacle layouts that a geometric planner does not represent. Second, the descriptor is extended with leg length and payload mass, because for a legged robot these parameters change the achievable dynamics, not only the collision geometry, and a planner that only checks geometry cannot anticipate them. CeRLP [31], from the same research line, converts heterogeneous visual observations into a unified geometric representation with scale-corrected depth and height-adaptive scan abstraction before local planning; its insight that sensor height must be normalized out of the observation motivates the sensor-height axis of the randomization here. Hu et al. [32] learned adaptive safety margins for visual navigation, showing that fixed inflation trades collision risk against detours; the clearance-margin results in Section 5.3 expose the same trade-off for configuration-conditioned skills.

2.5 Simulation and evaluation

GPU-parallel simulators, from Isaac Gym [33] to the Orbit framework [34] that became Isaac Lab, make it feasible to instantiate many morphologies concurrently. Evaluation in simulation must nonetheless be interpreted with care: Kadian et al. [35] found that simulation rankings can correlate poorly with real-world rankings when agents exploit simulator artifacts, and Truong et al. [36] found that higher-fidelity simulation does not necessarily improve sim-to-real transfer. The metrics follow the success-centered conventions of Anderson et al. [37], complemented by clearance and sample-efficiency measures specific to transfer.

03 Problem Formulation

3.1 Configuration, envelope, and held-out set

A quadruped embodiment is described by a configuration vector

$$c = (\underbrace{l_f, l_r, w, h_c, h_s, \bar{v}_x, \bar{v}_y, \bar{\omega}}_{c_A}, \underbrace{\ell, m_p}_{c_L}) \in \mathbb{R}^{10}, \quad (1)$$

where c_A is the AgniNav descriptor [8] and c_L its legged extension. Here l_f and l_r are the front and rear extents of the collision footprint measured from the base frame origin, w is its half width, h_c is the collision-relevant height (the maximum height of the body, payload, and sensor above the ground in nominal stance), h_s is the sensor mounting height, $\bar{v}_x, \bar{v}_y$, and $\bar{\omega}$ are the forward, lateral, and yaw-rate command limits, ℓ is the leg length (hip-to-foot distance at full extension), and m_p is the payload mass.

The fleet morphologies differ in three independent physical parameters $\xi = (\ell, m_p, h_s)$. All other entries of c are deterministic functions of ξ given the fixed trunk geometry, so $c = \Gamma(\xi)$, where Γ evaluates the scaled kinematic model in nominal stance and the velocity-limit model of Section 4.3. The training envelope is the axis-aligned box

$$\Xi = [0.42, 0.58] \text{ m} \times [0, 12] \text{ kg} \times [0.35, 0.55] \text{ m}, \quad (2)$$

spanned by the four fleet morphologies. The held-out set Ξ_{ho} consists of morphologies not seen during training that lie outside the training support, with leg length increased by 22% and a payload of 15 kg, so that transfer requires extrapolation beyond Ξ .

3.2 Navigation task

For a fixed configuration c , local navigation is a partially observed Markov decision process $\mathcal{M}_c = (\mathcal{S}, \mathcal{O}, \mathcal{A}, P_c, r_c, \gamma)$. The state $s_t \in \mathcal{S}$ comprises the robot's base pose and twist, joint states, and the scene. The observation is

$$o_t = (H_t, g_t, \nu_t, a_{t-1}), \quad (3)$$

where $H_t \in \mathbb{R}^{64 \times 64}$ is a robot-centric height map built from LiDAR returns, $g_t = (\rho_t, \cos \phi_t, \sin \phi_t)$ is the goal in polar coordinates in the base frame, $\nu_t = (v_x, v_y, \omega)$ is the estimated base twist, and a_{t-1} is the previous action. The action $a_t = (v_x^{\text{cmd}}, v_y^{\text{cmd}}, \omega^{\text{cmd}}) \in \mathcal{A}$ is a twist command to the locomotion controller, and the admissible set $\mathcal{A}$ depends on c . The transition kernel P_c depends on c through the locomotion dynamics (leg length and payload), the collision geometry (footprint and h_c), and the observation model (sensor height h_s changes occlusion). An episode succeeds if $\rho_t < 0.3$ m before a 30 s timeout without a collision of the trunk, payload, or sensor with the environment and without a fall.

3.3 Learning objective and transfer problem

A configuration-conditioned policy $\pi_\theta(a \mid o, c)$ is trained to maximize the expected return over the randomization distribution $p(\xi)$ supported on Ξ :

$$\max_{\theta} \mathbb{E}_{\xi \sim p(\xi)} \mathbb{E}_{\tau \sim \pi_\theta, \mathcal{M}_{\Gamma(\xi)}} \left[\sum_{t=0}^T \gamma^t r_{\Gamma(\xi)}(s_t, a_t) \right]. \quad (4)$$

At transfer time, a target configuration $c^* = \Gamma(\xi^*)$ with $\xi^* \in \Xi_{\text{ho}}$ is given together with a budget of N rollout episodes on

that embodiment, with N between 200 and 600. An adaptation operator produces $\theta^* = \text{Adapt}(\theta, c^*, N)$, and performance is measured on $\mathcal{M}_{c^*}$.

3.4 Metrics

Let $S(\theta; c)$ denote the success rate over the 20 scenarios and 5 seeds. The clearance of the footprint $\mathcal{F}(x_t, c)$ at base pose x_t is the distance to the nearest obstacle point q in the obstacle set $\mathcal{Q}$,

$$d_t = \min_{q \in \mathcal{Q}} \text{sd}(q; \mathcal{F}(x_t, c)), \quad (5)$$

where sd is the signed distance to the footprint rectangle. The clearance margin of an episode is $\min_t d_t$, averaged over successful episodes. Fine-tune sample efficiency is measured as

$$N_{90} = \min\{n : S_n \geq 0.9 S_{\text{final}}\}, \quad (6)$$

the number of target-embodiment rollout episodes after which the adaptation curve S_n first reaches 90% of its final value S_{final} . Transfer gain is reported in percentage points relative to morphology-specific training; its aggregation is discussed with the results.

04 Method

4.1 Control hierarchy

The system has three layers running at fixed rates (Table 1). At the top, the navigation policy π_θ maps (o_t, c) to a twist command every 50 ms. In the middle, a configuration-conditioned locomotion policy tracks that twist every 20 ms by outputting joint position targets. At the bottom, joint PD controllers run at the 5 ms physics step. The locomotion policy is trained beforehand with the same randomization of leg length and payload as in Eq. (12), following the massively parallel recipe of [4] with a velocity-tracking reward, and it is frozen while the navigation skill is trained and fine-tuned. The velocity interface is chosen over end-to-end joint control because it makes the configuration-dependent command limits explicit and lets the navigation skill be studied independently of gait quality. Between the navigation policy and the locomotion controller sits a footprint-aware velocity governor, described in Section 4.4.

4.2 Descriptor encoding and FiLM conditioning

The descriptor is normalized by the envelope statistics, $\hat{c} = (c - \mu_\Xi) \oslash \sigma_\Xi$, where μ_Ξ and σ_Ξ are the mean and standard deviation of $\Gamma(\xi)$ under uniform sampling of Ξ . Held-out configurations outside the envelope therefore appear as normalized values beyond the training range, which keeps extrapolation explicit rather than clipped. An encoder ψ maps $\hat{c}$ to an embedding $z = \psi(\hat{c}) \in \mathbb{R}^{64}$.

The height map is encoded by a convolutional network into a 128-dimensional feature that is concatenated with g_t, ν_t , and a_{t-1} to form the trunk input h_0 . The trunk has $K = 3$ layers of width 256, each modulated by the configuration through feature-wise linear modulation [14]:

$$h_{k+1} = \text{ELU}(\gamma_k(z) \odot (W_k h_k + b_k) + \beta_k(z)), \quad (7)$$

$$k = 0, \dots, K - 1,$$

where γ_k and β_k are linear maps from z and $\odot$ is the element-wise product. A Gaussian head outputs the mean $\mu_\theta(h_K)$ and a state-independent diagonal covariance Σ_θ of a pre-squashing action u_t .

Multiplicative conditioning is chosen over concatenating z to the input for two reasons. First, the effect of the configuration on navigation is largely a rescaling of decision thresholds: whether a gap is traversable depends on the ratio of gap width to w , and whether an overhang must be avoided depends on the ratio of its height to h_c . A gain $\gamma_k(z)$ represents such rescaling directly. Second, the conditioning parameters $(\psi, \{\gamma_k, \beta_k\})$ form a compact subset of the network through which the configuration acts, which is the natural subset to adapt when a new embodiment is encountered (Section 4.8).

4.3 Configuration-scaled action space

The policy does not output velocities directly. It outputs a fraction of the embodiment's capability,

$$a_t = \bar{u}(c) \odot \tanh(u_t), \quad u_t \sim \mathcal{N}(\mu_\theta(h_K), \Sigma_\theta), \quad (8)$$

with $\bar{u}(c) = (\bar{v}_x, \bar{v}_y, \bar{\omega})$. The forward limit follows dynamic similarity for legged locomotion: robots of different leg length move similarly at equal Froude number $\text{Fr} = v^2/(g\ell)$, so the attainable speed scales as $\sqrt{g\ell}$. The payload reduces the attainable speed approximately in proportion to the added fraction of body mass, giving

$$\bar{v}_x(c) = \sqrt{\text{Fr}^* g \ell} \left(1 - \kappa \frac{m_p}{M_b}\right), \quad \bar{v}_y = \frac{1}{2} \bar{v}_x, \quad \bar{\omega} = \frac{\bar{v}_x}{r_{\min}}, \quad (9)$$

where g is gravitational acceleration, M_b is the unloaded robot mass, $\text{Fr}^* = 0.45$ is the design Froude number, $\kappa = 0.5$ is a payload derating coefficient, and $r_{\min} = 1$ m is the turning radius at the forward limit. For $\ell = 0.5$ m and no payload, Eq. (9) gives $\bar{v}_x \approx 1.49$ m/s. These limits are entries of c_A , so the same model sets both the descriptor values the policy sees and the bounds its outputs respect. The design keeps action semantics comparable across embodiments: an output of $\tanh(u_t) = 1$ always means full forward capability, whatever the body, so the policy does not need to relearn the meaning of its action space for each morphology.

4.4 Footprint-aware velocity governor

To carry AgniNav's explicit collision reasoning into the learned stack, each command passes through a governor that rolls out the constant-twist arc for a horizon $T_h = 1.0$ s at $M = 10$ samples and scales the command by the largest factor that keeps the footprint clear:

$$\tilde{a}_t = \lambda^* a_t,$$

$$\lambda^* = \max \{ \lambda \in [0, 1] : d(x_{t+jT_h/M}(\lambda a_t), c) \geq d_{\min}, \quad j = 1, \dots, M \}, \quad (10)$$

where $x_{t+\tau}(a)$ is the base pose reached by holding twist a for time τ , $d(\cdot, c)$ is the clearance of Eq. (5) evaluated on the current height map, and $d_{\min} = 0.05$ m. The governor uses the true configuration c in every variant studied, including the ablation without the descriptor input, so that differences between variants reflect the policy rather than the safety layer. It is active during training as well as evaluation, and the policy is penalized for interventions (Eq. (11)) so that it learns to issue commands the governor does not need to alter. The governor is a last-resort filter, not the planner: it cannot steer around obstacles, only slow down before them.

4.5 Reward and termination

The per-step reward is

$$r_t = w_p(\rho_{t-1} - \rho_t) + w_g\mathbb{I}[\rho_t < 0.3] - w_c\mathbb{I}[\text{col}_t] - w_d \max(0, d_{\text{ref}} - d_t) - w_\lambda(1 - \lambda_t^*) - w_s \|a_t - a_{t-1}\|^2, \quad (11)$$

with progress weight $w_p = 1.0$ per meter, goal bonus $w_g = 10$, collision penalty $w_c = 10$, clearance shaping $w_d = 2.0$ per meter below $d_{\text{ref}} = 0.25$ m, governor-intervention penalty $w_\lambda = 0.1$, and smoothness weight $w_s = 0.01$. Episodes terminate on success, collision, a fall (base roll or pitch exceeding 1 rad), or timeout. The clearance term uses the configuration-dependent footprint, so the same obstacle layout yields different shaping for different bodies, which forces the policy to use c .

4.6 Skill-level configuration randomization

At each environment reset, the physical parameters are drawn from a mixture that oversamples the envelope boundary,

$$p(\xi) = (1 - \eta)\mathcal{U}(\Xi) + \eta\mathcal{U}(\partial\Xi), \quad \eta = 0.2, \quad (12)$$

where $\mathcal{U}(\partial\Xi)$ is uniform over the faces of the box. Boundary oversampling is motivated by the evaluation: held-out morphologies lie beyond the envelope, and the policy’s behavior there is shaped mostly by what it learned near the boundary. Each sample is instantiated in Isaac Lab by scaling the thigh and shank links by ℓ/ℓ_0 relative to a nominal leg length ℓ_0 , attaching the payload as a rigid body on the trunk with a center-of-mass offset drawn uniformly within ± 5 cm, and mounting the LiDAR at h_s . Because articulated assets cannot be rescaled during a rollout, a bank of procedurally generated assets drawn from $p(\xi)$ is assigned to the parallel environments and regenerated periodically. Ground friction and sensor noise are randomized per episode in the standard manner [10, 4]. The descriptor seen by the policy is perturbed as $\hat{c} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.02^2 I)$, so that the policy tolerates small measurement errors in the configuration.

4.7 Pretraining

The navigation policy is trained with PPO [38] using the clipped surrogate

$$\mathcal{L}_{\text{PPO}}(\theta) = -\mathbb{E}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon_c, 1 + \epsilon_c) \hat{A}_t \right) \right], \quad (13)$$

where $r_t(\theta) = \pi_\theta(a_t | o_t, c) / \pi_{\theta_{\text{old}}}(a_t | o_t, c)$ and $\epsilon_c = 0.2$. Advantages $\hat{A}_t$ are computed with generalized advantage estimation [39] ($\gamma = 0.99$, $\lambda_{\text{GAE}} = 0.95$). The critic is asymmetric [40]: in addition to (o_t, c) it receives privileged information unavailable at deployment, namely the unoccluded terrain heights around the robot, the ground friction coefficient, the payload center-of-mass offset, and foot contact states. Privileged inputs reduce the variance of value estimates across the randomized configurations, which is otherwise dominated by unobserved differences in dynamics. A curriculum over scenario difficulty raises obstacle density and narrows gaps as success on the current level increases.

4.8 Anchored few-shot fine-tuning

For a target embodiment c^* , the descriptor is set to the measured configuration and the policy is fine-tuned on rollouts

Table 1: Control and training settings.

Setting	Value
Physics step / joint PD	5 ms
Locomotion policy period	20 ms
Navigation policy period	50 ms
Height map	64 x 64 cells, from LiDAR
Episode timeout / goal radius	30 s / 0.3 m
Trunk / descriptor embedding	3 x 256 (FiLM) / 64
PPO clip / discount / GAE	0.2 / 0.99 / 0.95
Boundary mixture weight	0.2
Descriptor noise (normalized)	0.02
Governor horizon / samples	1.0 s / 10
Fine-tune KL weight	0.05
Fine-tune budget	200 to 600 episodes

of that embodiment only. The parameters are partitioned as $\theta = (\theta_T, \theta_A)$, where θ_T holds the height-map encoder and trunk weights $\{W_k, b_k\}$, and $\theta_A = (\psi, \{\gamma_k, \beta_k\}, \mu_\theta, \Sigma_\theta)$ holds the conditioning and output layers. Only θ_A is updated, under the objective

$$\mathcal{L}_{\text{FT}}(\theta_A) = \mathcal{L}_{\text{PPO}}(\theta) + \beta_{\text{KL}} \mathbb{E}_{o \sim \mathcal{D}_{c^*}} \left[D_{\text{KL}}(\pi_\theta(\cdot | o, c^*) \parallel \pi_{\theta_0}(\cdot | o, c^*)) \right], \quad (14)$$

where θ_0 are the pretrained parameters, $\mathcal{D}_{c^*}$ is the distribution of observations visited on the target, and $\beta_{\text{KL}} = 0.05$. The critic is initialized from the pretrained critic and updated in full; its first updates use only the initial batch of target rollouts so that advantage estimates are calibrated before the actor moves. Rollouts are drawn uniformly over the 20 scenarios, and the total budget is between 200 and 600 episodes.

Each element of the fine-tune has a specific purpose. Restricting updates to θ_A reduces the number of adapted parameters to the part of the network through which the configuration acts, so the few hundred episodes available are spent correcting how the body is represented rather than re-learning obstacle perception. The KL anchor keeps the adapted policy close to the pretrained one on states the target visits, protecting the obstacle-avoidance behavior learned across the envelope from being overwritten by high-variance gradients early in fine-tuning. The pretrained descriptor conditioning places the initial policy near the right behavior for c^* , so the fine-tune mainly corrects the residual mismatch that the descriptor cannot express, chiefly differences in contact dynamics outside the training envelope. The full procedure is summarized below.

PROCEDURE SUMMARY

- (1) Train the configuration-conditioned locomotion controller with leg-length and payload randomization; freeze it. (2) Train π_θ with PPO and an asymmetric critic, sampling $\xi \sim p(\xi)$ per reset (Eq. (12)), with the governor active. (3) For each held-out embodiment, set c^* from measurement and run anchored fine-tuning (Eq. (14)) for 200 to 600 episodes. (4) Evaluate on the 20 scenarios with 5 seeds.

05 Experiments

5.1 Setup and platform

All experiments run in Isaac Lab. The fleet consists of four quadruped morphologies with leg lengths between 0.42 and

0.58 m, payloads between 0 and 12 kg, and sensor heights between 0.35 and 0.55 m; together they span the envelope Ξ of Eq. (2). Training uses configuration randomization over the full envelope. Evaluation uses held-out morphologies outside the training support, with leg length increased by 22% and a payload of 15 kg, and each held-out morphology is fine-tuned with 200 to 600 rollout episodes. Every method is evaluated on 20 navigation scenarios with 5 seeds. Table 2 summarizes the testbed.

5.2 Baselines

Three baselines isolate the components of the method. **Morphology-specific training (per morphology)** trains a separate navigation policy for each morphology without configuration input or randomization, which is the standard single-body practice. **Configuration randomization, no fine-tune** is the pretrained configuration-conditioned policy deployed on each held-out morphology with its measured descriptor but without any target-specific update, which measures zero-shot transfer. **Fine-tune from scratch (per morphology)** trains the navigation policy on each held-out morphology without configuration-randomized pretraining, using the same locomotion controller and the same target-episode budget, which measures what the target data alone can achieve. All methods share the locomotion controller, governor, observation space, and reward. For rows in which no target-specific update is part of the deployed method, the sample-efficiency column reports the adaptation curve recorded when that initialization is trained on the target embodiment.

5.3 Main results

Table 3 reports the main comparison.

Success. The proposed method reaches 82.6% success on held-out morphologies, 9.5 points above morphology-specific training (73.1%), 4.8 points above fine-tuning from scratch (77.8%), and 10.3 points above the same pretrained policy deployed without fine-tuning (72.3%). The zero-shot result is informative in its own right: a single configuration-conditioned policy, never trained on the held-out bodies, comes within 0.8 points of the morphology-specific policies. Zero-shot transfer alone, however, does not outperform them, and fine-tuning from scratch improves on morphology-specific training by only 4.7 points. The largest gain appears when the two are combined, which suggests that the randomized prior and the target data contribute complementary information: the prior supplies obstacle-avoidance competence that generalizes across bodies, and the fine-tune supplies the target-specific correction that the prior cannot extrapolate.

Transfer gain. The transfer-gain column is the per-morphology gain statistic relative to the morphology-specific reference, as recorded by the evaluation server. Its ordering agrees with the success column for the proposed method, which has the largest gain (8.4 points), but its values do not reduce to differences of the pooled success rates in the first column; for instance, the pooled difference between the proposed method and morphology-specific training is 9.5 points. The spread of the gain statistic across methods (7.4 to 8.4 points) is also much narrower than the spread of pooled success (72.3% to 82.6%). The pooled success rate is therefore treated as the primary measure, and the gain column as a secondary indicator reported verbatim.

Sample efficiency. The proposed fine-tune reaches 90% of its final success after 420 episodes on average, compared with 481 for morphology-specific training, 486 for the randomized policy without the anchored fine-tune, and 455 for fine-tuning from scratch. Relative to morphology-specific training, this is 61 fewer episodes (12.7% fewer); relative to fine-tuning from scratch, 35 fewer (7.7% fewer). Because N_{90} is normalized by each method’s own final success, these numbers compare how quickly each method approaches its plateau, and the proposed method approaches a higher plateau faster. All four values lie inside the 200 to 600 episode budget, so the comparison is not truncated by the budget.

Clearance. The proposed method has the smallest clearance margin, 17.9 cm, compared with 19.1 cm for morphology-specific training and zero-shot randomization and 20.1 cm for fine-tuning from scratch; the differences are 1.2 cm and 2.2 cm, respectively. The margin is computed on successful episodes, so it describes how closely the robot passes obstacles when it succeeds, not how often it collides. A plausible reading is that the configuration-conditioned policy, knowing its footprint, accepts tighter passages that the other methods avoid or fail in; this would also contribute to its higher success. The governor’s $d_{\min} = 0.05$ m bounds how small the margin can become, but a lower average margin still reduces tolerance to footprint or localization errors on a physical robot, a trade-off of the kind studied by adaptive-margin methods [32].

5.4 Ablation

Table 4 removes one component at a time; every variant is fine-tuned with the same anchored procedure.

Removing the configuration descriptor causes the largest drop, 16.0 points to 66.6%, which is 6.5 points below even morphology-specific training. This is the expected consequence of randomizing morphology without telling the policy which body it controls: the same height map and goal require different actions for different bodies, and a policy without the descriptor can only learn a compromise across the envelope. The result supports the central design choice of feeding the AgniNav descriptor to the skill rather than relying on randomization alone. Removing payload randomization costs 10.1 points (72.5%), and removing sensor-height randomization costs 7.4 points (75.2%). Among the randomization axes, payload is therefore the most important, consistent with the failure analysis below. Sensor-height randomization matters less but remains substantial, which agrees with the observation in CeRLP [31] and AgniNav [8] that sensor height changes what a height-map-based obstacle representation contains.

5.5 Failure cases

Three recorded failure cases delimit the method.

Leg-length extrapolation. On the morphology with leg length increased by 22%, outside the support, success is 74.1%, which is 8.5 points below the held-out average of 82.6%. The descriptor extrapolates: the normalized leg length lies beyond the training range, and the velocity limits of Eq. (9) and the footprint are computed correctly for the longer leg. The contact model does not. The locomotion dynamics of a longer-legged body, including foot clearance, step timing, and the coupling between commanded twist and actual base motion, are outside what the frozen locomotion controller and the navigation prior have experienced, and an explicit descriptor cannot supply that information. This failure marks the boundary between what ex-

Table 2: Experimental testbed.

Item	Setting
Simulator	Isaac Lab
Fleet	Four quadruped morphologies
Leg length range	0.42 to 0.58 m
Payload range	0 to 12 kg
Sensor height range	0.35 to 0.55 m
Evaluation protocol	20 navigation scenarios x 5 seeds
Training	Configuration randomization over the full envelope
Fine-tune budget	200 to 600 rollout episodes per held-out morphology
Held-out morphologies	Outside the training support (leg length +22%, payload 15 kg)

Table 3: Main comparison on held-out morphologies (20 scenarios x 5 seeds). Higher is better for success and transfer gain; lower is better for fine-tune episodes.

Method	Held-out morphology success (%)	Transfer gain vs morphology-specific (pts)	Fine-tune episodes to 90% of final	Clearance (cm)	margin
Morphology-specific training (per morphology)	73.1	7.4	481	19.1	
Configuration randomization, no fine-tune	72.3	8.0	486	19.1	
Fine-tune from scratch (per morphology)	77.8	7.8	455	20.1	
Ours	82.6	8.4	420	17.9	

Table 4: Ablation on held-out morphologies. The change is relative to the full method.

Variant	Held-out morphology success (%)	Change (pts)
Ours (randomization + fine-tune)	82.6	—
w/o configuration descriptor	66.6	-16.0
w/o payload randomization	72.5	-10.1
w/o sensor-height randomization	75.2	-7.4

licit configuration can transfer, namely geometry and limits, and what it cannot, namely contact dynamics.

Heavy payload. The 15 kg payload fine-tune needed 580 episodes to reach 90% of its final performance, compared with the 420-episode average, which is 160 more episodes, or about 38% more. Mass dominates the envelope: a heavier payload changes achievable acceleration, braking distance, and the locomotion controller’s tracking error simultaneously, and Eq. (9) captures only the first-order effect on the speed limit. The 580-episode figure lies close to the upper end of the 200 to 600 episode budget, so a still heavier payload would likely exceed it.

Payload is the critical randomization axis. Without payload randomization, the held-out morphology carrying a 12 kg payload degrades by 12.3 points, more than the 10.1-point average loss of that ablation. Together with the ablation ranking, this identifies payload as the single most important randomization axis: a navigation skill that has not experienced the dynamics of a loaded body during pretraining cannot recover them from a few hundred episodes, even when the descriptor reports the payload mass.

5.6 Discussion

The results support a division of labor. Explicit configuration handles what is geometric and can be measured, such as footprint, sensor height, and velocity limits, and it makes zero-shot transfer competitive with morphology-specific training. Randomization over the envelope teaches the policy how those quantities change the navigation problem, and is ineffective without the descriptor. The anchored fine-tune corrects what neither can express, which is predominantly the dynamics of mass and contact. The failure cases indicate that this last com-

ponent is also where transfer is most fragile: both the leg-length and payload failures are dynamic rather than geometric. The evidence is limited to simulation and to one platform family, and the results should be read with that scope.

06 Limitations

The evaluation is entirely in simulation. Isaac Lab is the study’s testbed, and, as Kadian et al. [35] and Truong et al. [36] caution, rankings obtained in simulation may not hold on hardware; the contact-model failure under leg-length extrapolation is exactly the kind of effect that a physical robot could make worse. The fleet comprises four morphologies of one quadruped family, and the held-out morphologies vary leg length, payload, and sensor height; other changes, such as trunk dimensions, actuator limits, or different leg topologies, are not tested. The descriptor is assumed to be known at deployment; errors larger than the small noise injected during training, for example an unreported payload, are not studied, and online estimation of the configuration would be required in practice. The reported results are means over 5 seeds without per-seed dispersion, so small differences, such as the 0.8-point gap between zero-shot randomization and morphology-specific training, should not be interpreted as significant. The transfer-gain statistic does not reduce to differences of the pooled success rates, as noted in Section 5.3, and conclusions are drawn primarily from pooled success. The method trades clearance for success, with the smallest margin of the compared methods, which would need to be revisited before deployment near people or fragile infrastructure. Finally, the locomotion controller is frozen during navigation fine-tuning; adapting it jointly might address the contact-model failure but was not evaluated.

07 Conclusion

This paper extended configuration randomization from locomotion to local navigation skills. A navigation policy conditioned on an extended AgniNav descriptor and trained across a fleet’s configuration envelope, then fine-tuned for each new embodiment with an anchored update of its conditioning and output layers, reached 82.6% success on held-out quadruped morphologies outside the training support, compared with 73.1% for

morphology-specific training and 77.8% for fine-tuning from scratch, and it reached 90% of its final success in 420 episodes on average. The ablation showed that the descriptor is indispensable once morphology is randomized and that payload is the most important randomization axis. The failure cases showed where the approach stops: explicit configuration transfers geometry and limits, but not contact dynamics under leg-length extrapolation or heavy payloads, and the method passes obstacles with a smaller clearance margin. Natural next steps are physical validation across a real fleet, online estimation of the configuration, joint adaptation of the locomotion controller, and adaptive clearance budgets for configuration-conditioned skills.

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. arXiv:1901.08652, 2019.
- [2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. arXiv:2010.11251, 2020.
- [3] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Robust Perceptive Locomotion for Quadrupedal Robots in the Wild. arXiv:2201.08117, 2022.
- [4] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. arXiv:2109.11978, 2021.
- [5] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter. Learning a State Representation and Navigation in Cluttered and Dynamic Environments. arXiv:2103.04351, 2021.
- [6] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced Skills by Learning Locomotion and Local Navigation End-to-End. arXiv:2209.12827, 2022.
- [7] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. arXiv:2306.14874, 2023.
- [8] T. Zang, S. Cheng, H. Huang, S. Wang, and W. Zhang. AgniNav: Configuration-Driven Cross-Embodiment Local Planning for Robot Navigation. arXiv:2606.10903, 2026.
- [9] 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. arXiv:1703.06907, 2017.
- [10] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. arXiv:1710.06537, 2017.
- [11] OpenAI, I. Akkaya et al.. Solving Rubik’s Cube with a Robot Hand. arXiv:1910.07113, 2019.
- [12] G. Feng et al.. GenLoco: Generalized Locomotion Controllers for Quadrupedal Robots. arXiv:2209.05309, 2022.
- [13] 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:2409.06366, 2024.
- [14] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. FiLM: Visual Reasoning with a General Conditioning Layer. arXiv:1709.07871, 2017.
- [15] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot Parkour Learning. arXiv:2309.05665, 2023.
- [16] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme Parkour with Legged Robots. arXiv:2309.14341, 2023.
- [17] J. Truong, D. Yarats, T. Li, F. Meier, S. Chernova, D. Batra, and A. Rai. Learning Navigation Skills for Legged Robots with Learned Robot Embeddings. arXiv:2011.12255, 2020.
- [18] W. Yu, J. Tan, C. K. Liu, and G. Turk. Preparing for the Unknown: Learning a Universal Policy with Online System Identification. arXiv:1702.02453, 2017.
- [19] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. arXiv:2107.04034, 2021.
- [20] C. Finn, P. Abbeel, and S. Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. arXiv:1703.03400, 2017.
- [21] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A General Navigation Model to Drive Any Robot. arXiv:2210.03370, 2022.
- [22] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. ViNT: A Foundation Model for Visual Navigation. arXiv:2306.14846, 2023.
- [23] A. Sridhar, D. Shah, C. Glossop, and S. Levine. NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration. arXiv:2310.07896, 2023.
- [24] A. Eftekhari et al.. The One RING: a Robotic Indoor Navigation Generalist. arXiv:2412.14401, 2024.
- [25] J. Zhang et al.. Embodied Navigation Foundation Model. arXiv:2509.12129, 2025.
- [26] J. Zhang, Y. Du, X. Guo, S. Sun, X. Liu, Y. Sun, G. Lu, W. Sui, and J. Li. EA-Nav: Learning Safe Visual Navigation Policies with Embodiment Awareness. arXiv:2607.19880, 2026.
- [27] Open X-Embodiment Collaboration et al.. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. arXiv:2310.08864, 2023.
- [28] T. Chen, A. Murali, and A. Gupta. Hardware Conditioned Policies for Multi-Robot Transfer Learning. arXiv:1811.09864, 2018.
- [29] W. Huang, I. Mordatch, and D. Pathak. One Policy to Control Them All: Shared Modular Policies for Agent-Agnostic Control. arXiv:2007.04976, 2020.
- [30] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine. Scaling Cross-Embodied Learning: One Policy for Manipulation, Navigation, Locomotion and Aviation. arXiv:2408.11812, 2024.
- [31] H. Xi, M. Tan, X. Zhang, S. Cheng, S. Wang, Y. Gu, X. Shen, and W. Zhang. CeRLP: A Cross-embodiment Robot Local Planning Framework for Visual Navigation. arXiv:2603.19602, 2026.
- [32] J. Hu, S. Yuan, G. C. R. Bethala, A. Tzes, and Y. Fang. Learning Adaptive Safety Margins for Visual Navigation. arXiv:2607.18200, 2026.
- [33] V. Makoviychuk et al.. Isaac Gym: High Performance GPU-Based Physics Simulation for Robot Learning. arXiv:2108.10470, 2021.
- [34] M. Mittal et al.. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. arXiv:2301.04195, 2023.
- [35] A. Kadian et al.. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? arXiv:1912.06321, 2019.
- [36] J. Truong, M. Rudolph, N. H. Yokoyama, S. Chernova, D. Batra, and A. Rai. Rethinking Sim2Real: Lower Fidelity Simulation Leads to Higher Sim2Real Transfer in Navigation. arXiv:2207.10821, 2022.
- [37] P. Anderson et al.. On Evaluation of Embodied Navigation Agents. arXiv:1807.06757, 2018.

- [38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. arXiv:1707.06347, 2017.
- [39] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. arXiv:1506.02438, 2015.
- [40] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel. Asymmetric Actor Critic for Image-Based Robot Learning. arXiv:1710.06542, 2017.