2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

The One RING: a Robotic Indoor Navigation Generalist

Ainaz Eftekhar [1,2] **Rose Hendrix** [2] **Luca Weihs**[2] **Jiafei Duan** [1,2]
Ege Caglar[1] **Jordi Salvador** [2] **Alvaro Herrasti** [2] **Winson Han**[2]
Eli VanderBil[2] **Aniruddha Kembhavi** [1,2] **Ali Farhadi** [1,2] **Ranjay Krishna**[1,2]
Kiana Ehsani[*2] **Kuo-Hao Zeng**[*2]
[1]University of Washington, [2]Allen Institute for AI
one-ring-policy.allen.ai

***Abstract*— Modern robots vary significantly in shape, size, and sensor configurations used to perceive and interact with their environments. However, most navigation policies are embodiment-specific—a policy trained on one robot typically fails to generalize to another, even with minor changes in body size or camera viewpoint. As custom hardware becomes increasingly common, there is a growing need for a single policy that generalizes across embodiments, eliminating the need to (re-)train for each specific robot. In this paper, we introduce RING (Robotic Indoor Navigation Generalist), an embodiment-agnostic policy that turns any mobile robot into an effective indoor semantic navigator. Trained entirely in simulation, RING leverages large-scale randomization over robot embodiments to enable robust generalization to many real-world platforms. To support this, we augment the AI2-THOR simulator to instantiate robots with controllable configurations, varying in body size, rotation pivot point, and camera parameters. On the visual object-goal navigation task, RING achieves strong cross-embodiment (XE) generalization—72.1% average success rate across 5 simulated embodiments (a 16.7% absolute improvement on the CHORES-S benchmark) and 78.9% across 4 real-world platforms, including Stretch RE-1, LoCoBot, and Unitree Go1—matching or even surpassing embodiment-specific policies. We further deploy RING on the RB-Y1 wheeled humanoid in a real-world kitchen environment, showcasing its out-of-the-box potential for mobile manipulation platforms.**

INTRODUCTION

Robot embodiments are diverse and are constantly evolving to better suit new environments and tasks. This range in body configurations—differences in size, shape, wheeled or legged locomotion, and sensor configurations—not only shapes how robots perceive the world but also how they act in it. A robot with a wide field of view (FoV) or multiple cameras can scan its surroundings quickly, while one with a narrower view might need to more actively explore a room. A small robot can squeeze through tight spaces, a low-profile one can duck under furniture, while a larger robot needs to follow more conservative routes. The influence of embodiment on behavior means a policy trained on one design, or even a few, often fails to generalize out of domain.

There has been progress towards scalable cross-embodiment training [1], [2], [3], [4], [5]. While these methods demonstrate some transfer to unseen embodiments, they still suffer from performance degradation with relatively small changes in embodiment (e.g., camera pose modification on the same robot) [6], [7]. Potentially, this is due to these methods relying on the small amount of real-world data available in public datasets-around 20 embodiments in total [1]. Similarly, general-purpose navigation policies [8], [9], [10] are trained on datasets with relatively few embodiments (e.g., 8 robots in [9]), limiting their generalization. A more comprehensive solution is needed—one that can robustly handle the full spectrum of possible embodiments without retraining or additional adaptation.

We introduce RING, a **R**obotic **I**ndoor **N**avigation **G**eneralist. RING is trained *exclusively in simulation*, without any use of real-world robot embodiments. In other words, all robot platforms we evaluate on (i.e., Stretch RE-1, LoCoBot, Unitree's Go1, RB-Y1) are *unseen* by RING during training. We leverage simulation to randomly sample **1 Million** agent body configurations, varying the robot's camera parameters, collider sizes, and center of rotation. Concretely, each embodiment consists of a collider box of varying dimensions and cameras with randomized parameters, placed randomly within the collider box. Fig.1-A presents a t-SNE[11] visualization of body parameters for 30k such agents. Our approach builds on the success of prior works that achieve strong real-world performance through large-scale simulation-only training [12], [13], [14]. Simulation enables training across a vast distribution of environments (150k ProcTHOR houses [15]) and objects (40k+ 3D objects in Objaverse [16]) in the AI2-THOR simulator. Extensive domain randomization on visual observations and the use of pre-trained visual encoders then allows simulation-trained policies to bridge the sim-to-real gap. We follow the training procedure in FLaRe [14], first training our policy on expert trajectories collected from 1M randomized embodiments and subsequently fine-tuning it with on-policy reinforcement learning (RL) in the simulator.

Our results demonstrate generalization to *truly unseen embodiments*. RING transfers to diverse real-world embodiments without any adaptation, despite being trained entirely in simulation without access to the real robot configurations. We evaluate in a zero-shot setting across Stretch RE-1, LoCoBot, Unitree's Go1, and RB-Y1 wheeled humanoid. RING achieves 72.1% average success rate in simulation (16.7% absolute improvement on CHORES-S benchmark) and 78.9% on real robot platforms—matching or even sur-

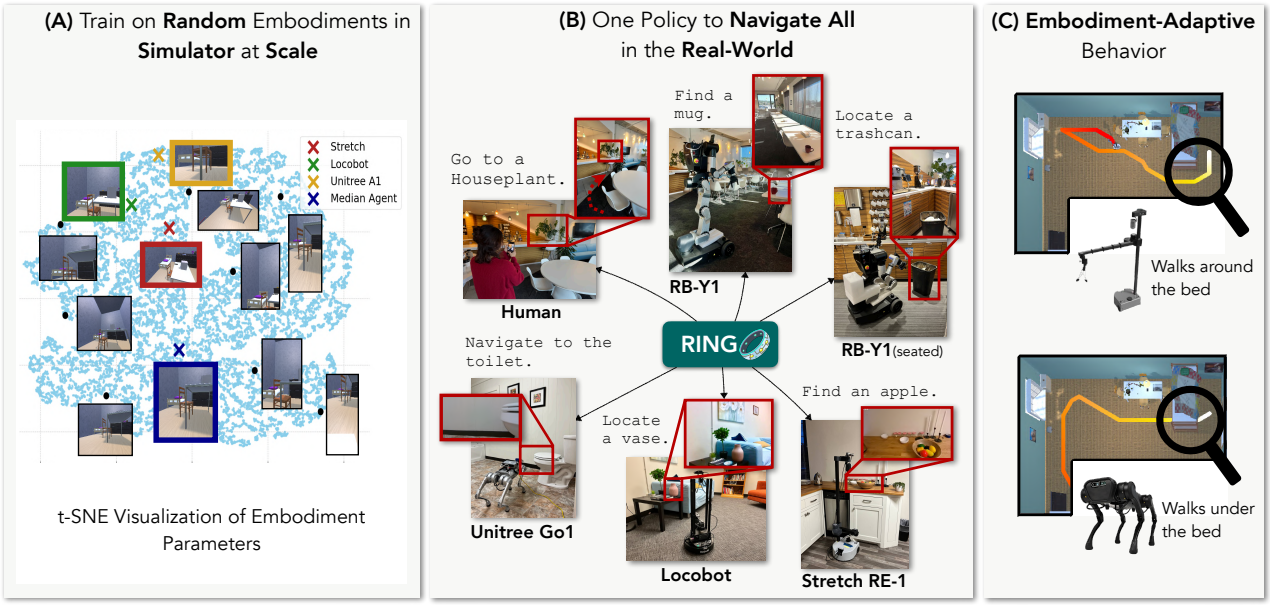

Fig. 1: We show that training on *one million* randomly generated embodiments in simulation (varying camera configurations, body size, and rotation pivot point) results in RING, a generalist navigation policy that works across various robot embodiments in the real world. (A) A t-SNE visualization of embodiment parameters for 30k random agents and three real robots (*we do not train on any real robot embodiment parameters*). Egocentric views from the first camera are shown for 10 sample agents. (B) RING transfers zero-shot to a wide range of embodiments in the real-world including Stretch RE-1, LoCoBot, Unitree Go1, RB-Y1 wheeled humanoid. (C) The policy displays embodiment-adaptive behavior, adjusting its navigation strategy based on its embodiment.

passing embodiment-specific policies. RING can be further adapted to an *embodiment-specialized* policy with even better performance (up to 10% absolute improvement) with minimal finetuning. RING is easy to install, and is ready for use by the community. We will release our pretrained models, code, and data.

RELATED WORK

Cross-embodiment. Cross-embodiment training has received substantial attention from the research community. Arguably the most representative of a large body of recent work [3], [4], [5], [17], [18], [19], [20], [21], [22], [23], [24], Open-X-Embodiment (OXE) is the fruit of a large collaboration to cover many robotic tasks, with special emphasis in manipulation. Its usage in RT-X results in a notable performance gain in emergent skill evaluations in comparison to RT-2 [25]. Despite the 1.5 million trajectories across 22 embodiments present in their dataset, the enormous cost of data collection in the real world makes further scaling challenging. CrossFormer [3] trains a transformer-based policy on 900k trajectories spanning 30 robots, drawing from OXE, navigation data from GNM [8], manipulation data from DROID [26], and additional sources. However, the limited diversity of embodiments and focus on low-level control highlight the need for denser embodiment coverage. GET-zero [27], focused on dexterous manipulation, incorporates embodiment structure via a connectivity graph to guide attention. In contrast, we generate an arbitrarily large set of randomized embodiments during training, allowing our policy to generalize zero-shot to novel embodiments without requiring access to their structure.

Foundational navigation policies Following the success in recent developments for point-goal navigation [28], locomotion [29], [30], [31], [32], agile control [33], exploration [34], [35], [36], and social navigation [37], comparable results in more nuanced tasks like semantic or object-goal navigation (ObjectNav) [38], [39], [40], [41], [42], [43], [44] remain elusive due to a lack of efficient exploration and semantic understanding capabilities. Recently, with powerful pretrained vision models [45], [46] and large-scale procedurally generated virtual environments [16], notable progress in end-to-end ObjectNav policy learning *for specific embodiments* has been achieved by means of imitation learning (IL) from shortest-path trajectories [12], RL [13], or combinations thereof [14]. In image-goal navigation, NoMaD [10], which extends ViNT [9], uses a diffusion policy to control a single embodiment. With the same goal in mind, GNM [8] trains navigation policies across 6 embodiments using IL. In contrast, our policy benefits from RL fine-tuning, improving robustness to compounding errors. Leveraging large-scale training with randomized embodiments in simulation, RING learns a single policy that generalizes to *any* embodiment, including *truly* unseen robot platforms in the real world. Unlike NoMaD, ViNT, GNM, and Mobility VLA [47], which rely on topological map or graph reconstruction for high-level planning, our approach is fully end-to-end and

can explore dynamic novel scenes without requiring an explicit map. While prior work [48], [49], [50] has explored embodiment-agnostic policies using LLMs or VLMs, these methods are limited to short-horizon navigation and single-step prediction. In contrast, RING incorporates temporal context via a transformer decoder.

RING

With the growing diversity of robots used in research labs and real-world applications, there remains a need for a policy that can operate a wide range of embodiments and transfer, in a zero- or few-shot manner, to unseen robots. We introduce RING, a generalist policy for indoor visual navigation that *learns from a broad spectrum of embodiments, trained exclusively in simulation, without any direct use of actual robot embodiments*. We show that training on an extensive range of ∼1M random embodiments results in a robust navigation policy, enabling zero-shot transfer to unseen real-world robots. To train RING, we define the space of random embodiments (Sec.-B), generate expert trajectories for random embodiments in simulation (Sec.-C), and use state-of-the-art architecture designs (Sec.-D) to train with a combination of IL and RL methods (Sec.-E).

A. Problem formulation

We define the space of possible embodiments as E, where each embodiment $e \in E$ is characterized by a configuration vector $\mathbf{c}_e$, including parameters such as camera settings, agent collider size, and center of rotation. Each task can be modeled as a Partially Observable Markov Decision Process (POMDP), denoted as $(S, A, E, O_e, T_e, R, L, P(s_0), \gamma)$, where S and A are the state and action spaces. The observation space O_e varies across embodiments due to differences in camera parameters. The observation at time t for embodiment e, $o_t^e = O_e(s_t, \mathbf{c}_e)$, is a function of both the state s_t and embodiment parameters $\mathbf{c}_e$. Given an action a_t, the next state follows the transition dynamics $s_{t+1} \sim T_e(s_{t+1}|s_t, a_t, \mathbf{c}_e)$, which depends on the embodiment (due to variations in collider size and rotation center). Fig. 2 shows trajectories from two different embodiments starting at the same location and following the same sequence of actions. They have distinct visual observations and follow different transition dynamics—one agent moves under the table, while the other collides with it.

Except where otherwise specified, we assume that all embodiments share the same discrete action space {`MoveBase(±20cm)`, `RotateBase(±6°, ±30°)`, `Done`}. These actions are executed using robot-specific low-level controllers during deployment. This simple and platform-agnostic action space enables effective cross-embodiment (XE) transfer, as demonstrated across both holonomic and differential-drive robots in our experiments. Investigating more expressive XE action spaces beyond this sufficient version is left for future work.

B. Embodiment randomization at scale

Domain randomization [51] is a class of methods in which policies are trained across a wide range of simulated *environmental* parameters; the aim is to enable robustness to unseen environments. Our approach is complementary yet orthogonal; we apply embodiment randomization to train policies on a diverse set of *robot body* parameters, enabling robust deployment to unseen real-world robots.

Parameters	Training Range
Collider Size $(\alpha_x, \alpha_y, \alpha_z)$	$[0.2, 0.5]$, $[0.3, 1.5]$, $[0.2, 0.5]$
Rotation Center (o_x, o_y, o_z)	$[-\alpha_x/2, \alpha_x/2]$, $[-\alpha_y/2, \alpha_y/2]$, $[-\alpha_z/2, \alpha_z/2]$
Vertical FoV (cam1, cam2)	$[40, 100]$, $[40, 100]$
Horizontal FoV (cam1, cam2)	$[40, 120]$, $[40, 120]$
Camera Pitch (cam1)	$[-20, 40]$
Camera Pitch (cam2)	$[-20, 60]$
Camera Yaw (cam1, cam2)	always 0, $[0, 360]$
Camera Position (x) (cam1, cam2)	$[-\alpha_x/2, \alpha_x/2]$, $[-\alpha_x/2, \alpha_x/2]$
Camera Position (y) (cam1, cam2)	$[0.3, \alpha_y]$, $[0.3, \alpha_y]$
Camera Position (z) (cam1, cam2)	$[-\alpha_z/2, \alpha_z/2]$, $[-\alpha_z/2, \alpha_z/2]$
RGB dimensions (H, W)	$[112, 448]$, $[112, 448]$

TABLE I: **Random Embodiment Parameters.** 1M embodiments are sampled from these ranges.

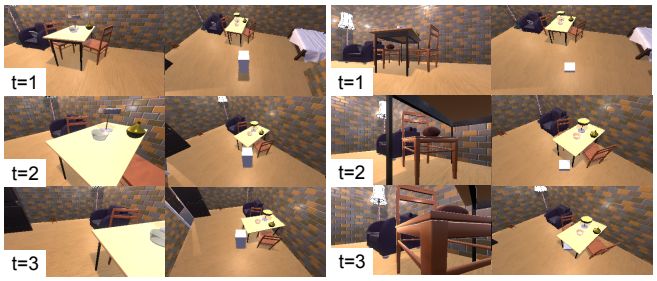

Fig. 2: **Different embodiments exhibit different behaviors.** We show egocentric view from the main camera and third-person view of the 2 agents–white boxes indicate the robot colliders. Embodiment B can go under the table to get to the chair but Embodiment A collides with the table and has to go around.

We model the body of the agent as an invisible collider box in the AI2-THOR [52] simulator. Each agent can have 1 or 2 RGB cameras placed at a random pose within the collider box. Both body and camera parameters are randomly sampled from the ranges specified in Tab. I. We detail the parameters varied in our embodiment randomization: **Collider size** $(\alpha_x, \alpha_y, \alpha_z)$. The agent's body is modeled as a collider box. We use three scale factors $(\alpha_x, \alpha_y, \alpha_z)$ to scale the box along x, y, z axis. **Rotation center** (o_x, o_y, o_z). These coordinates define the agent's pivot point. While this center is typically near (0,0), it can vary across different robots. We sample o_x from the range $[-\frac{\alpha_x}{3}, \frac{\alpha_x}{3}]$ and o_y from the range $[-\frac{\alpha_y}{3}, \frac{\alpha_y}{3}]$, with the sampling ranges determined by the collider size. **Camera parameters**. Each agent is equipped with two RGB cameras placed within the collider box. We randomize several camera parameters, including position, rotation, FoV, and aspect ratio. While the first camera always faces forward, the second camera can rotate up to 360° in z-axis– facing forward, to the sides, or backward.

C. Data Generation with Expert Planners

In order to incorporate policy *safety*, the expert planners should account for the embodiment (e.g. collider size) to avoid approaching obstacles along the way. We use A* search

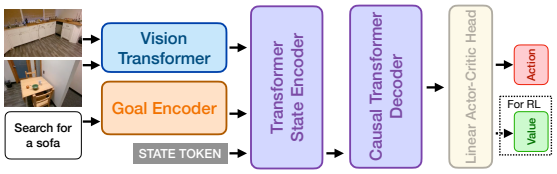

Fig. 3: **RING architecture** accepts visual observations and a language instruction as inputs and predicts an action to execute. At RL finetuning, RING also predicts a value estimate.

[53], [54] to generate safe navigation trajectories for training, following these steps: **1)** Extract reachable locations in a scene on a finely spaced grid, ensuring that the agent's collider does not intersect with any object's collider. Thus, different embodiments yield different reachable locations according to their collider. **2)** Compute a clipped Euclidean distance to the nearest obstacle. Then, for each location, set the cost of visiting it as the inverse of the third power of the distance. **3)** Construct a grid-like graph where each reachable location is a node connected to its immediate neighbors. For each connection, assign a cost equal to the maximum cost of visiting either of the two connected nodes. **4)** Extract a minimum-cost path connecting the reachable positions in the graph nearest to the source and to the target via A*. **5)** Extract waypoints by skipping over points in the A* path as long as skipping them doesn't increase the total path cost from the latest waypoint. **6)** The expert linearly interpolates between waypoints up to the precision reachable by the action space to generate each trajectory.

D. Architecture

With this rich dataset of expert trajectories for random embodiments, a deep, high-capacity architecture is essential to learn a robust policy (Fig. 3). At each timestep, RING uses N RGB images (one per camera) and a language instruction l to predict an action distribution over a discrete action space. RING's architecture, inspired by PoliFormer [13], consists of a Visual Encoder, a Goal Encoder, a Transformer State Encoder, and a Causal Transformer Decoder with a linear actor-critic head. The Visual and Goal Encoders are frozen pre-trained models (a ViT and a language model, respectively) that encode the RGB observations and instruction l into visual and goal token embeddings. Projections of these embeddings, along with a special `STATE` token vector, are stacked along the token axis and processed by the multi-layer Transformer State Encoder, which summarizes the observations at each timestep as the state embedding corresponding to the `STATE` token. Finally, the Causal Transformer Decoder performs explicit memory modeling over time, producing the current belief by causal attention on the state embeddings stacked along the temporal axis. The linear actor-critic head further predicts action logits over the action space as well as a value estimate. *(More details on the project website)*

E. Training paradigm

Our policy is pretrained with imitation learning on the expert trajectories collected from randomized embodiments and then fineuned with on-policy RL using the same random embodiments in the AI2-THOR simulator [52].

Large-scale imitation learning with random embodiments: We train our policy using expert trajectories collected from 1M randomized embodiments across 50k procedurally generated PROCTHOR houses [15], containing approximately 40k annotated 3D objects [55]. At each time step, the linear actor-critic head in the Causal Transformer Decoder predicts action logits, and a cross-entropy loss is computed between the predicted logits π^t and the expert action. We use a batch size of 240 trajectories, each with a temporal context window of 100 steps. Training is conducted on 8× H100 GPUs (80 GB each) using the AdamW optimizer with a learning rate of $2 \cdot 10^{-4}$ for 80k iterations.

Large-scale RL finetuning with random embodiments: Following the training recipe in FLaRe [14], we perform large-scale RL fine-tuning using AllenAct [56] on randomized embodiments in simulation. This fine-tuning phase is critical for enabling the policy to learn through trial and error how to navigate diverse embodiments. We use DD-PPO with 64 parallel environments and 128 rollout steps across 4 machines (each with 8× H100 GPUs), training for 40M steps using the AdamW optimizer with a learning rate of $2 \cdot 10^{-5}$. As in FLaRe [14], we disable the entropy term in the PPO loss to prevent catastrophic forgetting. For fair comparison, we adopt the reward function from [15]: $r_t = \max(0, min\Delta_{0:t-1} - \Delta_t) + s_t - \rho$, where $\min \Delta_{0:t-1}$ is the minimum L2 distance between the agent and the target object up to time $t-1$, Δ_t is the current L2 distance, s_t is a success reward, and $\rho = 0.01$ is a step penalty encouraging task efficiency. The agent must explicitly issue `Done` to receive the success reward ($s_t = 10$); otherwise, $s_t = 0$.

EXPERIMENTS

Our experiments show that RING operates effectively across a wide range of embodiments, including *Stretch RE-1*, *LoCoBot*, *Unitree Go1*, and *RB-Y1*, despite being trained exclusively in simulation **without** any direct exposure to real robot embodiments. Our key results are: 1) RING generalizes **zero-shot** to 5 *truly* unseen embodiments, despite never being trained on them, and achieves state-of-the-art performance across multiple benchmarks (Sec. -F). 2) Our policy, trained solely in simulation on randomized embodiments, transfers directly to the **real-world**, on 5 real robots (Sec. -G). 3) RING can be easily adapted to **embodiment-specialized** policies with minimal finetuning. It achieves better performance on each specific robot (Sec. -H). 4) RING shows **embodiment-adaptive behavior**, adjusting its strategies based on the agent's body (Sec. -I). 5) We perform a **collision analysis** showing that RING remains as safe—*and in some cases even safer*—than embodiment-specific policies(Sec. -J).

F. RING *generalizes zero-shot to unseen embodiments*

We perform zero-shot evaluations of all policies on four robot embodiments: Stretch RE-1 (with 1 or 2 cameras), LoCoBot, and Unitree A1 in simulation.

Model	Loss	Train Embodiment	Benchmark Embodiment: Success (SEL)					
			Stretch	**Stretch** (Nav Cam)	**Stretch** (Factory Config)	**LoCoBot**	**Unitree A1**	Average
SPOC [12]	IL only	Stretch	57.0 (38.1)*	37.9 (19.0)	33.0 (19.3)	16.2 (5.4)	2.1 (1.6)	29.2 (16.7)
SPOC-2.3M			60.0 (30.3)*	37.5 (17.9)	46.0 (19.4)	24.0 (7.9)	10.0 (5.2)	35.5 (16.1)
POLIFORMER [13]	RL only	Stretch	81.0 (58.1)*	65.0 (35.5)	47.5 (25.6)	27.5 (14.8)	42.6 (25.1)	52.7 (31.8)
		LoCoBot	56.0 (32.9)	56.5 (34.7)	52.0 (27.7)	61.5 (44.7)*	50.5 (34.2)	55.4 (34.9)
		Unitree A1	40.0 (25.2)	39.0 (22.5)	35.5 (20.9)	30.0 (17.4)	55.3 (48.2)*	40.0 (26.8)
FLARE [14]	IL + RL	Stretch	82.0 (63.5)*	55.5 (37.9)	38.0 (19.6)	21.5 (10.9)	27.0 (15.1)	44.8 (29.4)
RING-ZERO-SHOT	IL + RL	RING-Random	**76.0 (55.9)**	**74.0 (52.5)**	**72.0 (52.7)**	**66.5 (45.3)**	**72.0 (58.6)**	**72.1 (53.0)**

TABLE II: **Zero-shot Results.** RING shows zero-shot generalization to four unseen embodiments. Unless otherwise specified, "Stretch" refers to the two-camera variant of the RE-1 platform, used in[12]. All prior methods fail to generalize effectively to embodiments beyond those seen during training. Gray* numbers indicate evaluation on the training embodiment; all others reflect zero-shot performance on unseen embodiments.

Baselines. We select prior works from both imitation learning (IL) and reinforcement learning (RL) for comparison. Each baseline is trained on a specific embodiment and evaluated in a zero-shot setting across four different embodiments. SPOC [12] is a supervised IL baseline trained on shortest-path expert trajectories in AI2-THOR. PoliFormer [13] is a state-of-the-art transformer-based policy for object goal navigation, trained from scratch using RL. FLaRe [14] combines IL and RL for efficient policy fine-tuning. All baselines use similar architectures and comparable data, except for SPOC (reason to include SPOC 2.3M). Specifically, SPOC (SPOC-2.3M) is trained with IL on Stretch RE-1 using 100k (2.3M) expert trajectories; Poliformer [13] is trained from scratch on each embodiment individually over 300M RL steps (more than other baselines in terms of training frames); and FLaRe [14] finetunes SPOC on Stretch RE-1 with an additional 20M RL steps.

Experimental details. RING is first trained with IL on 1M expert trajectories collected from randomized embodiments in simulation, followed by finetuning with RL for an additional 40M steps on the randomized embodiments. *Note that all four target embodiments were unseen during training*. We evaluate on the navigation benchmark in CHORES-$\mathbb{S}$ [12], a simulation benchmark for household robot with 200 tasks across 200 scenes. For Unitree A1, we create a new, similar benchmark with 200 tasks adjusted for the robot's lower height to ensure that all targets are feasible.

Results. Tab. II presents the zero-shot evaluation of all policies across four embodiments. We compare *Success Rate* and *Success Weighted by Episode Length (SEL [39])*, a metric measuring efficiency. The results indicate that all single-embodiment baselines struggle to generalize effectively to new embodiments, with performance declining as embodiment differences increase. In contrast, RING exhibits strong generalization across all embodiments, despite not being trained on any of them, achieving an average absolute improvement of 16.7% in Success Rate. In some cases, it outperforms the baseline trained on the target embodiment: PoliFormer trained on LoCoBot *(*61.5 $\rightarrow$ 68.5*)* and Unitree A1 (55.3 $\rightarrow$ 72.0). These 2 more challenging embodiments (lower FoV, low camera placement) make RL from scratch less effective. RING benefits from more efficient learning by training across random embodiments at scale with more diverse navigation behaviors.

Model	Train Embodiment	Eval Embodiment			
		Stretch	**Stretch (FC)**	**LoCoBot**	**Unitree Go1**
SPOC [12]	Stretch	50.0	-	-	-
POLIFORMER [13]	Stretch	83.3	33.3	-	-
	LoCoBot	-	-	80.0	-
	Unitree Go1	-	-	-	41.7
FLARE [14]	Stretch	94.4	-	-	-
RING-ZERO-SHOT	RING-Random	83.3	**72.2**	**80.0**	**80.0**

TABLE III: **Real-world Results**. RING transfers zero-shot to the real-world without any finetuning. Gray numbers are evaluated on same training embodiment.

G. RING *transfers to real-world embodiments despite being purely trained in simulation*

We zero-shot evaluate our policy on 4 unseen robots in a multi-room real-world apartment without any real-world-specific finetuning (Tab. III). We use the same evaluation set of 15 tasks for LoCoBot[13], [15], [57] (3 start poses × 5 targets) and 18 tasks for Stretch RE-1 [12], [13], [14] (3 poses × 6 goals). For Unitree Go1, we create a new set with 3 start poses and 4 objects (*toilet, sofa, TV, trashcan*) placed to match its lower viewpoint. RING matches or outperforms specialized policies, likely due to cross-embodiment (XE) training enabling robust sim-to-real transfer in the presence of real-world noise. We also deploy RING on the RB-Y1 wheeled humanoid in an unstructured kitchen, where it successfully navigates to targets (*trashcan, apple, houseplant, mug*) at two different heights (standing/seated), using an iPhone 16 Pro camera mounted on the robot to stream visual observations (Fig. 1-B). (*Videos on project website.*)

H. RING *can efficiently adapt to an embodiment-specialized policy with minimal finetuning*

Although RING generalizes zero-shot across diverse embodiments, some scenarios benefit from embodiment-specialized policies for optimal performance. Here, we show that RING can be easily adapted to a *robot-specialized policy* through minimal fine-tuning. **Baselines.** We compare with FLaRe [14], which is pretrained on Stretch RE-1 and finetuned on each of the three test embodiments using up to 20M RL steps. **Implementation.** We finetune RING, pretrained on randomized embodiments, on each robot for

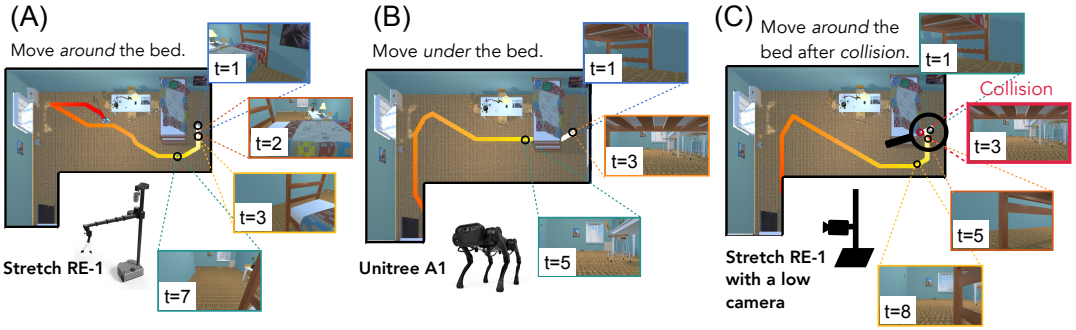

Fig. 4: RING exhibits **embodiment-adaptive** behavior, adjusting its navigation strategy based on the robot's physical configuration. The shorter quadruped robot (B) walks under the bed, while the taller Stretch-RE1 (A) navigates around it. In (C), an agent with the same height as Stretch-RE1 but a lower camera position initially attempts to go under the bed, mistakenly assuming it can fit. After a collision, it adapts and reroutes around the bed, similar to Stretch-RE1.

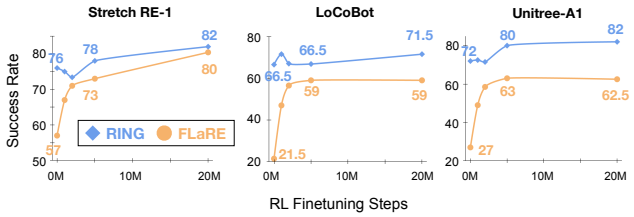

Fig. 5: **Embodiment-Specialized Adaptation.** RING, pretrained on randomized embodiments, adapts efficiently to individual robots with minimal fine-tuning.

up to 20M RL steps, using the same hyperparameters as FLaRe for fair comparison. Following FLaRe, we repurpose `RotateBase(±6°)` to `TiltCamera(±30°)` for LoCoBot, enabling camera control not available during zero-shot evaluation. **Results.** As shown in Fig. 5, RING adapts efficiently, achieving superior performance with minimal fine-tuning. For LoCoBot and Unitree-A1, FLaRe underperforms compared to Stretch RE-1, suggesting that pretraining on a single embodiment limits generalizability. This underscores the value of policies like RING that can adapt quickly and consistently to new embodiments.

I. RING *changes its behavior across different embodiments*

Ideally, an optimal policy would modify its behavior depending on the embodiment. For instance, a thinner robot can navigate through narrow hallways or under furniture, and a wider agent may need to take more conservative paths. Our qualitative results show that RING exhibits such embodiment-adaptive behavior. In Fig. 4-A,B, both Stretch RE-1 and Unitree A1 begin behind a bed. The low-profile quadruped moves directly under it, while Stretch RE-1 navigates around—demonstrating that RING *implicitly* infers embodiment characteristics from visual input, without access to privileged body information. Visual input reveals cues like camera specs and, in some cases, the agent's height. However, vision alone can be ambiguous, prompting the agent to rely on physical interactions—such as collisions—to refine its understanding. (Collision feedback may come from actual impacts or from sensors that anticipate collisions before they occur.) In Fig. 4-C, an agent with a low-mounted camera but tall body misjudges its own height, initially attempts to go under the bed, collides, and then reroutes like Stretch RE-1. This behavior is not present in the expert data but emerges during reinforcement learning through training across diverse embodiments.

J. Collision analysis

RING is trained on random body dimensions with no explicit embodiment information. Regardless, the learned policy remains as safe—*and in some cases even safer*—than embodiment-specific policies (Tab. IV). Collision-avoidance behavior can be further improved by incorporating a small collision penalty into the RL reward (Tab. V).

RING learns to take conservative paths without explicit knowledge of its collider size. Expert trajectories are generated using A*, finding minimum-cost paths where the cost is defined as the inverse of the Euclidean distance to the nearest obstacles. During RL, collisions slow down the agent's progress, leading to lower rewards through a step penalty. Since they produce no useful state change and only waste time, the policy learns to avoid them for more efficient task completion. We evaluate RING against embodiment-specific baselines using the metric *Safe Episodes*—the percentage of episodes completed without any collisions. As shown in Tab. IV, RING achieves a higher percentage of Safe Episodes compared to the baselines. By training across a large number of embodiments and without access to exact body size information, RING learns to take more conservative actions through reinforcement learning, promoting safer navigation.

Include collision penalty to take safer routes. Adding a small collision penalty of 0.1 to the reward function can further reduce collision rate (*CR*) by 50%. The resulting policy is more conservative, regardless of embodiment size. To quantify these results, we created a custom benchmark similar to CHORES-$\mathbb{S}$ [12], consisting of 2,000 random embodiments across 2,000 scenes. We evaluate 2 different versions of our policy on this benchmark, comparing metrics such as *Success Rate*, *Success Weighted by Collision (SC)*, *Collision Rate (CR)*, and *Safe Episode*. As shown in Tab. V, adding the collision penalty reduces the collision rate (CR) (7.77% $\rightarrow$ 4.03%) as well as increases the percentage of trajectories without collisions (46.90% $\rightarrow$ 60.57%).

Model	Train Embodiment	Safe Episode ↑		
		Stretch	**LoCoBot**	**Unitree A1**
PoliFormer [13]	Stretch	45.0	-	-
	LoCoBot	-	42.0	-
	Unitree A1	-	-	**49.25**
FLaRe [14]	Stretch	62.0	-	-
Ring-Zero-Shot	Ring-Random	**67.0**	**64.5**	48.5

TABLE IV: Ring has more **Safe Episodes** (with no collisions) compared to embodiment-specific baselines.

Model	Collision Penalty	Metrics				
		Success ↑	SEL ↑	SC ↑	CR ↓	Safe Episode ↑
Ring	✗	**67.62**	56.24	42.53	7.77	46.90
	✓	66.33	**56.87**	**49.05**	**4.03**	**60.57**

TABLE V: **Collision Penalty.** Adding a small collision penalty (0.1) to the reward function results in 50% less collision, forcing the policy to be even more conservative.

Conclusion

We introduce Ring (**R**obotic **I**ndoor **N**avigation **G**eneralist), an embodiment-agnostic policy trained entirely in simulation on 1 million diverse, randomly initialized embodiments. Ring demonstrates strong zero-shot generalization to a wide range of unseen real robots without any finetuning.

Acknowledgment

AI Tool Disclosure: The authors are responsible for all content in this article. AI tools (ChatGPT) were used in a limited capacity (minor grammar enhancement).

References

[1] A. O'Neill et al., "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[2] O. M. Team et al., "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[3] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine, "Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation," *arXiv preprint arXiv:2408.11812*, 2024.

[4] J. Yang et al., "Pushing the Limits of Cross-Embodiment Learning for Manipulation and Navigation," in *arXiv preprint arXiv:2402.19432*, 2024.

[5] L. Wang, X. Chen, J. Zhao, and K. He, "Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers," in *NeurIPS*, 2024.

[6] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox, "The colosseum: A benchmark for evaluating generalization for robotic manipulation," *arXiv preprint arXiv:2402.08191*, 2024.

[7] A. Xie, L. Lee, T. Xiao, and C. Finn, "Decomposing the generalization gap in imitation learning for visual robotic manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 3153–3160.

[8] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "Gnm: A general navigation model to drive any robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 7226–7233.

[9] D. Shah et al., "ViNT: A foundation model for visual navigation," *arXiv preprint arXiv:2306.14846*, 2023.

[10] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *ICRA*, 2023.

[11] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[12] K. Ehsani et al., "SPOC: Imitating Shortest Paths in Simulation Enables Effective Navigation and Manipulation in the Real World," in *CVPR*, 2024, pp. 16 238–16 250.

[13] K.-H. Zeng et al., "PoliFormer: Scaling On-Policy RL with Transformers Results in Masterful Navigators," in *CoRL*, 2024.

[14] J. Hu et al., "FLaRe: Achieving Masterful and Adaptive Robot Policies with Large-Scale Reinforcement Learning Fine-Tuning," *arXiv preprint arXiv:2409.16578*, 2024.

[15] M. Deitke et al., "ProcTHOR: Large-Scale Embodied AI Using Procedural Generation," in *NeurIPS*, 2022.

[16] M. Deitke et al., "Objaverse: A Universe of Annotated 3D Objects," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 142–13 153, 2022.

[17] Octo Model Team et al., "Octo: An Open-Source Generalist Robot Policy," in *RSS*, 2024.

[18] M. Xu et al., "Flow as the cross-domain manipulation interface," in *CoRL*, 2024.

[19] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song, "Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers," in *CoRL*, 2024.

[20] L. Y. Chen et al., "Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning," in *CoRL*, 2024.

[21] K. Kedia, P. Dan, and S. Choudhury, "One-Shot Imitation under Mismatched Execution," *arXiv preprint arXiv:2409.06615*, 2024.

[22] M. J. Kim et al., "OpenVLA: An Open-Source Vision-Language-Action Model," in *CoRL*, 2024.

[23] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song, "Xskill: Cross embodiment skill discovery," in *CoRL*, 2023.

[24] K. Zakka, A. Zeng, P. Florence, J. Tompson, J. Bohg, and D. Dwibedi, "Xirl: Cross-embodiment inverse reinforcement learning," in *CoRL*, 2022.

[25] A. Brohan et al., "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," *CoRR*, vol. abs/2307.15818, 2023.

[26] A. Khazatsky et al., "DROID: A large-scale in-the-wild robot manipulation dataset," in *arXiv preprint arXiv:2403.12945*, 2024.

[27] A. Patel and S. Song, "GET-Zero: Graph Embodiment Transformer for Zero-shot Embodiment Generalization," *CoRR*, vol. abs/2407.15002, 2024.

[28] E. Wijmans et al., "DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.

[29] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, 2024.

[30] N. Bohlinger et al., "One Policy to Run Them All: Towards an End-to-end Learning Approach to Multi-Embodiment Locomotion," in *CoRL*, 2024.

[31] M. Shafiee, G. Bellegarda, and A. Ijspeert, "Manyquadrupeds: Learning a single locomotion policy for diverse quadruped robots," in *ICRA*, 2024.

[32] C. Ying et al., "PEAC: Unsupervised Pre-training for Cross-Embodiment Reinforcement Learning," *CoRR*, vol. abs/2405.14073, 2024.

[33] W. Xiao, H. Xue, T. Tao, D. Kalaria, J. M. Dolan, and G. Shi, "AnyCar to Anywhere: Learning Universal Dynamics Model for Agile and Adaptive Mobility," *arXiv preprint arXiv:2409.15783*, 2024.

[34] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," *ICLR*, 2020.

[35] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, IEEE, 1997, pp. 146–151.

[36] J. Ye, D. Batra, A. Das, and E. Wijmans, "Auxiliary Tasks and Exploration Enable ObjectNav," *CoRR*, vol. abs/2104.04112, 2021. arXiv: 2104.04112.

[37] X. Puig et al., "Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots," *CoRR*, vol. abs/2310.13724, 2023.

[38] D. Batra et al., "ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects," *CoRR*, vol. abs/2006.13171, 2020.

[39] A. Eftekhar, K.-H. Zeng, J. Duan, A. Farhadi, A. Kembhavi, and R. Krishna, "Selective Visual Representations Improve Convergence and Generalization for Embodied AI," in *ICLR*, 2023.

[40] K.-H. Zeng, L. Weihs, A. Farhadi, and R. Mottaghi, "Pushing it out of the way: Interactive visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[41] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 340–32 352, 2022.

[42] S. Wani, S. Patel, U. Jain, A. Chang, and M. Savva, "Multion: Benchmarking semantic map memory using multi-object navigation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9700–9712, 2020.

[43] N. Yokoyama, R. Ramrakhya, A. Das, D. Batra, and S. Ha, "HM3D-OVON: A Dataset and Benchmark for Open-Vocabulary Object Goal Navigation," *IROS*, 2024.

[44] M. Khanna et al., "Goat-bench: A benchmark for multi-modal lifelong navigation," in *CVPR*, 2024.

[45] M. Oquab et al., *DINOv2: Learning Robust Visual Features without Supervision*, 2023.

[46] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid Loss for Language Image Pre-Training," *ICCV*, vol. abs/2303.15343, 2023.

[47] Z. Xu et al., "Mobility VLA: Multimodal Instruction Navigation with Long-Context VLMs and Topological Graphs," in *CoRL*.

[48] S. Nasiriany et al., "Pivot: Iterative visual prompting elicits actionable knowledge for vlms," *ICML*, 2024.

[49] W. Yuan et al., "RoboPoint: A Vision-Language Model for Spatial Affordance Prediction for Robotics," *arXiv preprint arXiv:2406.10721*, 2024.

[50] M. Ahn et al., "Do as i can, not as i say: Grounding language in robotic affordances," *CoRL*, 2022.

[51] X. Chen, J. Hu, C. Jin, L. Li, and L. Wang, "Understanding Domain Randomization for Sim-to-real Transfer," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022.

[52] E. Kolve et al., "AI2-THOR: An Interactive 3D Environment for Visual AI," *ArXiv*, vol. abs/1712.05474, 2017.

[53] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, pp. 100–107, 1968.

[54] A. A. Hagberg, D. A. Schult, P. Swart, and J. Hagberg, "Exploring Network Structure, Dynamics, and Function using NetworkX," *Proceedings of the Python in Science Conference*, 2008.

[55] A. I. for AI, *ObjaTHOR: Python package for importing and loading external assets into ai2thor*, https://github.com/allenai/objathor, 2024.

[56] L. Weihs et al., "AllenAct: A Framework for Embodied AI Research," *arXiv preprint arXiv:2008.12760*, 2020.

[57] M. Deitke, R. Hendrix, L. Weihs, A. Farhadi, K. Ehsani, and A. Kembhavi, *Phone2Proc: Bringing Robust Robots Into Our Chaotic World*, 2022.