The One RING: a Robotic Indoor Navigation Generalist

Ainaz Eftekhar [1,2] **Luca Weihs** [2] **Rose Hendrix**[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 Artificial Intelligence

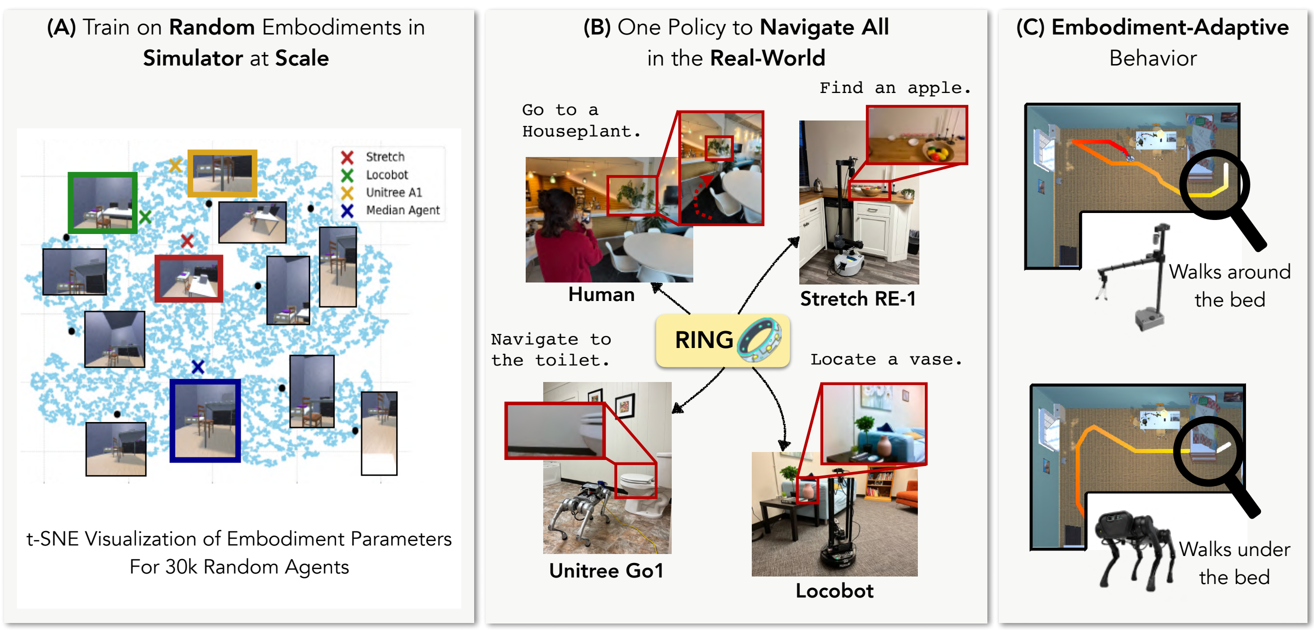

Figure 1. (A) We train on *one million* randomly generated embodiments in simulation varying camera configurations, body size, and rotation pivot point. The plot shows the t-SNE visualization of embodiment parameters $\mathbf{c}_e \in \mathbb{R}^{19}$ for 30k random agents and three specific robots (robots are for visualization-we do not train on any real robot embodiment parameters). Egocentric views from the first camera are shown for 10 sample agents. (B) Our trained policy transfers zero-shot to a wide range of embodiments in the real-world including Stretch RE-1, LoCoBot, and Unitree Go1, as well as a human embodiment. (C) The RING policy displays embodiment-adaptive behavior, adjusting its navigation strategy based on its embodiment.

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 learned using one robot's configuration does not typically gracefully generalize to another. Even small changes in the body size or camera viewpoint may cause failures. With the recent surge in custom hardware developments, it is necessary to learn a single policy that can be transferred to other embodiments, eliminating the need to (re-)train for each specific robot. In this paper, we introduce RING *(**R**obotic **I**ndoor **N**avigation **G**eneralist), an embodiment-agnostic policy, trained solely in simulation with diverse randomly initialized embodiments at scale. Specifically, we augment the AI2-THOR simulator with the ability to instantiate robot embodiments with controllable configurations, varying across body size, rotation pivot point, and camera configurations. In the visual object-goal navigation task,* RING *achieves robust performance on real unseen robot platforms (Stretch RE-1, LoCoBot, Unitree's Go1), achieving an average of* 72.1% *and* 78.9% *success rate across* 5 *embodiments in simulation and*

4 *robot platforms in the real world.*

1. 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, and a larger robot may need to follow more conservative routes. The influence of embodiment on behavior means a policy trained on one design, or even several, often does not perform well out of domain [62].

There has been progress towards scalable cross-embodiment training [34, 45] and in developing general-purpose navigation policies [13, 42, 43, 56]. While these methods demonstrate some transfer to unseen embodiments, they require construction of topological maps or graphs and suffer performance degradation with relatively small changes in embodiment (*e.g.*, camera position modification on the same robot). Potentially, this is due to these methods relying on the small amount of real-world data available in public datasets-only around 20 embodiments in total. This highlights the need for a more comprehensive solution that reliably covers the wide range 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 A1) 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 FoV and dimensions, placed randomly within the collider box. Fig. 1-A shows a t-SNE [46] visualization of body parameters for 30k random agents in our generated data.

Our method is inspired by the recent success in real-world experiments while training at large-scale only in simulation [17, 24, 62]. Simulation training is able to benefit from the vast scale of scenes (150k ProcTHOR houses [12]) and objects (40k+ annotated 3D objects from Objaverse [11]) 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 outlined in FLaRe [24], first training our policy on expert trajectories collected from 1M randomized embodiments and subsequently fine-tuning it with on-policy reinforcement learning (RL) within the simulator.

Our results demonstrate generalization to *truly unseen embodiments*. RING generalizes to diverse real-world embodiments without any adaptation, despite being trained exclusively in simulation without access to the real robot configurations. We evaluate our policy in a zero-shot setting across a variety of embodiments, including the Stretch RE-1, LoCoBot, Unitree's A1, and even "Navigation Assistants", wherein a human user captures ego-centric observations on their phone and prompt RING policy to predict actions to navigate. RING achieves 72.1% and 78.9% success rate on average, outperforming the best baseline significantly, both in simulation and the real-world.

We highlight three key characteristics of RING: **1)** it displays zero-shot generalization to unseen embodiments, keeping a consistently high performance (Sec. 4.1); **2)** it can zero-shot transfer to the real-world without any adaptation or real-world-specific finetuning. (Sec. 4.2); **3)** it can be adapted to an embodiment-specialized policy with even better performance with minimal finetuning (Sec. 4.3); and **4)** at inference, it dynamically adjusts its behavior based on the embodiment (Sec. 4.4, Fig. 6). RING can be directly deployed to navigate any robot platform, is easy to install, and is ready for use by researchers in the community. We will release our pretrained models, generated data, and training code.

2. 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, 7, 13, 14, 19, 21, 23, 25, 28, 30, 33, 47, 52, 53, 56, 60], Open-X-Embodiment (OXE) [9] 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 [5]. 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 [13] trains a transformer-based policy on 900k trajectories across 30 robots, including a subset of OXE, navigation data from GNM [42], manipulation data from DROID [27], and additional collected data. Due to the relatively sparse amount of embodiments observed during training and the target low-level control, it does not generalize to unseen embodiments. GET-zero [36] focuses on dexterous manipulation, and proposes to inform the policy with the structure of the embodiment via a connectivity graph to bias the attention. In contrast, we generate an arbitrarily large amount of embodiments for training

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 1. **Random Embodiment Parameters.** We generate 1M different embodiments sampled from the ranges above.

our policy, enabling zero-shot deployment to new embodiments without accessing the embodiment structure.

Foundational navigation policies. Following the success in recent developments for point-goal navigation [50], locomotion [4, 39, 41], agile control [51], exploration [6, 55, 57], and social navigation [37], comparable results in more nuanced tasks like semantic or object-goal navigation (ObjectNav) [2, 15, 26, 31, 40, 48, 58, 61] remain elusive due to a lack of efficient exploration and semantic understanding capabilities. Recently, with powerful pretrained foundational vision models [35, 63] and large-scale procedurally generated virtual environments [11], 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 [17], RL [62], or combinations thereof [24]. In image-goal navigation, NoMaD [44], which extends ViNT [43], uses a diffusion policy to control a single embodiment. With the same goal in mind, GNM [42] trains navigation policies across 6 embodiments using IL. In contrast, our policy benefits from finetuning with RL, improving resilience to compounding errors. Additionally, thanks to training with large-scale randomized embodiments in simulation, RING learns a single policy to navigate *any* embodiment, generalizing to *truly* unseen robot platforms in the real world. Furthermore, NoMaD, ViNT, GNM, and Mobility VLA [54] all require topological map or graph reconstruction for high-level planning, whereas our policy is fully end-to-end and explores novel scenes without an explicit map. While several efforts [1, 32, 59] focus on learning embodiment-agnostic policies using LLMs or VLMs, they address only short-horizon navigation tasks and perform single-step predictions. In contrast, RING models temporal information through a transformer decoder.

3. 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 robot embodiments. To train RING, we define the space of random embodiments (Sec. 3.2), enable generation of expert trajectories for random embodiments in simulation (see Appendix), and use state-of-the-art architecture designs (Sec. 3.3) to train with a combination of IL and RL methods (Sec. 3.4).

3.1. Problem formulation

Learning a navigation policy across multiple embodiments is a multi-task robotic problem. We define the space of possible embodiments as $\mathcal{E}$, where each embodiment $e \in \mathcal{E}$ is characterized by a configuration vector $\mathbf{c}_e$, which includes parameters such as camera settings, agent collider size, center of rotation, etc. Each task can be modeled as a Partially Observable Markov Decision Process (POMDP), denoted as $(\mathcal{S}, \mathcal{A}, \mathcal{E}, \mathcal{O}_e, \mathcal{T}_e, R, \mathcal{L}, P(s_0), \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces. The observation space $\mathcal{O}_e$ varies across embodiments due to differences in camera parameters and sensor configurations. The observation at time t for embodiment e, $o_t^e = \mathcal{O}_e(s_t, \mathbf{c}_e)$, 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 $s_{t+1} \sim \mathcal{T}_e(s_{t+1}|s_t, a_t, \mathbf{c}_e)$, which depends on the embodiment, as different embodiments interact with the environment in distinct ways (due to variations in collider size and center of rotation). Fig. 2 shows example 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`}, and use robot-specific low-level controllers to execute these actions during deployment.

3.2. Embodiment randomization at scale

Domain randomization [8] 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.

We model the body of the agent as an invisible collider box in the AI2-THOR [29] simulator. Each agent can have

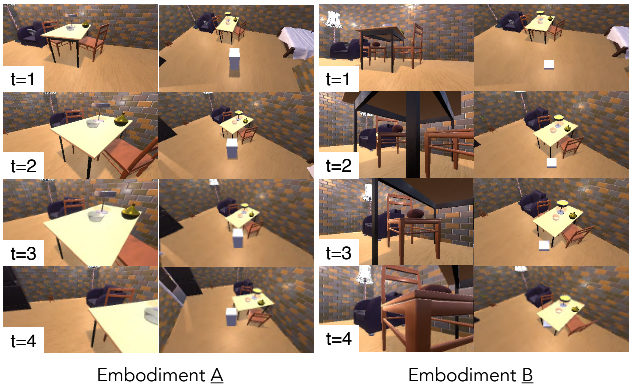

Figure 2. **Different embodiments exhibit different behaviors.** For each embodiment in these sample trajectories, the left column shows the first-person view from the main camera and the second one a third-person view of the agent –white boxes indicate the robot colliders. Embodiment A (shown on the left) has a bigger body size compared to Embodiment B (shown on the right). As a result, B can go under the table to get to the chair but A collides with the table and has to go around.

1 or 2 RGB cameras placed at a random pose within the collider box. Parameters corresponding to both the body and the cameras are sampled randomly from the ranges specified in Table 1. We also modify the process of generating expert trajectories to account for the diversity of embodiments, for details see Appendix. Below, 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. We sample α_x and α_y uniformly from the range $[0.2, 0.5]$m and sample α_z (the height of the agent) from the range $[0.3, 1.5]$m. These ranges adequately capture the variability among most robots.

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. The sampling ranges for these parameters are shown in Table 1. While the first camera always faces forward, the second camera can rotate up to 360° in z-axis, enabling it to face forward, to the sides, or backward.

For visualization purposes, we define an embodiment configuration vector $\mathbf{c}_e \in \mathbb{R}^{19}$ for each embodiment, representing the camera and body parameters. Fig. 1-A shows a t-SNE visualization of the vectors $\mathbf{c}_e$ for 30k of our random embodiments, along with the corresponding vectors for Stretch RE-1, LoCoBot, and Unitree A1. The figure also includes the egocentric view from the first camera for 10 random embodiments, and the three robots. This demonstrates that our randomization spans a wide range of possible embodiments, covering the real-world robot platforms of interest. In total, we gather **1M** trajectories across 50k houses, each with a randomly sampled embodiment.

3.3. Architecture

With this rich dataset of expert trajectories for random embodiments, a deep, high-capacity architecture is essential to learn a robust policy. In this section, we introduce our model architecture, shown in 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. To account for different dimensions, we pad the RGB observations to make them square and resize them to 256×256 before feeding them to the model. RING's architecture, inspired by PoliFormer [62], 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. We provide more details about the architecture in the Appendix.

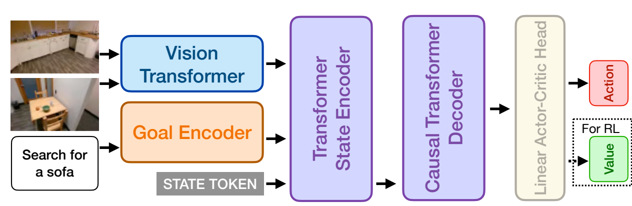

Figure 3. **Our RING model architecture**. It accepts visual observations and a language instruction as inputs and predicts an action to execute. At RL finetuning, RING also predicts a value estimate.

3.4. Training Paradigm

Recently, SPOC [17] showed that training policies with Behavior Cloning on large-scale expert trajectories in simulation leads to policies that effectively generalize to the real world. FLaRe [24] further introduced a robust and scalable method for finetuning such pretrained policies with On-policy Reinforcement Learning. RL finetuning introduces error recovery behaviors and mitigates the compounding errors typically encountered in imitation learning, leading to a substantial performance boost. We adopt the same recipe of first pretraining our policy on expert trajectories collected from randomized embodiments (Sec. 3.2), followed by finetuning with on-policy RL using the randomized embodiments in the AI2-THOR simulator [29].

Large-scale imitation learning with random embodiments. We train our policy using the architecture outlined in Sec. 3.3 on the collected trajectories. At each time step, the linear actor-critic head in Causal Transformer Decoder predicts the action logits. The cross-entropy loss is computed between the logits π^t and the expert action. At training, we use a batch size of 240 trajectories and each trajectory has a temporal context window of 100 steps. We train our model on 8× H100 GPUs (80 GB memory per GPU) using the AdamW optimizer with a learning rate of $2{\cdot}10^{-4}$ for 80k iterations.

RL finetuning with random embodiments. Following the training recipe in FLaRe [24], we further perform a large-scale RL finetuning using AllenAct [49] on the randomized embodiments in simulation. Our training includes 1M random embodiments, 50k procedurally generated ProcTHOR houses [12] with ∼40k annotated 3D objects [18]. RL finetuning is specifically important for the policy to learn to navigate a diverse set of embodiments through trial-and-error. In particular, as the Ring policy lacks explicit information about its embodiment, it must implicitly infer it, which requires extensive exploration and trial-and-error. We use DD-PPO with 64 parallel environments and 128 rollout steps across 4 machines (each with 8× H100 GPUs) using the AdamW optimizer with a learning rate of $2{\cdot}10^{-5}$ for 40M training steps. Following FLaRe, we turn off the entropy term in the PPO loss to avoid catastrophic forgetting. For fair comparison, we use the reward function of [12],

$$r_t = max(0, min\Delta_{0:t-1} - \Delta_t) + s_t - \rho, \qquad (1)$$

where $min\Delta_{0:t-1}$ denotes the minimum L2 distance between the agent and any target object up to time $t-1$, Δ_t is the most recent L2 distance, s_t is a success reward, and ρ denotes the step penalty of 0.01 to encourage the policy to finish the task efficiently. The agent must issue `Done` to indicate that it has found the target object to get the success reward $s_t = 10$, otherwise $s_t = 0$.

4. Experiments

Our experiments show that Ring operates effectively across a wide range of embodiments, including actual robots (*Stretch RE-1*, *LoCoBot*, and *Unitree Go1*) and human evaluation with *Navigation Assistants*, despite being trained exclusively in simulation **without** any direct exposure to real robot embodiments. Our key results are:

1. Ring generalizes **zero-shot** to 4 *truly* unseen embodiments, despite never being trained on them, and achieves state-of-the-art performance across multiple benchmarks (Sec. 4.1).
2. Our policy, trained solely in simulation on randomized embodiments, transfers directly to the **real-world**, on 3 real robots and navigation assistants (human evaluation). (Sec. 4.2).
3. Ring can be easily adapted to **embodiment-specialized** policies with minimal finetuning. It achieves better performance on each specific robot (Sec. 4.3).
4. Ring shows **embodiment-adaptive behavior**, adjusting its strategies based on the agent's body (Sec. 4.4).
5. We present ablation studies and explore finetuning with collision penalties to enable the policy to take more conservative actions (Sec. 4.5).

4.1. Ring generalizes zero-shot to unseen embodiments

In this section, 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.

Baselines. For our baselines, we selected prior works in both imitation learning (IL) and reinforcement learning (RL). Each baseline is trained on a specific embodiment and evaluated in a zero-shot setting on four different embodiments. SPOC [16] is a supervised IL baseline trained on shortest-path expert trajectories in AI2-THOR. PoliFormer [62] is a state-of-the-art transformer-based policy in object goal navigation, trained from scratch using RL. FLaRe [24] is a approach for efficient policy finetuning that combines IL and RL. Specifically, SPOC [17] is trained with IL on Stretch RE-1 using 100k expert trajectories; SPOC-2.3M is trained on more expert trajectories; Poliformer [62] is trained from scratch on each embodiment individually over 300M RL steps; and FLaRe [24] 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 (examples shown in Fig. 1-A and Fig. 2). *Note that all four target embodiments were unseen during training, and information on embodiment is not provided during evaluation.* We evaluate on the navigation benchmark in Chores-$\mathbb{S}$ [17], 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. Table 2 presents the zero-shot evaluation of all

Model	Loss	Train Embodiment	Benchmark Embodiment					
			Stretch	**Stretch** (Nav Cam)	**Stretch** (Factory Config)	**LoCoBot**	**Unitree A1**	Average
SPOC [17]	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	IL only	Stretch	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 [62]	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 [24]	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 2. **Zero-shot Results.** Ring shows 0-shot generalization on 4 unseen embodiments. Stretch RE-1 and LoCoBot are evaluated on the Chores-$\mathbb{S}$ [17] ObjectNav benchmark (Unitree A1 is evaluated on a different benchmark which accounts for the agent's lower height). "Stretch" without further qualification refers to the 2-camera variation on an RE-1 platform, as in [17]. All previous methods drastically fail to generalize to embodiments other than their training. Gray* numbers are evaluated on the training embodiment, otherwise evaluated zero-shot on an unseen embodiment.

policies across four embodiments. We compare *Success Rate* and *Success Weighted by Episode Length (SEL [15])*, 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. For example, SPOC trained on the Stretch RE-1 with two cameras shows a gradual decrease in performance as the evaluation embodiment diverges from left to right on the top row in Table 2. It is worst when evaluated on the Unitree A1, which has a substantial height difference. 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 → 68.5) and Unitree A1 (55.3 → 72.0). This shows that Ring benefits from training across random embodiments at scale, leading to a more effective navigation policy which even outperforms some embodiment-specialized policies.

4.2. Ring transfers to real-world embodiments despite being purely trained in simulation

Robot Evaluation We zero-shot evaluate our policy on 3 unseen robots in a real-world apartment. All evaluations are performed directly in a large-scale apartment (Fig. 4), without any further adaptation or real-world-specific finetuning. We used the same evaluation set of 15 tasks for LoCoBot [10, 12, 62] (3 different starting poses with 5 different targets), and 18 tasks for Stretch RE-1 [17, 24, 62] (3 different starting poses with 6 different goal specifications), respectively. We create a new evaluation set for Unitree Go1 with 3 starting poses and 4 objects (*toilet, sofa, TV, trashcan*) positioned to accommodate the robot's lower height, ensuring that the objects can be visible from its lower viewpoint.

Human Evaluation To further demonstrate our policy's generalization capability to unseen embodiments, we evaluate it as a navigation assistant with humans as new, unseen embodiments. We asked 5 participants to navigate in a real-world kitchen area, following the policy's output actions on their phones. Each individual has unique characteristics, including step size, height, rotation angles, and camera-holding posture. Each person navigates to three different objects (`Mug`, `Apple`, `Houseplant`), resulting in a total of 15 trajectories. We compare Ring with FLaRe [24],

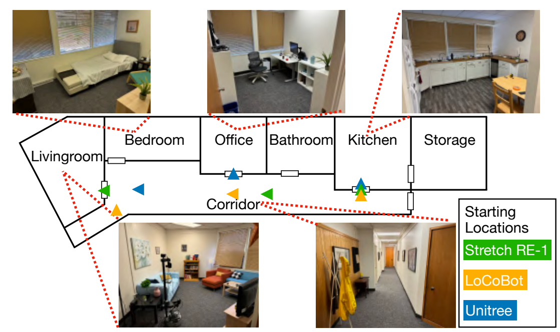

Figure 4. **Real Evaluation Environment**. Our real-world evaluations are performed in a multi-room apartment with a long corridor, shown here with the three starting locations for three different robots' evaluations.

Model	Train Embodiment	Eval Embodiment			
		Stretch	**Stretch (FC)**	**LoCoBot**	**Unitree Go1**
ProcTHOR [12]	LoCoBot	-	-	26.7	-
Phone2Proc [10]	LoCoBot	-	-	66.7	-
SPOC [16]	Stretch	50.0	-	-	-
PoliFormer [62]	Stretch	83.3	33.3	-	-
	LoCoBot	-	-	80.0	-
	Unitree Go1	-	-	-	41.7
FLaRE [24]	Stretch	**94.4**	-	-	-
Ring-Zero-Shot	Ring-Random	83.3	**72.2**	**80.0**	**80.0**

Table 3. **Real-world Results**. Ring transfers zero-shot to the real-world without any finetuning. Gray numbers are evaluated on same embodiment as their training. Ring achieves 78.9% success rate on average across 4 real-world robots.

Model	Train Embodiment	Object	Human Participants					Average
			H1	**H2**	**H3**	**H4**	**H5**	
FLARE [24]	Stretch RE-1	🍎	✓	✗	✗	✗	✗	40.0%
		🌱	✗	✓	✓	✓	✓	
		☕	✗	✗	✗	✗	✓	
RING-ZERO-SHOT	RING-Random	🍎	✓	✗	✗	✗	✓	**73.3%**
		🌱	✓	✓	✓	✓	✓	
		☕	✗	✓	✓	✓	✓	

Table 4. **Human Evaluation.** Five individuals navigate to 3 different objects (🍎 `Apple`, 🌱 `Houseplant`, ☕ `Mug`) following the policy's output actions on their phones in a kitchen area (example trajectories in Fig. 5). RING-ZERO-SHOT shows much better generalization to human embodiment than the FLaRE baseline trained on Stretch RE-1.

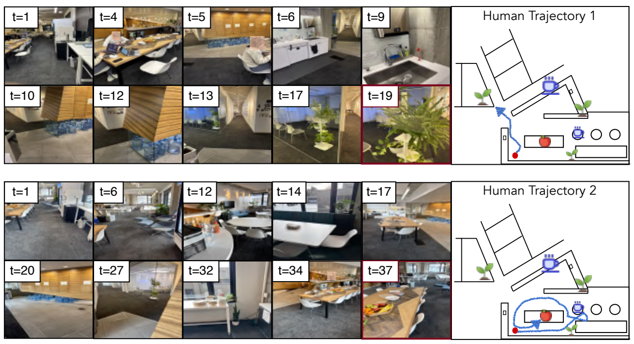

Figure 5. **Human Trajectories**. Two sample trajectories from two individuals navigating to a houseplant and an apple using RING.

which was trained exclusively on the Stretch RE-1. Tab. 4 shows that RING consistently outperforms FLaRe across target objects and different participants. Fig. 5 shows two qualitative results achieved by RING.

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

Although RING is a universal policy that works zero-shot across a broad range of embodiments, there are often cases where an embodiment-specialized policy is needed to achieve the best performance. In this section, we demonstrate that RING can be easily adapted to a *robot-specialized policy* with minimal fine-tuning, resulting in better performance on the target embodiment.

Baselines. We use FLaRe [24] as a baseline. It has shown successful adaptation to new tasks and embodiments. This baseline is pretrained on Stretch RE-1 and finetuned on each of the three embodiments using up to 20M RL steps.

Implementation Details. We finetune RING, pretrained on randomized embodiments, on each individual robot for up to 20M RL steps, while keeping all hyperparameters consistent with FLaRe to ensure a fair comparison. Following FLaRe, we repurpose two actions `RotateBase(±6°)` to `TiltCamera(±30°)` to allow camera movements for LoCoBot. Note that this movement is not allowed during the zero-shot evaluation.

Results. Fig. 7 shows that RING adapts efficiently to specific embodiments with minimal finetuning, leading to embodiment-specialized policies with even better performance. For LoCoBot and Unitree-A1, FLaRe's performance remains lower than on Stretch RE-1, indicating that pretraining on one embodiment and finetuning across embodiments cannot achieve the best results. This highlights the need for a policy that can consistently adapt to any embodiment with lightweight finetuning.

4.4. RING has embodiment-adaptive behavior.

The behavior of optimal navigation policy, $\pi^*_\theta(a_t \mid o^e_t,)$, should be strongly shaped by the agent's body. For instance, an agent with a narrow FOV must explore more to effectively perceive its surroundings compared to an agent with a wider FOV. A smaller agent can navigate through narrow hallways or under furniture, and a larger agent may need to take more conservative paths. A question arises: does the policy show the same navigation behaviors across different embodiments, or does it adjust its strategy accordingly?

Our qualitative results confirm that the policy has an **embodiment-adaptive behavior**. In Fig. 6-A,B, both Stretch RE-1 and Unitree A1 start from the same pose behind the bed. The quadruped robot directly moves under the bed because of its lower height, while the Stretch RE-1 bypasses it. We observe that RING *implicitly* infers embodiment parameters from visual observations and transition dynamics, dynamically adjusting its navigation strategy accordingly. It does not have access to any privileged information about its current body.

Visual observation reveals parameters, such as camera specs and, in some cases, agent height. However, visual information alone can be insufficient, leading the agent to rely on collisions to infer body dimensions. In Fig. 6-C, the agent matches the Stretch RE-1's height but has a camera positioned as low as the quadruped's. Initially, it assumes a lower height and attempts to go under the bed, but after colliding, it adjusts to maneuver around the bed, similar to the Stretch RE-1. This embodiment-adaptive navigation strategy adjustment is an interesting emergent behavior that would not have been possible without training across the exhaustive space of embodiments at scale.

4.5. Ablation Studies

A More Powerful Pretrained Visual Encoder. The default vision encoder used in our policies is the pretrained SIGLIP-VIT-B/16. In this section, we examine the impact of using a more powerful visual encoder on RING's performance. We train RING-LARGE using OpenAI's VIT-L/14 336PX CLIP model [38]. Table 5 compares the results, showing that a stronger visual encoder significantly improves zero-shot performance across all four embodi-

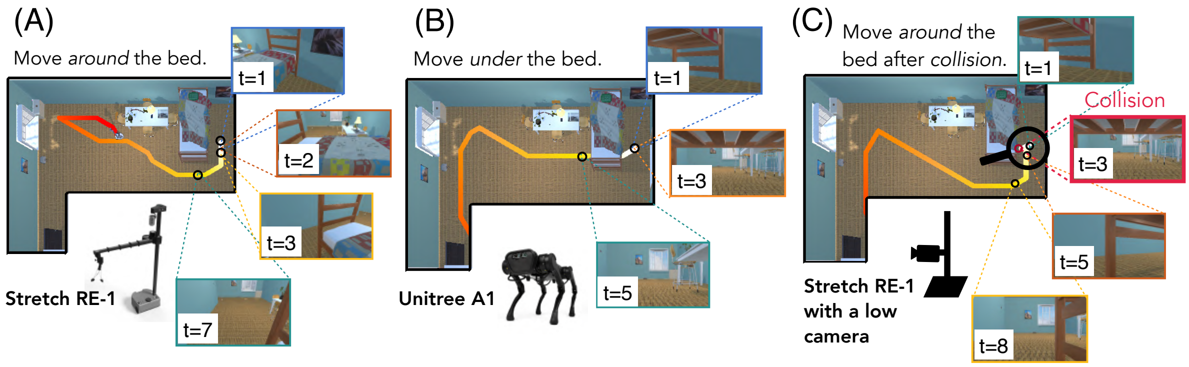

Figure 6. RING has **embodiment-adaptive** behavior, adjusting its navigation strategy based on the embodiment. The quadruped robot (B), due to its lower height, walks under the bed, while the taller Stretch-RE1 robot (A) navigates around it. In (C), an agent with the same height as Stretch-RE1 but a lower camera position initially attempts to move under the bed, assuming a shorter height. After colliding, it adapts its strategy and navigates around the bed, similar to Stretch-RE1.

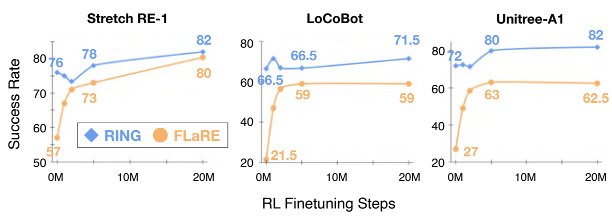

Figure 7. **Embodiment-Specialized Adaptation.** RING, pretrained on randomized embodiments, shows efficient adaptation to robot-specialized policies with minimal fine-tuning. Baseline performance on LoCoBot and Unitree-A1 remains lower as they are fine-tuned on a different embodiment than the one used in pretraining. In contrast, RING policy achieves consistent performance across all 3 embodiments, highlighting its capability for robot-specialized adaptation.

Model	Visual Encoder	Benchmark Embodiment			
		Stretch	**Stretch** (Nav)	**LoCoBot**	**Unitree A1**
RING	SIGLIP-VIT-B/16	76.0	74.0	66.5	72.0
RING-LARGE	VIT-L/14 336PX CLIP	**83.8**	**77.7**	**75.3**	**79.9**

Table 5. **A Stronger Visual Encoder**. Using a more powerful vision encoder significantly improves the zero-shot performance across all embodiments.

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 6. **Collision Penalty.** Adding a small collision penalty (0.1) to the reward function results in 50% less collision, forcing the policy to take more conservative paths.

ments (approximately 9% improvement on average). A larger visual encoder is particularly beneficial in our policy, as the visual observations are highly varied due to randomized camera parameters. To ensure fair comparison with the baselines and because VIT-L/14 is more computationally demanding, we chose to use the VIT-B/16 encoder for our main experiments. We will release the training code for the community for those interested in training with the larger visual encoder.

Include collision penalty to take safer routes. As our agents have randomized body dimensions and are not explicitly informed about their embodiment, they may occasionally collide before understanding their correct body size. In this section, we demonstrate that adding a small collision penalty of 0.1 to the reward function can reduce collisions rate by 50% (lowering the collision rate *CR* from 7% to 4%). The resulting policy is more conservative, regardless of embodiment size.

To quantify these results, we created a custom benchmark similar to CHORES-$\mathbb{S}$ [17], 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 (percentage of episodes without any collisions)*. As shown in Table 6, adding the collision penalty reduces the collision rate (CR) ($7.77\% \rightarrow 4.03\%$) as well as increases the percentage of trajectories without collisions ($46.90\% \rightarrow 60.57\%$).

5. Conclusion

In this paper, we introduce RING (**R**obotic **I**ndoor **N**avigation **G**eneralist), an embodiment-agnostic policy, trained solely in simulation with diverse randomly initialized embodiments at scale (1M embodiments). RING displays zero-shot generalization capability to various unseen embodiments, maintaining consistent performance across all. Our experimental results demonstrate that RING achieves state-of-the-art results on novel embodiments, including in some cases improving over embodiment-specific policies. It can be directly deployed to the real-world despite being solely trained in simulation. Finally, RING is able to dynamically adjust its behavior based on its embodiment and interactions with the environment.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *CoRL*, 2022. 3

[2] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav revisited: On evaluation of embodied agents navigating to objects. *CoRR*, abs/2006.13171, 2020. 3

[3] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *ECCV*, 2024. 2

[4] Nico Bohlinger, Grzegorz Czechmanowski, Maciej Piotr Krupka, Piotr Kicki, Krzysztof Walas, Jan Peters, and Davide Tateo. One policy to run them all: Towards an end-to-end learning approach to multi-embodiment locomotion. In *CoRL*, 2024. 3

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael S. Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong T. Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: vision-language-action models transfer web knowledge to robotic control. *CoRR*, abs/2307.15818, 2023. 2

[6] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *ICLR*, 2020. 3

[7] Lawrence Yunliang Chen, Chenfeng Xu, Karthik Dharmarajan, Zubair Irshad, Richard Cheng, Kurt Keutzer, Masayoshi Tomizuka, Quan Vuong, and Ken Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. In *CoRL*, 2024. 2

[8] Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei 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. 3

[9] Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani,

Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *ICRA*, 2024. 2

[10] Matt Deitke, Rose Hendrix, Luca Weihs, Ali Farhadi, Kiana Ehsani, and Aniruddha Kembhavi. Phone2Proc: Bringing robust robots into our chaotic world, 2022. 6, 4

[11] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A Universe of Annotated 3D Objects. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13142–13153, 2022. 2, 3

[12] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-scale embodied AI using procedural generation. In *NeurIPS*, 2022. 2, 5, 6

[13] Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. *arXiv preprint arXiv:2408.11812*, 2024. 2

[14] Jiafei Duan, Wentao Yuan, Wilbert Pumacay, Yi Ru Wang, Kiana Ehsani, Dieter Fox, and Ranjay Krishna. Manipulate-anything: Automating real-world robots using vision-language models. *arXiv preprint arXiv:2406.18915*, 2024. 2

[15] Ainaz Eftekhar, Kuo-Hao Zeng, Jiafei Duan, Ali Farhadi, Ani Kembhavi, and Ranjay Krishna. Selective visual representations improve convergence and generalization for embodied ai. In *ICLR*, 2023. 3, 6

[16] Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, et al. Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. In *CVPR*, 2024. 5, 6, 4

[17] Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, et al. Spoc: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. In *CVPR*, pages 16238–16250, 2024. 2, 3, 4, 5, 6, 8, 1

[18] Allen Institute for AI. ObjaTHOR: Python package for importing and loading external assets into ai2thor. `https://github.com/allenai/objathor`, 2024. 5

[19] Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. In *CoRL*, 2024. 2

[20] Aric A. Hagberg, Daniel A. Schult, Pieter Swart, and JM Hagberg. Exploring Network Structure, Dynamics, and Function using NetworkX. *Proceedings of the Python in Science Conference*, 2008. 2

[21] Sudarshan Harithas and Srinath Sridhar. Motionglot: A multi-embodied motion generation model. *arXiv preprint arXiv:2410.16623*, 2024. 2

[22] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.*, 4:100–107, 1968. 2

[23] Joey Hejna, Chethan Bhateja, Yichen Jian, Karl Pertsch, and Dorsa Sadigh. Re-mix: Optimizing data mixtures for large scale imitation learning. In *CoRL*, 2024. 2

[24] Jiaheng Hu, Rose Hendrix, Ali Farhadi, Aniruddha Kembhavi, Roberto Martin-Martin, Peter Stone, Kuo-Hao Zeng, and Kiana Ehsan. Flare: Achieving masterful and adaptive robot policies with large-scale reinforcement learning finetuning. *arXiv preprint arXiv:2409.16578*, 2024. 2, 3, 4, 5, 6, 7

[25] Kushal Kedia, Prithwish Dan, and Sanjiban Choudhury. One-shot imitation under mismatched execution. *arXiv preprint arXiv:2409.06615*, 2024. 2

[26] Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. Goat-bench: A benchmark for multi-modal lifelong navigation. In *CVPR*, 2024. 3

[27] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *arXiv preprint arXiv:2403.12945*, 2024. 2

[28] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. In *CoRL*, 2024. 2

[29] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Kumar Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *ArXiv*, abs/1712.05474, 2017. 3, 5

[30] Antonio Loquercio, Ana I Maqueda, Carlos R Del-Blanco, and Davide Scaramuzza. Dronet: Learning to fly by driving. In *RA-L*, 2018. 2

[31] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. ZSON: zero-shot object-goal

navigation using multimodal goal embeddings. In *NeurIPS*, 2022. 3

[32] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. *ICML*, 2024. 3

[33] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *RSS*, 2024. 2

[34] Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 2

[35] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2023. 3

[36] Austin Patel and Shuran Song. GET-Zero: Graph Embodiment Transformer for Zero-shot Embodiment Generalization. *CoRR*, abs/2407.15002, 2024. 2

[37] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, Vladimir Vondrus, Théophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots. *CoRR*, abs/2310.13724, 2023. 3

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 7

[39] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 2024. 3

[40] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5163–5173. IEEE, 2022. 3

[41] Milad Shafiee, Guillaume Bellegarda, and Auke Ijspeert. Manyquadrupeds: Learning a single locomotion policy for diverse quadruped robots. In *ICRA*, 2024. 3

[42] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023. 2, 3

[43] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023. 2, 3

[44] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *ICRA*, 2023. 3

[45] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 2

[46] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 2

[47] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *NeurIPS*, 2024. 2

[48] Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. MultiON: Benchmarking Semantic Map Memory using Multi-Object Navigation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3

[49] Luca Weihs, Jordi Salvador, Klemen Kotar, Unnat Jain, Kuo-Hao Zeng, Roozbeh Mottaghi, and Aniruddha Kembhavi. AllenAct: A framework for embodied ai research. *arXiv preprint arXiv:2008.12760*, 2020. 5

[50] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. 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. 3

[51] Wenli Xiao, Haoru Xue, Tony Tao, Dvij Kalaria, John M Dolan, and Guanya Shi. Anycar to anywhere: Learning universal dynamics model for agile and adaptive mobility. *arXiv preprint arXiv:2409.15783*, 2024. 3

[52] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery. In *CoRL*, 2023. 2

[53] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In *CoRL*, 2024. 2

[54] Zhuo Xu, Hao-Tien Lewis Chiang, Zipeng Fu, Mithun George Jacob, Tingnan Zhang, Tsang-Wei Edward Lee, Wenhao Yu, Connor Schenck, David Rendleman, Dhruv Shah, et al. Mobility vla: Multimodal instruction navigation with long-context vlms and topological graphs. In *CoRL*. 3

[55] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Sym-*

posium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation', pages 146–151. IEEE, 1997. 3

[56] Jonathan Yang, Catherine Glossop, Arjun Bhorkar, Dhruv Shah, Quan Vuong, Chelsea Finn, Dorsa Sadigh, and Sergey Levine. Pushing the limits of cross-embodiment learning for manipulation and navigation. In *arXiv preprint arXiv:2402.19432*, 2024. 2

[57] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectnav. *CoRR*, abs/2104.04112, 2021. 3

[58] Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. Hm3d-ovon: A dataset and benchmark for open-vocabulary object goal navigation. *IROS*, 2024. 3

[59] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024. 3

[60] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *CoRL*, 2022. 2

[61] Kuo-Hao Zeng, Luca Weihs, Ali Farhadi, and Roozbeh Mottaghi. Pushing it out of the way: Interactive visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3

[62] Kuo-Hao Zeng, Zichen Zhang, Kiana Ehsani, Rose Hendrix, Jordi Salvador, Alvaro Herrasti, Ross Girshick, Aniruddha Kembhavi, and Luca Weihs. Poliformer: Scaling on-policy rl with transformers results in masterful navigators. In *CoRL*, 2024. 2, 3, 4, 5, 6

[63] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *ICCV*, abs/2303.15343, 2023. 3