MOPA: Modular Object Navigation with PointGoal Agents Supplemental Materials

Sonia Raychaudhuri[1], Tommaso Campari[2,3], Unnat Jain[4], Manolis Savva[1], Angel X. Chang[1,5]
[1]Simon Fraser University, [2]University of Padova, [3]FBK, [4]Meta AI, [5]Amii
https://3dlg-hcvc.github.io/mopa

A. Supplementary Material

In this supplemental document, we provide some additional statistics on the MultiON 2.0 dataset (Appendix B), details of the Object detection module in MOPA (Appendix C), and additional experiments on MultiON (Appendix D) and ObjectNav (Appendix E). For MultiON, we first study the performance of MOPA on natural objects (NAT-objects) in Appendix D.1 to understand how the increased visual complexity of the target objects (compared to CYL-objects) influences performance. Then we discuss our findings on the different Navigation (Appendix D.2) and Exploration methods (Appendix D.3), and investigate the impact of having distractor objects on the OracleSem agent in Appendix D.4. We also discuss more about generalizability on n-ON in Appendix D.5 and effect of spatial map on longer-horizon task planning in Appendix D.6. We also show visualizations of episode rollouts of OracleSem on 5ON, PredSem on CYL and NAT-objects in Appendix D.7.

B. MultiON 2.0 Dataset

Fig. 1 compares the path length of MultiON 2.0 validation set episodes against episodes from other datasets. The episodes we generate are more complex that those in the original MultiON dataset.

Fig. 2 shows that while the original MultiON dataset contains a set of cylinder (CYL) objects of same size and shape but varying colors, we additionally have a set of more natural (NAT) looking objects of varying shape, size and color in MultiON 2.0.

C. MOPA object detection

For detecting cylinders, we fine-tune a FasterRCNN [18] and use KNN classifier to identify the color of the cylinder. Specifically, in offline training, we fine-tune a FasterRCNN model pretrained on MS-COCO on a set of 2k frames collected by an oracle agent following the shortest path to the goal. We then use a k-nearest neighbors classifier to distinguish between different categories. We choose the k-NN classifier as it is has been found to be effective in prior work in vision and robotics [1, 9, 13, 15, 20]. Concretely, we sample the RGB value from the center of each bounding box and use it to find the k-closest neighbors. We pick the color label of the most frequent nearest neighbor, *i.e.* if $\alpha_{\text{KNN}}\%$ of the nearest neighbors is of the same color, we select that as the label. For our experiments, we used ($k = 10$) for the number of nearest neighbors, and we set α_{KNN} to 80% (*i.e.* if 8 of the 10 nearest neighbors is of the same category, we select that as the label).

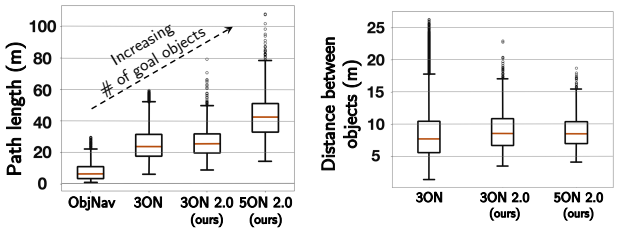

Figure 1. **Comparing path lengths across tasks.** (a) shows that 3ON2.0 has longer episodes than both Habitat ObjectNav 2021 [2] and the original 3ON [21] (~26m vs. ~23m), with 5ON2.0 having the longest average episode length. (b) shows that the average distance between the object-goal pairs is greater in 3ON2.0 than 3ON. With more object-goals, 5ON2.0 has more closely-spaced objects. These plots show that MultiON 2.0 contains harder episodes than Habitat ObjectNav 2021 and 3ON, with longer average shortest path and with object-goals placed farther apart.

D. MultiON experiments

Here we provide results for experiments on both the validation and test sets. We also compare the performance of MOPA with CYL and NAT objects.

D.1. Performance with natural objects

In Tab. 1, we present the results for CYL and NAT objects with predicted (PredSem) and oracle semantics, using agents with Uniform exploration policy and PointNav navigation.

	Object Types	MOPA Modules $\mathcal{O}$	$\mathcal{M}$	$\mathcal{E}$	$\mathcal{N}$	Validation Success	Progress	SPL	PPL	Test Success	Progress	SPL	PPL
PredSem	CYL	FRCNN	[5]	U	PN	50 (± 2)	65 (± 2)	21 (± 1)	26 (± 1)	52 (± 2)	66 (± 2)	21 (± 1)	27 (± 2)
	NAT	FRCNN	[5]	U	PN	28 (± 2)	47 (± 2)	11 (± 1)	18 (± 1)	29 (± 2)	45 (± 2)	11 (± 1)	17 (± 1)
OracleSem	CYL	GT	[5]	U	PN	80 (± 2)	87 (± 2)	35 (± 1)	38 (± 1)	81 (± 2)	87 (± 2)	37 (± 1)	39 (± 1)
	NAT	GT	[5]	U	PN	80 (± 2)	85 (± 2)	35 (± 1)	38 (± 1)	81 (± 2)	87 (± 2)	37 (± 1)	39 (± 1)
OracleMap	CYL	GT	GT	U	PN	84 (± 2)	90 (± 2)	37 (± 1)	41 (± 1)	81 (± 2)	85 (± 2)	36 (± 1)	39 (± 1)
	NAT	GT	GT	U	PN	84 (± 2)	90 (± 2)	37 (± 1)	41 (± 1)	81 (± 2)	85 (± 2)	36 (± 1)	39 (± 1)

Table 1. **MOPA performance on MultiON 2.0.** We observe that the PredSem agent, which builds a map ($\mathcal{M}$) using predicted semantic labels ($\mathcal{O}$), performs better on cylinder ('CYL') objects than natural ('NAT') objects. We compare its performance with two oracle agents, OracleMap and OracleSem where ground-truth ('GT') is provided for either the mapping or object semantics. As expected, the performance are mostly identical for the two object types for OracleMap and OracleSem, since the placement of the objects are the same for both, with OracleMap outperforming OracleSem. These methods use Uniform ('U') as the Exploration ($\mathcal{E}$) module and PointNav [17] ('PN') as the Navigation ($\mathcal{N}$) module.

Figure 2. **MultiON 2.0 vs MultiON objects.** The original dataset MultiON contains only cylinder objects, whereas MultiON 2.0 additionally contains more natural looking objects varying in shape, size and color. These easily blend in the HM3D houses, thus requiring better visual understanding for the agent. We use freely available 3D models from Sketchfab.

With predicted semantics (PredSem), the performance for the NAT objects drops compared with CYL since it is more challenging to detect these objects than different colored cylinders. When we use the OracleMap (the ground truth map) and OracleSem (where we use ground-truth semantic labels for the Object detection module), the performance on CYL and NAT objects are similar. The same observation holds when we compare different Navigation (Tab. 2) and Exploration (Tab. 3) methods for CYL and NAT objects. The performance variance for OracleSem in some cases for CYL vs NAT datasets is due to the randomness in the Navigation and Exploration modules.

D.2. Navigation performance

Tab. 2 provides the full comparison of the four different navigation modules for both CYL and NAT objects, for both the validation and test sets. In these experiments, we use the OracleSem mapping module with Uniform exploration. The performance on validation is similar to that of the test set, with PointNav agents having the highest *Success* while the Shortest Path Follower (SPF) has the highest SPL as it has access to the ground-truth navigation meshes.

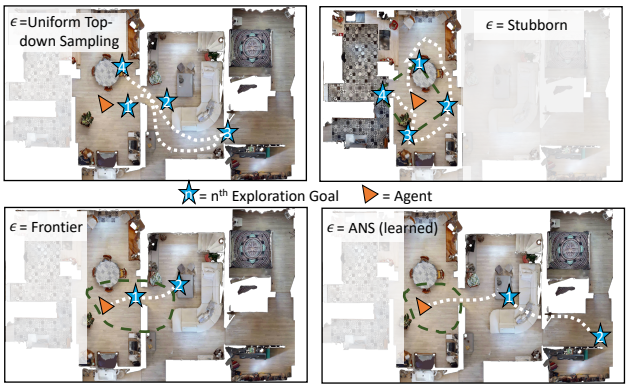

Figure 3. **Different Exploration strategies.** In *Uniform*, the agent uniformly samples exploration goals inside a local grid around itself, whereas in *Stubborn*, the agent selects each of the four corners of a local grid around itself. In *Frontier*, the agent samples a goal at the frontier, *i.e.*, the boundary between the explored and the unexplored areas. *ANS* is a learned exploration policy to predict distant goals to maximize coverage.

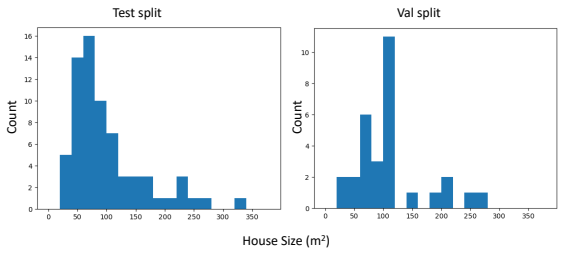

Figure 4. **HM3D scenes.** Majority of HM3D scenes are small.

D.3. Exploration performance

Tab. 3 provides the full comparison of four different exploration modules for both CYL and NAT objects, for both the validation and test sets. In these experiments, we use the

Method	MOPA Modules				Objects	Validation				Test			
	$\mathcal{O}$	$\mathcal{M}$	$\mathcal{E}$	$\mathcal{N}$		Success	Progress	SPL	PPL	Success	Progress	SPL	PPL
OracleSem	GT	[5]	Uniform	PointNav [17]	CYL	**80**	**87**	35	38	**81**	**87**	**37**	39
				BFS Path Planner [7]		27	41	19	29	21	44	12	22
				Shortest Path Follower* [19]		74	82	**39**	**43**	71	79	**37**	**42**
				Fast Marching Method [4]		19	37	13	25	18	36	11	21
				PointNav [17]	NAT	**80**	**85**	35	38	**81**	**87**	**37**	39
				BFS Path Planner [7]		27	41	19	29	21	44	12	22
				Shortest Path Follower* [19]		72	82	**38**	**43**	71	79	**37**	**42**
				Fast Marching Method [4]		19	37	13	25	18	36	11	21

Table 2. **Navigation module performance.** A learned PointNav, when used as the Navigation module in MOPA, outperforms analytical path planners (BFS, Shortest Path Follower and Fast Marching Method) on the 3ON task for both CYL and NAT datasets. We study the contribution of the Navigation module by using the ground truth (GT) semantic labels in the Object detection module, Map building from [5] (M) and Uniform (Uniform) as the Exploration module. We use * to indicate that the Shortest Path Follower has access to the ground truth navigation meshes from the Habitat simulator.

Method	MOPA				Objects	Validation				Test			
	$\mathcal{O}$	$\mathcal{M}$	$\mathcal{N}$	$\mathcal{E}$		Success	Progress	SPL	PPL	Success	Progress	SPL	PPL
OracleSem	GT	[5]	PN	Uniform		**80**	**87**	**35**	**38**	**81**	**87**	**37**	**39**
				Uniform w/o Fail-Safe	CYL	78	84	**35**	37	72	80	33	36
				Stubborn		75	82	**35**	**38**	72	80	33	36
				Stubborn w/o Fail-Safe [11]		69	77	25	27	66	75	23	26
				Frontier [25]		75	81	**35**	37	72	80	33	35
				ANS [4]		75	81	34	37	76	83	35	38
				Uniform		**80**	**85**	**35**	**38**	**81**	**87**	**37**	**39**
				Uniform w/o Fail-Safe	NAT	78	84	**35**	37	72	80	33	36
				Stubborn		75	82	**35**	**38**	72	80	33	36
				Stubborn w/o Fail-Safe [11]		69	77	25	27	66	75	23	26
				Frontier [25]		75	81	**35**	37	72	80	33	35
				ANS [4]		75	81	34	37	76	83	35	38

Table 3. **Exploration module performance.** Uniform strategy outperforms other heuristic-based and learned exploration strategies in MOPA on the 3ON task for both CYL and NAT datasets. We study the contribution of the Exploration module by using the ground truth (GT) semantic labels in the Object detection module, Map building from [5] (M) and PointNav (PN) as the Navigation module.

OracleSem mapping module with the PointNav agent. We illustrate how the different methods select goals in Fig. 3. For the exploration policies, it is possible to select a goal that is not navigable. To compensate for this, it is important to limit the number of steps the agent takes toward the exploration goal and select another exploration goal once this limit (α_{exp}) is reached. We conducted experiments with and without this threshold (w/o Fail-Safe) and found that this fail-safe is critical for good performance for both the Uniform and Stubborn approaches. Fig. 4 shows that most of the HM3D scenes are small having less than $100m^2$.

Delving into frontier. We investigate the popularly adopted Frontier [25] based exploration as a third strategy, where the agent selects an exploration goal at the boundary of the explored and the unexplored area. While the original paper selects the nearest accessible frontier as the exploration goal, we found that the distance at which we sample the exploration goal affects our agent performance (Fig. 5). The agent achieves the best performance when the exploration goal is sampled at a distance of 2m (10 grid cells away with each cell corresponding to 0.2m) from the boundary of explored area. This can be attributed to the fact that the MultiON task has a maximum step limit and thus expects the agent to

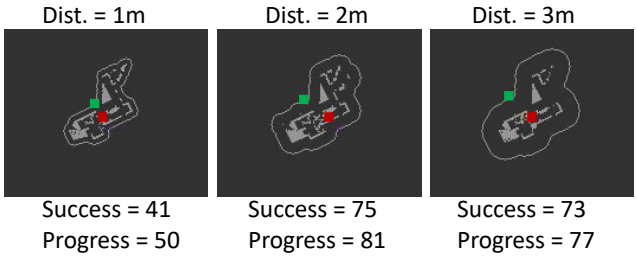

Figure 5. **Analysis on Frontier.** Agent performance varies with the distance at which the exploration goal is sampled in Frontier-based exploration. We achieve the best performance when the distance is 2m.

Distractors	Validation				Test			
	Success	Progress	SPL	PPL	Success	Progress	SPL	PPL
X	86	89	41	42	81	90	37	40
✓	85	89	39	40	81	87	37	39

Table 4. **Effect of distractors on OracleSem performance.** We observe that the MOPA agent performs equally well in the presence of distractors. This can be attributed to our target location retrieval method from the semantic map comparing directly with the next goal category.

effectively explore larger areas of the environment to find goals. We find that the agent is able to explore larger areas when we sample a goal farther away from the agent rather than sampling multiple goals nearer to the agent. However, we find that our simple Uniform strategy still outperforms the more sophisticated Frontier based exploration in the MultiON task.

D.4. MultiON 2.0 distractors vs. no distractors

Our MultiON 2.0 dataset additionally contains distractor objects in both CYL and NAT-objects episodes to make the episodes more challenging in terms of distinguishing between the goals and the distractors. We thus perform experiments to study the effect of having distractors for our MOPA. We evaluate our OracleSem agent on both validation and test sets for 3ON with and without distractors. Tab. 4 shows that the MOPA performs equally well in the presence of distractors. This is intuitive since we select the target location on the global map containing semantic categories of both the targets and the distractors by directly comparing with the next goal category given as input to the agent. However, distractors enable us to have cluttered environments thus making MultiON 2.0 closer to a more realistic setting. And our results demonstrate that our MOPA is invariant in the presence of clutter (distractors) in the environment.

D.5. Generalization of MOPA on n-ON

We study the generalization of MOPA to n-ON (1ON, 3ON, 5ON) episodes. MOPA allows us to use the same modules for any n-ON tasks without retraining. This is very efficient and generalizable compared to end-to-end approaches[21] that need to be retrained every time we introduce more objects. To study this, we evaluate the OracleSem agent on 1ON, 3ON, and 5ON episodes from both the validation and test sets. Although the performance decreases as we introduce more target objects (Tab. 5), with 1ON being the best and 5ON being the worst, the agent still performs considerably well across all n-ONs. The agent achieves a progress of 95% on 1ON, 87% on 3ON, and 76% on 5ON for the test set. We note that the progress values on 3ON and 5ON are comparable to the expected performance if we were to treat each of the goals independently and reset the agent after it finds each goal, with expected progress of $95\%^3 = 86\%$ for 3ON and $95\%^5 = 77\%$ for 5ON. However, the actual success rate on 3ON (81%) and 5ON (66%) are lower than 86% and 77% respectively. This can be explained by the fact that we keep the step limit fixed at 2500 for 1ON, 3ON, and 5ON, and so the task gets more challenging since the agent needs to find more objects within the same number of steps.

D.6. Effect of spatial map on exploration and navigation.

We analyse the importance of having spatial maps for exploration and navigation when we need to backtrack in MultiON. We find that when the agent already observes future goals and store them in the map it can efficiently navigate back to them. Tab. 6 shows the Path Length (PL) and Accuracy (Acc) with which the agent is able to reach the k^{th} goal if it was observed ('Seen') before $(k-1)^{\text{th}}$ goal was reached. We notice that for 'Seen', the path length is much shorter in Shortest Path Follower (last row) compared to PointNav (first row). We also find that Uniform covers the most area before the first goal has been reached. When comparing different exploration methods, we find that although the path length varies for 'Unseen' goals, it stays almost unchanged for 'Seen' goals.

D.7. Qualitative examples

Fig. 6 shows a rollout of the OracleSem policy with PointNav and Uniform. During the first phase of the rollout, we can see that the agent keeps exploring the environment since it has not yet discovered the first goal. Once the agent has found and navigated to every goal, the episode terminates successfully.

Fig. 7 shows a rollout of the OracleSem agent on one of the episodes from the 5ON test set. At each step the agent receives the egocentric depth and semantic observations along with the current goal category as inputs (column 1) and builds a top-down semantic map (column 3) from the egocentric object categories that it observes using the depth image. The agent switches between the Exploration and Navigation modes depending on whether it has seen the current target object. From the example, we see that the agent mostly explores the environment in the initial phase of the rollout. Once it starts discovering target objects, it navigates to them sequentially. Once it is able to successfully find all 5 objects, the episode terminates.

Fig. 8 and Fig. 9 show rollouts of the PredSem agent on the 3ON test set episodes with CYL and NAT objects respectively. Here the agent has access to the RGB and depth observations and the current goal category as inputs (column 1). The agent predicts the egocentric semantic category of the objects from the RGB image (column 2 shows the bounding box for the predicted object) and progressively builds a top-

	Dataset	Goals#	Max Steps	$\mathcal{O}$	$\mathcal{M}$	$\mathcal{N}$	$\mathcal{E}$	Validation				Test			
								Success	Progress	SPL	PPL	Success	Progress	SPL	PPL
1)	MultiON 2.0	1ON	2500	GT	[5]	PN	U	**96**	**96**	**36**	36	**95**	**95**	35	35
2)		3ON	2500	GT	[5]	PN	U	80	87	35	**38**	81	87	**37**	**39**
3)		5ON	2500	GT	[5]	PN	U	68	78	33	36	66	76	32	36
4)		1ON	500	GT	[5]	PN	U	69	69	34	34	68	68	34	34
5)	ObjNav [22]	1ON	500	GT	[5]	PN	U	64	-	30	-	-	-	-	-

Table 5. **Generalization of MOPA on n-ON.** Performance deteriorates as we increase the number of target objects on MultiON, for a fixed step limit (rows 1-3). We also notice that our approach performs similarly on the Habitat ObjectNav 2022 [22] and MultiON 2.0 1ON val set (rows 4,5) when we set the step limit to 500 steps, following ObjectNav task setting.

MOPA	First goal (k = 1)			Second goal (k = 2)							Third goal (k = 3)						
	Not reached		Reached	N	Seen			Not seen			N	Seen			Not seen		
	N_s	N_n	Cov		G_r	Acc	PL	G_r	Acc%	PL		G_r	Acc%	PL	G_r	Acc	PL
PN + Uniform	35	49	37	890	654	73	124	236	27	520	820	725	88	107	95	12	563
PN + ANS	26	72	35	861	620	72	122	241	28	545	785	696	89	106	89	11	649
PN + Frontier	25	167	29	714	523	73	121	191	27	449	629	563	90	111	66	10	545
PN + Stubborn	26	92	30	710	509	72	122	201	28	488	618	550	89	107	68	11	621
FMM + Uniform	30	150	34	678	490	72	130	188	28	697	451	397	88	141	54	12	734
SPF* + Uniform	33	110	36	803	563	70	71	240	30	548	742	637	86	70	105	14	594

Table 6. **Goal Discovery of k^{th} goal in 3ON.** Note: N_s: Number of goals seen but not reached, N_n: Number of goals not seen, Cov: Area covered (sqm) till reaching 1st goal, N: Total number of goals reached, G_r: Goals reached, Acc: Accuracy (%), PL: Avg path length to reach k^{th} goal after $(k-1)^{\text{th}}$ goal was reached. Observations: (1) PointNav vs Shortest Path Follower: For 'Seen', path length is much shorter in Shortest Path Follower. (2) Uniform covers most area before the 1st goal was reached. (3) For 'Seen' goals, the path length does not vary much for different exploration methods.

down semantic map (column 4) with the object categories using depth image. These examples also demonstrate that the agent mostly explores the environment in the first phase of the episodes, later switching to the Navigation mode once it discovers the target objects.

E. ObjectNav experiments

In this section, we report more results on ObjectNav task. We perform our experiments on both Habitat ObjectNav 2022 and 2021 challenge datasets[1]. ObjectNav 2022 challenge dataset is based on HM3D scenes and consists of 6 object categories: chair, couch, potted plant, bed, toilet and tv. We use HM3D-Sem v0.2 for our experiments. On the other hand, ObjectNav 2021 challenge dataset is based on MP3D scenes [3], consists of 21 object categories and contain 2195 validation episodes. In ObjectNav, the agent is allowed a maximum of 500 steps and the success is measured as whether the agent is able to navigate to and stop near any instance of the goal object. More specifically, each episode contains a list of viewpoints sampled at a distance of 1m from the goal object bounding box, and the episode is considered to be successful if the agent reaches within 0.1m of any of these viewpoints.

[1] https://aihabitat.org/challenge/2022, https://aihabitat.org/challenge/2021

	Method	Object Detection	Exp	Nav	Validation	
					Succ	SPL
1)	OracleSem (Ours)	GT	Uniform	PN	65	29
2)	PredSem (Ours)	Detic[26]	Uniform	PN	15	**12**
3)	EmbCLIP[10]	CLIP[16]	end-to-end w/ DD-PPO		**19**	9
4)	ZSON[12]	CLIP[16]	end-to-end w/ DD-PPO		15	5
5)	CoW[8]	OWL[14]	Frontier[25]	A*	7	4
6)	CoW[8]	CLIP[16] +GradRel[6]	Frontier[25]	A*	9	5
7)	OVRL[24]*	Self-supervised pretraining + ObjectNav finetuning			29	7

Table 7. **ObjectNav performance on Habitat ObjectNav 2021 challenge dataset.** PredSem with the Detic detector outperforms recent methods on the SPL metric.

MOPA performance on ObjectNav 2021 challenge dataset. Tab. 7 shows that our PredSem achieves better SPL than the prior works on the 2021 challenge dataset. We note that both EmbCLIP [10] and ZSON [12] requires training an action policy. In contrast, our modular approach makes use of pretrained modules and does not require any specific ObjectNav training. CoW [8] is also a modular

approach that uses Frontier based exploration and a target-driven planner based on vision-language models for visual features. Since the ObjectNav 2021 challenge is focused on just 21 object categories, we use Detic as our object detector. Our method (PredSem) is able to outperform CoW significantly. PredSem also outperforms OVRL [23], which is a fully supervised SOTA method, on SPL (while being lower on success rate).

References

[1] Aayush Bansal, Yaser Sheikh, and Deva Ramanan. PixelNN: Example-based image synthesis. In *ICLR*, 2018. 1

[2] 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

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *Intl. Conf. on 3D Comput. Vis.*, 2017. 5

[4] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using Active Neural SLAM. In *ICLR*, 2019. 3

[5] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object Goal Navigation using Goal-Oriented Semantic Exploration. In *NeurIPS*, volume 33, pages 4247–4258, 2020. 2, 3, 5

[6] Hila Chefer, Shir Gur, and Lior Wolf. Transformer Interpretability Beyond Attention Visualization. In *CVPR*, pages 782–791, 2021. 5

[7] Matt Deitke, Dhruv Batra, Yonatan Bisk, Tommaso Campari, Angel X Chang, Devendra Singh Chaplot, Changan Chen, Claudia Pérez D'Arpino, Kiana Ehsani, Ali Farhadi, et al. Retrospectives on the Embodied AI Workshop. *arXiv preprint arXiv:2210.06849*, 2022. 3

[8] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. CoWs on Pasture: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation. In *CVPR*, pages 23171–23181, 2023. 5

[9] Niv Granot, Ben Feinstein, Assaf Shocher, Shai Bagon, and Michal Irani. Drop the GAN: In defense of patches nearest neighbors as single image generative models. In *CVPR*, 2022. 1

[10] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but Effective: CLIP Embeddings for Embodied AI. *CVPR*, 2022. 5

[11] Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A Strong Baseline for Indoor Object Navigation. *arXiv preprint arXiv:2203.07359*, 2022. 3

[12] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. ZSON: Zero-Shot Object-Goal Navigation using Multimodal Goal Embeddings. *NeurIPS*, 35:32340–32352, 2022. 5

[13] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *ICCV*, 2011. 1

[14] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple Open-Vocabulary Object Detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022. 5

[15] Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021. 1

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 5

[17] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3D dataset (HM3d): 1000 large-scale 3D environments for embodied AI. In *NeurIPS Datasets and Benchmarks Track (Round 2)*, 2021. 2, 3

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *NeurIPS*, 28, 2015. 1

[19] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied AI research. In *ICCV*, pages 9339–9347, 2019. 3

[20] Nam Vo, Nathan Jacobs, and James Hays. Revisiting IM2GPS in the deep learning era. In *ICCV*, pages 2621–2630, 2017. 1

[21] Saim Wani, Shivansh Patel, Unnat Jain, Angel Chang, and Manolis Savva. MultiON: Benchmarking Semantic Map Memory using Multi-Object Navigation. *NeurIPS*, 33:9700–9712, 2020. 1, 4

[22] Karmesh Yadav, Santhosh Kumar Ramakrishnan, John Turner, Aaron Gokaslan, Oleksandr Maksymets, Rishabh Jain, Ram Ramrakhya, Angel X Chang, Alexander Clegg, Manolis Savva, Eric Undersander, Devendra Singh Chaplot, and Dhruv Batra. Habitat Challenge 2022. https://aihabitat.org/challenge/2022/, 2022. 5

[23] 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. *arXiv preprint arXiv:2204.13226*, 2022. 6

[24] 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 *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023. 5

[25] Brian Yamauchi. A Frontier-Based Approach for Autonomous Exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA). 'Towards New Computational Principles for Robotics and Automation'*, pages 146–151, 1997. 3, 5

[26] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähen-

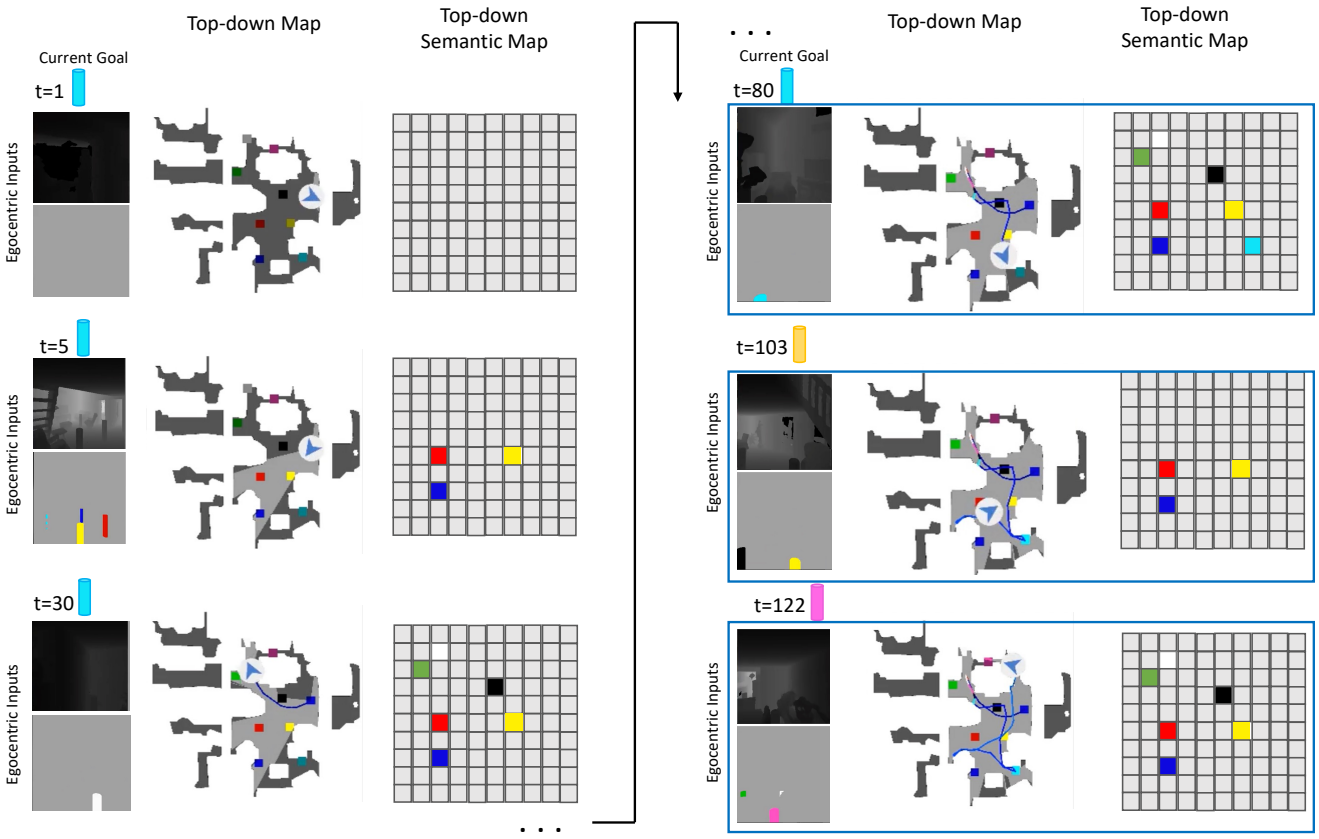

Figure 6. **OracleSem episode rollout** We visualize an episode rollout of the OracleSem agent over time (t). At $t = 1$, the agent has not yet observed the current goal (cyan cylinder). It keeps exploring and building the semantic map (third column) until it observes the current goal and navigates to it at $t = 80$. This process continues until it finds all the subsequent goals (yellow and pink). The Blue outline indicates that the agent executed the *found* action. The agent does not have access to the top-down obstacle map (second column) which is for visualization only.

bühl, and Ishan Misra. Detecting Twenty-thousand Classes using Image-level Supervision. In *ECCV*, 2022. 5

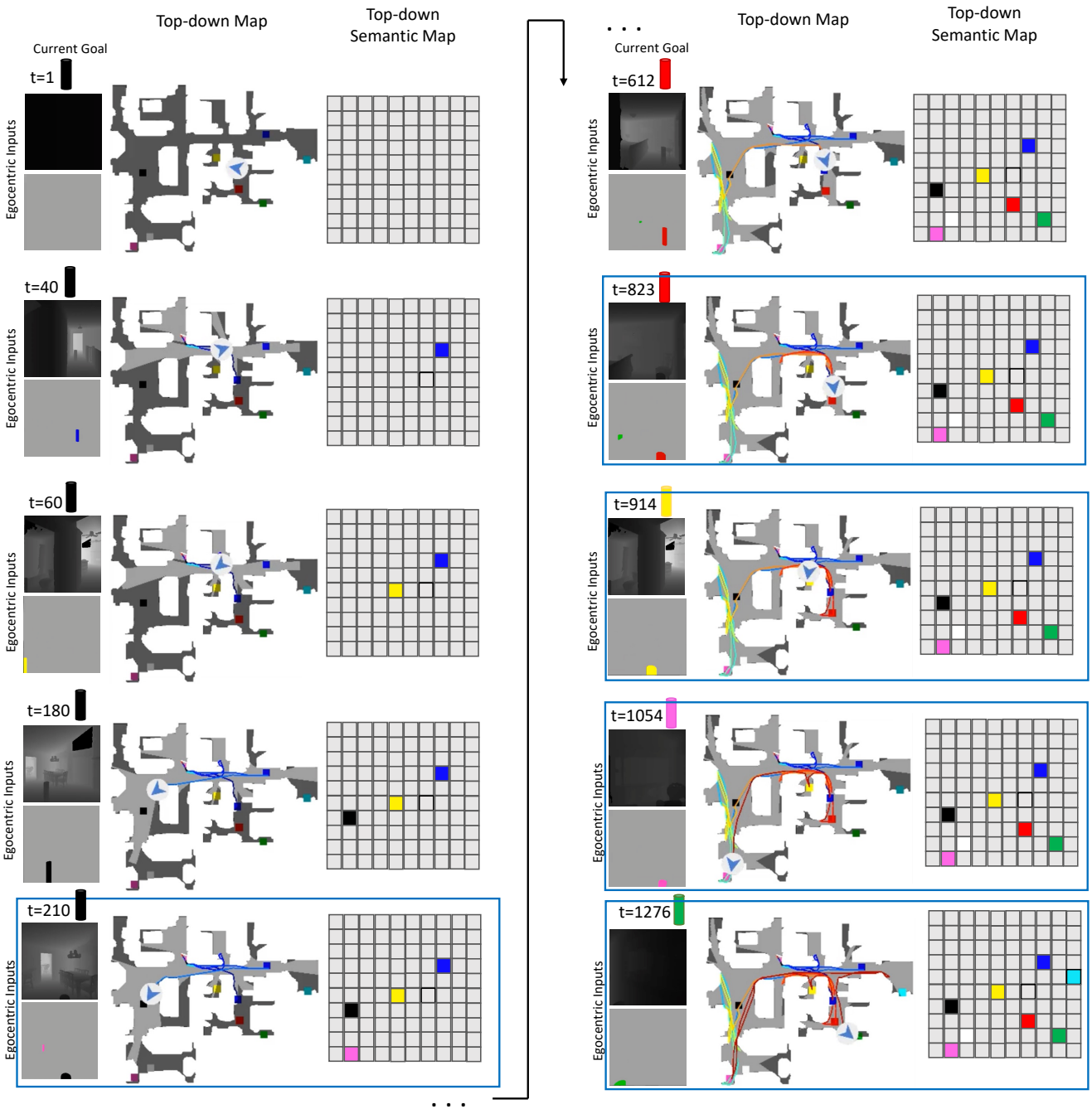

Figure 7. **Qualitative results: 5ON.** Rollouts of our OracleSem with PointNav and Uniform show that the agent explores over time (t) and discovers objects and progressively builds the semantic map using egocentric depth observations. The goal sequence is (black, red, yellow, pink, and finally green,). The top-down obstacle map is for visualization only; this agent does not have access to it. Blue outline indicates that the agent executed the *found* action. The agent has a 100% Success, 100% Progress, 39% SPL and 39% PPL in this episode.

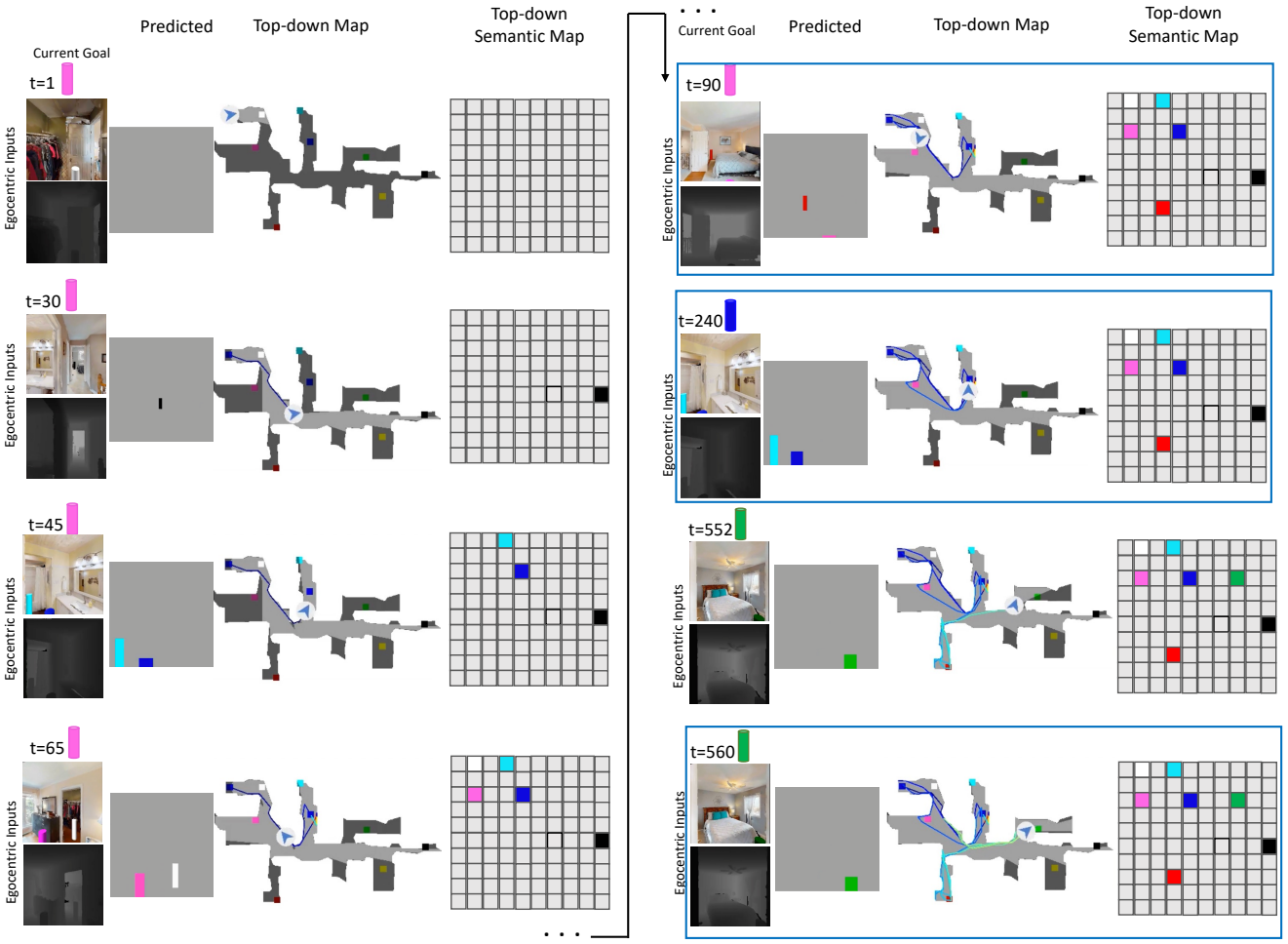

Figure 8. **Qualitative results: CYL objects.** Rollouts of our OracleSem with PointNav and Uniform show that the agent explores over time (t) and detects objects ('Predicted' column) and progressively builds the semantic map using egocentric depth observations. The goal sequence is (pink, blue, and finally green). The top-down obstacle map is for visualization only; this agent does not have access to it. Blue outline indicates that the agent executed the *found* action. The agent has a 100% Success, 100% Progress, 21% SPL and 21% PPL in this episode.

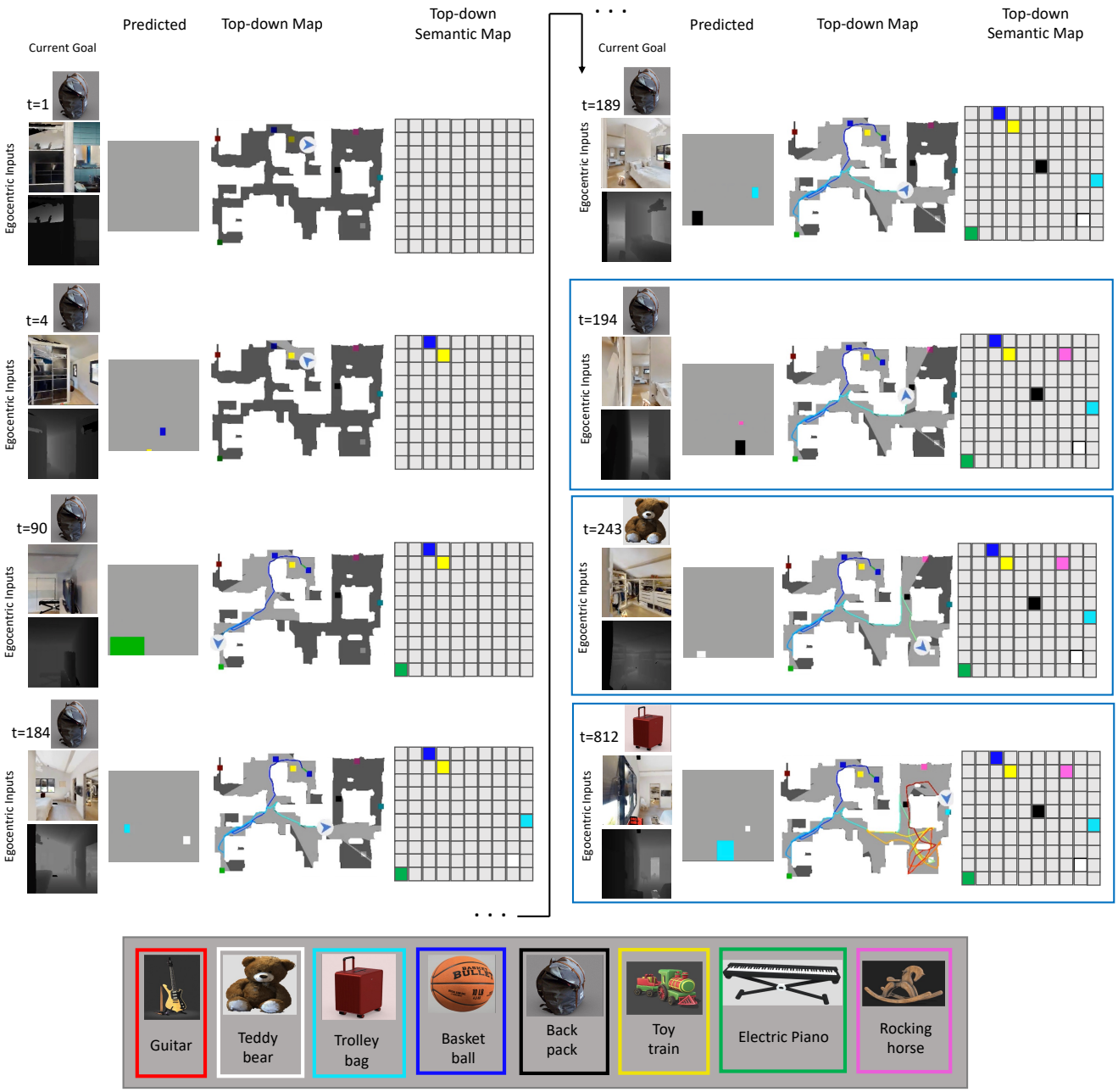

Figure 9. **Qualitative results: Natural objects.** Rollouts of our OracleSem with PointNav and Uniform show that the agent explores over time (t) and discovers target objects and progressively builds the semantic map using egocentric depth observations. The goal sequence is (backpack (black), teddy bear (white), and finally trolleybag (cyan)). The top-down obstacle map is for visualization only; this agent does not have access to it. Blue outline indicates that the agent executed the *found* action. The agent has a 100% Success, 100% Progress, 17% SPL and 17% PPL in this episode.