How to Train PointGoal Navigation Agents on a (Sample and Compute) Budget

Extended Abstract

Erik Wijmans
Georgia Institute of Technology
Facebook AI Research
United States
etw@gatech.edu

Irfan Essa
Georgia Institute of Technology
Google Atlanta
United States

Dhruv Batra
Georgia Institute of Technology
Facebook AI Research
United States

ABSTRACT

PointGoal navigation has seen significant recent interest and progress, spurred on by the Habitat platform and associated challenge [21]. In this paper, we study PointGoal navigation under both a *sample* budget (75 million frames) and a *compute* budget (1 GPU for 1 day). We conduct an extensive set of experiments, cumulatively totaling over 50,000 GPU-hours, that let us identify and discuss a number of ostensibly minor but significant design choices – the advantage estimation procedure (a key component in training), and visual encoder architecture. Overall, these design choices lead to considerable and consistent improvements. Under a sample budget, performance for RGB-D agents improves 3 SPL on Gibson (4% relative improvement) and *20* SPL on Matterport3D (43% relative improvement). Under a compute budget, performance for RGB-D agents improves by 3 SPL on Gibson (5% relative improvement) and *15* SPL on Matterport3D (50% relative improvement). Our findings and recommendations will serve to make the community's experiments more efficient – to reach 50 SPL with RGB-D on Matterport3D, they reduce the samples needed by 3x and the training time 2x.

KEYWORDS

Embodied AI; Navigation; Reinforcement Learning

ACM Reference Format:

Erik Wijmans, Irfan Essa, and Dhruv Batra. 2022. How to Train PointGoal Navigation Agents on a (Sample and Compute) Budget: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

Galvanized by fast simulation platforms [15, 20, 21, 29], large rich 3D datasets [4, 25, 30], and the success of deep reinforcement learning [18, 24, 26], training virtual robots (embodied agents) in simulation has garnered considerable interest in recent years. Works have developed a rich set of tasks, ranging from PointGoal navigation [1], to grounded instruction following [2, 23] and question answering [8, 11, 27].

In this rich space of tasks, PointGoal navigation with GPS+Compass has emerged as a test-bed problem due to its property of 'easy to get off the ground, but difficult to fully solve'. Specifically, Wijmans *et al.* [28] show that good performance can be obtained with under a week of GPU time but near-perfect performances currently

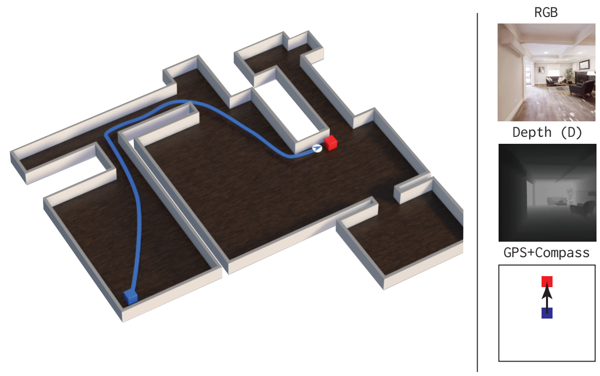

Figure 1: PointGoal navigation [1]. An agent is initialized in a novel environment (blue square) and task with navigation to a point specified relative to the start location (red square) – *e.g.* (5, 2) means go 5 meters forward and 2 meters right. It must do so from egocentric inputs – RGB-D and GPS+Compass– and without a map.

requires *half-a-year* of GPU time. This property makes PointNav an ideal test-bed for empirical studies as it provides ample dynamic range – we can be confident that improves are statically significant as the difference in performance is large. Moreover still, findings and improvements in PointNav with GPS+Compass translate to improved performance on other tasks (PointNav without GPS+Compass [9, 19], ObjectGoal navigation [3, 6], RoomGoal navigation [17]) and to navigation by real robots [14]. The model architecture developed in Chaplot *et al.* [5] has directly lead improvements on ObjectNav [6], ImageNav [7], and PointNav without GPS+Compass [19].

Given this interest, we present a systematic analysis of what matters and what what doesn't matter for learning PointNav with GPS+Compass. We identify and discuss a number of ostensibly minor but significant design choices – the advantage estimation procedure (a keep component in training) and visual encoder architecture – that have large impacts on agent performance.

We examine these differences in two contexts, i) *sample* efficiency, and ii) *compute* efficiency. To study sample efficiency, we train all agent variants for a fixed number of samples – 75 million steps, a high but feasible number of samples [5, 10, 21, 22, 31]. To study compute efficiency, we ask a subtly different but important question: ***How far can we get with 1 GPU for 1 day?*** We instead train for

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

Table 1: Results on PointNav at 75 million. Performance on Gibson and Matterport3D validation sets (validation used to reduce exposure to test). Checkpoint selection done by validation SPL for each run independently. We find ResNet18 improves performance and normalized advantage either harms performance or has no effect. Mean and 95% CI from 5 runs.

			SimpleCNN				ResNet18			
			Gibson		Matterport3D		Gibson		Matterport3D	
#	Sensors	Norm Adv.	Success ↑	SPL ↑	Success ↑	SPL ↑	Success ↑	SPL ↑	Success ↑	SPL ↑
1	RGB	✓	14.2±24.97	11.2±19.56	00.0±0.00	00.0±0.00	79.3±1.57	68.0±1.26	53.1±2.03	40.1±0.86
2		-	78.5±2.20	63.2±1.76	55.1±3.28	37.6±1.77	81.8±2.14	68.4±1.38	50.8±1.46	37.9±1.40
3	RGB-D	✓	85.1±1.29	77.4±1.32	55.5±31.39	45.9±26.00	87.4±0.75	80.5±0.77	78.8±1.02	66.7±1.07
4		-	87.3±1.39	78.0±0.79	75.2±2.56	60.9±1.82	88.6±0.80	80.3±0.36	78.3±0.98	64.7±1.05
5	Depth	✓	88.6±1.67	81.9±0.93	78.3±0.29	65.9±0.50	91.8±1.35	83.9±0.58	81.8±1.17	69.3±1.49
6		-	93.1±0.59	84.4±0.44	80.1±1.42	66.5±1.14	93.0±0.57	84.2±0.55	80.5±1.79	66.9±0.97

a fixed amount of computation – *i.e.* comparisons at a different sample budget but the same computation budget.

Specifically, we contend that when that when training in simulation compute efficiency should be an equally important objective. For instance, an agent architecture or training regime that *increases* the number of samples required 2-fold but *decreases* the compute required 6-fold would be desirable when training in simulation. On the other hand, an architecture or training regime that *reduces* the number of samples required 2-fold but *increases* the compute required 6-fold would not be desirable.

As a concrete example, consider auxiliary tasks. These methods are known to improve sample efficiency but do so at the cost of increased computation. If an auxiliary task either requires a large decoder (such as when predicting in pixel space), it may increase the computational cost more than it improves sample efficiency, leading to overall slower training time. While such a trade-off is worthwhile when training in reality, it isn't when training in simulation.

Under these two objectives, we conduct an extensive set of experiments, cumulatively totalling over 50,000 GPU hours. These design choices lead to considerable and consistent improvements. Under a fixed sample budget, performance for RGB-D agents improves over the Habitat baselines by 3 SPL on Gibson (8% relative improvement) and *20* SPL on Matterport3D (43% relative improvement). Under a fixed compute budget (1 GPU-day), performance for RGB-D agents improves by 3 SPL on Gibson (5% relative improvement) and *15* SPL on Matterport3D (50% relative improvement).

2 RESULTS ON A SAMPLE BUDGET

In this section we discuss our results when considering *sample* efficiency. Tab. 1 shows our results. The rows show the 6 different agent settings studied – {RGB, RGB-D, Depth }×{normalized advantage, unnormalized advantage}. The columns show the 4 different settings each agent is trained under – {Gibson, Matterport3D} ×{SimpleCNN, ResNet18}. We focus our analysis on the setting with normalized advantage initially (as this is the standard practice) and then focus on results without normalized advantage. We refer to changes in performance as {+,-}X/{+,-}Y SPL to indicate an {increase, decrease} of X SPL on Gibson and a {increase, decrease} of Y SPL on Matterport3D.

ResNet18 improves performance. The largest visible difference between Savva *et al.* [21] and Wijmans *et al.* [28] is the choice of visual encoder. Savva *et al.* [21] use a simple 3-layer CNN that has its origins in Atari experiments [16] and contains none of the features of modern CNNs – *e.g.* no skip-connections [12] nor normalization layers [13]. Due to the visual complexity of Gibson and Matterport, a better CNN improves performance considerably – +57/+40 SPL for RGB (row 1)[1], +3/+21 SPL for RGB-D (row 3), and +2/+4 SPL for Depth (row 5). As we transition from RGB to RGB-D to Depth, the improvements due to ResNet18 decrease, indicating that Depth is already a highly conducive for PointNav with GPS+Compass.

The gap between RGB-D and Depth closes. One intriguing trend of Savva *et al.* [21] is the difference in performance between the RGB-D and Depth agents, particularly on Matterport3D (gap of 20 SPL, row 3 vs. 5). The RGB-D agent could clearly do better if it simply ignored RGB but fails to do so. We find that given a better visual encoder, this gap closes considerably, to 3 SPL (row 6 vs. 10, right). Specifically, the RGB-D performance on Matterport3D with ResNet18 is 66 SPL, while the performance with Depth is 69 SPL.

Normalized advantage harms performance (for SimpleCNN). We find that normalized advantage harms performance for SimpleCNN in almost all cases and *never* improves performance – +54/+38 SPL for RGB (row 2 vs. 1), +0/+15 SPL for RGB-D (4 vs. 3), and +2/+0 SPL for Depth (6 vs. 5). For ResNet18, normalized advantage neither harms nor improves the performance of the best checkpoint by a statically significant margin. For both, normalized advantage introduces instability – *i.e.* agent performance will spuriously collapse before recovering shortly. Despite its prevalence, we find clear evidence that this method is harmful.

In the extended version – available in the auxiliary material or arxiv.org/abs/2012.06117 – we discuss our results on a compute budget, examine generality of our findings, and provide a conjecture for why normalizing advantage harms performance.[2]

[1] We comment on the divergence with SimpleCNN in the supplement.

[2] EW is supported in part by an ARCS fellowship. The Georgia Tech effort was supported in part by NSF, AFRL, DARPA, ONR YIPs, ARO PECASE. 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. 2018. On Evaluation of Embodied Navigation Agents. *arXiv preprint arXiv:1807.06757* (2018).

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[3] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. 2020. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171* (2020).

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. *International Conference on 3D Vision (3DV)* (2017). MatterPort3D dataset license available at: http://kaldir.vc.in.tum.de/matterport/MP_TOS.pdf.

[5] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. 2020. Learning to explore using active neural slam. *Proceedings of the International Conference on Learning Representations (ICLR)* (2020).

[6] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. 2020. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems (NIPS)* 33 (2020).

[7] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. 2020. Neural topological slam for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12875–12884.

[8] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied Question Answering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[9] Samyak Datta, Oleksandr Maksymets, Judy Hoffman, Stefan Lee, Dhruv Batra, and Devi Parikh. 2020. Integrating Egocentric Localization for More Realistic Point-Goal Navigation Agents. *Proceedings of the Conference on Robot Learning (CoRL)* (2020).

[10] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. 2020. RoboTHOR: An Open Simulation-to-Real Embodied AI Platform. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3164–3174.

[11] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. 2018. Iqa: Visual question answering in interactive environments. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4089–4098.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[13] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[14] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. 2019. Are We Making Real Progress in Simulated Environments? Measuring the Sim2Real Gap in Embodied Visual Navigation. *IEEE Robotics and Automation Letters (RA-L)* (2019).

[15] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv* (2017).

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[17] Medhini Narasimhan, Erik Wijmans, Xinlei Chen, Trevor Darrell, Dhruv Batra, Devi Parikh, and Amanpreet Singh. 2020. Seeing the Un-Scene: Learning Amodal Semantic Maps for Room Navigation. *arXiv preprint arXiv:2007.09841* (2020).

[18] OpenAI. 2018. OpenAI Five. https://blog.openai.com/openai-five/.

[19] Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. 2020. Occupancy Anticipation for Efficient Exploration and Navigation. In *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 400–418.

[20] Manolis Savva, Angel X. Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. 2017. MINOS: Multimodal Indoor Simulator for Navigation in Complex Environments. *arXiv:1712.03931* (2017).

[21] 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. 2019. Habitat: A Platform for Embodied AI Research. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.

[22] Alexander Sax, Jeffrey O Zhang, Bradley Emi, Amir Zamir, Silvio Savarese, Leonidas Guibas, and Jitendra Malik. 2019. Learning to navigate using mid-level visual priors. *Proceedings of the Conference on Robot Learning (CoRL)* (2019).

[23] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1912.01734

[24] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354.

[25] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. 2019. The Replica Dataset: A Digital Replica of Indoor Spaces. *arXiv preprint arXiv:1906.05797* (2019).

[26] Yuandong Tian, Jerry Ma, Qucheng Gong, Shubho Sengupta, Zhuoyuan Chen, James Pinkerton, and C Lawrence Zitnick. 2019. ELF OpenGo: An Analysis and Open Reimplementation of AlphaZero. *arXiv preprint arXiv:1902.04522* (2019).

[27] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. 2019. Embodied question answering in photorealistic environments with point cloud perception. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6659–6668.

[28] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. 2020. DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[29] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. 2020. Interactive Gibson Benchmark: A Benchmark for Interactive Navigation in Cluttered Environments. *IEEE Robotics and Automation Letters* 5, 2 (2020), 713–720.

[30] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. 2018. Gibson env: Real-world perception for embodied agents. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Gibson dataset license agreement available at https://storage.googleapis.com/gibson_material/Agreement%20GDS%2006-04-18.pdf.

[31] Joel Ye, Dhruv Batra, Erik Wijmans, and Abhishek Das. 2020. Auxiliary Tasks Speed Up Learning PointGoal Navigation. *Proceedings of the Conference on Robot Learning (CoRL)* (2020).