Curriculum Learning via Task Selection for Embodied Navigation

Anonymous CVPR submission

Paper ID

1. Introduction

Deep Reinforcement Learning (RL) has shown exciting progress on many tasks in the embodied AI community including object navigation [1, 2], rearrangement [3, 4], language-guided navigation [5, 6] and instruction following [7], and question answering [8–12]. However, deep RL approaches are sample inefficient and require carefully engineered dense rewards so that the learned policy exhibits the desired behavior [13]. Engineering the 'right' reward is difficult, frequently requires multiple computationally expensive iterations, and is not generalizable across tasks. A natural alternative is to use sparse rewards, *i.e.* the agent will only be rewarded when it succeeds at the task. Such rewards require little engineering and are applicable to all tasks that have a well-defined goal state. Training agents using sparse rewards with deep RL is, however, empirically difficult: the lack of training signal during early stages of training means learning struggles to get off the ground.

Prior works [14–18] have addressed this challenge by using Curriculum Learning (CL) [19], a learning paradigm where the difficulty of the training tasks grows progressively as the agent improves. This paradigm has proven effective in guiding the learning process, allowing agents to learn more effectively and efficiently. Recent works have proposed Automatic Curriculum Learning (ACL) [15, 20–22] as a promising approach to alleviate the burden of designing a curriculum of tasks manually. In ACL, a curriculum generator learns to calibrate the complexity and sequence of the generated tasks tailored to the agent's (curriculum student) current capabilities.

In this work, we study the use of ACL for training long-horizon embodied AI tasks. We focus on ACL methods which generate their curriculum via *task selection*, *i.e.* methods which select training tasks for the agent from a predefined dataset of existing tasks of varying complexity [20–22]. These methods have the advantage that they can be easily integrated into existing training pipelines: rather than sampling uniformly from an existing training dataset we bias this sampling to improve learning efficiency. We present a simple approach, ONACL, which samples the next training task so that the predicted probability of the agent's success

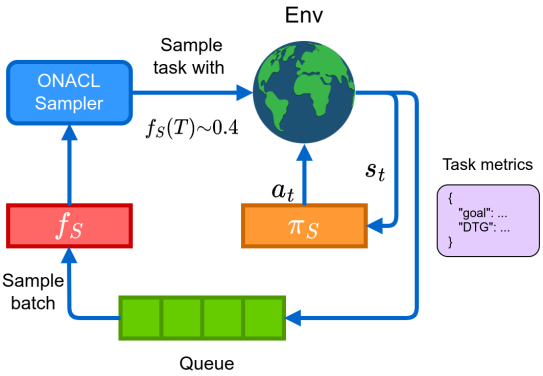

Figure 1. ONACL training framework

on task is near some threshold value. See Fig. 1 for an overview of ONACL. As training progresses this naturally results in the agent being presented increasingly difficult tasks. In particular, suppose that the agent has underwent $S > 0$ total steps of training in ONACL. We train a linear model f_S, using agent trajectories collected up to this point in training, to estimate the task success probability of the agent's current policy π_S. Using f_S we then sample the tasks of intermediate difficulty for further training as these tasks provide more learning signal than random tasks for policy improvement [23]. The agent is trained on these tasks using standard DD-PPO RL approach [13] and the above process is repeated.

We present an empirical study of ACL on the ObjectGoal Navigation (ObjectNav) [24] task in the ProcTHOR [25] and HM3D [26] home environments. We observe that policies trained without ACL using deep RL with sparse rewards fail to get off the ground during training. We find with a simple curriculum learning approach like ONACL, the agent achieves a significant improvement in performance and sample efficiency. Surprisingly, however, we find that the commonly held belief that sparse reward training in HM3D obtains near 0% success is largely incorrect: if we simply add a sufficiently large number of 'easy' episodes during policy training then (evaluation set) performance dramatically improves. We hypothesize this happens due to the emergence of an implicit curriculum during training and present an analysis supporting the claim. This suggests that,

in some cases, curriculum learning approaches may simply be correcting for needlessly difficult training datasets.

2. Experimental Setup

ObjectGoal Navigation (ObjectNav). In ObjectNav [24], an agent receives RGB+D observations and is tasked with navigating to any object instance of a goal category (*e.g.* *'Find a chair'*) from a randomly sampled start position. The goal is specified using a unique category ID. The navigation agent uses six discrete actions: MOVE_FORWARD, TURN_LEFT, TURN_RIGHT, LOOK_UP, LOOK_DOWN, STOP. An agent is successful if it calls STOP action within 1m of any instance of goal category.

Training. We use EmbCLIP [27] agent architecture. We train agents using RL with sparse rewards using DD-PPO [13] for 150 million frames of experience using AI2-THOR [28] and Habitat [29] simulator. We use ProcTHOR-10k [25] scenes in AI2-THOR and HM3D scenes [26] in Habitat for our experiments.

3. Experimental Findings

Sampling Method	Eval Dataset			
	ProcTHOR-10k-Hard		ProcTHOR-10k	
	SR ↑	SPL ↑	SR ↑	SPL ↑
Uniform	0.16	0.11	0.57	0.43
ONACL	0.56	0.33	0.56	0.41

Table 1. ObjectNav evaluation performance of policies trained on ProcTHOR-10k-Hard and ProcTHOR-10k.

Sampling Method	Eval Dataset			
	HM3D		HM3D-Easy	
	SR ↑	SPL ↑	SR ↑	SPL ↑
Uniform	0.0	0.0	0.34	0.18
ONACL	0.02	0.02	0.46	0.22

Table 2. ObjectNav evaluation performance of policies trained on HM3D and HM3D-Easy.

Effectiveness of ONACL. We compare the performance of agents trained using RL with sparse rewards with uniform episode sampling to ONACL sampling. We present results on ProcTHOR-10k-Hard, a dataset with large houses *i.e.* 4-10 room houses, and ProcTHOR-10k [25], a dataset with 1-10 room houses, in Tab. 1. We find agents trained with ONACL perform significantly better on the ProcTHOR-10k-hard dataset and generalize better to unseen scenes. The uniform sampling baseline (Tab. 1, row 1) obtains a 40% drop in success rate and 22% drop in SPL on the ProcTHOR-10k-hard dataset compared to ONACL (Tab. 1, row 2). This demonstrates that policies trained with a simple curriculum that samples tasks on the frontier of predicted success rates outperforms standard uniform sampling. Surprisingly, we find that training with ONACL doesn't help improve performance on the ProcTHOR-10k [25] dataset which has 1-10 room houses. We hypothesize ONACL doesn't help with ProcTHOR-10k [25] due to an implicit curriculum that emerges from having a large amount of easier training tasks that come from having 1-3 room houses in ProcTHOR-10k dataset.

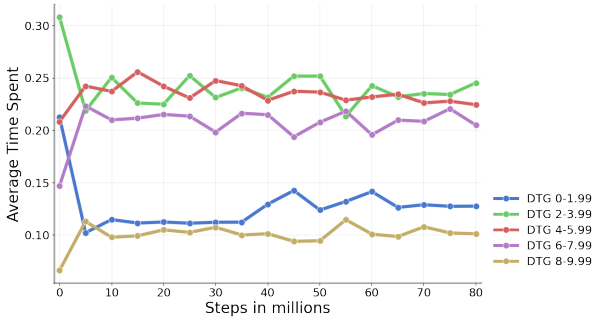

Figure 2. Average time spent for episodes of varying difficulty during training for policies trained on ProcTHOR-10k [25] with random sampling.

Suprising Effectiveness of Easy Tasks. Next, we compare uniform sampling *vs*. ONACL on HM3D [26, 30] in Tab. 2. We find that agents trained with uniform sampling using sparse rewards with RL do not get off the ground and achieve 0% success on HM3D-Val split. However, even with ONACL, policy performance improves only to 2.5% success on HM3D-Val. We hypothesize that ONACL fails is largely due to the lack of 'easy' training episodes where a policy can succeed by taking only a few actions to reach the target object *i.e.* distance to target is less than 1m. Without such episodes, our ONACL cannot efficient learn to predict which tasks to give the agent. To test this hypothesis, we generate HM3D-Easy, a new dataset for training ObjectNav agents in HM3D scenes which includes 'easy' episodes. We observe, simply adding easier training episodes enables learning with sparse rewards for ObjectNav. We hypothesize this is due to the fact that early on during training the policy takes random actions which have a low probability of succeeding at a long horizon task like ObjectNav but by simply adding 'easy' episodes the probability of succeeding increases which leads to enough learning signal for training with sparse rewards.

Easier Training Tasks Leads to a Implicit Curriculum. Fig. 2 shows the average number of steps spent in episodes of varying difficulty during training within ProcTHOR-10k scenes when uniformly sampling tasks. Note that an implicit curriculum emerges: the agent quickly spends far less time in easy episodes (as it completes them quickly) and instead spends the vast majority of its training time within more challenging episodes. For instance, the easy DTG 0-1.99 episodes account for >20% of training episodes but the agent, by the end of training, spends only 13% of its training time in such episodes; similarly the hard DTG 8-9.99 episodes account for only 2% of all episodes but, by training's end, almost 10% of the agent's training steps.

References

[1] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, "MINOS: Multimodal indoor simulator for navigation in complex environments," *arXiv preprint arXiv:1712.03931*, 2017. 1

[2] P. Anderson, A. X. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018. 1

[3] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, M. Savva, and H. Su, "Rearrangement: A Challenge for Embodied AI," *arXiv preprint arXiv:2011.01975*, 2020. 1

[4] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi, "Visual room rearrangement," in *CVPR*, 2021. 1

[5] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *CVPR*, 2018. 1

[6] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," in *ECCV*, 2020. 1

[7] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks," in *CVPR*, 2020. 1

[8] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied Question Answering," in *CVPR*, 2018. 1

[9] E. Wijmans, S. Datta, O. Maksymets, A. Das, G. Gkioxari, S. Lee, I. Essa, D. Parikh, and D. Batra, "Embodied Question Answering in Photorealistic Environments with Point Cloud Perception," in *CVPR*, 2019. 1

[10] A. Das, F. Carnevale, H. Merzic, L. Rimell, R. Schneider, J. Abramson, A. Hung, A. Ahuja, S. Clark, G. Wayne, and F. Hill, "Probing emergent semantics in predictive agents via question answering," in *ICML*, 2020. 1

[11] A. Das, *Building agents that can see, talk, and act*. PhD thesis, Georgia Institute of Technology, 2020. 1

[12] L. Yu, X. Chen, G. Gkioxari, M. Bansal, T. L. Berg, and D. Batra, "Multi-target embodied question answering," in *CVPR*, 2019. 1

[13] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames," in *ICLR*, 2020. 1, 2

[14] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," in *Advances in Neural Information Processing Systems*, 2017. 1

[15] I. Gur, N. Jaques, K. Malta, M. Tiwari, H. Lee, and A. Faust, "Adversarial environment generation for learning to navigate the web," in *NeurIPS*, 2021. 1

[16] J. Chen, Y. Zhang, Y. Xu, H. Ma, H. Yang, J. Song, Y. Wang, and Y. Wu, "Variational automatic curriculum learning for sparse-reward cooperative multi-agent problems," in *NeurIPS*, 2021. 1

[17] M. Fang, T. Zhou, Y. Du, L. Han, and Z. Zhang, "Curriculum-guided hindsight experience replay," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), 2019. 1

[18] S. Sukhbaatar, Z. Lin, I. Kostrikov, G. Synnaeve, A. Szlam, and R. Fergus, "Intrinsic motivation and automatic curricula via asymmetric self-play," in *ICLR*, 2018. 1

[19] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*. 1

[20] S. Narvekar, J. Sinapov, and P. Stone, "Autonomous task sequencing for customized curriculum design in reinforcement learning," in *IJCAI*, 2017. 1

[21] D. Feng, C. P. Gomes, and B. Selman, "A novel automated curriculum strategy to solve hard sokoban planning instances," *NeurIPS*, 2021. 1

[22] Y. Zhang, P. Abbeel, and L. Pinto, "Automatic curriculum learning through value disagreement," 2020. 1

[23] R. Portelas, C. Colas, L. Weng, K. Hofmann, and P. Oudeyer, "Automatic curriculum learning for deep RL: A short survey," 2020. 1

[24] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "ObjectNav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020. 1, 2

[25] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, J. Salvador, K. Ehsani, W. Han, E. Kolve, A. Farhadi, A. Kembhavi, and R. Mottaghi, "Procthor: Large-scale embodied ai using procedural generation," in *NeurIPS*, 2022. 1, 2

[26] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet, J. Turner, A. Gokaslan, N. Maestre, A. X. Chang, D. Batra, M. Savva, *et al.*, "Habitat-matterport 3d semantics dataset," *arXiv preprint arXiv:2210.05633*, 2022. 1, 2

[27] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but Effective: CLIP Embeddings for Embodied AI," in *CVPR*, 2022. 2

[28] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017. 2

[29] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, *et al.*, "Habitat: A platform for embodied AI research," in *ICCV*, 2019. 2

[30] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, "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