

# Emergence of Implicit System Identification via Embodiment Randomization

Pranav Putta<sup>\*1</sup> Gunjan Aggarwal<sup>\*1</sup> Roozbeh Mottaghi<sup>2</sup> Dhruv Batra<sup>1,2</sup>  
Naoki Yokoyama<sup>1</sup> Joanne Truong<sup>1</sup> Arjun Majumdar<sup>1</sup>

<sup>1</sup>Georgia Institute of Technology <sup>2</sup>FAIR, Meta AI <sup>\*</sup>equal contribution

## 1. Introduction

Modern robots come in various shapes and sizes, and use different sensor suites (*e.g.*, different cameras with different camera parameters) to observe their environment. By contrast, when training embodied agents (*i.e.*, virtual robots) in simulation, these differences are typically ignored and a single robot embodiment (height, radius, camera parameters, *etc.*) is used. As a result, behavior policies learned in such simulations only work for the fixed embodiment used in training, and generalize poorly when even minor changes to the robot’s embodiment are introduced, such as changing the robot’s height or camera field-of-view. However, such variations are inevitable as new robots are developed or existing robots are updated and modified for new applications. How do we develop embodied agents that can adapt, as needed, to new embodiments during deployment?

Recent work has proposed a number of techniques for addressing this embodiment generalization challenge [3, 8, 11, 14]. One line of work trains “universal” navigation policies that are conditioned on a learned robot embedding vector [11]. Alternatively, Shah *et al.* [8] train a navigation policy using data from multiple robot embodiments by conditioning the policy on past observations (used to infer a robot’s embodiment) and using a normalized action space (to generalize actions across embodiments).

This work studies a simpler alternative: *embodiment randomization*. Akin to domain randomization [10], in embodiment randomization the robot configuration is randomly sampled at the beginning of each training episode. In simulation, modifying a robot’s embodiment is trivial. Thus, embodiment randomization is an inexpensive and intuitive technique for addressing the embodiment generalization challenge.

We empirically investigate embodiment randomization using the image-goal navigation (ImageNav) task [15]. We observe that, on this task, policies trained using a fixed embodiment catastrophically fail to generalize to new robot configurations. However, with embodiment randomization, agents recover a substantial portion of this lost performance.

Furthermore, we discover that policies trained with embodiment randomization implicitly learn to perform system

identification. Specifically, we find that many (but not all) configuration parameters can be predicted from intermediate representations within agents trained with embodiment randomization, allowing these agents to adapt to new embodiments during deployment. Our findings suggest that embodiment randomization is a simple but effective method for training navigation agents that are generalizable to new embodiments zero-shot.

## 2. Embodiment Randomization

Agent Property	Train Embodiment		Eval Embodiment	
	LoCoBot	Multi-Embodiment	In-Dist.	Out-of-Dist.
Height (m)	0.61	[0.61, 1, 1.5]	[0.75, 1.25]	[0.25, 2]
Radius (m)	0.18	[0.1, 0.18, 0.3]	[0.15, 0.25]	[0.05, 0.4]
Step Size (m)	0.2	[0.1, 0.2, 0.3]	[0.15, 0.25]	[0.05, 0.4]
Turn Angle (deg.)	30	[15, 30, 45, 60]	[22.5, 37.5, 52.5]	[7.5, 72]
Camera Tilt (deg.)	0	[-30, -15, 0, 15, 30]	[-22.5, 22.5]	[-45, 45]
Camera FOV (deg.)	55	[55, 90, 120]	[75, 105]	[30, 150]

Table 1. Agent parameters used for embodiment randomization.

We use embodiment randomization to train a single policy network that can perform well across a range of embodiment configurations. In training, an agent embodiment is sampled at the start of every episode from the combinatorial space of configurations shown in Table 1 column 3. Specifically, we vary the robot’s height, radius, step size, turn angle, camera tilt, and camera field-of-view (FoV) to simulate different robot embodiments.

## 3. Experimental Setup

**Image-Goal Navigation (ImageNav).** In ImageNav [15], an agent receives RGB-D observations, and is tasked with navigating from a randomly sampled initial position to a goal location. The goal is represented as an RGB image constructed from the goal location. We consider navigation agents that use four discrete actions: `STEP_FORWARD`, `TURN_LEFT`, `TURN_RIGHT`, and `STOP`. Agents are successful if they call `STOP` within 1m of the goal. We report two navigation metrics: Success Rate (SR  $\uparrow$ ) and Success weighted by Path Length (SPL  $\uparrow$ ) [1].

**Navigation Agent.** We adapt the agent architecture from OVRL-v2 [13]. Specifically, image observations are rendered (by a simulator) at a 640x480 resolution, then

dowsampled to 160x120, which is used as input to the policy. Our policy processes RGB observations using a pre-trained ViT from [13] and depth observations using a randomly-initialized ResNet-18 [4]. The encoded sensor data is projected using a linear layer, and consumed by a recurrent network (a GRU [2]) to predict actions. We train agents with reinforcement learning using DD-PPO [12] for 1 billion frames of experience in the Habitat simulator [7,9] using 800 training scenes from HM3D [6].

#### 4. Experimental Findings

We compare the performance of an agent trained with embodiment randomization to an agent trained with a single embodiment. The single embodiment agent uses the LoCoBot specifications [5] (Table 1 column 2), and the multi-embodiment agent uses the parameters specified in Table 1 column 3. The multi-embodiment setting includes the LoCoBot parameters to evaluate whether simply expanding the space of robot configurations used during training can be beneficial for generalization to novel embodiments. We evaluate both agents using 3 embodiments at test-time: (a) LoCoBot embodiment, (b) embodiments in-distribution to the multi-embodiment training parameters (Table 1 column 4), and (c) embodiments out-of-distribution to the multi-embodiment training parameters (Table 1 column 5).

Train Embodiment	Eval Embodiment					
	LoCoBot		In-Dist.		Out-of-Dist.	
	SR ↑	SPL ↑	SR ↑	SPL ↑	SR ↑	SPL ↑
LoCoBot	0.59	0.38	0.11	0.06	0.02	0.01
Multi-Embodiment	0.55	0.34	0.60	0.41	0.12	0.06

Table 2. ImageNav evaluation performance for single and multi-embodiment trained policies.

In Table 2, we observe that the policy trained with embodiment randomization (row 2) generalizes to configurations not seen during training and substantially outperforms the single embodiment agent (row 1). Specifically, the single embodiment agent fails to generalize to novel embodiments – success rate drops from 59% to 11% (-48%) when evaluating using the multi-embodiment in-distribution setting. More significantly, success rate drops to 2% (-57%) when evaluated in the multi-embodiment out-of-distribution setting. This demonstrates that policies trained with a single embodiment are sensitive to agent parameters, and the policy must be re-trained for each embodiment expected at test-time. By contrast, the multi-embodiment policy is robust to changes in embodiment parameters and achieves a strong success rate of 60% SR when evaluated in-distribution, and 12% SR when evaluated out-of-distribution.

Next, we examine why the multi-embodiment policy generalizes to new embodiments. Specifically, after policy training, we train separate *non-linear probes* to predict em-

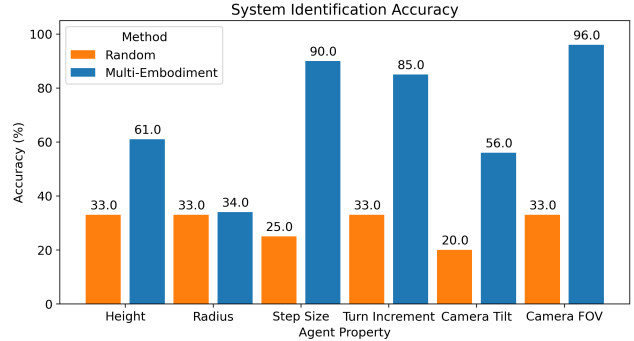


Figure 1. System identification accuracy for the multi-embodiment policy, compared to random chance.

bodiment parameters such as camera FoV. Each probe consists of a 2-layer MLP that takes as input the hidden state of the GRU and outputs a predicted embodiment parameter.

From Figure 1, we see that the non-linear probes are able to accurately predict many of the embodiment parameters, suggesting that the agent performs *implicit system identification*. Particularly, parameters such as step size, turn increment, and camera FoV are detected with high accuracy (90%, 85%, and 96% respectively). This suggests that embodiment randomization can be a simple technique for training embodiment-aware agents, which allow zero-shot generalization to novel embodiments during deployment. Lastly, we note that radius is difficult to predict accurately (34% accuracy, compared to 33% accuracy for random chance). We hypothesize that to accurately predict the radius, the agent would need to collide often with the environment. However, we found our policy mostly avoids collisions, only colliding 3 times per episode on average.

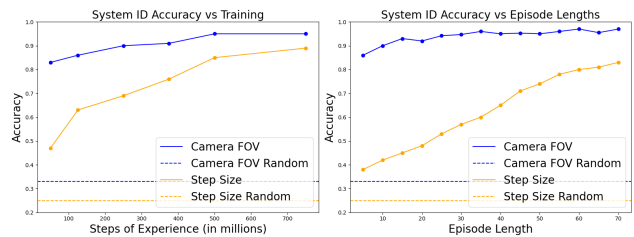


Figure 2. System identification over steps of experience and episode length for camera FOV and step size parameters.

We additionally examine the evolution of system identification. In Figure 2, we plot the system identification accuracy for over steps of experience in training and over episode length during evaluation. Certain parameters, such as Camera FoV are detected early in the episode (within 20 steps), while step size is identified as the agent steps through the environment, and can take more than 50 steps to detect. More details can be found in Appendix A.

Overall, we find that embodiment randomization is a simple, scalable, and effective method for training visual navigation agents that can be zero-shot deployed to various embodiments.

## Acknowledgements

The Georgia Tech effort was supported in part by ONR YIP and ARO PECASE. JT was supported by an Apple Scholars in AI/ML PhD Fellowship. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government, or any sponsor.

## References

- [1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. [1](#)
- [2] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv e-prints*, page arXiv:1409.1259, Sept. 2014. [2](#)
- [3] Gilbert Feng, Hongbo Zhang, Zhongyu Li, Xue Bin Peng, Bhuvan Basireddy, Linzhu Yue, ZHITAO SONG, Lizhi Yang, Yunhui Liu, Koushil Sreenath, and Sergey Levine. Genloco: Generalized locomotion controllers for quadrupedal robots. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1893–1903. PMLR, 14–18 Dec 2023. [1](#)
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [2](#)
- [5] Adithyavairavan Murali, Tao Chen, Kalyan Vasudev Alwala, Dhiraj Gandhi, Lerrel Pinto, Saurabh Gupta, and Abhinav Gupta. Pyrobot: An open-source robotics framework for research and benchmarking. *arXiv preprint arXiv:1906.08236*, 2019. [2](#)
- [6] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. [2](#)
- [7] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *International Conference on Computer Vision (ICCV)*, 2019. [2](#)
- [8] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. GNM: A General Navigation Model to Drive Any Robot. In *arXiv*, 2022. [1](#)
- [9] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [2](#)
- [10] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017. [1](#)
- [11] Joanne Truong, Denis Yarats, Tianyu Li, Franziska Meier, Sonia Chernova, Dhruv Batra, and Akshara Rai. Learning Navigation Skills for Legged Robots with Learned Robot Embeddings. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020. [1](#)
- [12] Erik Wijmans, Abhishek Kadian, Ari S. 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 *International Conference on Learning Representations (ICLR)*, 2020. [2](#)
- [13] Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav. *arXiv preprint arXiv:2303.07798*, 2023. [1](#), [2](#)
- [14] Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems (RSS)*, 2017. [1](#)
- [15] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Kumar Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. *ICRA*, 2017. [1](#)

