Comparison of Model-Free and Model-Based Learning-Informed Planning for PointGoal Navigation

Yimeng Li*, Arnab Debnath*, Gregory J. Stein, and Jana Košecká
Department of Computer Science
George Mason University United States
yli44, adebnath, gjstein, kosecka@gmu.edu

Abstract: In recent years several learning approaches to point goal navigation in previously unseen environments have been proposed. They vary in the representations of the environments, problem decomposition, and experimental evaluation. In this work, we compare the state-of-the-art Deep Reinforcement Learning based approaches with Partially Observable Markov Decision Process (POMDP) formulation of the point goal navigation problem. We adapt the (POMDP) subgoal framework proposed by [1] and modify the component that estimates frontier properties by using partial semantic maps of indoor scenes built from images' semantic segmentation. In addition to the well-known completeness of the model-based approach, we demonstrate that it is robust and efficient in that it leverages informative, learned properties of the frontiers compared to an optimistic frontier-based planner. We also demonstrate its data efficiency compared to the end-to-end deep reinforcement learning approaches. We compare our results against an optimistic planner, ANS [2] and DD-PPO [3] on Matterport3D [4] dataset using the Habitat Simulator [5]. We show comparable, though slightly worse performance than the SOTA DD-PPO approach, yet with far fewer data.

Keywords: Point Goal Navigation, Model-Based Planning, Deep Learning

1 Introduction

In the point goal navigation task, a robot is deployed in a previously-unseen environment and navigates to a point goal in the egocentric camera coordinate frame. This task has a long history in robotics [6] and embodied computer vision, and is also an essential component for many downstream tasks including object rearrangement [7], or disaster response scenarios [8]. With the recent emergence of visually rich simulators [5] and abundance of real-world environments [4, 9], several learning-based approaches [3, 10, 11] have been proposed. These learning-driven strategies for point-goal navigation [3, 12] are usually model-free, in the form of an end-to-end architecture, and trained via deep reinforcement learning. These policies are learned directly from images or intermediate representations of indoor scenes and are data-intensive to train, requiring millions or billions of frames of training data.

Instead of learning policies end-to-end directly from observations, another class of RL-based approaches [2, 13, 11] adopts a hierarchical approach. A high-level planner that predicts the waypoints on the map in the unknown area is approximated using a neural network and trained through reinforcement learning. Local navigation to predicted waypoints is handled by traditional path finding and following algorithms. Despite some empirical robustness, the waypoints are still predicted via a black-box neural network, and there is no guarantee that they are reachable.

*These authors contributed equally to this work

Recently, Stein et al. [1] proposed a learning-augmented model-based approach for point goal navigation. They introduce a high-level abstraction by assigning subgoals to frontiers (boundaries between free and unknown space) and define high-level actions corresponding to traveling to the point goal via a subgoal. This abstraction allows for model-based planning over the available frontiers and approximates a complex POMDP as a much simpler Markov Decision Process (MDP). They use a simple fully-connected neural network to learn the properties of the subgoals from local laser scans. Conditioned on accurate mapping and reliable low-level navigation, planning via this high-level model-based approach is robust. However, they only conduct experiments in office-like environments with a robot equipped with a laser sensor, limiting the information the agent can use to make predictions of unseen space.

In this work, we adapt Learning over Subgoals Planner (LSP) [1] and evaluate its performance on a large-scale pseudo-realistic indoor dataset (Matterport3D [4]) with the Habitat simulator [5]. Motivated by [14], we modify the learning module of LSP (designed initially for a laser range finder) by using visual sensing. We use U-Net deep learning architecture to learn the properties of the frontiers from semantic segmentation of images, projected on the semantic map using depth maps. We compare our LSP-UNet approach with an optimistic classical planner, a state-of-the-art end-to-end RL navigator (DD-PPO [3]), and a modular hierarchical RL-based approach (ANS [2]). We evaluate the performance of the LSP-UNet and optimistic planner using both the ground truth semantic map and the semantic map built from sensor observations.

2 Related Work

Embodied Agent Navigation Embodied agent navigation [15] refers to an agent's ability to navigate and interact with its environment in a way similar to how a human would. In recent years, there has been a surge of research in embodied navigation thanks to the availability of photo-realistic simulators [16, 17, 18, 9] for large-scale indoor environments [4, 19, 20]. The proposed tasks include PointGoal Navigation [12], ObjectGoal Navigation [21, 22, 23, 24], ImageGoal Navigation [10, 25, 26, 27] Vision-and-Language Navigation (VLN) [28, 29], and Object Rearrangement [30, 31, 7]. These efforts mainly aimed to develop learning-based strategies that enable the agent to carry out the tasks mentioned above in novel, previously unseen environments. The existing approaches typically use end-to-end RL-based models without explicit memory and train a policy using direct raw sensor measurements [3], or intermediate representations of the visual input [32, 33, 34] (e.g., depth maps, semantic segmentation, local occupancy maps). The traditional non-learning-based approaches typically start with exploration [35] and SLAM [36] to build a map of the environment, followed by path planning and trajectory following. On the other hand, modular-based approaches exploiting SLAM [37], path planning, trajectory following [38], and exploration [39]. The adopted LSE approach is a modular-based framework with a mapping module, a high-level path planner, and a local navigator.

End-to-End Point Goal Navigation Recent deep reinforcement learning methods [3, 40, 41] learn a policy to predict low-level actions directly from raw RGB-D observations or local occupancy maps to improve navigation performance or exploration coverage. These end-to-end RL approaches are usually sample inefficient and require a large number of episodes for convergence.

Intermediate Goals Alternative deep reinforcement learning methods [2, 13, 11, 42] use hierarchical models that know global policies that predict long-term goals and use local navigation modules to reach the long-term goal. However, such methods still choose long-term goals greedily, and these long-term goals are usually located in an unknown area that may not be reachable by the agent. Instead, Stein et al. [1] use a learning-augmented frontier-based abstraction for point goal navigation. The robot is guaranteed to reach the goal, and the learning module requires significantly less training data than the model-free RL approaches under their model-based formalism.

3 Learning-Augmented Model-Based PointGoal Navigation

We adapt the Learning over Subgoal Planning (LSP) approach [1] to tackle the task of reaching a point goal in unseen space in a minimum expected distance. LSP is a high-level planning framework in which navigation in a partially-mapped environment is formulated as a Partially Observable Markov Decision Process (POMDP). LSP introduces an abstraction to simplify the calculation of the expected reward in long-horizon planning. Under this abstraction, high-level actions correspond to *subgoals*, each associated with travel to and navigation beyond a *frontier*—a boundary between free and unseen space—in an effort to reach the unseen goal. Under this high-level planning abstraction, a Bellman Equation (Eq. 1) is used to represent the expected cost of executing a high-level action a_t from the belief b_t, represented as a three-element tuple consisting of the partially observed map m_t, the robot pose q_t, and the subgoal set $\mathcal{A}_t : b_t = \{m_t, q_t, \mathcal{A}_t\}$. The expected cost is defined as:

$$\begin{aligned} Q(\{m_t, q_t, \mathcal{A}_t\}, a_t) = D(m_t, q_t, q_{a_t}) + P_{\mathrm{S}}(m_t, q_{\mathrm{G}}, a_t) R_{\mathrm{S}}(m_t, q_{\mathrm{G}}, a_t) \\ + (1 - P_{\mathrm{S}}(m_t, q_{\mathrm{G}}, a_t)) \left[R_{\mathrm{E}}(m_t, a_t) + \max_{a \in \mathcal{A}_t \setminus a_t} Q(\{m_t, q_{a_t}, \mathcal{A}_t \setminus a_t\}, a) \right] \end{aligned} \tag{1}$$

Each action a_t, and its corresponding subgoal, is characterized by three properties, each a statistic of the unseen space to which it corresponds: the probability of the subgoal leading to the point goal P_{S}, expected cost of reaching the point goal via the subgoal R_{S}, and expected cost of failure (i.e., exploration) in case the subgoal doesn't lead to the point goal R_{E}. We train a U-Net [43] to estimate the subgoal properties for each subgoal-action, P_{S}, R_{S} and R_{E}, based on the currently observed map m_t and the relative point goal location q_{G}. $D(m_t, q_t, q_{a_t})$ is the cost of traveling from the current location q_t to the centroid of the frontier q_{a_t}. Note that this motion occurs entirely in the explored area of the map, so we use Dijkstra's algorithm to calculate this cost.

While LSP is not specific to navigation planning, it relies on frontier abstraction to simplify the decomposition of the problem and calculate the expected reward in long-horizon planning. The set of frontiers is usually a small finite set. The frontier is a unique abstraction for 2D occupancy maps specific to navigation. To extend LSP to other problems, e.g., object rearrangement [30], suitable simplifying abstractions need to be found.

3.1 Hierarchial Navigational Model

Similar to [10, 13, 11, 44], our planner is hierarchical: high-level planning involves selection among frontier subgoals via Eq. 1, while low-level planning involves navigation to the chosen frontier. Planning with our model to reach a point goal involves two stages: in the first stage of navigation where the point goal remains in the unknown area of the map, we make a high-level plan to reach the subgoal: (i) we compute the set of frontiers and the respective action set $\mathcal{A}(b_t)$ given the map m_t, (ii) we run the U-Net to estimate the properties P_{S}, R_{S} and R_{E} of each frontier, (iii) we compute the expected cost for each action according to Eq. (1), (iv) we choose the action a_t^* that minimizes the expected cost. The second stage of navigation involves low-level planning where, (v) we compute a motion plan to travel to the centroid of the frontier associated with action a_t^* (q^*) and finally (vi) we select the primitive action that best moves along the computed path and updates the map given the newly observed area. This process repeats at each timestep until there is a navigable path to the point goal on the explored area of the map m_t.

An illustration of our procedure can be found in Fig. 1.

4 Map and Frontier Computation

Occupancy and Semantic Mapping The occupancy map is a metric map of size $m \times m$ where each cell of the map can have three values: 0 (unknown), 1 (occupied), and 2 (free space). The semantic map is a metric map of size $m \times m$ where each cell's value is in the range $[0, N]$. Each

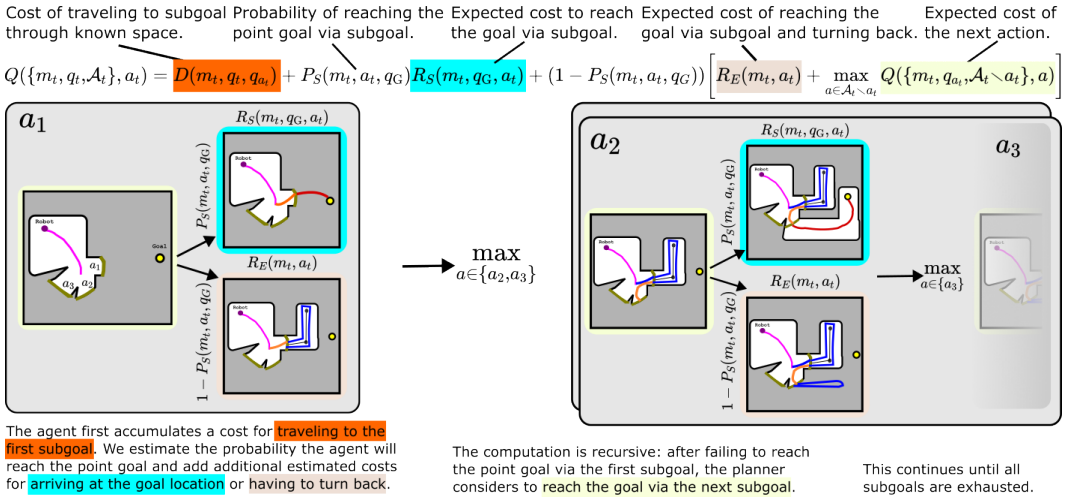

Figure 1: This diagram gives an overview of the LSP-UNet algorithm, which allows us to compute the expected cost of each action in a computationally feasible way for planning through an unknown environment. Compared to the original LSP [1] approach which estimates the properties of each frontier separately from laser scan, we use a camera sensor and use U-Net to estimate the terms P_S, R_S and R_E from the map, thereby introducing prior knowledge about environment regularities into the decision-making procedure.

value from 1 to N corresponds to one of the N object categories, and 0 refers to the undetected class. Each cell is a $5cm \times 5cm$ region in the real world. Given an RGB-D view, we build the occupancy map and semantic map by projecting semantic segmentation images to a 3D point cloud using the available depth maps and the robot poses, discretizing the point cloud into a voxel grid and taking the top-down view of the voxel grid. The occupancy map depends on the height of the points projected to each cell. The semantic map depends on the majority category of the points located at the top grid of each cell.

Frontier Computation Frontiers are the boundary between unknown and free space on the partial map. A frontier is computed as a set of connected cells on the boundary that follow an eight-neighbourhood connection. The center of the frontier is the centroid of the points belonging to that frontier. We further reduce the number of frontiers by filtering out frontiers unreachable from the robot on the current occupancy map. Fig. 2 shows the computed frontiers on the occupancy map.

Local Navigation The local navigation module uses the Fast Marching Method [45] to plan a path to the subgoal or the point goal from the current location on the occupancy map. It takes deterministic actions (move forward by 25cm, turn left/right by 30 degrees) to reach the subgoal. We update the map after each step and replan.

5 Learning-Based Modules

5.1 Training Data Generation

To plan via Eq. 1, we must estimate the values of the subgoal properties P_S, R_S and R_E. We do so via a neural network, trained with around 230k frames of data, compared to the 2.5 billion frames required to train the state-of-the-art DD-PPO. To generate training data, we use an optimistic planner with an RGB-D camera sensor to navigate from random start and goal positions in roughly 5,000 trials in each of the 61 environments. As the robot moves, revealing new portions of the environment, it maintains occupancy and semantic maps, extracts the newly updated frontiers, and computes labels of P_S, R_S and R_E used to train a neural network from the underlying "true" map of the environment, known to the robot at training time.

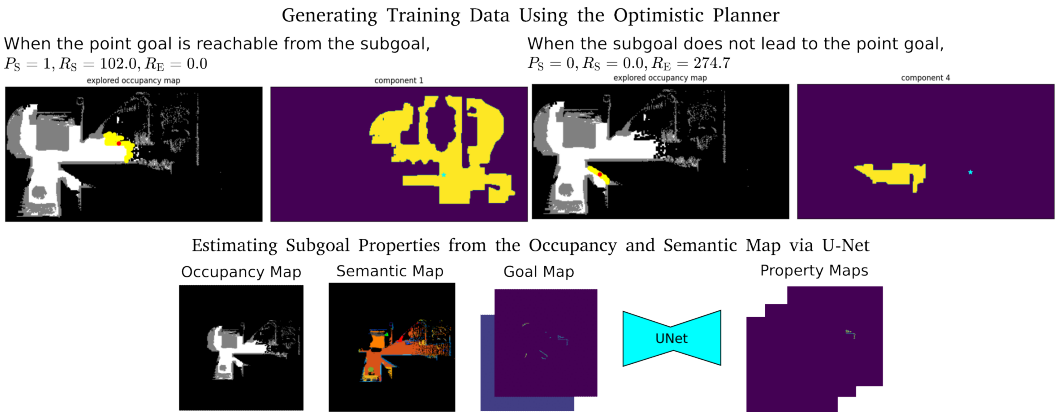

Figure 2: We generate training data by having an optimistic planner navigating the environment, maintaining an occupancy and semantic map, extracting the frontiers from the map, and computing their properties. The U-Net model takes in the local occupancy map, local semantic map, and the goal map and estimates properties for all the visible frontiers.

If a frontier leads to the goal through the unknown space, P_S is labeled 1, and R_S is calculated as the path length to the goal via the unknown space. If a frontier does not lead to the goal through the unknown space, P_S is labeled 0, and R_E, the traveling distance to explore beyond a frontier and coming back, is computed on the skeleton of the ground truth occupancy map on the unknown space. For each subgoal (each corresponding to a frontier), we take the neighboring connected component on the unknown region and generate a skeleton of the surrounding ground truth free space. To compute R_E, we compute the subgraph of the skeleton and run a TSP solver (Travelling Salesman Problem solver) on the subgraph to get the trajectory length of visiting every node once and returning to the frontier.

We label the calculated values P_S, R_S and R_E on frontier pixels. In the end, we achieve three maps m_{P_S}, m_{R_S} and m_{R_E} only having values on the frontier pixels. For the convenience of computation, we create three masks, mask_{P_S}, mask_{R_S} and mask_{R_E}, to denote if the pixel has the ground truth label or not. To indicate the position of the point goal, we create a 2-channel goal map m_{q_G} by computing the relative coordinates between the point goal and each frontier centroid. For each frame, we collect the data tuple $(m_t, m_{q_G}, m_{P_S}, m_{R_S}, m_{R_E}, \text{mask}_{P_S}, \text{mask}_{R_S}, \text{mask}_{R_E})$ for training the learning module as described in the section below.

5.2 Learning Module Architecture

We use a U-Net [43] architecture that takes the current observed map m_t and the goal map m_{q_G} as input and outputs a same-sized map with three channels. Each channel of the output map represents the predictions of P_S, R_S, and R_E at each pixel. We train the U-Net through supervised learning. We add a sigmoid activation after the output of P_S and use a Binary Cross-Entropy loss $\mathcal{L}_{\text{BCE}}$ to train it. The output of R_S and R_E is trained with the L1-loss $\mathcal{L}_{\text{L}_1}$. The loss for R_S is only applied when ground truth P_S is 1 and the loss for R_E is only applied when ground truth P_S is 0. The three mask values indicate the positions for which the losses are calculated. Our final objective is:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{BCE}} + \lambda * \mathcal{L}_{\text{L}_1} \tag{2}$$

These losses are computed only on the frontier pixels indicated by the generated masks. λ is setup as 10^{-2} in practice.

Fig. 2 shows the data collection process and the U-Net architecture.

Table 1: Comparison of Point Goal Navigation Methods

Dataset	MP3D Test		
Method	Success (%)	SPL (%)	SoftSPL (%)
ANS [2]	59.3	49.6	-
DD-PPO [3, 41]	**89.0**	**80.0**	-
Optimistic [35]	75.9	60.1	64.2
LSP-UNet [1]	82.0	60.5	63.4
Optimistic* [35]	87.5	70.5	72.7
LSP-UNet* [1]	88.2	74.7	77.0

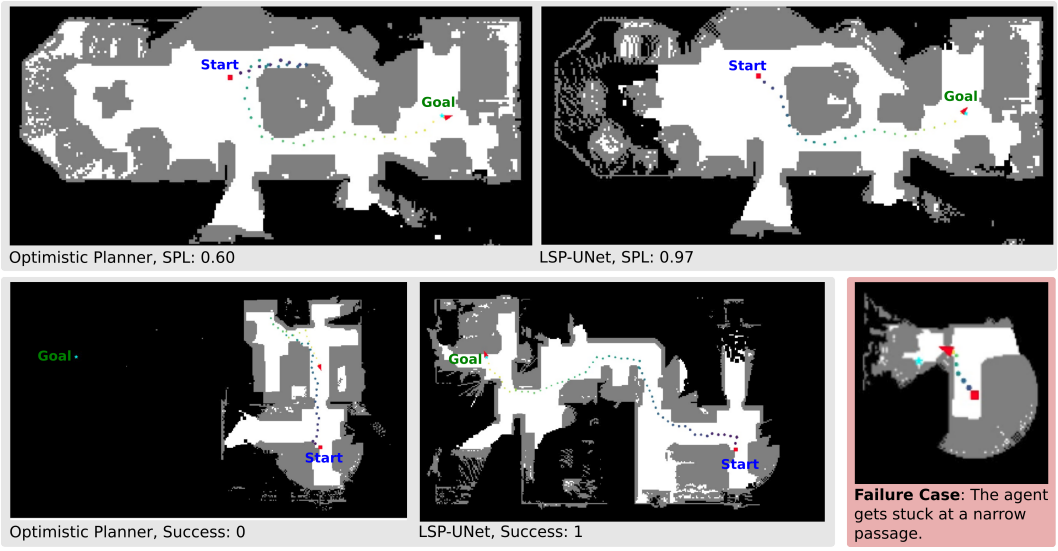

Figure 3: Success and failure examples running the *Optimistic* and *LSP-UNet* planner. The *LSP-UNet* planner generally knows to avoid dead ends and achieves better-expected performance than the *Optimistic* planner. The typical failure cases are the fault of the local planner.

6 Experiments

Datasets + Simulated Environments We use the Habitat [5] simulator with the Matterport (MP3D) [4] dataset for our experiments. MP3D consists of 90 scenes that are 3D reconstructions of real-world environments, and semantic segmentations (40 semantic categories) of egocentric views are available. We split the MP3D scenes into train:val:test following the standard 61:11:18 ratio to train the perception models and evaluate the navigators. We compare different Reinforcement Learning approaches using the standard 1008 testing episodes generated for the Habitat PointGoal task [5]. For the experiments below, noise-free poses and ground truth egocentric semantic segmentations are provided by the simulator.

Task Setup The robot observation space consists of RGB, depth, and semantic segmentation images. Even though the LSP approach is supposed to perform better with a panoramic camera sensor, to compare with the existing approaches, our agent is equipped with an egocentric 90° FoV sensor with observations of 256×256 resolution. The action space contains four short-term actions: `MOVE_FORWARD` by 25cm, `TURN_LEFT` and `TURN_RIGHT` by 30° and `STOP`.

We follow the point goal task setup from [46] where given a goal location, the agent needs to navigate to the goal and stop within a 0.2m radius. The agent is allowed to move 500 steps to reach the goal. We use the standard metrics for evaluation [46]; *Success*: percentage of successful episodes, *SPL*: success rate normalized by path length, and *SoftSPL*: success rate normalized by the progress to the goal.

Implementation Details For the training of the U-Net, we pre-compute 230,000 train / 2,000 val data tuples. We train the model using PyTorch for six epochs and use the Adam optimizer with a learning rate of 0.001 decayed by a factor of 10 after every two epochs.

6.1 Point Goal Navigation with an Egocentric 90° FoV Sensor

We compare the LSP-UNet approach with the following three methods:
Active Neural SLAM (ANS) [2]: A learned exploration model trained with model-free Deep RL using coverage as rewards. We use this model as an example for modular model-free RL approaches. It has a global policy module for selecting long-term goals given the currently observed occupancy map and agent-visited location history as input. We use the results evaluated on MP3D from the original paper [2].
DD-PPO [3]: DD-PPO essentially solves the point goal navigation task on the Gibson Environments. We include it as the example work for end-to-end model-free RL models. We report the numbers using their latest results trained on the MP3D scenes [41].
Frontier-Based Optimistic Planner (Optimistic) [35]: A non-learned greedy algorithm – which plans by assuming all unknown space is unoccupied and uses the path length from the agent to the point goal through each subgoal as the heuristic. We use the same optimistic planner for data generation in Sec. 5.

We show the evaluation results in Table 1. The *Optimistic* and *LSP-UNet* methods outperform *ANS* by over 15% in *Success* but are at least 7% worse than *DD-PPO*. We observe that because the implemented mapping module (described in detail at Sec. 4) does not involve learning, it is prone to the perception noise from the simulator. It sometimes misrecognizes free locations as occupied, which dramatically affects the performance of the entire navigation framework. To reduce the effect of the mapper, we further evaluate *Optimistic* and *LSP-UNet* with an oracle mapper that produces occupancy maps with no errors. We name the two approaches as *Optimistic** and *LSP-UNet** in Table 1. The performance of *LSP-UNet** is similar to *DD-PPO* but slightly worse in *SPL*. *DD-PPO* performs the best in the existing approaches without an explicit map. We analyze the failure cases of *Optimistic** and *LSP-UNet** and illustrate some example success and failure cases in Fig. 3. A failure case usually happens when the local navigation module has difficulty passing through a narrow passage. Considering that the typical failure cases are the fault of the local planner, improving the local planner will be the subject of future work. We can use A* with motion primitives instead.

7 Conclusion

We adapted an existing model-based point goal navigation approach, augmented its learning module with a deep convolutional neural network, and compared its performance with a set of SOTA RL-based approaches. Exploiting advances in visual sensing, semantic mapping, and availability of large datasets of rendered 3D models of real indoor scenes, we showed how to train a deep convolutional neural network to learn three informative properties of the frontiers: (i) likelihood of the frontier leading to the goal, (ii) expected cost of reaching the goal via the frontier and (iii) expected cost of exploring the frontier. Our experiments suggest that our model is comparable to DD-PPO in point goal navigation in performance and requires much less training data. Another desirable property is that the representations used by our approach, e.g., semantic map, are reusable for other navigation tasks.

References

[1] G. J. Stein, C. Bradley, and N. Roy. Learning over subgoals for efficient navigation of structured, unknown environments. In *Conference on Robot Learning*, pages 213–222. PMLR, 2018.

[2] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov. Learning to explore using active neural mapping. In *International Conference on Learning Representations*, 2020.

[3] 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. *arXiv preprint arXiv:1911.00357*, 2019.

[4] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[5] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019.

[6] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, et al. Minerva: A second-generation museum tour-guide robot. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 3. IEEE, 1999.

[7] J. Gu, D. S. Chaplot, H. Su, and J. Malik. Multi-skill mobile manipulation for object rearrangement. *arXiv preprint arXiv:2209.02778*, 2022.

[8] C. Bradley, A. Pacheck, G. J. Stein, S. Castro, H. Kress-Gazit, and N. Roy. Learning and planning for temporally extended tasks in unknown environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4830–4836. IEEE, 2021.

[9] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018.

[10] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta. Neural topological slam for visual navigation. In *CVPR*, 2020.

[11] G. Georgakis, B. Bucher, A. Arapin, K. Schmeckpeper, N. Matni, and K. Daniilidis. Uncertainty-driven planner for exploration and navigation. *arXiv preprint arXiv:2202.11907*, 2022.

[12] R. Partsey, E. Wijmans, N. Yokoyama, O. Dobosevych, D. Batra, and O. Maksymets. Is mapping necessary for realistic pointgoal navigation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17232–17241, 2022.

[13] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman. Occupancy anticipation for efficient exploration and navigation. In *European Conference on Computer Vision*, pages 400–418. Springer, 2020.

[14] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman. PONI: Potential functions for ObjectGoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18890–18900, 2022.

[15] M. Deitke, D. Batra, Y. Bisk, T. Campari, A. X. Chang, D. S. Chaplot, C. Chen, C. P. D'Arpino, K. Ehsani, A. Farhadi, et al. Retrospectives on the embodied ai workshop. *arXiv preprint arXiv:2210.06849*, 2022.

[16] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[17] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[18] 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. *arXiv preprint arXiv:1712.05474*, 2017.

[19] 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*, 2021. URL https://arxiv.org/abs/2109.08238.

[20] 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. URL https://arxiv.org/abs/2210.05633.

[21] G. Georgakis, Y. Li, and J. Kosecka. Simultaneous mapping and target driven navigation. *arXiv preprint arXiv:1911.07980*, 2019.

[22] G. Georgakis, B. Bucher, K. Schmeckpeper, S. Singh, and K. Daniilidis. Learning to map for active semantic goal navigation. *arXiv preprint arXiv:2106.15648*, 2021.

[23] A. Pal, Y. Qiu, and H. Christensen. Learning hierarchical relationships for object-goal navigation. In *Conference on Robot Learning*, pages 517–528. PMLR, 2021.

[24] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33: 4247–4258, 2020.

[25] N. Savinov, A. Dosovitskiy, and V. Koltun. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018.

[26] M. Hahn, D. S. Chaplot, S. Tulsiani, M. Mukadam, J. M. Rehg, and A. Gupta. No rl, no simulation: Learning to navigate without navigating. *Advances in Neural Information Processing Systems*, 34:26661–26673, 2021.

[27] Y. Li and J. Košecka. Learning view and target invariant visual servoing for navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 658–664. IEEE, 2020.

[28] J. Krantz, A. Gokaslan, D. Batra, S. Lee, and O. Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15162–15171, 2021.

[29] J. Krantz, S. Banerjee, W. Zhu, J. Corso, P. Anderson, S. Lee, and J. Thomason. Iterative vision-and-language navigation. *arXiv preprint arXiv:2210.03087*, 2022.

[30] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020.

[31] G. Sarch, Z. Fang, A. W. Harley, P. Schydlo, M. J. Tarr, S. Gupta, and K. Fragkiadaki. Tidee: Tidying up novel rooms using visuo-semantic commonsense priors. In *European Conference on Computer Vision*, pages 480–496. Springer, 2022.

[32] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson. Visual representations for semantic target driven navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8846–8852. IEEE, 2019.

[33] A. Sax, B. Emi, A. R. Zamir, L. Guibas, S. Savarese, and J. Malik. Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies. *arXiv preprint arXiv:1812.11971*, 2018.

[34] W. B. Shen, D. Xu, Y. Zhu, L. J. Guibas, L. Fei-Fei, and S. Savarese. Situational fusion of visual representation for visual navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2881–2890, 2019.

[35] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997.

[36] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[37] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2616–2625, 2017.

[38] A. Kumar, S. Gupta, D. Fouhey, S. Levine, and J. Malik. Visual memory for robust path following. *Advances in neural information processing systems*, 31, 2018.

[39] Y. Li, A. Debnath, G. Stein, and J. Kosecka. Learning-augmented model-based planning for visual exploration. *arXiv preprint arXiv:2211.07898*, 2022.

[40] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 538–547, 2019.

[41] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

[42] C. Chen, S. Majumder, Z. Al-Halah, R. Gao, S. K. Ramakrishnan, and K. Grauman. Learning to set waypoints for audio-visual navigation. *arXiv preprint arXiv:2008.09622*, 2020.

[43] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[44] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293, 2021.

[45] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[46] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.