Selective Visual Representations Improve Convergence and Generalization for Embodied AI

Ainaz Eftekhar[*1,2] **Kuo-Hao Zeng**[*2] **Jiafei Duan**[1,2]
Ali Farhadi[1,2] **Ani Kembhavi**[1,2] **Ranjay Krishna**[1,2]

[1]University of Washington, [2]Allen Institute for Artificial Intelligence

Abstract

Embodied AI models often employ off the shelf vision backbones like CLIP to encode their visual observations. Although such general purpose representations encode rich syntactic and semantic information about the scene, much of this information is often irrelevant to the task at hand. This introduces noise within the learning process and distracts the agent's focus from task-relevant visual cues. Inspired by selective attention in humans—the process through which people filter their perception based on their task at hand—we introduce a parameter-efficient approach to filter visual stimuli for embodied AI. Our approach induces a task-conditioned bottleneck using a small learnable codebook module. This codebook is trained jointly to optimize task reward and acts as a task-conditioned selective filter over the visual observation. Our experiments showcase state-of-the-art performance for object goal navigation and object displacement across 5 *benchmarks, ProcTHOR, ArchitecTHOR, RoboTHOR, AI2-iTHOR, and ManipulaTHOR. The filtered representations produced by the codebook also generalize better and converge faster when adapted to other simulation environments such as Habitat. Our qualitative analyses show that agents explore their environments more effectively and their representations retain task-relevant information like target object recognition while ignoring superfluous information about other objects. (project page)*

1. Introduction

Human visual perception is not a passive reception of all available stimuli; it selectively tunes itself to process visual cues relevant to the task at hand [4, 5, 8]. For instance, when searching for our misplaced keys, we tend to overlook many visual details in the scene and concentrate only on surfaces where we usually place our keys.

Embodied-AI agents are tasked with similar goal-directed behaviors such as navigation [3, 14], instruction following [1, 13, 17], manipulation [9, 19], and rearrangement [2, 18]. Conventional frameworks use general-purpose visual backbones [11, 20] to extract visual representations and fuse it with goal embeddings to construct a goal-conditioned representation $E \in \mathcal{R}^D$, where D is often as large as a 1574. However, this general-purpose representation often contains task-irrelevant information, distracting the agent from more pertinent visual cues and introducing unnecessary noise into the learning process.

In this paper, we leverage insights from cognitive psychology to create task-specific representations for embodied AI agents, filtering out irrelevant sensory input and only preserving essential stimuli. We introduce a parameter-efficient **codebook module** into our agent's architecture, which includes 256 learnable latent codes of 10 dimensions each. This module takes visual embedding $E \in \mathcal{R}^{1574}$ as input and selects from these codes through an attention mechanism to form the bottlenecked representation $\hat{E}$, a weighted combination of the selected codes (Figure 1). This approach effectively narrows down the visual information to the most task-relevant cues by using a bottleneck of limited low-dimensional latent codes.

2. Method

Background. Embodied-AI frameworks typically include: a *visual encoder* that turns inputs like RGB into a visual representation v; a *goal encoder* that converts task objectives, such as GPS locations or instructions, into a goal embedding g; and a *previous-action encoder* that captures the most recent action in an embedding α. These are fused into a *task-conditioned representation* E. The architecture also includes a *recurrent state encoder* that aggregates all past representations into a single state and an *actor-critic head* that predicts next action based on the current state.

The need for task-bottlenecked visual representations. EmbCLIP [11] is today's state-of-the-art model for the tasks that we consider which encodes the three representations in $E \in \mathcal{R}^D$ ($D = 1574$). This embedding contains CLIP features, which were trained for general-purpose vision tasks. Therefore, when presented with the input image, this representation identifies a large number of object categories, their attributes, etc. These additional pieces of information are also sent over to the policy, which is often designed as

[*]Equal contribution

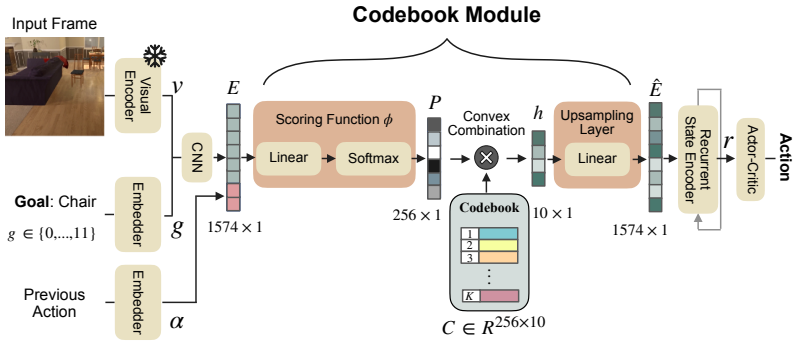

Figure 1. **An overview of EmbCLIP-Codebook.** The 3 representations corresponding to the input frame, the goal, and the previous action get concatenated to form $E \in \mathcal{R}^{1574}$. The codebook module takes E and generates a probability simplex $\mathcal{P} \in \mathcal{R}^{256}$ over the latent codes. The hidden compact representation $h \in \mathcal{R}^{10}$ is a convex combination of the codes weighted by $\mathcal{P}$. The final task-bottlenecked codebook representation $\hat{E}$ is derived by upsampling h which is subsequently passed to the recurrent state encoder and the policy to produce an action.

an RNN followed by a small actor-critic module. These few parameters must serve two purposes: 1) identify what information is useful for the task at hand and 2) what action to take given that information.

The codebook module. We introduce a module that decouples the two objectives. The input encoders and the codebook focus on extracting essential information for the task from the visual input, whereas the policy can focus on taking actions conditioned on this filtered information. The codebook is a parameter-efficient module to transform the general-purpose representation E into a compact task-bottlenecked one $\hat{E}$ (Fig. 1). This module contains a set of latent vectors $C = [c_1, c_2, ..., c_K] \in \mathcal{R}^{K \times D_c}$ (K denotes codebook's size and D_c is the dimension of each latent code). To create a strong bottleneck, we set $D_c = 10$ and $K = 256$. These codes are initialized randomly via a normal distribution and optimized along with the overall training.

The module contains a scoring function $\phi(.)$ to generate a probability simplex over the K latent codes $\phi(E) = \mathcal{P} = [p_i]_{i=1}^{K}$ such that $\sum_{i=1}^{K} p_i = 1$. The scoring function ϕ is a single-layer MLP followed by a softmax function. This forces the agent to select which latent code(s) are more useful for representing the current frame. Next, the hidden compact representation h is a convex combination of the learnable codes $\{c_i\}_{i=1}^{K}$ weighted by their corresponding p_i: $h = \mathcal{P}^T C = \sum_{i=1}^{K} p_i.c_i$. Finally we upsample the hidden embedding h to the *task-bottlenecked codebook representation* $\hat{E}$. All modules are trained to optimize the task reward.

3. Experiments

All models are trained using (PPO) [16] in AllenAct framework . We follow [7] to pretrain the *EmbCLIP* baseline and *EmbCLIP-Codebook* on ProcTHOR-10k houses. **1.** We show state-of-the-art **zero-shot performances** on two Embodied-AI tasks—object goal navigation (ObjNav) [6] and object displacement (ObjDis [10])—across five benchmarks (ProcTHOR[7], ArchitecTHOR, RoboTHOR [6], AI2-iTHOR[12], and ManipulaTHOR [9])(Table 1). **2.** We demonstrate that our bottlenecked embeddings **generalize** well to new visual domains (Habitat environments [15]) with minimal finetuning (Table 2). **3.** We confirm that the codebook-bottlenecked representation captures the most **task-relevant** information and ignores distractions. **4.** We observe noticeable improvements in agent's behavior in the form of **smoother trajectories** and more **efficient exploration strategies**.

Table 1. We outperform the baselines in zero-shot evaluation on 4 Object Goal Navigation benchmarks and 1 Object Displacement benchmark.

Benchmark	Model	Object navigation SR(%)↑	EL↓	Curvature↓	SEL↑
ProcTHOR-10k (validation)	EmbCLIP	67.70	182.00	0.58	36.00
	+codebook	**73.72**	**136.00**	**0.23**	**43.69**
ArchitecTHOR (0-shot)	EmbCLIP	55.80	222.00	0.49	20.57
	+Codebook	**58.33**	**174.00**	**0.20**	**28.31**
RoboTHOR (0-shot)	EmbCLIP	51.32	-	-	-
	+Codebook	**55.00**	-	-	-
AI2-iTHOR (0-shot)	EmbCLIP	70.00	121.00	0.29	21.45
	+Codebook	**78.40**	**86.00**	**0.16**	**26.76**

Benchmark	Model	Object displacement PU(%)↑	SR(%)↑
ManipulaTHOR	m-VOLE	81.20	59.60
	+Codebook	**86.00**	**65.10**

Table 2. Models are trained on ProcTHOR and evaluated with fine-tuning on Habitat. We show much better adaptation with lightweight finetuning.

Benchmark	Fine-tuning parts	Model	Object goal navigation SR(%)↑	SPL↑	Invalid Actions(%)↓	Curvature↓
Habitat challenge 2022(HM3D Semantics)	Adaptation Module	EmbCLIP	36.45	18.18	28.10	0.53
		+Codebook	**50.25**	**25.76**	**21.50**	**0.26**

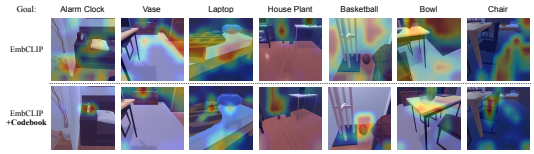

Figure 2. **GradCAM Visualization.** While EmbCLIP is distracted by different objects and other visual cues even though the target object is visible in the frame, EmbCLIP-Codebook is able to ignore such distractions and only focus on the object goal.

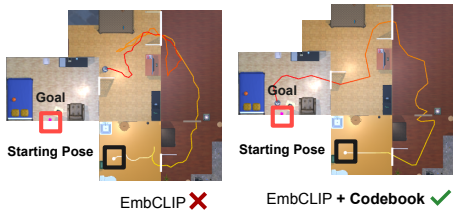

Figure 3. **Sample Trajectory.** EmbCLIP agent takes many redundant rotations, resulting in a high average curvature, whereas ours navigates more smoothly (see curvature metric in Tab. 1).

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018. 1

[2] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. *ArXiv*, 2020. 1

[3] 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. *arXiv preprint arXiv:2006.13171*, 2020. 1

[4] Bergen R Bugelski and Delia A Alampay. The role of frequency in developing perceptual sets. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 15(4):205, 1961. 1

[5] Timothy J Buschman and Earl K Miller. Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *science*, 315(5820):1860–1862, 2007. 1

[6] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *CVPR*, 2020. 2

[7] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, et al. Procthor: Large-scale embodied ai using procedural generation. *arXiv preprint arXiv:2206.06994*, 2022. 2

[8] Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18 (1):193–222, 1995. 1

[9] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In *CVPR*, 2021. 1, 2

[10] Kiana Ehsani, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. Object manipulation via visual target localization. In *ECCV*, 2022. 2

[11] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *CVPR*, pages 14829–14838, 2022. 1

[12] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017. 2

[13] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 104–120. Springer, 2020. 1

[14] Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to objects specified by images. In *ICCV*, 2023. 1

[15] 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 *ICCV*, 2019. 2

[16] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv*, 2017. 2

[17] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *CVPR*, 2020. 1

[18] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5922–5931, 2021. 1

[19] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020. 1

[20] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline visual representation learning for embodied navigation. In *ICLR Workshop on Reincarnating Reinforcement Learning*, 2023. 1