

Embodiment-Conditioned Goal Navigation for Mixed Quadruped-Humanoid Groups

Ivan Sokolov • Yating Feng • Rachel Kim • Xiaotong Zhang

Robotics Institute, Georgia Institute of Technology, Atlanta, GA, USA

PUBLISHED

August 5, 2026

REGISTERED

August 8, 2026

RECORD TYPE

Preprint

ABSTRACT

Mixed groups of quadruped and humanoid robots that navigate toward a common goal differ in footprint, speed envelope, and sensor viewpoint, so a single shared navigation policy is ill-posed unless these differences are made explicit. This paper presents an embodiment-conditioned goal-navigation policy for such groups. Building on the configuration descriptor of AgniNav, each agent receives a normalized embodiment vector encoding footprint, velocity limits, and sensor height, together with egocentric perception, its goal, and the group-relative states and descriptors of its neighbors. A single network maps these inputs to normalized motion commands that are scaled by the agent's own limits and passed through a lightweight resolver enforcing pairwise separation, velocity-bound, and deadlock-breaking invariants. The policy is trained in Isaac Lab with embodiment randomization over the full fleet envelope, while one morphology per platform family is withheld for zero-shot evaluation. Across 25 goal-navigation scenes with 5 seeds and 40 episodes each, groups of two to six agents reach the goal in 88.1 percent of episodes with 0.27 collisions per episode and a 95th-percentile goal time of 41 s, compared with 82.9 percent for per-embodiment policies and 84.0 percent for a shared policy that receives the embodiment as a constant input. On held-out morphologies, zero-shot success is 79.4 percent, versus 68.4 percent and 68.7 percent for these two baselines. In ablations, removing embodiment randomization causes the largest drop. Remaining failures concentrate in doorway clearance and in corridor throughput for six-agent groups.

INDEX TERMS multi-robot navigation, cross-embodiment learning, quadruped robots, humanoid robots, embodiment conditioning, collision avoidance, zero-shot transfer

KAGG NOTE A single goal-navigation policy can serve quadrupeds and humanoids in the same group, including morphologies it has not seen, when embodiment is an explicit, randomized input and pairwise safety is enforced by invariants rather than learned implicitly.

AUTHOR AFFILIATIONS

Ivan Sokolov, Yating Feng, and Rachel Kim: Robotics Institute, Georgia Institute of Technology, Atlanta, GA, USA. Xiaotong Zhang: Robotics and Autonomous Systems Lab, Shanghai Jiao Tong University, Shanghai, China.

01 Introduction

Legged robots are increasingly deployed as teams rather than as single platforms. A facility inspection crew may combine quadrupeds, which are fast, low, and stable on rough ground, with humanoids, which carry sensors at head height, pass through spaces designed for people, and can eventually manipulate doors and fixtures. When such a mixed group is sent to a common goal, every member must solve the same local problem—reach the goal region without colliding with the environment or with teammates—yet the members differ in exactly the quantities that determine the solution: how much space they occupy, how fast and how sharply they can move, and what their sensors can see from where they are mounted.

The prevailing practice is to train or tune one navigation stack per platform. This is workable for a fixed fleet, but it scales poorly: each new morphology, payload configuration,

or sensor remount requires a new policy, and policies trained in isolation have no common convention for yielding to one another. The obvious alternative, a single shared policy, is ill-posed if the policy cannot tell the bodies apart: an observation of a narrow doorway calls for a different action from a compact quadruped than from a humanoid whose shoulders nearly fill it. Cross-embodiment navigation models such as GNM, ViNT, and NoMaD aggregate data across robots [3, 4, 5], and recent planners such as AgniNav make the robot configuration an explicit input to a shared local planner [1]. These systems, however, address a single robot at a time. In a mixed group, embodiment differences also govern *interaction*: who yields in a doorway, how much clearance a pair requires, and which agent can absorb an avoidance maneuver without losing balance.

This paper studies the gap between configuration-driven single-robot navigation and heterogeneous multi-agent goal navigation. Its premise is that embodiment differences should be inputs, not hidden variables. We adopt the AgniNav configuration descriptor [1] as the conditioning channel of a single goal-navigation policy, extend it to describe neighbors as well as the ego agent, and pair the learned policy with a small set of collision-avoidance invariants whose responsibility split is itself computed from the descriptors. Training randomizes embodiments over the full envelope of the fleet; one morphology per platform family is withheld to measure zero-shot transfer.

The contributions are as follows.

1. A formulation of mixed quadruped-humanoid goal navigation in which a normalized embodiment vector (footprint, velocity limits, and sensor height), derived from the AgniNav configuration descriptor, conditions one shared policy and is also exchanged as part of each agent’s group-relative neighbor state.
2. An embodiment-conditioned policy architecture that combines an egocentric range encoder, a permutation-invariant neighbor encoder, and feature-wise modulation by the descriptor, and outputs normalized commands scaled by each agent’s own velocity limits.
3. A collision-avoidance resolver with three invariants—pairwise separation via a decentralized barrier condition with agility-weighted responsibility, velocity bounds, and priority-based deadlock breaking—that is always feasible and is kept in the loop during training.
4. An evaluation in Isaac Lab over 25 scenes, 5 seeds, and 40 episodes per scene and seed, with groups of two to six agents, comparing against per-embodiment policies and two shared-policy baselines, reporting held-out zero-shot transfer, ablations, and an analysis of recorded failure cases.

02 Related Work

2.1 Configuration-driven and cross-embodiment navigation

AgniNav [1] is the closest antecedent and the direct basis of this work. It formulates local planning over an abstract robot configuration—body geometry, sensor placement, and velocity limits—so that a single planning stack can be instantiated on different embodiments by changing the configuration rather than retraining or retuning the planner. Its central design decision is to separate what belongs to the environment (free space perceived from sensors) from what belongs to the robot (the collision envelope and motion limits) and to recombine them only at the point where feasibility is decided. We adopt this decomposition and its descriptor, but differ in three respects. First, AgniNav plans for one robot; here the descriptor is also attached to each neighbor, so that the ego policy reasons about the bodies of its teammates. Second, AgniNav consumes the configuration in a planner; here the descriptor conditions a learned policy, which must therefore interpolate and extrapolate over the descriptor space, and we test this explicitly with held-out morphologies. Third, we use the descriptor not only for feasibility but also to allocate collision-avoidance responsibility between agents of different agility. CeRLP [2] pursues a related geometric route for visual local planning, mapping heterogeneous camera observations to a unified, height-adaptive scan representation before planning; our egocentric range encoder is in the same spirit, but with the sensor height passed explicitly in the descriptor.

Data-driven navigation generalists take the complementary route of aggregating experience across robots. GNM trains a goal-conditioned policy on trajectories from six robots with a normalized action space [3]; ViNT scales this to a Transformer foundation model with adaptable goal modalities [4]; and NoMaD adds a diffusion action head that unifies goal-directed navigation and exploration [5]. These models show that a shared policy can drive different robots, but embodiment enters only implicitly through the data and action normalization; they neither expose collision geometry nor address multi-robot interaction. The One RING [6] randomizes body size, rota-

tion pivot, and camera configuration in simulation and finds that this randomization is key to generalization across indoor platforms, a finding that our ablation on randomization corroborates for legged groups. NavFoM [7] trains a navigation foundation model across quadrupeds, drones, wheeled robots, and vehicles, using identifiers for camera views and temporal context; we instead use measured physical parameters, which are interpretable and permit evaluation on morphologies outside the training set.

2.2 Embodiment-conditioned policies and cross-embodiment locomotion

The idea of conditioning a policy on a hardware description predates navigation foundation models. Hardware Conditioned Policies [8] encode a manipulator’s kinematic structure as an explicit vector input and demonstrate transfer to unseen arms, the closest conceptual precedent for our conditioning design. Shared Modular Policies [9] instead factorize a controller over the agent’s morphology graph with message passing, achieving agent-agnostic locomotion without an explicit vector. In robot learning at scale, Yang et al. [10] show that one policy can be co-trained across manipulation and navigation embodiments, CrossFormer [11] extends a single Transformer policy to manipulation, navigation, locomotion, and aviation, and the Open X-Embodiment collaboration [12] assembles a large multi-robot dataset whose positive transfer motivates shared models. For legged locomotion specifically, GenLoco [13] trains a single controller across randomized quadruped morphologies, ManyQuadrupeds [14] learns one locomotion policy for diverse quadrupeds via central pattern generators, and URMA [15] handles quadrupeds, hexapods, and humanoids with a morphology-agnostic encoder. These works establish that morphology randomization and explicit morphology inputs can support zero-shot transfer at the joint level. The present work operates one level higher—on the navigation commands sent to such controllers—and adds the multi-agent interaction that single-robot studies do not consider.

2.3 Learned multi-agent collision avoidance

Learning-based decentralized collision avoidance has a long line of work. CADRL [16] learns a value function for pairwise non-communicating avoidance, and GA3C-CADRL [17] extends it to arbitrary numbers of neighbors with a recurrent encoder. Long et al. [18] train a sensor-level multi-robot avoidance policy with PPO from range scans, and Fan et al. [19] combine it with a hybrid control scheme for complex scenarios. SARL [20] uses attention to model crowd-robot interaction, and PRIMAL [21] combines reinforcement and imitation learning for large-scale multi-agent pathfinding. In cooperative multi-agent RL, MADDPG [23] introduced centralized critics with decentralized actors, and MAPPO [22] showed that PPO with parameter sharing and a centralized value function is a strong baseline. Almost all of these assume homogeneous agents, often disks with identical speed limits, so that a shared policy is well-posed by construction. Our setting removes that assumption: the neighbor encoder receives each neighbor’s descriptor, and the training procedure follows MAPPO’s centralized-critic, shared-actor structure.

2.4 Safety filters and invariants

Control barrier functions (CBFs) provide a principled way to render a safe set forward invariant by constraining the control

input in a quadratic program [24]. Decentralized multi-robot CBF formulations split the pairwise constraint between the two agents; we use such a split but weight it by agility derived from the descriptors, so that a quadruped absorbs more of an avoidance maneuver than a humanoid with limited lateral acceleration. Placing a filter in the loop during learning, rather than only at deployment, changes what the policy learns; our comparison between a baseline trained without invariants and an ablation that removes them only at deployment speaks to this point.

2.5 Legged locomotion and navigation in simulation

Massively parallel simulation has made learned legged control routine. Isaac Gym [36] and Orbit [35], the predecessor of Isaac Lab, provide GPU-parallel physics and environment management; Rudin et al. [31] train quadruped locomotion in minutes with thousands of parallel environments. Robust and perceptive locomotion has been demonstrated with teacher-student training [28, 29], rapid motor adaptation [27], and egocentric vision [30], and humanoid locomotion has been learned in simulation and transferred to hardware [34]. Rudin et al. [32] and ANYmal Parkour [33] learn local navigation together with or on top of locomotion skills. We follow the hierarchical pattern of the latter: a navigation policy issues velocity commands that family-specific locomotion controllers track. Domain randomization [25] and dynamics randomization [26] motivate our embodiment randomization, which can be viewed as randomizing the agent rather than the environment.

2.6 Evaluation of embodied navigation

Anderson et al. [41] standardized success and path-efficiency metrics for embodied navigation, and SCT [42] argued that completion time, not path length, is the relevant efficiency measure when agents have different dynamics. We report success and the 95th percentile of goal time for the same reason: agents with different speed limits make path-length metrics hard to compare. Kadian et al. [43] showed that simulation rankings can fail to predict real-world performance; since all results here are in simulation, we treat this as an explicit limitation (Section 6).

03 Problem Formulation

3.1 Agents and embodiments

A group consists of N agents indexed by $i \in \mathcal{N} = \{1, \dots, N\}$, with $2 \leq N \leq 6$ in our experiments. Each agent belongs to a platform family $f_i \in \{Q, H\}$ (quadruped or humanoid) and has a planar pose $(\mathbf{p}_i, \theta_i) \in \mathbb{R}^2 \times \mathbb{S}^1$. Following AgniNav [1], the embodiment of agent i is summarized by a configuration vector

$$\mathbf{e}_i = (\ell_i, w_i, \bar{v}_i^x, \bar{v}_i^y, \bar{\omega}_i, \bar{a}_i, h_i) \in \mathbb{R}^7, \quad (1)$$

where ℓ_i and w_i are the length and width of the body’s ground-projected bounding box (m), $\bar{v}_i^x$ and $\bar{v}_i^y$ are the maximum forward and lateral speeds (m/s), $\bar{\omega}_i$ is the maximum yaw rate (rad/s), $\bar{a}_i$ is the maximum planar acceleration that the locomotion controller tracks reliably (m/s²), and h_i is the height of the primary range sensor above ground (m). The footprint is conservatively approximated by the enclosing disk of radius

$$\rho_i = \frac{1}{2} \sqrt{\ell_i^2 + w_i^2}. \quad (2)$$

The *fleet envelope* is the axis-aligned box $\mathcal{E} = [\mathbf{e}^{\text{lo}}, \mathbf{e}^{\text{hi}}]$ that contains the descriptors of all morphologies of both families. Each family f occupies a sub-box $\mathcal{E}_f \subset \mathcal{E}$, and a held-out morphology $\mathbf{e}_f^* \in \mathcal{E}_f$ with a neighborhood $\mathcal{H}_f$ is excluded from training.

3.2 Commands and dynamics

Each agent is driven by a family-specific locomotion controller that tracks a body-frame planar twist command $\mathbf{c}_i = (v_i^x, v_i^y, \omega_i)$. The navigation layer runs with control period $\Delta t = 50$ ms and treats the closed-loop locomotion system as velocity-controlled,

$$\dot{\mathbf{p}}_i = \mathbf{R}(\theta_i) \begin{bmatrix} v_i^x \\ v_i^y \end{bmatrix}, \quad \dot{\theta}_i = \omega_i, \quad (3)$$

subject to the box constraint $|\mathbf{c}_i| \leq \bar{\mathbf{c}}_i := (\bar{v}_i^x, \bar{v}_i^y, \bar{\omega}_i)$ (elementwise) and the rate constraint $\|\mathbf{c}_{i,t} - \mathbf{c}_{i,t-1}\|_\infty \leq \bar{a}_i \Delta t$, where $\mathbf{R}(\theta)$ is the planar rotation matrix. Tracking error of the locomotion controller is not modeled at this layer; it is absorbed by the safety margin introduced in Section 4.5.

3.3 Task, observations, and objective

The group is assigned a common goal region $\mathcal{G}$, a disk of radius $r_{\mathcal{G}}$ around $\mathbf{g}$; at episode start, each agent receives a slot $\mathbf{g}_i \in \mathcal{G}$ by minimum-cost assignment so that slots do not overlap. An episode succeeds if every agent reaches its slot within the time limit $T_{\max}$. A collision is any contact between an agent and an obstacle or between two agents; collisions are counted but do not terminate the episode. Agent i observes, at time t ,

$$\mathbf{o}_{i,t} = (\mathbf{m}_{i,t}, \mathbf{g}_{i,t}^B, \mathbf{c}_{i,t-1}, \bar{\mathbf{e}}_i, \{\mathbf{z}_{ij,t}\}_{j \in \mathcal{N}_i(t)}), \quad (4)$$

where $\mathbf{m}_{i,t} \in \mathbb{R}^B$ is a 360° planar range scan of B beams taken at height h_i and truncated at range R_s , $\mathbf{g}_{i,t}^B$ is the goal slot expressed in the body frame, $\bar{\mathbf{e}}_i$ is the normalized descriptor defined in (6), $\mathcal{N}_i(t)$ is the set of teammates within communication radius R_c , and $\mathbf{z}_{ij,t}$ is the group-relative neighbor state defined in (7). All agents share one policy π_θ , and the objective is

$$\begin{aligned} \max_{\theta} \quad & \mathbb{E}_{\mathbf{e}_{1:N} \sim p_{\text{train}}, \xi} \left[\sum_{t=0}^T \gamma^t \sum_{i \in \mathcal{N}} r_{i,t} \right] \\ \text{s.t.} \quad & h_{ij}(t) \geq 0, \quad h_{ik}(t) \geq 0, \end{aligned} \quad (5)$$

where ξ denotes the scene and initial conditions, p_{train} is the training distribution over $\mathcal{E} \setminus \bigcup_f \mathcal{H}_f$, and h_{ij} and h_{ik} are the agent-agent and agent-obstacle separation functions of Section 4.5. Zero-shot transfer asks how well the trained policy performs when some $\mathbf{e}_i$ lie in $\mathcal{H}_f$, without any fine-tuning.

04 Method

4.1 Normalized embodiment vector

Raw descriptor entries span different units and ranges (meters for footprint and sensor height, rad/s for yaw rate). Each entry is mapped to $[-1, 1]$ with respect to the fleet envelope,

$$\bar{\mathbf{e}}_i = 2 (\mathbf{e}_i - \mathbf{e}^{\text{lo}}) \oslash (\mathbf{e}^{\text{hi}} - \mathbf{e}^{\text{lo}}) - \mathbf{1}, \quad (6)$$

where $\oslash$ denotes elementwise division. Normalizing against the joint envelope of both families, rather than per family, places quadrupeds and humanoids in one coordinate system, so that the policy can share structure where the families overlap (for example, in forward speed) and separate it where they do

not (for example, in sensor height). A held-out morphology whose parameters lie slightly outside the training range yields entries slightly outside $[-1, 1]$; the network sees such values without clipping. The family label is intentionally *not* provided: any family-specific behavior must be explained by the physical parameters, which is what permits transfer to morphologies with unfamiliar combinations of them.

4.2 Group-relative state

For each neighbor $j \in \mathcal{N}_i(t)$, the ego agent receives

$$\mathbf{z}_{ij} = \left(\mathbf{R}(\theta_i)^\top (\mathbf{p}_j - \mathbf{p}_i), \mathbf{R}(\theta_i)^\top (\dot{\mathbf{p}}_j - \dot{\mathbf{p}}_i), \cos(\theta_j - \theta_i), \sin(\theta_j - \theta_i), \bar{\mathbf{e}}_j, \hat{h}_{ij} \right), \quad (7)$$

where $\hat{h}_{ij} = \|\mathbf{p}_j - \mathbf{p}_i\| - (\rho_i + \rho_j)$ is the pairwise surface clearance. Two group-level features are appended to the ego state: the body-frame offset to the group centroid, $\mathbf{R}(\theta_i)^\top (\frac{1}{N} \sum_j \mathbf{p}_j - \mathbf{p}_i)$, and the fraction of teammates that have already reached their slots. Expressing all quantities in the ego frame makes the input invariant to global translation and rotation, which matters because the same doorway interaction occurs at many positions and orientations across the 25 scenes. Including $\bar{\mathbf{e}}_j$ lets the ego agent anticipate a neighbor's capabilities, for instance that a humanoid will not sidestep quickly.

4.3 Policy architecture

The policy has three encoders and a conditioned trunk.

- *Range encoder.* A 1-D convolutional network with circular padding maps $\mathbf{m}_{i,t}$ to a feature $\mathbf{u}_i^m$; circular padding respects the angular periodicity of the scan.
- *Neighbor encoder.* A shared MLP ϕ embeds each $\mathbf{z}_{ij}$, and a single-head attention pooling with the ego feature as query aggregates the embeddings into $\mathbf{u}_i^n = \sum_j \alpha_{ij} \phi(\mathbf{z}_{ij})$, with $\alpha_{ij} = \text{softmax}_j(\mathbf{q}_i^\top \mathbf{k}_{ij} / \sqrt{d})$ [40]. The pooling is permutation invariant [39] and handles group sizes from one to five neighbors with masking.
- *Ego encoder.* An MLP embeds $(\mathbf{g}_{i,t}^B, \mathbf{c}_{i,t-1})$ and the two group-level features into $\mathbf{u}_i^g$.

The concatenation $\mathbf{x}_i^{(0)} = [\mathbf{u}_i^m, \mathbf{u}_i^n, \mathbf{u}_i^g]$ passes through an L -layer MLP trunk whose hidden layers are modulated by the ego descriptor through feature-wise linear modulation (FiLM) [38]:

$$\mathbf{x}_i^{(l+1)} = \sigma \left(\gamma^{(l)}(\bar{\mathbf{e}}_i) \odot (\mathbf{W}^{(l)} \mathbf{x}_i^{(l)} + \mathbf{b}^{(l)}) + \beta^{(l)}(\bar{\mathbf{e}}_i) \right), \quad (8)$$

where $\gamma^{(l)}$ and $\beta^{(l)}$ are small MLPs of $\bar{\mathbf{e}}_i$, σ is the ELU non-linearity, and $\odot$ is the elementwise product. FiLM is preferred over plain concatenation because it lets the descriptor rescale every hidden layer, so that a change in footprint can change how strongly range features near the body are weighted rather than merely shifting the input to the first layer. A Gaussian head outputs the mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$ of the pre-squash action during training; at deployment the mean is used.

4.4 Action parameterization

The network output is squashed and scaled by the agent's own limits,

$$\mathbf{c}_{i,t}^{\text{nom}} = \text{diag}(\bar{v}_i^x, \bar{v}_i^y, \bar{\omega}_i) \tanh(\boldsymbol{\mu}_{i,t}), \quad (9)$$

so that the same network output means “a given fraction of what this body can do”. This normalization is the action-side counterpart of (6): it keeps the action distribution comparable across embodiments and guarantees the velocity-bound invariant by construction. Because all inputs (7) are in the ego frame and all outputs are fractions of ego limits, the resulting motions are group-relative: the policy reasons about where teammates are and what they can do, and responds within its own envelope.

4.5 Collision-avoidance invariants

The nominal command is passed through a per-agent resolver that enforces three invariants.

(I1) Pairwise separation. For agents i and j , define $\mathbf{d}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ and the barrier function

$$h_{ij} = \|\mathbf{d}_{ij}\|^2 - (\rho_i + \rho_j + \delta)^2, \quad (10)$$

with safety margin $\delta > 0$ covering locomotion tracking error and gait sway. Under (3), the CBF condition $\dot{h}_{ij} \geq -\alpha h_{ij}$ [24] with $\alpha > 0$ reads $2\mathbf{d}_{ij}^\top (\dot{\mathbf{p}}_i - \dot{\mathbf{p}}_j) \geq -\alpha h_{ij}$. It is split between the two agents as

$$-2\mathbf{d}_{ij}^\top \dot{\mathbf{p}}_i \leq \kappa_{ij} \alpha h_{ij}, \quad \kappa_{ij} = \frac{\bar{a}_i}{\bar{a}_i + \bar{a}_j}, \quad (11)$$

and symmetrically for j with $\kappa_{ji} = 1 - \kappa_{ij}$. Adding the two inequalities recovers the joint condition, so if both agents satisfy their share, $h_{ij} \geq 0$ is forward invariant under exact velocity tracking. The weight κ_{ij} assigns the larger share of the avoidance to the more agile agent: when a quadruped meets a humanoid, the quadruped, which can accelerate laterally more readily, deviates more. Each agent evaluates (11) from its own observation, because $\mathbf{d}_{ij}$, ρ_j , and $\bar{a}_j$ are contained in $\mathbf{z}_{ij}$; no neighbor velocity estimate is needed. Substituting (3), the constraint is linear in the body-frame command, $-2\tilde{\mathbf{d}}_{ij}^\top (v_i^x, v_i^y)^\top \leq \kappa_{ij} \alpha h_{ij}$ with $\tilde{\mathbf{d}}_{ij} = \mathbf{R}(\theta_i)^\top \mathbf{d}_{ij}$.

Static obstacles are handled in the same form. For the M closest scan returns $\mathbf{q}_{ik}$ (body-frame vectors from the agent center to the return), $h_{ik} = \|\mathbf{q}_{ik}\|^2 - (\rho_i + \delta)^2$ and, because the obstacle is static, $h_{ik} = -2\mathbf{q}_{ik}^\top (v_i^x, v_i^y)^\top$, giving the constraint $2\mathbf{q}_{ik}^\top (v_i^x, v_i^y)^\top \leq \alpha h_{ik}$ with full responsibility.

(I2) Velocity bounds. The resolver solves

$$\begin{aligned} \mathbf{c}_i^{\text{safe}} &= \arg \min_{\mathbf{c}, s \geq 0} (\mathbf{c} - \mathbf{c}_i^{\text{nom}})^\top \mathbf{Q} (\mathbf{c} - \mathbf{c}_i^{\text{nom}}) + \lambda s^2 \\ \text{s.t. } \mathbf{A}_i \mathbf{c} &\leq \mathbf{b}_i, \quad |\mathbf{c}| \leq \bar{\mathbf{c}}_i, \\ \|\mathbf{c} - \mathbf{c}_{i,t-1}\|_\infty &\leq \bar{a}_i \Delta t + s, \end{aligned} \quad (12)$$

where the rows of $\mathbf{A}_i \mathbf{c} \leq \mathbf{b}_i$ stack the agent and obstacle constraints of (I1), $\mathbf{Q}$ is a diagonal weight that trades translational against rotational deviation, and the slack s softens the rate limit only. The QP has three decision variables plus one slack and at most $(N - 1) + M$ separation constraints, so it is solved in well under one control period. *Feasibility:* if all $h_{ij}, h_{ik} \geq 0$ at time t , then $\mathbf{c} = \mathbf{0}$ satisfies every separation constraint (their right-hand sides are nonnegative) and the box constraint, and the rate constraint is met with a sufficiently large slack. The QP therefore always has a solution, and the worst case is a commanded stop. Hard separation takes precedence over smoothness by design: a jerky stop is preferable to contact.

(I3) Deadlock breaking. Barrier constraints can trap agents in symmetric configurations, most often in doorways. Agent i is *stalled* if $\|\mathbf{c}_i^{\text{safe}}\| < \epsilon_v$ while $\|\mathbf{c}_i^{\text{nom}}\| > \epsilon_c$ and $\mathbf{p}_i$ is outside its slot. If agent i and at least one neighbor are stalled for T_d consecutive seconds, the involved agents compute a priority order by ascending distance to the common goal center $\mathbf{g}$, with ties broken by agent index. Because the goal is common and relative positions are shared, every agent computes the same order. For the duration of the resolution, responsibility is reassigned within each stalled pair: the lower-priority agent takes $\kappa = 1$ and the higher-priority agent $\kappa = 0$, which preserves $\kappa_{ij} + \kappa_{ji} = 1$ and hence (I1); in addition, the lower-priority agent’s nominal command is replaced by a tangential escape, $(v^x, v^y)^{\text{nom}} \leftarrow \eta_i \mathbf{n}_{ij}$ and $\omega^{\text{nom}} \leftarrow 0$, where $\mathbf{n}_{ij} = \mathbf{R}(-\pi/2) \hat{\mathbf{d}}_{ij} / \|\hat{\mathbf{d}}_{ij}\|$ is the right-hand normal of the separation vector and $\eta_i = \min(\bar{v}_i^x, \bar{v}_i^y)$. The override ends when all involved agents make progress again.

4.6 Training

Embodiment randomization. At each episode reset, the group size N and the family composition are sampled, and for each agent $\mathbf{e}_i$ is drawn uniformly from $\mathcal{E}_{f_i} \setminus \mathcal{H}_{f_i}$. In simulation, the morphology is realized by rescaling the family’s base asset (link lengths and torso dimensions), the geometric entries of (1) are measured from the rescaled asset, the sensor is mounted at the corresponding height, and the velocity and acceleration limits are those of the command range over which the family’s locomotion controller has been trained. The locomotion controllers are themselves trained with morphology randomization over $\mathcal{E}_f$, in the manner of GenLoco and URMA [13, 15], so that every sampled body can track commands. Randomizing the agent, rather than only the environment, is the analog of domain randomization [25, 26] for the embodiment axis.

Reward. Each agent receives

$$\begin{aligned} r_{i,t} = & w_p (\|\mathbf{p}_{i,t-1} - \mathbf{g}_i\| - \|\mathbf{p}_{i,t} - \mathbf{g}_i\|) \\ & + w_g \mathbb{I}[\text{arrive}_i] + w_G \mathbb{I}[\text{all arrive}] - w_c \mathbb{I}[\text{contact}_i] \\ & - w_f \|\mathbf{c}_{i,t}^{\text{safe}} - \mathbf{c}_{i,t}^{\text{nom}}\|_{\mathbf{Q}}^2 - w_s \|\mathbf{c}_{i,t} - \mathbf{c}_{i,t-1}\|^2 - w_\tau, \end{aligned} \quad (13)$$

with nonnegative weights. The progress and arrival terms drive goal reaching; the team bonus w_G couples agents so that an agent near its slot has an incentive to clear the way for others; the filter-intervention term w_f penalizes nominal commands that the resolver must correct, so the policy learns to propose feasible motions instead of relying on the filter; and the time penalty w_τ discourages waiting indefinitely.

Optimization. The shared actor is trained with PPO [37] in the centralized-training, decentralized-execution pattern of MAPPO [22]: a centralized critic V_ψ receives the global state, including all agents’ poses, goals, and descriptors, while each actor receives only its local observation (4). Parameters are shared across all agents of both families. The resolver (12) is applied inside the environment step during training, so the executed action is $\mathbf{c}^{\text{safe}}$ and the PPO likelihood ratio is computed for the policy’s own pre-filter sample.

05 Experiments

5.1 Setup and platform

All experiments run in Isaac Lab, the successor of Orbit [35]. Groups contain two to six agents with mixed quadruped and humanoid counts. The evaluation suite comprises 25 goal-navigation scenes of three types—corridors, doorways, and open rooms with obstacles—each evaluated with 5 seeds and 40 episodes per seed, that is, $25 \times 5 \times 40 = 5,000$ episodes per method. Embodiment randomization spans the full fleet envelope during training, and one morphology per platform family is held out for zero-shot transfer. The held-out quadruped differs from the nominal quadruped by an 18 % increase in leg length.

Metrics. *Goal success* is the percentage of episodes in which all agents reach their slots within the time limit. *Collisions per episode* counts agent-agent and agent-obstacle contacts, averaged over episodes. *Held-out zero-shot success* is goal success for groups whose members use the held-out morphologies, with no fine-tuning. *Goal time p95* is the 95th percentile, over successful episodes, of the time at which the last agent reaches its slot; following the argument for completion-time metrics in heterogeneous settings [42], it captures the slowest member of the group rather than average path efficiency.

Baselines. All baselines use the same scenes, seeds, and locomotion controllers.

- *Per-embodiment policies (no shared model):* a separate navigation policy per platform family, with the same architecture minus the descriptor input, each trained with randomization over its own family envelope; at evaluation each agent runs its family’s policy.
- *Shared policy, embodiment as constant input:* the proposed architecture and resolver trained on randomized bodies, but with the descriptor input fixed to the nominal value of the agent’s family, so that it acts as a family indicator rather than a description of the actual body.
- *Shared policy, no collision invariants:* the proposed conditioned policy trained with embodiment randomization but without the resolver at training or evaluation time; separation is encouraged only by the contact penalty in (13).

5.2 Main comparison

Table 1 reports the main comparison. The proposed policy reaches the goal in 88.1 % of episodes, which is 5.2 percentage points above per-embodiment policies (82.9 %), 4.1 points above the shared policy with a constant embodiment input (84.0 %), and 5.6 points above the shared policy without collision invariants (82.5 %). The gap is larger on held-out morphologies: 79.4 % zero-shot success against 68.4 %, 68.7 %, and 74.5 %, that is, 11.0, 10.7, and 4.9 points, respectively.

Three observations follow. First, sharing a model is not by itself beneficial: the constant-input shared policy improves indistribution success by only 1.1 points over per-embodiment policies and is essentially tied with them on held-out morphologies (68.7 % versus 68.4 %). A shared network that cannot distinguish bodies within a family has to adopt behavior that is acceptable on average, which does not extrapolate to a new body. Second, the baseline that does receive the true descriptor but lacks the invariants transfers better than both (74.5 %), which indicates that the descriptor carries the information needed for

Table 1: Main comparison on 25 Isaac Lab scenes $\times$ 5 seeds $\times$ 40 episodes. Goal time p95 is the 95th percentile of the time for the last agent of the group to reach its slot.

Method	Goal success (%)	Collisions per episode	Held-out zero-shot success (%)	Goal time p95 (s)
Per-embodiment policies (no shared model)	82.9	0.28	68.4	42
Shared policy, embodiment as constant input	84.0	0.30	68.7	45
Shared policy, no collision invariants	82.5	0.31	74.5	46
Ours	88.1	0.27	79.4	41

Table 2: Ablation of the full system. Δ columns give the change in percentage points relative to the full system.

Variant	Goal success (%)	Δ	Held-out zero-shot success (%)	Δ
Ours (configuration-conditioned)	88.1	—	79.4	—
w/o embodiment randomization	77.6	−10.5	67.2	−12.2
w/o collision invariants	78.2	−9.9	72.7	−6.7
w/o group-relative state	83.3	−4.8	71.1	−8.3

zero-shot transfer; its in-distribution success (82.5 %) and collision count (0.31) are, however, the worst in the table, consistent with separation being harder to learn from a contact penalty alone in mixed groups. Third, the proposed method combines the transfer benefit of conditioning with the safety benefit of the invariants and has the lowest collision count (0.27) and the lowest goal time p95 (41 s). The collision differences across methods are small (0.27 to 0.31 per episode), and no confidence intervals were recorded, so the ordering on this metric should be read as indicative rather than conclusive.

5.3 Ablation

Table 2 removes one component at a time from the full system. Differences are computed from the recorded values.

The variants are defined as follows. *W/o embodiment randomization* trains the full architecture only on the nominal morphology of each family, so the descriptor takes two values during training. *W/o collision invariants* removes the resolver at evaluation from the policy that was trained with it in the loop. *W/o group-relative state* replaces the ego-frame neighbor features of (7) and the group-level features with world-frame positions and velocities.

Removing embodiment randomization causes the largest drop on both metrics (−10.5 and −12.2 points). With only two training descriptors, the conditioning pathway has nothing to interpolate between, and held-out success (67.2 %) falls to the level of the per-embodiment and constant-input baselines. Together with the constant-input baseline, which has randomization but an uninformative descriptor, this indicates that zero-shot transfer requires both: varied bodies during training and a descriptor that tells the policy which body it has.

Removing the invariants at deployment costs 9.9 points in-distribution, more than the 5.6-point gap between the full system and the baseline trained without invariants (Table 1). This is consistent with the policy having learned to rely on the resolver: a policy trained with the filter in the loop proposes commands that are close to, but not always inside, the safe set, and removing the filter exposes that reliance. The held-out drop (−6.7) is smaller than the in-distribution drop, which suggests that on held-out bodies the remaining failures are dominated by clearance misjudgment rather than by inter-agent conflicts

(see Section 5.5). We note that the two configurations without invariants—trained without them (82.5 %, 74.5 %) and with them removed at deployment (78.2 %, 72.7 %)—are distinct experiments and should not be compared as if they were repeated measurements of one condition.

Replacing group-relative with world-frame neighbor features has the smallest in-distribution effect (−4.8) but a larger held-out effect (−8.3). A plausible reading is that world-frame features force the network to memorize scene-specific interaction patterns that are partially reusable for training bodies but less so when a new body changes the clearances involved.

5.4 Per-embodiment breakdown

The records provide one per-embodiment comparison: the held-out quadruped morphology, with leg length increased by 18 %, succeeds in 79.4 % of episodes, against 88.1 % in distribution, a gap of 8.7 points. This value coincides with the aggregate held-out figure in Table 1. The gap is mostly attributable to doorway clearance. A plausible explanation is that the policy’s clearance judgment, interpolated from training bodies, is least accurate exactly where the margin is tightest. A separate breakdown for the held-out humanoid morphology was not recorded, and we do not report one.

5.5 Failure cases

Table 3 summarizes the recorded failure cases.

Doorway deadlock. In the doorway scene with two humanoids and two quadrupeds, one deadlock was recorded, with both humanoids backing into the threshold. This is the configuration (13) is designed for. A plausible mechanism is that the two humanoids have similar agility, so (11) splits responsibility roughly evenly and their mutually yielding commands cancel. The priority rule assigns full responsibility to one of them, and the deadlock is broken in 1.2 s. The case shows that (11) alone does not guarantee liveness, and that a deterministic, consistently computable priority is needed alongside the barrier.

Held-out clearance. The 8.7-point gap for the held-out quadruped is concentrated in doorways. Footprint is represented by the enclosing disk (2), which is conservative for elongated bodies; if the longer legs enlarge the measured footprint, the disk grows and narrow doorways become marginal under (11). A tighter footprint model, for example a capsule or an oriented box, would reduce this conservatism.

Six-agent corridor. With six agents in the corridor scene, goal time p95 rises to 83 s, roughly twice the 41 s aggregate p95 of Table 1. Agents queue rather than collide, and the episode still succeeds; the increase reflects the corridor’s throughput capacity rather than a failure of the policy. It does indicate that

Table 3: Recorded failure cases.

Scenario	Observation	Interpretation
Doorway scene, 2 humanoids + 2 quadrupeds	One deadlock: both humanoids back into the threshold; the invariant resolver breaks it in 1.2 s.	Symmetric barrier configuration resolved by (13).
Held-out quadruped morphology (leg length +18%)	Succeeds in 79.4% of episodes vs. 88.1% in distribution.	Gap is mostly doorway clearance.
Group of 6 in the corridor scene	Goal time p95 doubles (83 s).	Capacity limit, not a failure.

the policy does not reorder agents to exploit faster members, which a group-level scheduling layer could address.

06 Limitations

All results are from simulation. Isaac Lab locomotion and contact are not equivalent to hardware, and simulation rankings can fail to predict real-world rankings [43]; hardware validation remains future work. The evaluation reports point estimates without confidence intervals or per-seed variance, so small differences, especially in collisions per episode, should not be over-interpreted. The per-embodiment breakdown is partial: the held-out quadruped is reported separately, but the held-out humanoid is not. The descriptor is assumed to be known exactly; in practice, payloads and sensor remounts alter the effective footprint and limits, and an online estimate with uncertainty would be needed. The disk footprint is conservative for elongated bodies, which the held-out clearance gap reflects. The invariants assume that the locomotion controller tracks velocity commands within the margin δ ; large tracking errors, for example during humanoid stumbles, would void the forward-invariance argument. The deadlock rule is a heuristic that guarantees a consistent priority among agents with a common goal but does not guarantee liveness in general, and groups with different goals would require a different priority definition. Finally, the study is limited to two platform families, one held-out morphology per family, and groups of at most six agents.

07 Conclusion

This paper presented an embodiment-conditioned goal-navigation policy for mixed quadruped-humanoid groups. A normalized embodiment vector, derived from the AgniNav configuration descriptor, conditions a single shared policy and describes each neighbor in a group-relative state; a resolver with agility-weighted pairwise separation, velocity bounds, and priority-based deadlock breaking enforces safety around the learned commands. In Isaac Lab, across 5,000 episodes per method, the policy reaches the goal in 88.1% of episodes with 0.27 collisions per episode and a goal time p95 of 41 s, and transfers zero-shot to held-out morphologies with 79.4% success, compared with 68.4% for per-embodiment policies. The ablations indicate that the transfer gain depends jointly on embodiment randomization and on an informative descriptor, and that the invariants contribute substantially to in-distribution success. The main open directions are tighter footprint models for narrow passages, group-level scheduling for high-occupancy corridors, online descriptor estimation, and validation on hardware.

REPRODUCIBILITY NOTE

All quantitative results are from Isaac Lab with 25 scenes, 5 seeds, and 40 episodes per scene and seed. Record: <https://academy.kaggleyes.top/abs/KAGG:2608.00003>. License: CC BY 4.0.

References

- [1] T. Zang, S. Cheng, H. Huang, S. Wang, and W. Zhang. AgniNav: Configuration-Driven Cross-Embodiment Local Planning for Robot Navigation. arXiv:2606.10903, 2026.
- [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] A. Eftekhari et al.. The One RING: a Robotic Indoor Navigation Generalist. arXiv:2412.14401, 2024.
- [7] J. Zhang et al.. Embodied Navigation Foundation Model. arXiv:2509.12129, 2025.
- [8] T. Chen, A. Murali, and A. Gupta. Hardware Conditioned Policies for Multi-Robot Transfer Learning. arXiv:1811.09864, 2018.
- [9] W. Huang, I. Mordatch, and D. Pathak. One Policy to Control Them All: Shared Modular Policies for Agent-Agnostic Control. arXiv:2007.04976, 2020.
- [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 et al.. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. arXiv:2310.08864, 2023.
- [13] G. Feng et al.. GenLoco: Generalized Locomotion Controllers for Quadrupedal Robots. arXiv:2209.05309, 2022.
- [14] M. Shafiee, G. Bellegarda, and A. Ijspeert. ManyQuadrupeds: Learning a Single Locomotion Policy for Diverse Quadruped Robots. arXiv:2310.10486, 2023.
- [15] 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.
- [16] Y. F. Chen, M. Liu, M. Everett, and J. P. How. Decentralized Non-communicating Multiagent Collision Avoidance with Deep Reinforcement Learning. arXiv:1609.07845, 2016.
- [17] M. Everett, Y. F. Chen, and J. P. How. Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning. arXiv:1805.01956, 2018.

- [18] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan. Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning. arXiv:1709.10082, 2017.
- [19] T. Fan, P. Long, W. Liu, and J. Pan. Distributed Multi-Robot Collision Avoidance via Deep Reinforcement Learning for Navigation in Complex Scenarios. arXiv:1808.03841, 2018.
- [20] C. Chen, Y. Liu, S. Kreiss, and A. Alahi. Crowd-Robot Interaction: Crowd-aware Robot Navigation with Attention-based Deep Reinforcement Learning. arXiv:1809.08835, 2018.
- [21] G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. K. S. Kumar, S. Koenig, and H. Choset. PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning. arXiv:1809.03531, 2018.
- [22] C. Yu, A. Velu, E. Vinitzky, J. Gao, Y. Wang, A. Bayen, and Y. Wu. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. arXiv:2103.01955, 2021.
- [23] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. arXiv:1706.02275, 2017.
- [24] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control Barrier Functions: Theory and Applications. arXiv:1903.11199, 2019.
- [25] 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.
- [26] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. arXiv:1710.06537, 2017.
- [27] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. arXiv:2107.04034, 2021.
- [28] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. arXiv:2010.11251, 2020.
- [29] 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.
- [30] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged Locomotion in Challenging Terrains using Egocentric Vision. arXiv:2211.07638, 2022.
- [31] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. arXiv:2109.11978, 2021.
- [32] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced Skills by Learning Locomotion and Local Navigation End-to-End. arXiv:2209.12827, 2022.
- [33] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots. arXiv:2306.14874, 2023.
- [34] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath. Real-World Humanoid Locomotion with Reinforcement Learning. arXiv:2303.03381, 2023.
- [35] M. Mittal et al.. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. arXiv:2301.04195, 2023.
- [36] V. Makoviychuk et al.. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. arXiv:2108.10470, 2021.
- [37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. arXiv:1707.06347, 2017.
- [38] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. FiLM: Visual Reasoning with a General Conditioning Layer. arXiv:1709.07871, 2017.
- [39] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola. Deep Sets. arXiv:1703.06114, 2017.
- [40] A. Vaswani et al.. Attention Is All You Need. arXiv:1706.03762, 2017.
- [41] P. Anderson et al.. On Evaluation of Embodied Navigation Agents. arXiv:1807.06757, 2018.
- [42] N. Yokoyama, S. Ha, and D. Batra. Success Weighted by Completion Time: A Dynamics-Aware Evaluation Criteria for Embodied Navigation. arXiv:2103.08022, 2021.
- [43] A. Kadian et al.. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? arXiv:1912.06321, 2019.