

Cross-Embodiment Distillation of Local Navigation Planners from Humanoid to Quadruped Teams

Yifan Chen • Zhiyuan Wang • Haoran Liu • Xiaotong Zhang

Robotics and Autonomous Systems Lab, Shanghai Jiao Tong University, Shanghai, China

PUBLISHED

July 21, 2026

REGISTERED

July 25, 2026

RECORD TYPE

Preprint

ABSTRACT

Configuration-driven local planners such as AgniNav can serve several robot embodiments from a single planning stack, but the full planner is too expensive for the onboard computers of commodity quadrupeds. This preprint presents a cross-embodiment distillation pipeline in which a configuration-driven teacher planner generates local navigation trajectories in a shared, body-normalized configuration space, and a compact onboard student policy is distilled from these trajectories with an explicit embodiment descriptor as input. Training combines on-policy data aggregation, a finite-difference consistency regularizer that aligns the sensitivity of the student to descriptor changes with that of the teacher, and a held-out embodiment split used for model selection. The pipeline is evaluated in Isaac Lab with a Unitree Go2 and a 12-DoF H1-class humanoid over 20 indoor-outdoor transition scenes, 5 seeds, and 50 episodes per scene and seed, and onboard on a Go2 whose Jetson Orin computer runs the student. The distilled student reaches 90.4 percent navigation success, compared with 81.5 percent for a per-embodiment oracle teacher and 82.3 percent for a student without embodiment conditioning, with a teacher parity score of 93.1 percent and an onboard decision latency of 8.2 ms. Its mean path clearance margin of 18.7 cm is, however, the smallest among the compared methods. Ablations show drops of 11.3 to 15.3 percentage points when the descriptor input, the consistency regularizer, or the held-out split is removed. Recorded failures include reduced clearance on a steep entry ramp and degraded success under descriptor dropout.

INDEX TERMS cross-embodiment navigation, policy distillation, local planning, legged robots, humanoid robots, quadruped robots

KAGG NOTE A configuration-driven planner can be compressed into an onboard policy that serves unseen bodies only when the embodiment descriptor is an explicit input to the student and the student's sensitivity to that descriptor is trained and selected to match the teacher's.

01 Introduction

Mixed teams of legged robots are increasingly deployed in the same facilities: a humanoid may carry out manipulation-heavy tasks indoors while one or more quadrupeds patrol, inspect, or escort it through doorways, ramps, and outdoor yards. Each of these platforms needs a local navigation planner that converts a goal and a near-field map into a short, collision-free trajectory. Maintaining one hand-tuned planner per body multiplies engineering cost and makes team behavior hard to reason about. Configuration-driven planners address this problem by construction: AgniNav [1] describes the robot through an abstract configuration—collision envelope, sensor placement, and velocity limits—so that a single planning stack can run on bodies as different as a humanoid and a quadruped. Related geometric frameworks such as CeRLP [2] follow the same principle of separating environment structure from body parameters.

The difficulty is compute. A configuration-driven planner that evaluates a dense candidate library, checks every candidate against a body-specific envelope, and refines the best candidates by sampling-based optimization is comfortably real-time on a workstation or on the larger computers carried by research humanoids, but it competes for cycles with perception, state estimation, and locomotion control on the embedded computers of commodity quadrupeds such as the Unitree Go2. In our logs, the full teacher planner requires a 41 ms decision cycle.

Learned navigation policies, on the other hand, are cheap at inference time, and large multi-robot models such as GNM, ViNT, and NoMaD [5, 6, 7] show that data from several bodies can be shared. However, such policies usually treat embodiment implicitly, which leaves them exposed to the observation-action ambiguity documented by EA-Nav [3]: the same observation can call for different actions on bodies of different sizes.

This preprint studies the gap between these two lines of work: how to retain the embodiment generality of a configuration-driven planner while executing only a compact learned policy on the robot, including on embodiments for which the policy has never seen teacher demonstrations. The proposed approach distills a configuration-driven teacher into an embodiment-conditioned student. The teacher plans in a shared, body-normalized configuration space; the student receives the same configuration descriptor as an explicit input; and training is organized so that generalization to unseen descriptors is measured and selected for rather than assumed.

The contributions are as follows.

1. A formulation of cross-embodiment planner distillation in which the teacher, the student, and the feasibility constraints share a single descriptor-indexed configuration space, and generalization is evaluated on held-out embodiment descriptors.

2. A distillation objective that combines trajectory regression in body-normalized coordinates, soft-label distillation over a configuration-scaled trajectory library, a differentiable envelope-penetration penalty, and a finite-difference consistency regularizer that matches the student’s sensitivity to descriptor changes to that of the teacher.
3. A deployment structure for the Go2 onboard computer—student inference, single-trajectory envelope check, way-point tracking at 50 Hz, and a descriptor watchdog with hand-off to a conservative fallback.
4. An evaluation in Isaac Lab with a Unitree Go2 and a 12-DoF H1-class humanoid over 20 indoor–outdoor transition scenes, together with onboard Jetson Orin measurements on the Go2, ablations of each training component, and an analysis of the recorded failure cases.

02 Related Work

2.1 Configuration-driven and geometric cross-embodiment planning

AgniNav [1] is the planner family that this work distills. It defines a configuration-driven local planner that operates on an abstract robot configuration rather than on a specific platform. Body geometry is summarized by a compact collision envelope—a collision-relevant height together with front length, rear length, and half width—and the configuration also carries sensor and velocity-limit parameters. The collision-relevant height conditions how monocular observations are converted into a planar scan, and the planar envelope parameters configure collision checking inside the local planner. The consequence is that one planning stack can serve several embodiments by changing a configuration file instead of re-training or retuning. The present work adopts this abstraction as the shared space in which both teacher and student operate, but it differs from AgniNav in where computation happens: rather than running the configuration-driven planner online, the planner is used offline and in simulation as a teacher, and its behavior across configurations is compressed into a policy cheap enough for a quadruped’s onboard computer. A second difference is the treatment of configurations that were never planned for during training; AgniNav handles a new configuration exactly because it plans online, whereas a distilled student must interpolate or extrapolate, which motivates the consistency regularizer and the held-out split introduced in Section 4.

CeRLP [2] pursues a related decomposition for visual navigation: heterogeneous camera observations are converted into a unified geometric representation, including scale-corrected depth and a height-adaptive scan abstraction, before a common local planner is applied. CeRLP and AgniNav together establish that an explicit geometric interface between perception and planning makes cross-embodiment failures diagnosable. This work keeps that interface for the teacher and asks whether a learned student can inherit it. EA-Nav [3] approaches the problem from the learning side: it identifies observation–action ambiguity as the central failure of embodiment-agnostic imitation, conditions a visual navigation policy on embodiment geometry tokens, and trains risk-aware corrections on augmented high-risk trajectories. Our descriptor input is motivated by the same ambiguity argument; unlike EA-Nav, the supervision comes from a configuration-driven planner rather than from demon-

strations, so labels for arbitrary descriptors can be generated on demand. Classical local planning and its learned variants are surveyed by Xiao et al. [4]; within that taxonomy, the present method is a learned replacement for a classical planner that preserves the planner’s explicit parameter interface.

2.2 Cross-embodiment navigation policies

A second family learns a single policy from data collected on many robots. GNM [5] trains a goal-conditioned navigation model on trajectories from six robots using a normalized action space and temporal context, and shows that pooled heterogeneous data improve robustness over single-robot specialists. ViNT [6] scales this approach into a Transformer foundation model with a general goal-reaching objective and adaptation to new task modalities, and NoMaD [7] unifies goal-directed navigation and undirected exploration with a goal-masked diffusion decoder. These systems normalize actions across robots, which is analogous to our body-normalized coordinates, but they do not receive explicit body geometry and therefore cannot represent that the same gap is passable for one body and not for another. RING [8] addresses this by randomizing body size, rotation pivot, and camera configuration in simulation, and NavFoM [9] trains a unified model across quadrupeds, drones, wheeled robots, and vehicles with embodiment-related identifiers for camera views and temporal context. Our descriptor sampling over a continuous family is closest in spirit to RING’s randomization; the difference is that each sampled descriptor is paired with teacher labels computed for exactly that descriptor, so the student is supervised rather than reinforced.

Beyond navigation, Yang et al. [10] show that manipulation and navigation data can be co-trained in one policy, CrossFormer [11] trains one Transformer policy across manipulation, navigation, locomotion, and aviation embodiments with embodiment-specific readout tokens, and the Open X-Embodiment collaboration [12] assembles a large multi-robot dataset with RT-X models trained on it. These efforts emphasize breadth of data; our setting is narrower (two legged families and their descriptor perturbations) but requires that held-out bodies be handled safely in the near field, which the large-scale models do not explicitly guarantee.

2.3 Embodiment-conditioned control

Conditioning a policy on a description of the hardware has a longer history in control. Hardware Conditioned Policies [13] feed an explicit or learned hardware vector to the policy to transfer manipulation skills across kinematic structures. Shared Modular Policies [14] and MetaMorph [15] represent morphology as a graph or token sequence so that one controller can drive agents with different limb structures. In legged locomotion, GenLoco [16] trains a single controller across randomized quadruped morphologies, ManyQuadrupeds [17] learns one locomotion policy for diverse quadrupeds, and RMA [18] estimates a latent extrinsics vector online to adapt to unknown dynamics. Our descriptor plays the role of the hardware vector in these works, but it is defined in the planner’s configuration space rather than learned, which allows the teacher to consume exactly the same quantity. RMA’s online estimation is relevant to the dynamic part of our descriptor (Section 3), whose failure under an IMU fault is one of the recorded failure cases.

2.4 Distilling planners and privileged teachers into policies

Knowledge distillation [19] trains a compact student on the softened outputs of a larger teacher, and policy distillation [20] transfers this idea to control. Because a student trained only on teacher-visited states suffers from compounding errors, DAgger [21] aggregates teacher labels on states visited by the learner; we adopt this scheme. Several robotics systems distill an expensive or privileged expert into a sensorimotor policy: Pan et al. [22] distill an MPC expert for agile off-road driving, Learning by Cheating [23] distills a privileged driving agent into a vision-based one, Deep Drone Acrobatics [24] and Learning High-Speed Flight in the Wild [25] distill privileged trajectory planners into onboard quadrotor policies, and teacher-student training is standard in legged locomotion [26, 27]. These works distill for a single embodiment; the present work adds the embodiment descriptor as an input to both teacher and student and asks the student to generalize across it. Our consistency regularizer is related to Sobolev training [28], which matches a teacher’s derivatives in addition to its outputs; here the derivative is taken with respect to the descriptor and approximated by finite differences.

2.5 Legged locomotion and navigation

The students in this work command velocity-tracking locomotion controllers. Massively parallel reinforcement learning in GPU simulation [29] made such controllers practical for quadrupeds, and later work combined locomotion and local navigation end-to-end [30], learned state representations for navigation in cluttered and dynamic scenes [31], and produced agile parkour behaviors [32, 33, 34]. DTC [35] combines trajectory optimization with learned tracking. For humanoids, reinforcement learning has produced real-world walking controllers [36] and open simulation-to-reality pipelines [37]. These works mostly couple navigation and locomotion for one body. We deliberately keep the locomotion controller platform-specific and distill only the local planner, so that the shared component is the one for which a configuration abstraction exists.

2.6 Simulation, randomization, and safety filters

Our training and evaluation are conducted in Isaac Lab, which builds on Isaac Gym [38] and Orbit [39]. Domain randomization [40] and dynamics randomization [41] motivate randomizing the descriptor and the sensing pipeline during distillation. Kadian et al. [42] caution that simulation rankings can poorly predict real-world rankings, which is why onboard measurements are reported separately. The teacher’s refinement stage follows model predictive path integral control [43], and the onboard envelope check is a discrete-time analogue of the safety filters formalized with control barrier functions [44]. The student’s descriptor pathway uses feature-wise linear modulation (FiLM) [45].

03 Problem Formulation

Embodiment descriptor. An embodiment is described by a vector $\mathbf{c} = (\mathbf{c}^s, \mathbf{c}^d) \in \mathcal{C} \subset \mathbb{R}^{12}$. The static part

$$\mathbf{c}^s = (h, l_f, l_r, w, z_s, v_{\max}, \omega_{\max}, a_{\max}, \Delta h_{\text{step}}, \theta_{\max}) \quad (1)$$

contains the collision-relevant height h , the front length l_f , rear length l_r , and half width w of the planar collision envelope (the AgniNav envelope [1]), the sensor mounting height z_s , the

navigation limits on forward speed $v_{\max}$, yaw rate $\omega_{\max}$, and acceleration $a_{\max}$, the maximum traversable step height Δh_{step} , and the maximum traversable slope $\theta_{\max}$. The dynamic part $\mathbf{c}^d = (\eta, \phi)$ is computed online by the state estimator: $\eta \in (0, 1]$ is a speed derating factor and ϕ is the estimated terrain pitch under the body, obtained from the IMU attitude and leg kinematics. The derating models payload effects on achievable speed,

$$\eta = \max(\eta_{\min}, 1 - \kappa m_p / m_b), \quad (2)$$

where m_p is the payload mass, m_b the nominal body mass, κ a platform constant identified from velocity-tracking logs, and $\eta_{\min}$ a lower bound. The effective speed limit is $\eta v_{\max}$.

Observations and trajectories. At decision step t the robot observes $\mathbf{o}_t = (M_t, g_t, \nu_t)$, where $M_t \in \mathbb{R}^{64 \times 64}$ is an ego-centric elevation map at 0.1 m resolution built from LiDAR returns, $g_t \in \mathbb{R}^2$ is the local goal in the body frame, and $\nu_t = (v_x, v_y, \omega)$ is the body velocity. A local trajectory is a sequence of K body-frame poses $\tau = (p_1, \dots, p_K)$ with $p_k = (x_k, y_k, \psi_k)$ at times $k\Delta$. We use $K = 10$ and $\Delta = 0.2$ s, a 2 s horizon that matches the 5 Hz sensor rate of the Go2.

Feasibility. Let $B(\mathbf{c}) = [-l_r, l_f] \times [-w, w]$ be the planar envelope and $\mathcal{O}(M; \mathbf{c})$ the obstacle set extracted from M for embodiment $\mathbf{c}$: cells whose height above the local ground exceeds Δh_{step} and lies below h , together with cells whose local slope exceeds $\theta_{\max}$. The clearance of τ is

$$d(\tau; M, \mathbf{c}) = \min_{k=1, \dots, K} \text{sd}(T_{p_k} B(\mathbf{c}), \mathcal{O}(M; \mathbf{c})), \quad (3)$$

where T_{p_k} applies the pose p_k and sd is the signed distance between sets (negative under penetration). A trajectory is feasible for $\mathbf{c}$ if $d(\tau; M, \mathbf{c}) \geq d_{\text{safe}}(\mathbf{c})$ and it respects the kinematic limits

$$\begin{aligned} \frac{\|p_{k+1}^{xy} - p_k^{xy}\|}{\Delta} &\leq \eta v_{\max}, & \frac{|\psi_{k+1} - \psi_k|}{\Delta} &\leq \omega_{\max}, \\ \frac{\|p_{k+1}^{xy} - 2p_k^{xy} + p_{k-1}^{xy}\|}{\Delta^2} &\leq a_{\max}, \end{aligned} \quad (4)$$

for all k , with p_0 the current pose and p_{-1} extrapolated from ν_t .

Teacher and student. The teacher is a configuration-driven planner $\mathcal{T} : (\mathbf{o}, \mathbf{c}) \mapsto \tau^*$. The student is a parametric policy $\pi_\theta : (\mathbf{o}, \mathbf{c}) \mapsto \tau_\theta$. Let $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{test}}$ be pairwise disjoint descriptor sets. The distillation problem is

$$\begin{aligned} \min_{\theta} \mathbb{E}_{\mathbf{c} \sim \mathcal{D}_{\text{test}}} \mathbb{E}_{\mathbf{o} \sim \rho_{\pi_\theta, \mathbf{c}}} [\ell(\pi_\theta(\mathbf{o}, \mathbf{c}), \mathcal{T}(\mathbf{o}, \mathbf{c}))] \\ \text{s.t. } L(\pi_\theta) \leq L_{\text{budget}}, \end{aligned} \quad (5)$$

where $\rho_{\pi_\theta, \mathbf{c}}$ is the state distribution induced by the student on embodiment $\mathbf{c}$, ℓ is a trajectory discrepancy, and $L(\pi_\theta)$ is the onboard decision latency. Only $\mathcal{D}_{\text{train}}$ may provide supervised labels; $\mathcal{D}_{\text{val}}$ may be used only for model selection; $\mathcal{D}_{\text{test}}$ is used only for reporting.

Body-normalized coordinates. To make regression targets comparable across bodies, positions are expressed relative to the distance reachable within the horizon,

$$\bar{p}_k^{xy} = \frac{p_k^{xy}}{\eta v_{\max} K \Delta}, \quad \bar{\psi}_k = \frac{\psi_k}{\omega_{\max} K \Delta}, \quad (6)$$

so that any kinematically feasible trajectory lies in the unit box. The shared configuration space is the product of this normalized trajectory space and the descriptor space $\mathcal{C}$.

04 Method

4.1 Teacher: configuration-driven local planner

The teacher follows the AgniNav structure [1]: it plans over a candidate library scaled by the descriptor and checks collisions with the descriptor’s envelope. The library $\mathcal{L}(\mathbf{c}) = \{A_j(\mathbf{c})\}_{j=1}^J$ contains $J = 135$ constant-curvature arcs generated from a 9×15 grid of commanded speeds $v \in [0, \eta v_{\max}]$ and yaw rates $\omega \in [-\omega_{\max}, \omega_{\max}]$, each ramped from the current velocity under the acceleration limit $a_{\max}$. Every candidate is scored by

$$\begin{aligned} J(\tau) = & w_g \frac{\|p_K^{xy} - g\|}{\eta v_{\max} K \Delta} \\ & + w_c \sum_{k=1}^K e^{-d_k/\sigma_c} \\ & + w_s \sum_{k=1}^{K-1} \|\bar{p}_{k+1} - 2\bar{p}_k + \bar{p}_{k-1}\|^2 \\ & + w_\psi |\psi_K - \angle g|, \end{aligned} \quad (7)$$

where d_k is the per-pose clearance inside the minimum of (3), σ_c sets the decay of the clearance cost, $\angle g$ is the bearing of the goal, and w_g, w_c, w_s, w_ψ are fixed weights shared across embodiments. Candidates violating the feasibility conditions are discarded. The eight lowest-cost survivors initialize a refinement stage based on model predictive path integral control [43], which perturbs the velocity sequence with Gaussian noise and updates it by the exponentially weighted average

$$u_k \leftarrow u_k + \frac{\sum_n e^{-J(\tau^{(n)})/\lambda} \epsilon_k^{(n)}}{\sum_n e^{-J(\tau^{(n)})/\lambda}}, \quad (8)$$

where u_k is the commanded (v, ω) at step k , $\epsilon_k^{(n)}$ is the n -th noise sample, and λ is a temperature. The refined trajectory is projected back onto the limits (4). The refinement, together with the per-candidate envelope sweep against the elevation map, dominates the teacher’s cost and is responsible for its 41 ms decision cycle.

Besides the trajectory τ^* , the teacher emits a soft distribution over the library,

$$q_j^*(o, \mathbf{c}) = \frac{\exp(-J(A_j(\mathbf{c}))/\beta)}{\sum_{j'} \exp(-J(A_{j'}(\mathbf{c}))/\beta)}, \quad (9)$$

with infeasible candidates assigned zero mass. The soft labels transmit the teacher’s ranking of alternatives, which is informative near decision boundaries such as the choice of side around an obstacle [19].

For label generation the teacher is run with privileged ground-truth elevation maps from the simulator and an enlarged refinement budget. This reference configuration defines the targets τ^* and q^* and the reference trajectories for the teacher parity score (Section 5.1).

4.2 Student architecture

The student must run within a few milliseconds on a Jetson Orin. It consists of three parts.

- *Map encoder.* Four strided convolutional blocks reduce M_t (with a second channel holding the local slope) to a 256-dimensional feature.
- *Descriptor pathway.* The descriptor is standardized, $\bar{\mathbf{c}} = (\mathbf{c} - \mu_{\mathcal{C}}) \odot \sigma_{\mathcal{C}}$, concatenated with a binary validity mask for $\mathbf{c}^d$, and mapped by a two-layer MLP to an embedding e . FiLM layers [45] modulate every encoder block, $z \leftarrow \gamma(e) \odot z + \beta(e)$, so that the descriptor changes how the map is interpreted rather than being appended only at the output.
- *Heads.* A score head outputs logits $s_\theta \in \mathbb{R}^J$ over the same configuration-scaled library, and a residual head outputs a correction $\Delta \bar{\tau}_\theta \in \mathbb{R}^{3K}$ in normalized coordinates. The student trajectory is $\bar{\tau}_\theta = \bar{A}_j(\mathbf{c}) + \Delta \bar{\tau}_\theta$ with $\hat{j} = \arg \max_j s_{\theta,j}$, mapped back to metric coordinates by inverting (6).

Anchoring the output to a library arc keeps the residual small and keeps the student’s output inside the kinematic envelope of the target body even when the descriptor is unseen, because the arcs themselves are generated from $\mathbf{c}$ analytically.

4.3 Distillation objective

The training loss is

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}} + \lambda_{\text{feas}} \mathcal{L}_{\text{feas}} + \lambda_{\text{cons}} \mathcal{L}_{\text{cons}}, \quad (10)$$

with the following terms.

Trajectory regression in body-normalized coordinates, $\mathcal{L}_{\text{traj}} = \sum_{k=1}^K \alpha_k \|\bar{p}_k^\theta - \bar{p}_k^*\|^2$, where α_k decreases linearly with k because only the first segment is executed before replanning.

Soft-label distillation, $\mathcal{L}_{\text{KL}} = \text{KL}(q^* \parallel \text{softmax}(s_\theta))$.

Envelope penetration, $\mathcal{L}_{\text{feas}} = \sum_k [d_{\text{safe}}(\mathbf{c}) - \tilde{d}_k(\tau_\theta)]_+^2$, where $\tilde{d}_k$ is a differentiable clearance obtained by bilinear interpolation of a Euclidean distance transform of $\mathcal{O}(M; \mathbf{c})$ at sample points on the boundary of $T_{p_k} B(\mathbf{c})$.

Finite-difference consistency. Labels at individual descriptors do not constrain how the student’s output changes between them, which is precisely what matters on held-out descriptors. For each training sample a perturbed descriptor $\mathbf{c}' = \mathbf{c} + \delta$, $\delta \sim \mathcal{N}(0, \text{diag}(\sigma_\delta^2))$ restricted to $\mathcal{D}_{\text{train}}$, is drawn, the teacher is queried at $(o, \mathbf{c}')$, and

$$\mathcal{L}_{\text{cons}} = \left\| (\bar{\tau}_\theta(o, \mathbf{c}) - \bar{\tau}_\theta(o, \mathbf{c}')) - (\bar{\tau}^*(o, \mathbf{c}) - \bar{\tau}^*(o, \mathbf{c}')) \right\|^2. \quad (11)$$

To first order this matches the Jacobian of the student with respect to the descriptor to that of the teacher along δ , analogous to Sobolev training [28]. When the teacher’s choice switches homotopy class between $\mathbf{c}$ and $\mathbf{c}'$ (for example, a gap becomes impassable), the pair is excluded from (11), because the finite difference is then not informative about a derivative.

4.4 Data generation, aggregation, and held-out splits

Descriptor family. Descriptors are sampled from two platform-centered regions (quadruped-like and humanoid-like) and from the segment between them in standardized descriptor space, with envelope dimensions, velocity limits, step height, slope limit, and derating randomized around the platform values in Table 1. The regions are partitioned into axis-aligned boxes assigned to $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{test}}$; the evaluated Go2 and humanoid descriptors with their test-time perturbations lie in $\mathcal{D}_{\text{test}}$.

Table 1: Nominal static descriptor values for the two platforms. These are configuration settings of the planner, not measured results.

Parameter	Go2	H1-class humanoid
Envelope height h (m)	0.40	1.80
Front / rear length l_f / l_r (m)	0.38 / 0.32	0.20 / 0.15
Half width w (m)	0.16	0.30
Sensor height z_s (m)	0.30	1.60
Speed limit $v_{\max}$ (m/s)	1.5	1.0
Yaw-rate limit $\omega_{\max}$ (rad/s)	1.5	1.0
Acceleration limit $a_{\max}$ (m/s ²)	1.5	0.8
Step height Δh_{step} (m)	0.15	0.10
Slope limit $\theta_{\max}$ (deg)	30	15

Held-out embodiment training split. $\mathcal{D}_{\text{val}}$ is never used for gradient updates. After each aggregation round, the teacher parity of the current student on $\mathcal{D}_{\text{val}}$ is computed, and this quantity selects the checkpoint and anneals λ_{cons} (it is increased when parity on $\mathcal{D}_{\text{val}}$ lags parity on $\mathcal{D}_{\text{train}}$). Selecting on held-out embodiments rather than held-out episodes of training embodiments is the training-time counterpart of the evaluation protocol.

Aggregation. Training follows DAGger [21]. Round 0 rolls out the teacher; in round $i \geq 1$ the executed action is the teacher’s with probability $\beta_i = \beta_0^i$ and the student’s otherwise, and the teacher labels every visited state. Sensing is randomized during rollouts (LiDAR dropout, elevation-map noise, odometry drift) following domain randomization practice [40, 41]. The dynamic descriptor entries are masked with a small probability during training, with the validity mask set accordingly.

4.5 Onboard execution and watchdog

Decision loop. On each 5 Hz sensor frame (a 200 ms period) the Go2 updates M_t , assembles $\mathbf{c}$ from its configuration file and state estimator, and runs the student compiled with TensorRT in half precision. The output trajectory is checked once against the current map using (3). If it violates $d_{\text{safe}}(\mathbf{c})$, the library arcs are checked in decreasing order of s_θ and the first feasible one is used. This single-trajectory check acts as a discrete-time safety filter in the spirit of control barrier function filters [44], but at a cost that is linear in the number of arcs actually examined rather than in the full library with refinement.

Tracking. A waypoint tracker runs at 50 Hz (20 ms period) and produces body velocity commands

$$\begin{bmatrix} v_x^{\text{cmd}} \\ v_y^{\text{cmd}} \\ \omega^{\text{cmd}} \end{bmatrix} = \text{clip}_{\mathbf{c}} \left(\begin{bmatrix} v_x^{\text{ref}} \\ v_y^{\text{ref}} \\ \omega^{\text{ref}} \end{bmatrix} + K_p \begin{bmatrix} R(\psi)^{\top} (p_{\text{ref}}^{xy} - p^{xy}) \\ \psi_{\text{ref}} - \psi \end{bmatrix} \right), \quad (12)$$

where the reference is interpolated from τ_θ , $R(\psi)$ is the planar rotation, K_p is a diagonal gain, and $\text{clip}_{\mathbf{c}}$ enforces the limits of $\mathbf{c}$. The commands are passed to a platform-specific learned locomotion controller trained in Isaac Lab in the manner of [29] for the Go2 and [36, 37] for the humanoid. The latency budget matters because the plan is applied to a state that has moved on: at the Go2’s 1.5 m/s limit, a 41 ms decision delay corresponds to about 6.2 cm of travel, and an 8.2 ms delay to about 1.2 cm.

Watchdog. A watchdog monitors three conditions: (i) staleness or out-of-range values in $\mathbf{c}^d$, (ii) rejection of every exam-

ined library arc on two consecutive frames, and (iii) tracking error above a threshold. When any condition holds, it hands control to a fallback that decelerates at $a_{\max}$ to a stop and then replans with the teacher’s library stage using the nominal descriptor with $\eta = \eta_{\min}$, at a reduced rate. The watchdog’s detection window is finite, and behavior during that window is examined in Section 5.4.

IMPLEMENTATION NOTE

The teacher, the student, and the watchdog all read the same descriptor file format; changing the embodiment on the robot changes only this file. Teacher labels can be regenerated for any descriptor, so new training regions can be added without collecting robot data.

5 Experiments

5.1 Setup and protocol

Platforms. The study’s testbed is Isaac Lab with two simulated platforms: a Unitree Go2 and a 12-DoF research humanoid of the H1 class. Scenes are 20 indoor–outdoor transition scenes (doorways, thresholds, ramps, corridors, and turns between indoor and outdoor areas), each evaluated with 5 seeds and 50 episodes per seed, that is, $20 \times 5 \times 50 = 5,000$ episodes per evaluated configuration. Onboard evaluation uses the Go2 at a 5 Hz sensor rate with the student policy deployed on its onboard Jetson Orin computer, over 30 held-out runs.

Compared methods. (i) *Per-embodiment teacher planner (oracle)*: the configuration-driven teacher run online and instantiated with the exact descriptor of each test embodiment, using onboard-equivalent perception; its full decision cycle is 41 ms. (ii) *Student, no embodiment conditioning*: a student of the same architecture without the descriptor pathway, trained on teacher labels computed for a single conservative envelope that bounds both platforms, so that its labels are consistent across bodies. (iii) *Student, seen embodiments only*: a descriptor-conditioned student trained by DAGger on the two nominal platform descriptors only, without the descriptor family. (iv) *Ours*: the full pipeline of Section 4.

Metrics. *Navigation success* is the percentage of episodes that reach within a goal tolerance without collision, fall, or timeout. *Path clearance margin* is the minimum of (3) along the executed path, averaged over episodes. *Onboard decision latency* is the time from receipt of a sensor frame to availability of a trajectory on the Jetson Orin. *Teacher parity score* is the percentage of decision steps at which the executed trajectory stays within a fixed waypoint tolerance of the reference teacher trajectory (privileged map, enlarged budget) computed at the same state and passes obstacles on the same side. Because the oracle planner itself fails scene 20 for the humanoid, parity is computed over the 19 solvable scenes.

5.2 Main comparison

Table 2 reports the main comparison.

Success and parity. The full student reaches 90.4% navigation success, which is 8.1 percentage points above the

Table 2: Main comparison on held-out embodiment descriptors. Success, clearance, and parity are from the Isaac Lab protocol (20 scenes $\times$ 5 seeds $\times$ 50 episodes); latency is measured on the Go2’s Jetson Orin. Parity is computed over the 19 solvable scenes.

Method	Navigation success (%)	Path clearance margin (cm)	Onboard decision latency (ms)	Teacher parity score (%)
Per-embodiment teacher planner (oracle, 41 ms decision)	81.5	19.4	8.7	89.6
Student, no embodiment conditioning	82.3	20.3	9.4	86.3
Student, seen embodiments only	80.8	20.3	9.4	78.9
Ours	90.4	18.7	8.2	93.1

unconditioned student (82.3%), 9.6 points above the seen-embodiments-only student (80.8%), and 8.9 points above the oracle teacher (81.5%). It also has the highest teacher parity, 93.1%, against 89.6% for the oracle, 86.3% for the unconditioned student, and 78.9% for the seen-embodiments-only student. The 14.2-point parity gap between the full student and the seen-embodiments-only student is the largest difference in the table and is consistent with the view that training on a descriptor family, rather than on the two nominal bodies, is what allows the student to track the teacher on held-out descriptors.

That a distilled student exceeds the success rate of an oracle teacher is unusual and should be interpreted with care. Two factors in the recorded data are compatible with it. First, the oracle is not perfect: it fails scene 20 (a narrow U-turn) for the humanoid, which lowers its success on the full 20-scene protocol, whereas parity is restricted to the 19 solvable scenes. Second, the oracle’s parity of 89.6% shows that the online teacher, running with onboard-equivalent perception and a finite budget, already deviates from the reference teacher that generated the labels; a student distilled from the reference teacher on student-visited states can therefore be closer to the reference than the online teacher is. The data do not establish which factor dominates, and we do not claim that the student is generally a better planner than its teacher.

Clearance. The full student has the smallest mean clearance margin, 18.7 cm, compared with 19.4 cm for the oracle and 20.3 cm for both alternative students. The differences are small (at most 1.6 cm) but consistent in direction. One plausible contributor, which the recorded data do not isolate, is that the full student succeeds in more narrow passages, whose low clearances enter the average; another is that the unconditioned student is trained on a conservative bounding envelope by design. In either case, the gain in success is not accompanied by a gain in clearance, and the margin should be monitored in deployment.

Latency. On the Jetson Orin, the full student’s decision latency is 8.2 ms, against 9.4 ms for both alternative students and 8.7 ms in the latency column of the oracle row. The 8.2 ms latency is about 4% of the 200 ms sensor period. The oracle row records two different figures: a 41 ms decision time for the full teacher planner (row label) and 8.7 ms in the latency column. The available logs do not resolve the measurement scope of the 8.7 ms entry, so both values are reported verbatim and no latency comparison with the oracle is drawn from the table. Relative to the 41 ms teacher decision cycle, the student’s 8.2 ms corresponds to a factor of 5.0 by arithmetic; this comparison should be read with the same caveat. The difference between the full student and the other students (1.2 ms) is small; the latency advantage of the distilled approach lies mainly against the full teacher.

Table 3: Ablation of the distillation pipeline (navigation success, Isaac Lab protocol).

Variant	Navigation success (%)
Ours (full distillation)	90.4
w/o configuration descriptor input	75.1
w/o consistency regularizer	79.1
w/o held-out embodiment training split	77.6

5.3 Ablation

Table 3 removes one training component at a time from the full pipeline.

Removing the configuration descriptor input causes the largest drop, 15.3 percentage points (90.4% to 75.1%). This variant differs from the unconditioned baseline of Table 2 (82.3%): it is trained on embodiment-specific teacher labels, so identical observations receive different targets for different bodies. Its lower success is consistent with the observation–action ambiguity described by EA-Nav [3], which the conservative-envelope baseline avoids by construction. Removing the consistency regularizer costs 11.3 points (79.1%), supporting its role in shaping how the student’s output changes between training descriptors. Removing the held-out embodiment training split costs 12.8 points (77.6%). This variant falls below the seen-embodiments-only baseline (80.8%), even though it trains on the full descriptor family. The recorded data do not isolate the cause. A hypothesis consistent with the design is that, without selection and annealing on $\mathcal{D}_{\text{val}}$, the chosen checkpoint and the fixed consistency weight favor fit on training descriptors at the expense of held-out ones; confirming this would require parity curves on $\mathcal{D}_{\text{val}}$ that are not part of the reported results.

5.4 Failure cases

Steep entry ramp (scene 17). In the condition where the Go2 student is trained from teacher labels generated for the humanoid configuration, the clearance margin dropped to 9 cm in 2 of the 30 held-out runs at scene 17, which begins with a steep entry ramp. This is less than half of the full student’s mean margin of 18.7 cm. A ramp entry is where the descriptor’s slope limit, the low sensor height of the Go2, and the elevation-map classification of the ramp edge interact; humanoid-configured labels encode a different slope limit and sensor viewpoint, and the residual head appears unable to fully correct for this near the transition. The clearance stayed positive in these runs, but the margin is too small for comfort and indicates that the slope-related entries of the descriptor are among the least well covered by the training family.

Descriptor dropout (simulated IMU fault). When the dynamic descriptor entries drop out because of a simulated IMU fault, the student degrades by 11% in success before the watch-

dog hands off to the fallback. The degradation occurs inside the watchdog’s detection window, during which the student acts on stale or corrupted values of η and ϕ . Training-time masking of $\mathbf{c}^d$ did not prevent this degradation. Shortening the detection window, or making the student default to conservative values of $\mathbf{c}^d$ when the validity mask is uncertain, are direct remedies that remain to be tested.

Teacher failure on a narrow U-turn (scene 20). The oracle planner fails scene 20 for the humanoid. A narrow U-turn requires rotating a long envelope in a confined space; with a constant-curvature library and a yaw-rate limit, no candidate in the teacher’s library is feasible throughout the maneuver. Because a student cannot be expected to imitate a teacher that has no feasible plan, teacher parity is computed over the 19 solvable scenes. The case also shows that distillation inherits the teacher’s incompleteness: the library would need in-place rotation or multi-point turn primitives for the humanoid.

06 Limitations

The evaluation has several limits. The main comparison and the ablations are based on simulation in Isaac Lab; onboard measurements cover the Go2 only, over 30 held-out runs, and the humanoid is evaluated only in simulation. Only two platform families and descriptor perturbations around them are studied, so claims about generalization to substantially different bodies (wheeled bases, larger quadrupeds) are not supported by the present data. The results report point estimates without confidence intervals across seeds. The oracle row’s two latency figures (41 ms and 8.7 ms) could not be reconciled from the logs, which weakens the latency comparison with the teacher. The full student trades a slightly smaller clearance margin for its higher success, and the steep-ramp case shows that the margin can fall to 9 cm in held-out runs. The ablation leaves the mechanism behind the held-out split’s effect unresolved. Finally, the student inherits the teacher’s planning primitives, and therefore its failure on the narrow U-turn.

07 Conclusion

This preprint presented a pipeline for distilling a configuration-driven local planner into a compact, embodiment-conditioned onboard policy for legged robot teams. Teacher and student share a body-normalized configuration space and a single descriptor, and training combines DAGger, soft-label distillation over a configuration-scaled library, an envelope-penetration penalty, a finite-difference consistency regularizer, and model selection on held-out embodiments. In Isaac Lab with a Uni-tree Go2 and an H1-class humanoid, the student reached 90.4% navigation success with 93.1% teacher parity and an 8.2 ms onboard decision latency on the Go2’s Jetson Orin, while its mean clearance margin of 18.7 cm was the smallest among the compared methods. Each training component contributed between 11.3 and 15.3 points of success in the ablation. The recorded failures—reduced clearance on a steep ramp, degradation before watchdog hand-off under descriptor dropout, and inherited teacher incompleteness on a narrow U-turn—identify the next steps: better coverage of slope-related descriptor entries, faster descriptor fault detection, richer teacher primitives, and larger onboard evaluations that include the humanoid.

References

- [1] T. Zang, S. Cheng, H. Huang, S. Wang, and W. Zhang. Agni-Nav: Configuration-Driven Cross-Embodiment Local Planning for Robot Navigation. arXiv:2606.10903, 2026.
- [2] 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.
- [3] 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.
- [4] X. Xiao, B. Liu, G. Warnell, and P. Stone. Motion Planning and Control for Mobile Robot Navigation Using Machine Learning: a Survey. arXiv:2011.13112, 2020.
- [5] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A General Navigation Model to Drive Any Robot. arXiv:2210.03370, 2022.
- [6] 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.
- [7] A. Sridhar, D. Shah, C. Glossop, and S. Levine. NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration. arXiv:2310.07896, 2023.
- [8] A. Eftekhari et al.. The One RING: a Robotic Indoor Navigation Generalist. arXiv:2412.14401, 2024.
- [9] J. Zhang et al.. Embodied Navigation Foundation Model. arXiv:2509.12129, 2025.
- [10] J. Yang, C. Glossop, A. Bhorkar, D. Shah, Q. Vuong, C. Finn, D. Sadigh, and S. Levine. Pushing the Limits of Cross-Embodiment Learning for Manipulation and Navigation. arXiv:2402.19432, 2024.
- [11] 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.
- [12] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. arXiv:2310.08864, 2023.
- [13] T. Chen, A. Murali, and A. Gupta. Hardware Conditioned Policies for Multi-Robot Transfer Learning. arXiv:1811.09864, 2018.
- [14] W. Huang, I. Mordatch, and D. Pathak. One Policy to Control Them All: Shared Modular Policies for Agent-Agnostic Control. arXiv:2007.04976, 2020.
- [15] A. Gupta, L. Fan, S. Ganguli, and L. Fei-Fei. MetaMorph: Learning Universal Controllers with Transformers. arXiv:2203.11931, 2022.
- [16] G. Feng et al.. GenLoco: Generalized Locomotion Controllers for Quadrupedal Robots. arXiv:2209.05309, 2022.
- [17] M. Shafiee, G. Bellegarda, and A. Ijspeert. ManyQuadrupeds: Learning a Single Locomotion Policy for Diverse Quadruped Robots. arXiv:2310.10486, 2023.
- [18] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. arXiv:2107.04034, 2021.
- [19] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. arXiv:1503.02531, 2015.
- [20] A. A. Rusu et al.. Policy Distillation. arXiv:1511.06295, 2015.
- [21] S. Ross, G. J. Gordon, and J. A. Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. arXiv:1011.0686, 2010.
- [22] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Boots. Agile Autonomous Driving using End-to-End Deep Imitation Learning. arXiv:1709.07174, 2017.

- [23] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by Cheating. arXiv:1912.12294, 2019.
- [24] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza. Deep Drone Acrobatics. arXiv:2006.05768, 2020.
- [25] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza. Learning High-Speed Flight in the Wild. arXiv:2110.05113, 2021.
- [26] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. arXiv:2010.11251, 2020.
- [27] 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.
- [28] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu. Sobolev Training for Neural Networks. arXiv:1706.04859, 2017.
- [29] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. arXiv:2109.11978, 2021.
- [30] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced Skills by Learning Locomotion and Local Navigation End-to-End. arXiv:2209.12827, 2022.
- [31] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter. Learning a State Representation and Navigation in Cluttered and Dynamic Environments. arXiv:2103.04351, 2021.
- [32] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. arXiv:2306.14874, 2023.
- [33] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme Parkour with Legged Robots. arXiv:2309.14341, 2023.
- [34] Z. Zhuang et al.. Robot Parkour Learning. arXiv:2309.05665, 2023.
- [35] F. Jenelten, J. He, F. Farshidian, and M. Hutter. DTC: Deep Tracking Control. arXiv:2309.15462, 2023.
- [36] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath. Real-World Humanoid Locomotion with Reinforcement Learning. arXiv:2303.03381, 2023.
- [37] X. Gu, Y.-J. Wang, and J. Chen. Humanoid-Gym: Reinforcement Learning for Humanoid Robot with Zero-Shot Sim2Real Transfer. arXiv:2404.05695, 2024.
- [38] V. Makoviychuk et al.. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. arXiv:2108.10470, 2021.
- [39] M. Mittal et al.. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. arXiv:2301.04195, 2023.
- [40] 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.
- [41] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. arXiv:1710.06537, 2017.
- [42] A. Kadian et al.. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? arXiv:1912.06321, 2019.
- [43] G. Williams, A. Aldrich, and E. Theodorou. Model Predictive Path Integral Control using Covariance Variable Importance Sampling. arXiv:1509.01149, 2015.
- [44] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control Barrier Functions: Theory and Applications. arXiv:1903.11199, 2019.
- [45] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. FiLM: Visual Reasoning with a General Conditioning Layer. arXiv:1709.07871, 2017.