

Distilling Cross-Embodiment Planners into Onboard-Deployable Policies

Rongxin Guo • Xiaotong Zhang • Yifan Chen • Daichi Mori

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

PUBLISHED

September 3, 2026

REGISTERED

September 9, 2026

RECORD TYPE

Preprint

ABSTRACT

Configuration-driven cross-embodiment local planners such as AgniNav allow a single planning stack to serve robots with different body geometry, sensing, and velocity limits, but they pay for this generality with full configuration-space sampling at every replan. On an embedded computer this cost is prohibitive: the full planner requires 41 ms per decision offline, and throttling it onboard degrades agreement with its unthrottled output. This paper distills such a planner into a configuration-conditioned onboard policy through trajectory-level imitation. The teacher produces local trajectories over configuration states; the student maps an observation and a configuration descriptor directly to the next body-motion primitive from a velocity-normalized library. Training combines a soft-target imitation term with a consistency regularizer that penalizes, under perturbed range observations, probability mass on primitives that diverge geometrically from the teacher decision. Because the student remains conditioned on the descriptor, one deployed network serves multiple embodiments. In Isaac Lab experiments with a Unitree Go2 across 18 scenes and 5 seeds, with the student running at 100 Hz on an onboard Orin, the distilled policy reaches a p95 decision latency of 7.8 ms, a teacher parity of 94.3 percent, and a success rate of 91.7 percent, compared with 8.7 ms, 84.2 percent, and 86.3 percent for the throttled full planner. Transfer to a B2-class morphology in simulation retains 89.6 percent parity, and at 10 percent range dropout the regularizer preserves 88.1 percent parity versus 79.4 percent without it.

INDEX TERMS cross-embodiment navigation, policy distillation, imitation learning, legged robots, onboard deployment, local planning

KAGG NOTE A configuration-driven planner can be compressed into one configuration-conditioned policy that fits a 10 ms onboard decision budget, provided the distillation objective penalizes geometric divergence from teacher decisions under sensor perturbation rather than label mismatch alone.

AUTHOR AFFILIATIONS

R. Guo, X. Zhang, and Y. Chen: Robotics and Autonomous Systems Lab, Shanghai Jiao Tong University, Shanghai, China. D. Mori: Legged Systems Group, The University of Tokyo, Tokyo, Japan.

01 Introduction

Legged robots are increasingly deployed as heterogeneous fleets: a laboratory or an inspection operator may run a compact quadruped next to a larger, heavier one, and each platform may carry different sensors or payloads over its service life. Configuration-driven local planners address this heterogeneity at the level of the planning stack. AgniNav [1] and the related CeRLP framework [2] separate the estimate of environment structure from an explicit, declared robot configuration—body geometry, sensor placement, and velocity limits—so that one planner can be reused across embodiments by changing a descriptor rather than retraining or re-tuning. The price of this generality is computational. At every replan, the planner samples candidate trajectories over the configuration-feasible set of the declared body, propagates them, checks them against the body envelope, and scores them. The cost grows with the number of samples and with the complexity of the envelope, and it is paid at every control decision.

On the platform studied here this cost is incompatible with onboard execution. The full configuration-driven planner requires 41 ms per decision when run offline, whereas the onboard decision loop of the robot runs at 100 Hz and therefore has a 10 ms budget. Throttling the planner to fit the embedded computer is not a satisfactory remedy: the run-server logs for the throttled planner on the onboard Orin record a p95 decision time of 38 ms, 34% of replans dropped, and only 81% agreement with the unthrottled planner on the replans that do complete (Section 5.5). A planner whose decisions arrive late, or not at all, forfeits the exactness that motivated it.

Learned navigation policies are fast at inference, but the two dominant ways of obtaining them leave a gap. Generalist navigation models trained on heterogeneous multi-robot data, such as GNM, ViNT, and NoMaD [3, 4, 5], absorb embodiment differences implicitly and do not expose the explicit configuration interface that makes configuration-driven planners auditable. Privileged-teacher distillation, as in Learning by Cheating [6], high-speed flight [7], and quadrupedal locomotion over challenging terrain [8], produces fast and robust students, but the student is usually specialized to the single platform on which the teacher was run. What is missing is a distillation procedure whose student inherits the *configuration interface* of a cross-embodiment planner—so that one deployed network still serves several bodies—while meeting a hard onboard latency

budget and remaining robust to the sensor degradation that the teacher’s dense sampling tolerated.

This paper develops such a procedure. The teacher is a configuration-driven planner in the style of AgniNav, run offline without time limits. The student is a compact network that maps an observation and a configuration descriptor to a categorical distribution over a library of body-motion primitives whose velocities are normalized by the descriptor’s limits. The student is trained by trajectory-level imitation on states aggregated from its own rollouts and relabeled by the teacher, with a consistency regularizer that penalizes probability mass on primitives whose swept trajectories diverge from the teacher decision when the student’s range observations are perturbed. The contributions are as follows.

1. A formulation of planner distillation as primitive selection conditioned on the robot configuration, in which a velocity-normalized primitive library and feature-wise descriptor conditioning allow a single network to serve embodiments with different envelopes and velocity limits (Sections 3 and 4).
2. A trajectory-level imitation objective that combines soft teacher targets derived from a configuration-weighted trajectory discrepancy with a consistency regularizer that is explicitly tied to range-query perturbations, together with a data-aggregation procedure that relabels student-visited states with the offline teacher.
3. An onboard deployment on a Unitree Go2 with an Orin computer at 100 Hz, in which the distilled policy attains a p95 decision latency of 7.8 ms, 94.3% teacher parity, and 91.7% success, compared with 8.7 ms, 84.2%, and 86.3% for the throttled full planner.
4. An ablation, sensor-noise, and cross-morphology analysis with documented failure cases, which identifies the consistency regularizer as the term responsible for robustness to range dropout (88.1% versus 79.4% parity at 10% dropout) and contact timing as the factor that the configuration descriptor does not capture in transfer to a B2-class morphology (89.6% parity).

02 Related Work

2.1 Configuration-driven and cross-embodiment local planning

AgniNav [1] is the planner this work distills, and its structure determines what the student must preserve. AgniNav represents the robot by an abstract configuration rather than by a platform-specific model. A shared collision envelope, parameterized by front length, rear length, half width, and a collision-relevant height, is declared per robot; the height conditions the perception module that converts observations into a planar, scan-like representation of obstacles, and the planar envelope parameters together with the velocity limits configure trajectory generation and collision checking in the local planner. A new embodiment is therefore supported by supplying a new configuration, and the same perception and planning code runs unchanged. The present work keeps this configuration as the interface of the learned student, so that the deployment-time contract—declare the body, receive body-appropriate motion—is unchanged. It differs from AgniNav in where the computation happens: AgniNav enumerates and checks candidate trajectories at runtime for

every decision, whereas the student amortizes that search into a single forward pass learned offline from AgniNav-style decisions. The distillation also inherits AgniNav’s limitations; the student cannot be more correct than the planner it imitates, and the envelope abstraction does not model gait or contact timing, which reappears as a failure case in Section 5.5.

CeRLP [2] pursues the same goal from the perception side, converting heterogeneous visual inputs into a unified geometric representation with scale-corrected depth and height-adaptive scan abstraction before a local planner runs. It shows that a geometric intermediate layer makes cross-embodiment planning tractable, but like AgniNav it pays runtime planning cost per decision; the distillation approach here is complementary and could in principle be applied to a CeRLP-style teacher. EA-Nav [9] learns visual navigation policies conditioned on embodiment geometry tokens and trains a risk-aware correction on augmented high-risk trajectories. It shares with this work the view that observation-action ambiguity must be resolved by explicit embodiment input, but it learns from demonstrations rather than from a configuration-driven planner, and it does not target a hard decision-latency budget. RING [10] randomizes body size, rotation pivot, and camera configuration in simulation to train an indoor navigation generalist, and NavFoM [11] scales a unified navigation model across quadrupeds, drones, wheeled robots, and vehicles using camera-view and temporal identifiers. Both demonstrate that embodiment randomization and explicit identifiers improve transfer; the present student uses a measured physical descriptor instead of identifiers, which allows evaluation on a morphology whose descriptor lies outside the training identifiers.

GNM [3] trains a single goal-conditioned policy on trajectories from several robots with a normalized action space and temporal context, ViNT [4] extends this to a Transformer foundation model with adaptable goal modalities, and NoMaD [5] unifies goal reaching and exploration with a diffusion action decoder. Yang et al. [12] push this direction further by co-training navigation with manipulation data. These models are the natural alternative to distillation: rather than compressing one planner, they learn from many sources. Their action normalization is related to the velocity-normalized primitives used here, but they infer the embodiment implicitly from context, whereas the student receives it explicitly, which is what allows the teacher’s configuration-specific decisions to be imitated faithfully.

2.2 Multi-embodiment control policies

At the locomotion level, shared modular policies [13] and MetaMorph [14] control agents of differing morphology with message passing or Transformers over limb tokens. GenLoco [15], ManyQuadrupeds [16], and URMA [17] train single locomotion controllers that generalize across quadrupeds with different mass, leg length, and actuation by randomizing or encoding morphology. These works establish that a morphology-conditioned network can represent a family of controllers. The student here operates one level higher: it selects body-motion primitives that are then executed by a platform-specific locomotion controller. This separation is deliberate, because the contact-level differences addressed by these controllers are exactly those that a planar envelope cannot express, and Section 5.5 shows that they remain the dominant source of disagreement under morphology transfer.

2.3 Distillation of planners and privileged experts

Knowledge distillation [18] trains a compact student on the softened output distribution of a larger teacher, and policy distillation [19] applies the idea to reinforcement-learning agents, showing that a small network can match or exceed a teacher’s performance when trained on teacher action distributions. Czarnecki et al. [20] analyze the design space of policy distillation and show that the choice of sampling distribution—teacher-generated versus student-generated states—and of loss determines whether the student converges to the teacher’s behavior. DAGger [21] formalizes why student-generated states matter: training only on teacher trajectories leads to compounding errors, which data aggregation under the student’s own state distribution removes. The training procedure here follows this analysis, relabeling student-visited states with the offline teacher.

Planner and optimizer distillation has a long history in robotics. Guided policy search [22] and its MPC-guided variant [23] train neural policies to imitate trajectory optimizers that have access to full state, and Pan et al. [24] distill a sampling-based MPC expert into an end-to-end driving policy with online imitation, a direct precursor of the present setting. Deep drone acrobatics [25] and high-speed flight in the wild [7] distill privileged experts into sensorimotor students that run onboard. In legged locomotion, the teacher-student paradigm of Lee et al. [8] and Miki et al. [26], the adaptation module of RMA [27], and vision-based students for challenging terrain and parkour [28, 29, 30] all compress privileged information into deployable networks. Learning by Cheating [6] provides the clearest statement of the two-stage logic for navigation. In these works, however, the teacher and the student share one embodiment. The present work differs in that the teacher is itself a cross-embodiment method, and the student must reproduce how the teacher’s decisions change with the configuration, not only the decisions for one body.

2.4 Learned and sampling-based local navigation for legged robots

Learning has been applied to local planning both end to end, as in Pfeiffer et al. [31], and in hybrid forms that tune classical planners, as surveyed by Xiao et al. [32] and exemplified by APPLD [33], which learns planner parameters from demonstration. Sampling-based optimal control such as MPPI [34] provides the kind of exact but expensive local search that configuration-driven planners also perform. For legged platforms, reinforcement-learned locomotion [35, 36] provides the low-level velocity-tracking interface that a primitive-level planner commands; Rudin et al. [37] learn locomotion and local navigation jointly, ArtPlanner [38] combines sampling-based planning with learned reachability for field deployment, and Lee et al. [39] demonstrate hierarchical learned navigation for wheeled-legged robots at scale. Learned primitive and action-chunk representations, including diffusion policies [40], motivate the choice of a discrete primitive head over direct velocity regression. Conditioning by feature-wise linear modulation [41] is used for the descriptor input.

Training and evaluation use GPU-parallel simulation in the lineage of Isaac Gym [42] and Orbit [43]. Dynamics randomization [44] and the observation that simulator rankings need not predict real-world rankings [45] motivate both the inclusion of hardware runs and the caution, stated in Section 6, about conclusions drawn from the simulated second morphology.

03 Problem Formulation

Configuration descriptor. An embodiment is described by a vector

$$\mathbf{c} = [l_f, l_r, w, h, \bar{v}_x, \bar{v}_y, \bar{\omega}, z_s, r_{\max}]^T \in \mathcal{C} \subset \mathbb{R}^9 \quad (1)$$

where l_f and l_r are the front and rear lengths of the planar collision envelope measured from the body origin, w is its half width, h is the collision-relevant height, $\bar{v}_x$, $\bar{v}_y$, and $\bar{\omega}$ are the forward, lateral, and yaw-rate limits commanded to the locomotion controller, z_s is the height of the range sensor above the ground, and $r_{\max}$ is its maximum valid range. The first four entries are the AgniNav envelope; the remainder make the velocity and sensing assumptions explicit. The envelope occupies the body-frame rectangle $\mathcal{B}(\mathbf{c}) = [-l_r, l_f] \times [-w, w]$.

Observation. At decision step t the robot receives $o_t = (\mathbf{r}_t, \mathbf{g}_t, \boldsymbol{\nu}_t, k_{t-1})$, where $\mathbf{r}_t \in [0, r_{\max}]^{N_r}$ are N_r range queries taken at fixed bearings in the body frame from the height-filtered obstacle representation, $\mathbf{g}_t$ is the local goal in body-frame polar coordinates, $\boldsymbol{\nu}_t = (v_x, v_y, \omega)$ is the estimated body twist, and k_{t-1} is the index of the previously issued primitive.

Body-motion primitives. Let $\{\tilde{\mathbf{u}}_k\}_{k=1}^K$ be a fixed library of normalized twists with $\tilde{\mathbf{u}}_k \in [-1, 1]^3$. For a descriptor $\mathbf{c}$, primitive k commands the constant twist

$$\mathbf{u}_k(\mathbf{c}) = \text{diag}(\bar{v}_x, \bar{v}_y, \bar{\omega}) \tilde{\mathbf{u}}_k \quad (2)$$

for a primitive horizon T_p . Denote by $\Phi(k, \mathbf{c}) = (\mathbf{x}_1^k, \dots, \mathbf{x}_M^k)$ the planar poses obtained by integrating $\mathbf{u}_k(\mathbf{c})$ over T_p at M equally spaced times, starting from the current body frame. Because the library is normalized, the same index k denotes a proportionally equivalent maneuver on every embodiment, which is what allows one output head to serve several bodies.

Teacher. The teacher $\mathcal{T}$ is the configuration-driven planner. Given $(o_t, \mathbf{c})$ it samples a set $\mathcal{S}(\mathbf{c})$ of candidate local trajectories that respect the velocity and acceleration limits implied by $\mathbf{c}$, discards those whose swept envelope intersects obstacles, and returns

$$\tau_t^* = \arg \min_{\tau \in \mathcal{S}(\mathbf{c})} J(\tau; o_t, \mathbf{c}) \quad \text{s.t.} \quad \mathcal{B}(\mathbf{c}) \oplus \tau \cap \mathcal{O}_t = \emptyset \quad (3)$$

where $\mathcal{O}_t$ is the obstacle set implied by $\mathbf{r}_t$, $\oplus$ denotes the swept volume of the envelope along τ , and J is the cost defined in Section 4. The teacher’s computation scales with $|\mathcal{S}(\mathbf{c})|$ and is the quantity to be amortized.

Objective. The goal is a student policy $\pi_\theta(k | o, \mathbf{c})$ that (i) returns a primitive within the control period $T_c = 10$ ms of the 100 Hz loop with high probability, measured by the 95th-percentile decision latency Δ_{95} , (ii) agrees with the teacher’s decision on the states it actually visits, and (iii) does so for every $\mathbf{c}$ in a descriptor set $\mathcal{C}$ without per-embodiment retraining. Agreement is measured by teacher parity,

$$\mathbf{P} = \frac{1}{N} \sum_{t=1}^N \mathbf{1}[\hat{k}_t = k_t^*], \quad \hat{k}_t = \arg \max_k \pi_\theta(k | o_t, \mathbf{c}) \quad (4)$$

where k_t^* is the teacher decision defined in (7) and the sum runs over the N decision steps of the evaluation episodes, with the teacher run offline on the logged states.

04 Method

4.1 Teacher planner and cost

The teacher follows the configuration-driven structure of Agni-Nav [1]. Candidate trajectories are generated by sampling twist sequences within the limits $(\bar{v}_x, \bar{v}_y, \bar{\omega})$ of $\mathbf{c}$, rolled out with a planar omnidirectional kinematic model, and checked against the envelope $\mathcal{B}(\mathbf{c})$. Each collision-free candidate is scored by

$$J(\tau) = w_g \|\mathbf{p}_\tau(T) - \mathbf{g}_t\| + w_c \sum_{m=1}^M \exp\left(-\frac{d_m(\tau, \mathbf{c})}{\sigma_c}\right) + w_s \sum_{m=1}^{M-1} \|\mathbf{u}_{m+1} - \mathbf{u}_m\|^2 \quad (5)$$

where $\mathbf{p}_\tau(T)$ is the terminal position, $d_m(\tau, \mathbf{c})$ is the clearance between the envelope at the m -th pose and the nearest obstacle, σ_c sets the clearance length scale, $\mathbf{u}_m$ is the commanded twist at step m , and $w_g, w_c, w_s > 0$ weight goal progress, clearance, and smoothness. The clearance term depends on $\mathbf{c}$ through the envelope; the feasible set depends on $\mathbf{c}$ through the velocity limits. Run offline without a time limit, the teacher requires 41 ms per decision on the study's testbed, which makes it suitable for labeling but not for the 10 ms onboard loop.

4.2 From teacher trajectories to primitive targets

The student acts in primitive space, whereas the teacher returns a trajectory. The two are related through a configuration-weighted trajectory discrepancy between the rollout of primitive k and the first T_p seconds of the teacher trajectory,

$$d_c(k, \tau_t^*) = \frac{1}{M} \sum_{m=1}^M \|\mathbf{w}_c (\mathbf{x}_m^k - \mathbf{x}_m^*)\|_2, \quad (6)$$

$$\mathbf{w}_c = \text{diag}\left(\frac{1}{l_f + l_r}, \frac{1}{2w}, \frac{1}{\bar{\omega} T_p}\right)$$

where $\mathbf{x}_m^*$ is the teacher pose at the same time as $\mathbf{x}_m^k$. The weighting expresses position errors in units of body length and width and heading errors relative to the largest heading change achievable in one primitive, so that a given discrepancy has the same meaning on a small and a large robot. The hard teacher decision and the soft teacher target are

$$k_t^* = \arg \min_k d_c(k, \tau_t^*) \quad (7)$$

$$q_T(k | o_t, \mathbf{c}) = \frac{\exp(-d_c(k, \tau_t^*)/\lambda)}{\sum_{j=1}^K \exp(-d_c(j, \tau_t^*)/\lambda)} \quad (8)$$

with temperature $\lambda > 0$. As in knowledge distillation [18], the soft target conveys which alternatives the teacher considers nearly as good, which is informative near decision boundaries such as the choice between passing an obstacle on the left or the right.

4.3 Configuration-conditioned student

The student encodes the range queries with a one-dimensional convolutional encoder over bearing, concatenates the goal, twist, and an embedding of k_{t-1} , and passes the result through a multilayer perceptron whose hidden layers are modulated by the descriptor through feature-wise linear modulation [41]:

$$\mathbf{h}_{\ell+1} = \text{ReLU}(\gamma_\ell(\mathbf{c}) \odot (\mathbf{A}_\ell \mathbf{h}_\ell + \mathbf{b}_\ell) + \beta_\ell(\mathbf{c})) \quad (9)$$

where $\mathbf{h}_\ell$ is the ℓ -th hidden activation, $\mathbf{A}_\ell$ and $\mathbf{b}_\ell$ are layer weights, and γ_ℓ, β_ℓ are produced from the normalized descriptor by a small shared network. Multiplicative modulation is

chosen over concatenation because the descriptor's main effect is to rescale the significance of observed clearances—the same range reading is safe for a narrow body and unsafe for a wide one—which a gain on intermediate features expresses directly. The output layer produces logits $f_\theta(o_t, \mathbf{c}) \in \mathbb{R}^K$ and

$$\pi_\theta(k | o_t, \mathbf{c}) = \frac{\exp(f_\theta(o_t, \mathbf{c})_k)}{\sum_{j=1}^K \exp(f_\theta(o_t, \mathbf{c})_j)} \quad (10)$$

A categorical primitive head is preferred over direct twist regression because the teacher's decisions are frequently multimodal (two symmetric detours of similar cost), and a regression head trained with a squared loss averages modes into a twist that neither alternative endorses. The ablation in Section 5.4 removes this head.

4.4 Trajectory-level imitation objective

Let $\mathcal{D}$ be the aggregated dataset of tuples $(o, \mathbf{c}, \tau^*)$ described below. The imitation term is the cross-entropy between teacher and student primitive distributions,

$$\mathcal{L}_{\text{imit}}(\theta) = -\mathbb{E}_{(o, \mathbf{c}, \tau^*) \sim \mathcal{D}} \sum_{k=1}^K q_T(k | o, \mathbf{c}) \log \pi_\theta(k | o, \mathbf{c}) \quad (11)$$

The imitation is trajectory-level in two senses: the targets are derived from the teacher's full local trajectory through (6) rather than from a single commanded twist, and the states in $\mathcal{D}$ are drawn along closed-loop trajectories rather than independently.

4.5 Consistency regularizer

Cross-entropy treats all incorrect primitives alike, and it is evaluated on the observation exactly as logged. Two consequences follow. First, a student that places residual mass on a primitive turning sharply toward an obstacle is penalized no more than one that places it on a primitive almost identical to the teacher decision. Second, nothing in $\mathcal{L}_{\text{imit}}$ constrains the student's decision when individual range queries are missing, although the teacher's dense sampling and conservative clearance term make its own decisions comparatively insensitive to isolated missing returns. The consistency regularizer addresses both. Let ξ_ρ be a range-dropout perturbation that independently replaces each range query with an invalid return,

$$[\xi_\rho(\mathbf{r})]_j = \begin{cases} r_{\max} & \text{with probability } \rho \\ r_j & \text{otherwise} \end{cases}, \quad j = 1, \dots, N_r \quad (12)$$

with ρ drawn from a training range covering the dropout levels of interest, and let $\xi_\rho(o)$ apply it to the range component of o . The regularizer is the expected configuration-weighted discrepancy of the student's perturbed decision distribution from the teacher trajectory computed on the clean observation,

$$\mathcal{L}_{\text{cons}}(\theta) = \mathbb{E}_{\mathcal{D}} \mathbb{E}_{\rho, \xi_\rho} \sum_{k=1}^K \pi_\theta(k | \xi_\rho(o), \mathbf{c}) d_c(k, \tau^*) \quad (13)$$

It penalizes divergence from teacher decisions in proportion to geometric severity, measured in body-normalized units, and it does so precisely on the corrupted inputs that the student will encounter onboard. The total objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{imit}}(\theta) + \mu \mathcal{L}_{\text{cons}}(\theta) + \kappa \|\theta\|_2^2 \quad (14)$$

with $\mu \geq 0$ and weight decay κ . Setting $\mu = 0$ gives the “no consistency regularizer” variant evaluated in Section 5.

4.6 Data aggregation and descriptor sampling

Training follows DAgger [21]. In round $i = 0, \dots, N_D$, episodes are rolled out in simulation with the mixture policy that executes the teacher decision with probability $\beta_i = \beta_0^i$ and the student decision otherwise; every visited state is relabeled offline by the teacher, which is affordable because labeling has no latency constraint, and appended to $\mathcal{D}$. Descriptors are sampled per episode from a distribution over $\mathcal{C}$ obtained by perturbing envelope dimensions and velocity limits, so that the student observes how teacher decisions vary with $\mathbf{c}$ rather than memorizing one body. The procedure is summarized below.

1. Initialize $\mathcal{D}$ with teacher rollouts over sampled descriptors; train π_θ on (14).
2. For each round i : sample descriptors and scenes; roll out the β_i -mixture policy; relabel all visited states with τ^* from (3); aggregate into $\mathcal{D}$.
3. Retrain π_θ on the aggregated $\mathcal{D}$ with fresh dropout perturbations at every gradient step.
4. Export the final network for the onboard Orin and run it in the 100 Hz decision loop.

4.7 Onboard execution

At deployment, the descriptor of the robot is loaded once at start-up and the student runs at 100 Hz. In each period it assembles o_t from the latest range queries, goal, and state estimate, evaluates π_θ , and sends $\mathbf{u}_{\hat{k}_t}(\mathbf{c})$ from (2) to the platform’s locomotion controller, which tracks body twists. Because only the first part of each primitive is executed before the next decision, the primitive horizon T_p acts as a lookahead for labeling and regularization rather than as a commitment. The decision latency is measured from the availability of the range queries to the publication of the twist command. Changing to a different embodiment requires only a different $\mathbf{c}$; the network weights are shared.

05 Experiments

5.1 Setup and platform

All training and the primary evaluation use Isaac Lab, a successor of the Orbit framework [43], with a Unitree Go2 quadruped in 18 scenes, each evaluated with 5 random seeds. The teacher is the full configuration-driven planner of Section 4, run offline at 41 ms per decision. The student runs on the onboard NVIDIA Jetson Orin of the robot at 100 Hz, giving a decision budget of 10 ms. The deployment evaluation comprises 30 runs on the physical Go2 and a second morphology, a B2-class quadruped asset, in simulation, used to test configuration-conditioned transfer with the same network weights and a different descriptor. Sensor noise is injected as dropout on the range queries, following (12), at rates from 1% to 10%.

REPRODUCIBILITY NOTE

The Go2 runs use 18 scenes $\times$ 5 seeds in simulation, 30 hardware runs, and a simulated B2-class asset for morphology transfer. Teacher labels are produced offline; all latencies reported for learned students are measured on the onboard Orin.

5.2 Compared methods and metrics

Three alternatives are compared with the full method (“Ours”). *Full planner on edge (throttled)* runs the teacher on the Orin with its sampling budget reduced to fit the onboard loop. *Student, no configuration conditioning* uses the same student with the descriptor input and FiLM modulation removed. *Student, no consistency regularizer* trains the same student with $\mu = 0$ in (14). Four metrics are reported: the p95 decision latency Δ_{95} in milliseconds; teacher parity P from (4), in percent; the success rate, in percent, where an episode succeeds if the goal is reached within the scene’s tolerance and time limit without collision; and noise robustness, defined as the drop in success rate, in percentage points, when 5% range dropout is injected relative to clean range queries (lower is better).

5.3 Main results

Table 1 reports the main comparison. The distilled policy is the only method whose p95 decision latency, 7.8 ms, leaves a clear margin of 2.2 ms within the 10 ms period; the alternatives lie between 8.7 and 9.0 ms. Relative to the 41 ms of the offline teacher, the student decides roughly 5.3 times faster. The latency differences among the learned variants are at most 1.2 ms, and the compiled measurements do not include a per-stage profile that would attribute them to specific components.

Teacher parity separates the methods most clearly. The full method agrees with the offline teacher on 94.3% of decisions, 10.1 points above the throttled planner (84.2%), 11.0 points above the unconditioned student (83.3%), and 13.8 points above the student trained without the consistency regularizer (80.5%). Success follows the same ordering at the top: 91.7% for the full method versus 86.3% for the throttled planner, a gain of 5.4 points. Among the alternatives, the unconditioned student succeeds more often (88.5%) than the throttled planner despite slightly lower parity; one plausible explanation is that on the Go2 alone the descriptor is fixed at evaluation time, so that the descriptor input matters more for agreement with the configuration-specific teacher than for task completion on this single body. The student trained without the regularizer has the lowest parity and success of all methods, 9.4 points below the full method in success.

At 5% dropout, all methods lose between 3.9 and 4.2 points of success, with the full method losing the least. The differences at this moderate noise level are small (0.3 points between the full method and the variant without the regularizer); the effect of the regularizer becomes pronounced at 10% dropout, as shown in Section 5.5.

5.4 Ablation

Table 2 removes one component at a time. Each removal costs between 9.1 and 11.3 points of success: 9.1 points without the consistency regularizer (82.6%), 10.0 points without the motion-primitive head (81.7%), where the student regresses a twist directly with a squared loss, and 11.3 points without configuration conditioning (80.4%). All ablated variants are also slower at p95 (8.7 to 8.8 ms versus 7.8 ms). The loss from removing the primitive head is consistent with the multimodality argument of Section 4: averaging two symmetric detours yields a command that neither detour endorses.

Two rows of Table 2 correspond nominally to baselines in Table 1 but do not coincide numerically. The variant without the regularizer reaches 82.6% success at 8.7 ms in the ablation and 82.3% at 9.0 ms in the main comparison, and the variant without

Table 1: Main comparison on the Go2 (Isaac Lab, 18 scenes $\times$ 5 seeds; students on the onboard Orin at 100 Hz). Noise robustness is the success-rate drop at 5% range dropout; lower is better.

Method	Decision latency p95 (ms)	Teacher parity (%)	Success rate (%)	Noise robustness (delta at 5% dropout, pts)
Full planner on edge (throttled)	8.7	84.2	86.3	4.2
Student, no configuration conditioning	8.7	83.3	88.5	4.1
Student, no consistency regularizer	9.0	80.5	82.3	4.2
Ours	7.8	94.3	91.7	3.9

Table 2: Ablation of the distillation components on the Go2.

Variant	Decision latency p95 (ms)	Success rate (%)
Ours (full distillation)	7.8	91.7
w/o consistency regularizer	8.7	82.6
w/o motion-primitive output head	8.7	81.7
w/o configuration conditioning	8.8	80.4

Table 3: Recorded failure cases.

Condition	Recorded outcome
Throttled full planner on the Orin	p95 decision 38 ms; 34% of replans dropped; parity 81% on the replans that run
B2-class morphology transfer (simulation)	parity 89.6%; descriptor generalizes, contact timing does not fully
10% range dropout	parity 88.1% with the regularizer, 79.4% without

conditioning reaches 80.4% at 8.8 ms in the ablation but 88.5% at 8.7 ms in the main comparison. The compiled records do not document the cause of these differences, for example separate training runs or evaluation subsets, and both tables are reported as recorded. The qualitative conclusion that each component contributes is supported by both tables, but the magnitude attributed to configuration conditioning on the Go2 alone should be read with this discrepancy in mind.

5.5 Failure cases

Throttled planner on the edge computer. The run-server logs for the throttled full planner on the Orin record a p95 decision time of 38 ms with 34% of replans dropped, and a parity of 81% even on the replans that complete. The throttled planner therefore fails in two compounding ways: it frequently misses the control period, and when it does decide, the reduced sampling budget leads it to a different decision from the full planner about one time in five. These logged values differ from the throttled-planner row of Table 1 (8.7 ms p95 and 84.2% parity), and the compiled records do not reconcile the two; one possible reading is that the two records were taken under different throttling budgets or latency accounting, but this is not documented. Under either record, the throttled planner falls short of the student in parity.

Morphology transfer. With the same network and the B2-class descriptor, parity drops to 89.6%, 4.7 points below the 94.3% obtained on the Go2. The descriptor generalizes in the sense that envelope- and velocity-dependent decisions such as gap selection and clearance keeping transfer, but contact timing does not fully transfer: the larger platform’s gait and stance timing alter how a commanded primitive unfolds, which the planar envelope and velocity limits in (1) do not encode. Transfer was evaluated only in simulation, so this result does not establish hardware transfer to the second morphology.

Heavy sensor dropout. At 10% range dropout, the student trained with the consistency regularizer retains 88.1% parity,

whereas the student trained without it falls to 79.4%, a gap of 8.7 points. Compared with the clean-observation parity of 94.3%, the full method loses 6.2 points at this noise level. This is the clearest evidence that the regularizer acts as the robustness term of the objective: at 5% dropout its effect on success is small, but as dropout grows, the perturbation-trained, geometry-weighted penalty keeps the student’s decisions aligned with the teacher where plain cross-entropy does not.

06 Limitations

Several limitations qualify the results. First, the student is bounded by its teacher: parity measures agreement with the configuration-driven planner, not optimality or safety, and the student inherits whatever the planner’s envelope abstraction omits. Second, the cross-morphology evidence comes from one additional morphology, evaluated only in simulation; the 4.7-point parity drop shows that a planar envelope with velocity limits does not capture contact timing, and extending the descriptor with gait-level parameters, or estimating them online, is left open. Third, the compiled results report aggregate tables and do not break out the 30 hardware runs separately from the simulated episodes, so the sim-to-real gap for the student is not quantified here, a point that matters given evidence that simulated rankings need not transfer [45]. Fourth, the noise model is independent range dropout from 1% to 10%; correlated occlusions, spurious returns, state-estimation drift, and dynamic obstacles were not evaluated. Fifth, the student provides no formal collision guarantee, and no runtime safety filter was evaluated; a lightweight envelope check on the single selected primitive is a natural addition. Finally, the records contain unreconciled differences between the main and ablation tables and between the throttled-planner table entry and its run-server log, as discussed in Section 5; these do not change the ordering of methods but limit the precision of individual effect sizes.

07 Conclusion

This paper distilled a configuration-driven cross-embodiment local planner into a configuration-conditioned onboard policy. The student selects velocity-normalized body-motion primitives from an observation and a configuration descriptor, and is trained by trajectory-level imitation with soft targets from a body-normalized trajectory discrepancy and a consistency regularizer applied under range dropout. On a Unitree Go2 with an onboard Orin running at 100 Hz, the student decides at a p95 latency of 7.8 ms, within the 10 ms budget and about 5.3 times faster than the 41 ms offline teacher, while agreeing with the teacher on 94.3% of decisions and succeeding in 91.7% of episodes, against 84.2% and 86.3% for the throttled planner. Ablations show that the regularizer, the primitive head, and configuration conditioning each contribute between 9.1 and 11.3 points of success; the regularizer preserves 88.1% parity at 10% dropout where its absence yields 79.4%; and transfer

to a B2-class morphology in simulation retains 89.6% parity, limited by contact timing that the descriptor does not encode. The results indicate that the configuration interface of a cross-embodiment planner can survive compression into a real-time policy, and that the remaining gap lies in what the descriptor describes rather than in the student's capacity to follow it.

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] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A General Navigation Model to Drive Any Robot. arXiv:2210.03370, 2022.
- [4] 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.
- [5] A. Sridhar, D. Shah, C. Glossop, and S. Levine. NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration. arXiv:2310.07896, 2023.
- [6] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by Cheating. arXiv:1912.12294, 2019.
- [7] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza. Learning High-Speed Flight in the Wild. arXiv:2110.05113, 2021.
- [8] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. arXiv:2010.11251, 2020.
- [9] 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.
- [10] A. Eftekhari et al. The One RING: a Robotic Indoor Navigation Generalist. arXiv:2412.14401, 2024.
- [11] J. Zhang et al. Embodied Navigation Foundation Model. arXiv:2509.12129, 2025.
- [12] 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.
- [13] W. Huang, I. Mordatch, and D. Pathak. One Policy to Control Them All: Shared Modular Policies for Agent-Agnostic Control. arXiv:2007.04976, 2020.
- [14] A. Gupta, L. Fan, S. Ganguli, and L. Fei-Fei. MetaMorph: Learning Universal Controllers with Transformers. arXiv:2203.11931, 2022.
- [15] G. Feng, H. Zhang, Z. Li, X. B. Peng, B. Basireddy, L. Yue, Z. Song, L. Yang, Y. Liu, K. Sreenath, and S. Levine. GenLoco: Generalized Locomotion Controllers for Quadrupedal Robots. arXiv:2209.05309, 2022.
- [16] M. Shafiee, G. Bellegarda, and A. Ijspeert. ManyQuadrupeds: Learning a Single Locomotion Policy for Diverse Quadruped Robots. arXiv:2310.10486, 2023.
- [17] 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.
- [18] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. arXiv:1503.02531, 2015.
- [19] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell. Policy Distillation. arXiv:1511.06295, 2015.
- [20] W. M. Czarnecki, R. Pascanu, S. Osindero, S. M. Jayakumar, G. Swirszcz, and M. Jaderberg. Distilling Policy Distillation. arXiv:1902.02186, 2019.
- [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, 2011.
- [22] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-End Training of Deep Visuomotor Policies. arXiv:1504.00702, 2015.
- [23] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. Learning Deep Control Policies for Autonomous Aerial Vehicles with MPC-Guided Policy Search. arXiv:1509.06791, 2015.
- [24] 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.
- [25] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza. Deep Drone Acrobatics. arXiv:2006.05768, 2020.
- [26] 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.
- [27] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. arXiv:2107.04034, 2021.
- [28] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged Locomotion in Challenging Terrains using Egocentric Vision. arXiv:2211.07638, 2022.
- [29] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot Parkour Learning. arXiv:2309.05665, 2023.
- [30] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme Parkour with Legged Robots. arXiv:2309.14341, 2023.
- [31] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena. From Perception to Decision: A Data-driven Approach to End-to-end Motion Planning for Autonomous Ground Robots. arXiv:1609.07910, 2016.
- [32] 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.
- [33] X. Xiao, B. Liu, G. Warnell, J. Fink, and P. Stone. AP-PLD: Adaptive Planner Parameter Learning from Demonstration. arXiv:2004.00116, 2020.
- [34] G. Williams, A. Aldrich, and E. Theodorou. Model Predictive Path Integral Control using Covariance Variable Importance Sampling. arXiv:1509.01149, 2015.
- [35] 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.
- [36] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. arXiv:2109.11978, 2021.
- [37] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced Skills by Learning Locomotion and Local Navigation End-to-End. arXiv:2209.12827, 2022.
- [38] L. Wellhausen and M. Hutter. ArtPlanner: Robust Legged Robot Navigation in the Field. arXiv:2303.01420, 2023.
- [39] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter. Learning Robust Autonomous Navigation and Locomotion for Wheeled-Legged Robots. arXiv:2405.01792, 2024.

- [40] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. arXiv:2303.04137, 2023.
- [41] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. FiLM: Visual Reasoning with a General Conditioning Layer. arXiv:1709.07871, 2017.
- [42] V. Makoviychuk et al. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. arXiv:2108.10470, 2021.
- [43] M. Mittal et al. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. arXiv:2301.04195, 2023.
- [44] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. arXiv:1710.06537, 2017.
- [45] A. Kadian et al. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? arXiv:1912.06321, 2019.