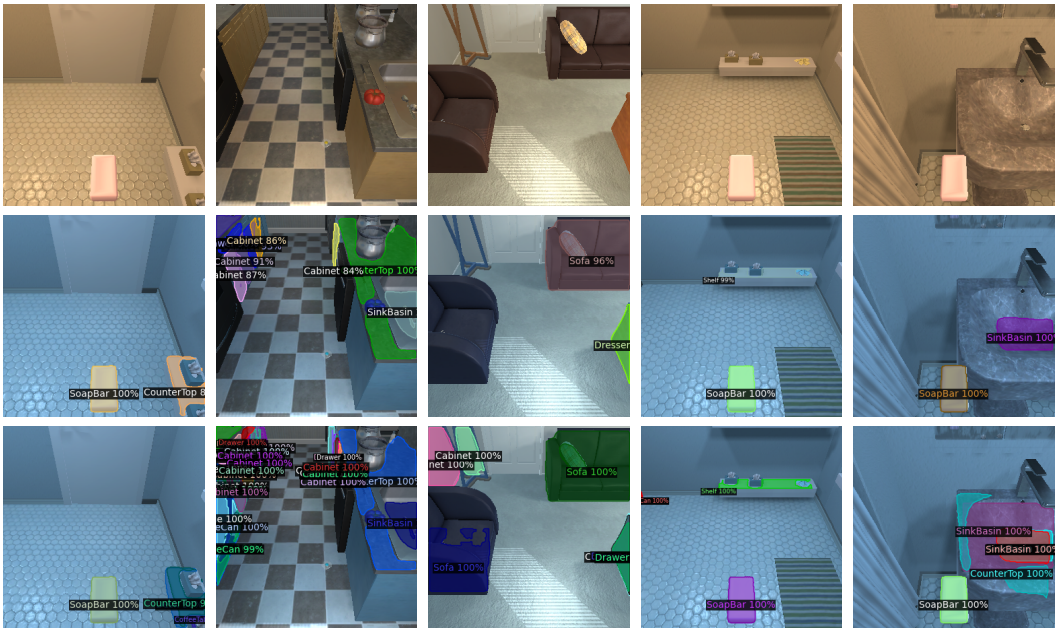

Figure 1: Concept learning visualization. From top to bottom: the original image, supervised instance segmentation map, and our concept learning results.

Appendix for "Embodied Concept Learner: Self-supervised Learning of Concepts and Mapping through Instruction Following"

Mingyu Ding
HKU

Yan Xu
CUHK

Zhenfang Chen
MIT-IBM Watson AI Lab

David Cox
MIT-IBM Watson AI Lab

Ping Luo
HKU

Joshua B. Tenenbaum
MIT BCS, CBMM, CSAIL

Chuang Gan
UMass Amherst
MIT-IBM Watson AI Lab

A ALFRED Dataset

We evaluate our method and its counterparts on ALFRED [1], which is a benchmark for connecting human language to actions, behaviors, and objects in interactive visual environments. Planner-based expert demonstrations are accompanied by both high- and low-level human language instructions in 120 indoor scenes in AI2-THOR 2.0 [2]. ALFRED [1] includes 25,743 English language directives describing 8,055 expert demonstrations averaging 50 steps each, resulting in 428,322 image-action pairs. The test set contains "Test seen" (1,533 episodes) and "Test unseen" (1,529 episodes); the scenes of the latter entirely consist of rooms that do not appear in the training set, while those of the former only consist of scenes seen during training. Similarly, the validation set contains "Valid seen" (820 episodes) and "Valid Unseen" (821 episodes). The success rate is the ranking metric used in the official leaderboard.

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

Table 1: Comparison of the semantic policies on the ALFRED benchmark. **Red** denotes the top success rate (SR) (ranking metric of the leaderboard) on the `test_unseen` set. We take our ECL w. depth as the baseline model and make comparison between our model and our model + learned semantic policy [3].

Method	Supervision			Test Seen				Test Unseen			
	Semantic	Depth	Policy	PLWGC (%)	GC (%)	PLWSR (%)	SR (%)	PLWGC (%)	GC (%)	PLWSR (%)	**SR (%)**
ECL	×	√	probability map	12.34	27.86	8.02	18.26	11.11	27.30	7.30	17.24
ECL + POLICY	×	√	learned map	12.74	27.98	8.67	18.79	11.52	27.75	7.45	**17.92**

Table 2: Performance by different task types of model ECL w. Depth on the validation set.

Task Type	Val Seen %		Val Unseen %	
	Goal-condition	Success Rate	Goal-condition	Success Rate
Overall	30.83	18.67	21.74	10.50
Examine	46.81	31.18	47.98	29.65
Pick & Place	21.36	23.72	3.67	8.49
Stack & Place	16.38	6.25	7.80	0.99
Clean & Place	41.44	24.77	29.50	8.85
Cool & Place	19.64	5.88	13.15	0.00
Heat & Place	35.75	19.27	31.00	13.67
Pick 2 & Place	34.48	19.67	19.14	11.84

B Evaluation of the Semantic Policy

In this section, we evaluate the semantic policy used in our model (average semantic probability map from demonstrations), in comparison to the learned semantic policy in FILM [3]. [3] learns a semantic policy model using additional map supervision. However, the policy in our work is freely available from the grounded average semantic probability map. From Tab. 1, we can see that: our average semantic probability map achieves good performance. With the learned semantic policy, the success rate improves slightly from 17.24% to 17.92%. To keep our framework clean with reduced supervision, we train our model without learning a semantic policy [3].

C Per-task Performance

We provide per-task performance (success rate and goal-condition success rate) in Tab. 2 to show ECL's strengths and weaknesses in different types of tasks. We have the following observations: 1) "Stack & Place" and "Cool & and Place" are the most challenging tasks, with a low success rate. 2) The "Examine" task is the easiest task, with a success rate over 30% and 46.81% goal-condition success rate. 3) A similar observation with FILM [3] regarding the number of subtasks and success rate is found: whereas "Heat & Place" and "Clean & Place" usually involve three more subtasks than "Pick & Place", the metrics of the former are higher than the latter. This is because "Heat & Place" only appears in kitchens, and "Clean & Place" only appears in toilets. And the room area of these two scenes is relatively small. The results show that the success of a task is highly dependent on the type and scale of the scene.

D Detailed Analysis of Concept Learning

In addition to the figure shown in our main paper, we also report the per-object concept grounding evaluation results (small) in Tab. 3. Objects "HandTowel", "KeyChain", "Bowl", and "Television" have over 80% concept learning accuracy because these objects often appear alone in the scene

Table 3: Concept grounding accuracy (small).

Category	Vase	Pillow	Plate	Laptop	FloorLamp	Newspaper	HandTowel	Box
Accuracy (%)	61.5	66.7	51.4	70.0	68.1	57.5	85.4	74.0
Category	Towel	Television	Mug	Book	Bowl	Tomato	Knife	KeyChain
Accuracy (%)	67.4	81.1	46.7	53.1	81.5	60.0	65.9	83.2
Category	Cloth	TeddyBear	CellPhone	BasketBall	Glassbottle	Apple	CD	Others
Accuracy (%)	21.4	18.1	13.2	0	1.3	50.2	38.6	57.0

Table 4: Concept grounding accuracy (large).

Category	Shelf	TVStand	Dresser	Fridge	Microwave	SinkBasin	BathtubBasin
Accuracy (%)	77.9	82.6	75.2	13.6	64.8	99.6	0
Category	CoffeeMachine	Cart	Cabinet	Desk	CoffeeTable	Safe	Drawer
Accuracy (%)	81.0	59.5	2.6	0	74.3	73.4	38.8
Category	Bed	Sofa	DiningTable	GarbageCan	Toilet	CounterTop	Others
Accuracy (%)	64.4	72.7	52.9	53.3	81.5	87.2	59.4

(easy to learn and less likely to be confused). Objects like "HandTowel", "KeyChain", "Bowl", and "Television" rarely appear in the environment, so their concepts are difficult to learn.

Likewise, we perform detailed evaluations and report the per-object concept grounding evaluation results (large) in Tab. 4. We notice two classes with ground accuracy of 0: "BathtubBasin" and "Desk". This is because all BathtubBasins are identified as SinkBasins by our grounding model (SinkBasins are shown more frequently than BathtubBasins). As for "Desk", the object proposals are identified as "CoffeTable", "DiningTable", etc. The average grounding accuracy for large objects is higher than for small objects, because large objects often appear alone in the scene (easy to learn and less confusing).

E Details of the deterministic policy

The deterministic policy is based on the Fast Marching Method [56]. If the object needed in the current subtask is observed in the current semantic map, the location of the object is selected as the goal; otherwise, we sample the location based on the distribution of the corresponding object class in our averaged semantic map as the goal. In both cases, we did not use any domain knowledge about ALFRED. We find the goal from the concept learner and plan the shortest path to the goal based on our semantic map. It's a very general solution that can be used in many other tasks or environments rather than a hand-coded policy for ALFRED.

F More Visualization Results

In this section, we show more visualizations of our concept learning results in Figure 1. We also demonstrate two examples of intermediate estimates by ECL when an agent tries to accomplish an instruction in Figure 2.

G Key Assumption for Other Simulators or Real Robots

Real robotics applications have been one of the longstanding motivations for this work and have been carefully considered by the authors in the design of ALFRED. When generalizing our model to real-world scenarios:

- The instruction parser is supervisedly trained, and can be directly employed in the real world.

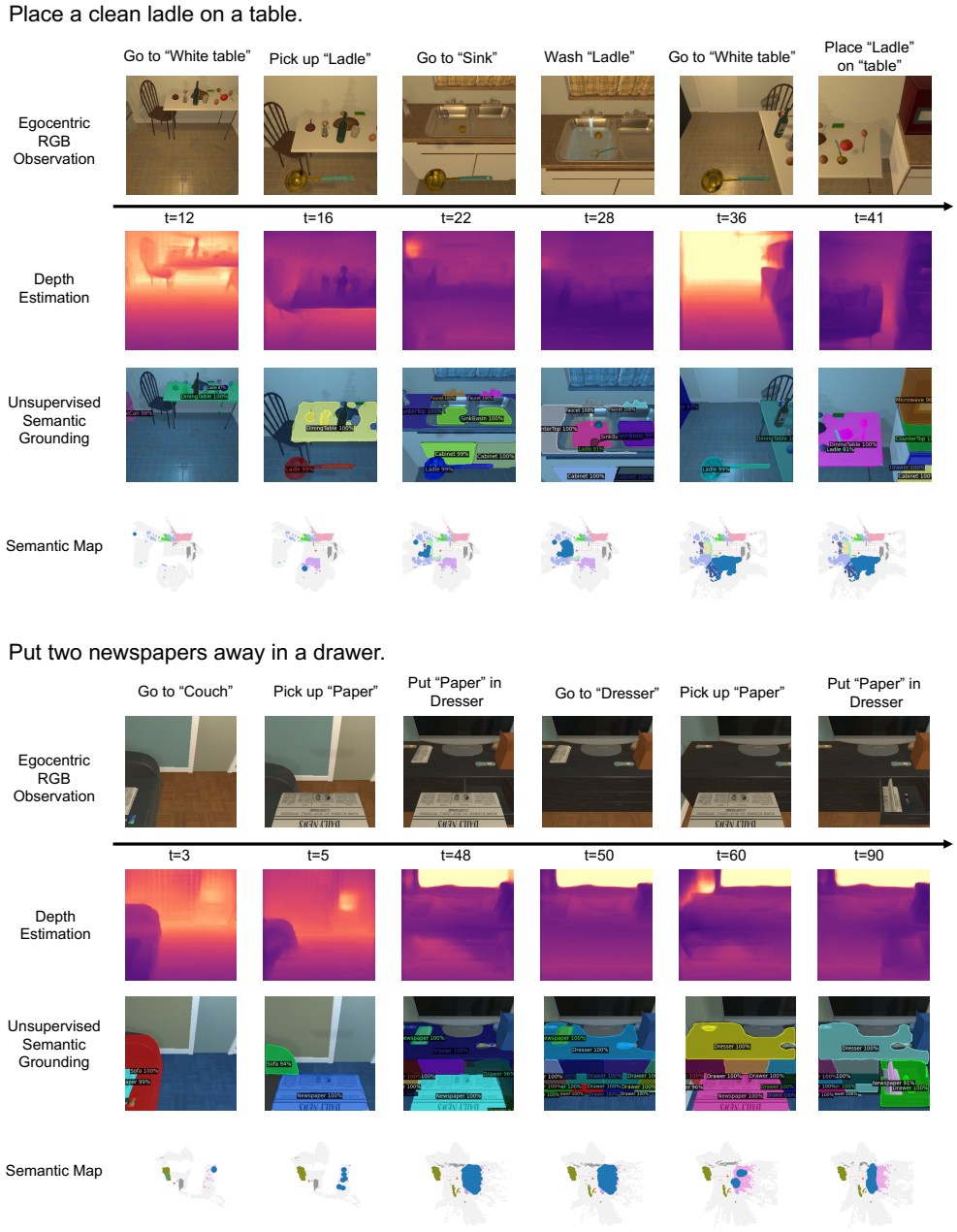

Figure 2: Two examples of intermediate estimates for ECL when the agent tries to accomplish the instructions. Based on RGB observations, our system estimates the depths and semantic masks. The BEV semantic map is gradually established with these estimates as exploration continues. The goals (sub-goal/final-goal) are represented by large blue dots in the semantic map, while the agent trajectories are plotted as small red dots.

- The embodied concept learner should work when there are real-world demonstrations. Currently, we have some assumptions based on the artifact of the AI2THOR environment. However, the assumptions are not strong and are still applied to real environments.

- Unsupervised depth and mapping are well-studied problems in the real world. We see this as a reasonable assumption for the time being.

- Still, there are some limitations in ALFRED that the action execution is not feasible, i.e, picking up an object by only one command without robot manipulation. However, the low-level control task and the current embodied instruction following task are orthogonal, which means the two tasks can still be decoupled in real-world scenarios, while our model focus on instruction following.

However, it's really challenging for a robot to perform instruction following in an unseen real-world environment, even in a simulated environment (test unseen success rate of only 23.6% even in our oracle model). To this end, ALFRED simplifies the hard problem of making meaningful progress through tight integration between visual perception, language instruction, and robotic navigation and manipulation. To the best of our knowledge, no other benchmarks contain language instructions in an interactive 3D environment with visual observation and navigation. As the field progresses, we are confident more works and benchmarks will be introduced, and we will take it as our future research direction.

H Related Work about Depth and Mapping

Depth estimation [4, 5, 6, 7, 8, 9, 10] has witnessed a boom since the emergence of deep learning. Compared with stereo matching [10, 11, 12] and sensor-based methods [13, 14], the monocular depth estimation only requires a single-view color image for depth inference, which is suitable for practical deployment given its low-cost nature. Following the supervised methods [15, 16], Zhou et al. [8] first demonstrated the possibility of depth learning in an unsupervised manner, inspired by the learning principle of humans. Afterwards, the unsupervised depth estimation are well explored in both indoor [17, 18, 19] and outdoor scenarios [9, 20, 4] due to its labeling-free advantage. In this work, we also follow their spirits to investigate the learning process of an agent baby. After depth estimation, a mapping module [21, 22, 23] is usually included in a robotic system to memorize the geometry layouts of the visited regions for path planning and navigation. Given different sensor properties and map representations, the mapping procedure could also differ. For instance, [24, 21] maintain reliable sparse landmarks, [22] constructs TSDF, and [25, 23] store voxel maps.

References

[1] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.

[2] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. Ai2-thor: An interactive 3d environment for visual ai, 2019.

[3] S. Y. Min, D. S. Chaplot, P. Ravikumar, Y. Bisk, and R. Salakhutdinov. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*, 2021.

[4] T.-W. Hui. Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1675–1684, 2022.

[5] X. Luo, J.-B. Huang, R. Szeliski, K. Matzen, and J. Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020.

[6] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12240–12249, 2019.

[7] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019.

[8] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.

[9] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[10] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[11] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019.

[12] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14194–14203, 2021.

[13] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30:1116–1129, 2020.

[14] Y. Xu, X. Zhu, J. Shi, G. Zhang, H. Bao, and H. Li. Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2811–2820, 2019.

[15] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016.

[16] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2017.

[17] N. Yang, L. v. Stumberg, R. Wang, and D. Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020.

[18] B. Li, Y. Huang, Z. Liu, D. Zou, and W. Yu. Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12663–12673, 2021.

[19] P. Ji, R. Li, B. Bhanu, and Y. Xu. Monoindoor: Towards good practice of self-supervised monocular depth estimation for indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12787–12796, 2021.

[20] S. Pillai, R. Ambruş, and A. Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9250–9256. IEEE, 2019.

[21] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[22] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[23] T. Shan and B. Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018.

[24] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007.

[25] J. Zhang and S. Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA, 2014.