LEVERAGING ONLINE SEMANTIC POINT FUSION FOR 3D-AWARE OBJECT GOAL NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Object goal navigation in unseen environments is a fundamental task for building intelligent embodied agents. Existing works tackle this problem with modular or end-to-end learning-based methods, which implicitly learn from 2D maps, sparse scene graphs or video sequences, ignoring the established fact that objects lie in 3D. Hence, in this work, we propose a dedicated 3D-aware online semantic point fusion algorithm that online aggregates 3D points along with their semantic predictions from RGB-D observations to form a high-efficient 3D point-based sparse map, which further enables us to check spatial semantic consistency. To leverage the 3D information for navigation while remaining sample efficient, we then propose a two-stage reinforcement learning framework that decomposes the object goal navigation into two complementary sub-tasks, namely *exploration* and *verification*, each learning in a different discrete action space. Thanks to the highly accurate semantic understanding and robust goal verification, our framework achieves the best performance among all modular-based methods on the Matterport3D and Gibson datasets. Furthermore, compared to mainstream RL based works, our method requires (5-28x) less computational cost for training. We will release the source code upon acceptance.

1 INTRODUCTION

As a vital task for intelligent embodied agents, object goal navigation (ObjectNav), asks the agents to find an object of a given category via exploring in an unmapped scene. A surge has occurred recently in the research community, that learns 3D scene priors over abstract representations, e.g., 2D maps (Ramakrishnan et al., 2022; Georgakis et al., 2022; Chaplot et al., 2020b), scene graphs (Zhu et al., 2021) or directly over RGBD sequences (Ye et al., 2021; Ramrakhya et al., 2022; Maksymets et al., 2021), to enable ObjectNav. We argue that semantic understanding plays a crucial role in ObjectNav, however existing works fall short of delivering highly accurate semantic predictions to guide the navigation, due to their unawareness of the 3D structure.

Given that objects naturally lie in 3D space, 3D scene understanding naturally offers more accurate, spatially dense and consistent semantic prediction than its 2D counterpart, as proved by Dai & Nießner (2018); Nekrasov et al. (2021); Vu et al. (2022). Hence if the agent could take advantage of the 3D structure derived from multi-view observations during navigation, it is expected that the agent will have a more comprehensive understanding of the surrounding 3D environments.

However, leveraging 3D scene representations in ObjectNav raises two main concerns: 1) building and querying 3D scene representation requires extensive computational cost (Zheng et al., 2019); 2) training reinforcement learning policy with 3D scene representation typically suffers from low sampling efficiency (Zhu et al., 2017; Lin et al., 2020). These issues collectively hinder the use of 3D representations in the ObjectNav task.

To tackle these issues, we propose a 3D-aware two-stage object goal navigation approach featured by two key designs.

First, we propose a highly efficient online semantic point fusion algorithm that online organizes the 3D points into a 3D sparse map and updates their semantic labels along with spatial semantic consistency in 3D space. Specifically, the unstructured points are organized into sparse blocks of 3D grids at the coarse level and per-point octree at a fine-grained level for fast querying and neighborhood searching, respectively. It's worth noting that, different from dense voxel-based representation Chaplot et al. (2021), 3D points are naturally more memory-efficient. Based on all these

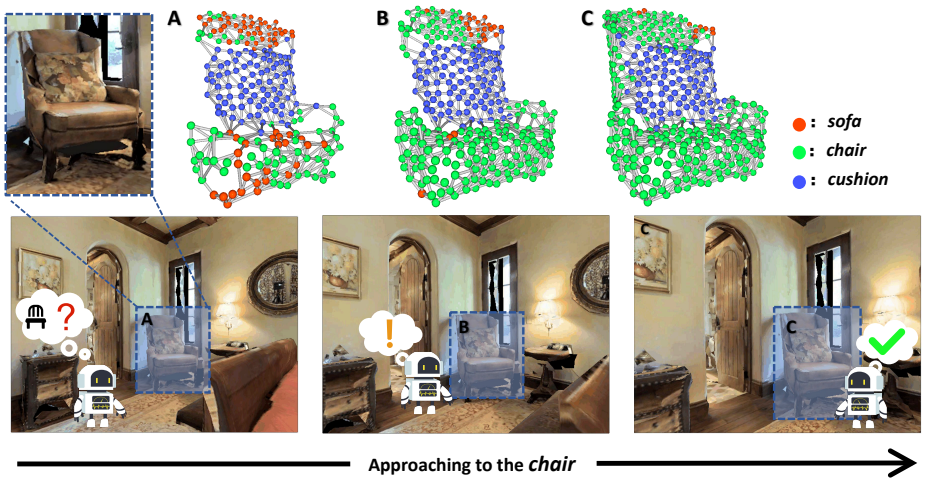

Figure 1: We present a two-stage navigation framework along with an online semantic point fusion: from A $\rightarrow$ B, explore the environment to find a potential target category object; from B $\rightarrow$ C, verify the correctness of the target category object and confirm whether to stop. As the robot approaches the *chair*, semantic prediction and spatial semantic consistency are improved, thanks to the fusion.

advantages, our method is able to achieve near real-time frame rates and requires extremely few memory resources.

Second, we use a two-stage navigation mechanism to drive the robot to reach the target goal. Unlike current popular methods, which directly learn to predict a global goal Ramakrishnan et al. (2022); Yadav et al. (2022), we separate the goal navigation task into two complementary sub-tasks: exploration and verification, each with different learning policies. The exploration policy, taking a coarse 2D map and fine-grained 3D points as input, learns to predict a discrete direction in a low dimension. It shares the benefit of avoiding back-and-forth paces with heuristic-based methods Luo et al. (2022), while efficiently leveraging the semantic priors. When the agent observes something that is predicted with goal semantic label and needs to decide whether marches to the place, our proposed verification policy learns to dynamically adjust a confidence threshold to accommodate different object categories, and we can further use the spatial semantic consistency to check whether we can trust the semantic prediction, which makes our navigation robust to semantic errors. Benefiting from the accurate and robust 3D understanding, our method significantly outperforms all the previous object goal navigation works while achieving a very high sampling efficiency, owing to the low-dimensional discrete action spaces and sub-tasks learning.

To summarize, our method incorporates online semantic point fusion with a two-stage policy learning to present a practically performing solution to 3D-aware object goal navigation. To the best of our knowledge, this work is the first 3D fusion-based ObjectNav method. The contributions are:

- We construct a 3D semantic point fusion framework that is able to on-the-fly update temporal semantic prediction and consistency. This framework requires satisfactory computational resources and enables a comprehensive 3D understanding of the environment.
- We develop a two-stage goal navigation method to improve sampling efficiency. Our method disentangles the object goal navigation task into two sub-tasks: an exploration stage and a verification stage, both with discrete action space for RL.
- Experiments on the photorealistic 3D environments of Gibson Xia et al. (2018) and Matterport3D Chang et al. (2017) validate the effectiveness of our key designs. Our method outperforms all the existing modular-based methods and requires (up to 10x) less time than mainstream modular RL-based methods Chaplot et al. (2020b); Georgakis et al. (2022).

2 Related Work

GoalNav with Visual Sequences. There are constantly emerging researches on object goal navigation. One line of recent works directly leverages RGBD sequences, called end-to-end RL meth-

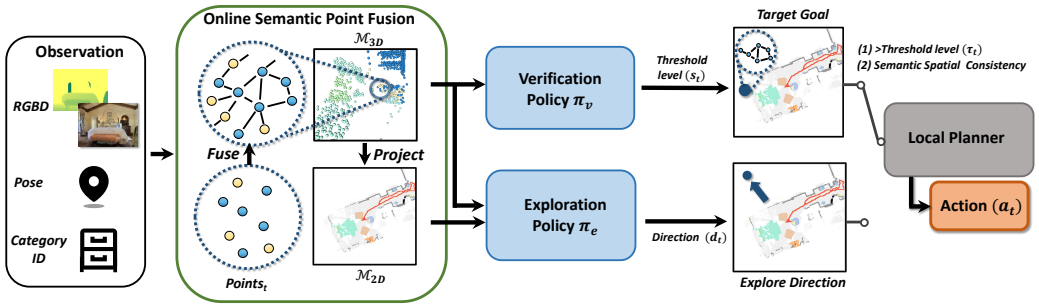

Figure 2: An overview of our approach. We take in posed RGB-D images at time step t and perform *online semantic point fusion* to organize the 3D points ($\mathcal{M}_{3D}$), along with a $\mathcal{M}_{2D}$ from semantics projection. Then, we leverage a two-stage policy, including *exploration policy* and *verification policy*, to predict a discrete direction and target goal (if exists). Finally, a local planner is used to drive the robot to the given direction or target goal.

ods Wijmans et al. (2019), which tends to implicitly encode the environment and predict low-level actions. These works benefit from visual representation Mousavian et al. (2019); Yang et al. (2018), auxiliary task Ye et al. (2021), and data augmentation Maksymets et al. (2021), demonstrating strong results on object goal navigation benchmarks Batra et al. (2020b); Yadav et al. (2022). However, by learning all skills, e.g., avoiding collisions, exploration, and stopping from scratch, it's well known that it suffers from sampling efficiency and generalizability. Ramakrishnan et al. (2022); Campari et al. (2020).

GoalNav with Explicit Scene Representations. To ease the burden of learning directly from visual sequences, another category of methods, called modular-based methods Chaplot et al. (2020a;b); Parisotto & Salakhutdinov (2018); Gupta et al. (2017); Georgakis et al. (2019), use explicit representations as a proxy for robot observations. By leveraging explicit scene representations like scene graph Zhu et al. (2021); Qiu et al. (2020) or 2D top-down map Ramakrishnan et al. (2022); Georgakis et al. (2022), modular-based methods benefit from the modularity and shorter time horizons. They are considered to be more sample efficient and generalizable Ramakrishnan et al. (2022); Georgakis et al. (2022). Recent progress in modular-based methods has proposed a frontier-based exploration strategy Ramakrishnan et al. (2022), a hallucinate-driven semantic mapping method Georgakis et al. (2022), and novel verification stage Luo et al. (2022). In contrast with prior map-based works, our method utilizes 3D spatial knowledge, including 3D point semantic prediction and consistency, enabling a more comprehensive understanding of the environments.

Online 3D Scene Segmentation. With the ability to on-the-fly construct scenes and predict semantic or instance labels, online scene segmentation methods have potential applications in embodied AI tasks. In this literature, the leading works perform the 3D convolutionZhang et al. (2020); Huang et al. (2021) or graph neural network Wald et al. (2020); Rosinol et al. (2020) on dense scene representation, *e.g.* voxels or patch-based graph Wu et al. (2021), demonstrating better semantic prediction results on existing room-level datasets Dai et al. (2017). Despite the efforts to improve efficiency Liu et al. (2022), these methods still require extensive computational resources under floor-level or building-level scenes Xia et al. (2018); Chang et al. (2017). To reduce the burden of 3D convolution, McCormac et al. (2017); Narita et al. (2019); Grinvald et al. (2019) directly back-project the 2D segmentation result to 3D space and perform temporal 3D fusion with heuristic design algorithms. In our work, intended for making online 3D scene understanding practically useable in GoalNav, we extend a point-based framework Zhang et al. (2020) with efficiently multi-view semantic label fusion while requiring satisfactory computational resources for RL.

3 Approach

3.1 Task Definition and Overview

Object Goal Navigation Task. In the Object Goal task, the robot is expected to navigate to an instance of a specific object category (e.g., *chair*) in an unknown environment. The robot is initialized at a random location with a target category object ID and does not have access to a pre-built environment map. At each time step t, the agent receives onboard sensor readings, including an RGBD

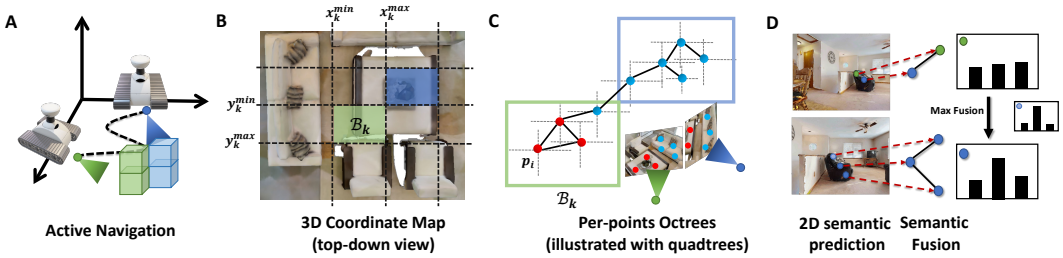

Figure 3: Illustration of online semantic point fusion. (**A**) A robot takes multi-view observations during navigation. (**B**) The online captured points are divided into blocks of a 3D coordinate map for fast querying. (**C**) The points are organized as per-point octrees which can be used to search neighborhood points of any given point. (**D**) The 2D semantic prediction from multi-view observations will be fused to obtain 3D semantic prediction.

camera (first-person RGB and depth images) and GPS+compass (location and orientation relative to the start of an episode). The agent then executes an action $a_t \in \mathcal{A}$, where $\mathcal{A}$ consists of four discrete actions: `move_forward`, `turn_left`, `turn_right` and `stop`. The agent is required to navigate within $d_s = 1.0m$ of the target object and take the `stop` action to complete the task. The agent is allowed to take no more than 500 steps.

Overview. We present a novel framework for object-goal navigation with two-stage reinforcement learning powered by an online semantic point fusion algorithm. During the navigation, our method consistently constructs a 3D coordinate map to on-the-fly organize 3D points from observed posed RGBD frames (Sec.3.2). Benefiting from fusing multi-view observation. We can online achieve more complete scene reconstruction and more accurate semantic prediction, which can be further used to measure the consistency of the target object. To efficiently leverage the semantic point fusion framework, we separate the object goal navigation into two complementary sub-tasks: exploration and verification (Sec.3.3). In the exploration stage, a policy implicitly learns the semantic and consistency priors from coarse-grained 2D maps and fine-grained 3D points. Then the policy predicts a discrete exploring direction, which drives the robot to explore the unobserved area where it could potentially find the target goal. For the verification stage, the agent exclusively takes the 3D points observation to predict a dynamic confidence threshold with a consistency check mechanism to confirm the final target goal. These two stages simultaneously perform during navigation. If the verification does not find any target goal, the agent will follow the predicted direction to explore the environment until a target goal is predicted and verified. To reach the given direction or target goal, we use a local planner to navigate the agent using analytical path planning. A visualization of the proposed pipeline can be found in Fig. 2.

3.2 Online Semantic Point Fusion

During navigation, the robot constantly obtains new observations while incrementally building a 3D scene representation to predict the next action. However, utilizing 3D scene representation in active learning-based methods is fairly challenging due to two major requirements: 1) memory efficiency and 2) fast updating and querying. In this section, we introduce our proposed online semantic point fusion algorithm to enable fast 3D semantic perception for robot learning while maintaining memory-efficient. Our algorithm leverages the way introduced by a point-based fusion framework Zhang et al. (2020) to organize the 3D points and further devise the semantic fusion and consistency estimation module tailored for navigation policy learning.

3D Sparse Map. Here, we briefly revisit a 3D point fusion algorithm Zhang et al. (2020). Given a sequence of posed color image I_t^c and depth images I_t^d at time step t along with the camera intrinsics, we can obtain the 3D points $p_t \mid p_t = (x_p, y_p, z_p)$ via back-projection. To facilitate fast point querying, the points are online organized in a 3D sparse map $\mathcal{M}_{3D}$. Specifically, we construct a 3D grid and the 3D sparse map $\mathcal{M}_{3D}$ is composed of the occupied 3D blocks $\{\mathcal{B}_k\}$ along with their indices k obtained by a tree-based method Jagadish et al. (2005). Each block $\mathcal{B}_k$ records the points inside it, *i.e.* the points within a given coordinate range:

$$x_p \in [X_k^{min}, X_k^{max}), y_p \in [Y_k^{min}, Y_k^{max}), z_p \in [Z_k^{min}, Z_k^{max}), \tag{1}$$

After constructing the 3D sparse map, we can achieve efficient point searching and neighborhood retrieval for any given 3D point p. However, the points sharing the same 3D block are still unstructured at the per-point level. To obtain the fine-grained relationship of points, we further build a one-level octree $\mathcal{O}_i$ for each point p_i. Specifically, we connect the point with the nearest points in the eight quadrants of the Cartesian coordinate system. Now, given any given point, we can search the nearest points in eight directions and expand the search region as large as we want. Please see Fig. 3 for more detail explanation.

Note that, under online scanning, there are considerable overlaps between consecutive frames. Therefore, we can reuse most of ($\sim 60\%$) the blocks and thus significantly boost the running efficiency. Additionally, we only insert the newly observed points that have a distance (greater than 4 cm) from all existing points, making the points as uniformly distributed as possible. This algorithm for organizing 3D points can run at 15 FPS while requiring reasonable memory resources. More details can be found in the appendix.

Online Semantic Fusion. Considering a sequence of RGBD observations ($I^c_{t=1..N}$ and $I^d_{t=1..N}$), our method first obtains the semantic prediction $S_{2D}(p_i|I^c_t)$ by a pre-trained 2D network Jiang et al. (2018), following existing works Ye et al. (2021); Ramakrishnan et al. (2022). Leveraging our 3D sparse map, we can easily fuse the predictions to lower the errors. For any 3D point, Our 3D sparse map enables us to efficiently 1) find the corresponding pixels cross multi-view observation and 2) search its nearest neighbor points, which enables us to perform the semantic fusion.

We thus propose to online aggregate the multi-view 2D semantic predictions $S_{2D}(p_i|I^c_t)$ using a max-fusion mechanism to obtain the final 3D semantic prediction:

$$S_{3D}(p_i|I^c_{t=1,..,N}) = \text{normalize}(\max(S_{2D}(p_i|I^c_{t=1}), ..., S_{2D}(p_i|I^c_{t=N}))), \tag{2}$$

where $\max$ is performed per semantic class followed by a normalization to linearly scale to 1. Note that, different from Huang et al. (2021); Zhang et al. (2020) that leverage 3D convolution for fusing the semantics, we propose to utilize this simple yet effective max fusion since simply incorporating 3D convolution into such a floor-level or building-level 3D map leads to a formidable computational cost, especially in the context of online reinforcement learning for navigation policy. Also, we find that directly aggregating the 2D semantic prediction in our semantic fusion algorithm already achieves impressive improvement on semantic accuracy with significantly higher memory efficiency and time efficiency. Similar findings have also been reported and exploited in relevant works Chaplot et al. (2021); Grinvald et al. (2019). Moreover, through experiment, we find that the max-fusion demonstrates better performance than Bayesian-fusion McCormac et al. (2017).

Spatial Semantic Consistency. In addition to fusing semantics, our 3D sparse map further enables us to check the spatial semantic consistency among neighboring points, which provides critical information for whether to trust the semantic prediction. We propose to model the spatial semantic consistency C_{3D} of object points by measuring the maximum KL-divergence between a given point and its connected points in octree $\mathcal{O}$. Notably, this consistency will be on-the-fly updated during the navigation. This also showcases that our 3D point-based method is more powerful for perceiving 3D space and enabling a more comprehensive scene understanding for the agent.

3.3 Two-stage Reinforcement Learning

Although our online semantic point fusion algorithm is designed to be very efficient, it still consumes far more time than naive 2D-map-based methods in the context of reinforcement learning for navigation policy. Therefore, to improve the sampling efficiency of reinforcement learning, we disentangle the whole navigation task into two complementary sub-tasks: exploration and verification, with each learning in different discrete action spaces.

Exploration Stage. In the exploration stage, the agent, driven by an exploration policy, attempts to explore the unobserved area where it could access the potential target category object. Existing map-based methods learn the semantic priors from the 2D map and predict a global goal within the current 2D map. These approaches implicitly encode the priors between the object distribution and scene layout, then predict a continuous global Chaplot et al. (2020b) or a discrete goal but with a large action space dimension Georgakis et al. (2022). These methods require extensive training time Chaplot et al. (2021) or complicated data preparation Ramakrishnan et al. (2022). Recently, Luo et al. (2022) proposed a heuristic goal selection strategy, which simply guides the agent following a clockwise varying direction. This strategy avoids repeated forward-backward moving and demonstrates high navigation efficiency. However, it is a sub-optimal design because the heuristic

direction selection strategy does not encode the object category information. Please see Fig.4 for a visual explanation.

Here, to take the best of both worlds, we define an exploration policy $d_t = \pi_e(x_t; \theta_e)$ that predicts a goal direction $d_t \in \mathcal{D}$ that drives the agent to find the target category object as soon as possible, where θ_e indicates the parameters of the policy, $\mathcal{D}$ contains four pre-defined target directions (Fig.4 middle) and , the state x_t is comprised of a coarse 2D top-down map $\mathcal{M}_{2D} \in (C_{2D} \times M \times M)$ and sampled 3D points from 3D sparse map $\mathcal{M}_{3D} \in (C_{3D} \times N)$. For a 2D map, the $M \times M$ denotes the map size, and the C_{2D} is composed of the obstacle map, explored map, and semantic channels. For 3D sparse map $\mathcal{M}_{3D}$, the N indicates the point number (4096), and C_{3D} includes position, spatial semantic consistency, and semantic channels. Here, the coarse (20 cm) 2D map is constructed to give a large perception view of the scenes, and 3D points perform as a fine-grained observation of objects. The experiment shows that the combination demonstrates better performance than any individual representation.

Once the direction is determined, the agent uses a deterministic local planner to plan a path to reach the predicted direction. Note that, our method also builds a fine-grained (5cm) 2D occupancy map for local planning only, which requires a small amount of memory without training. More details are offered in the appendix.

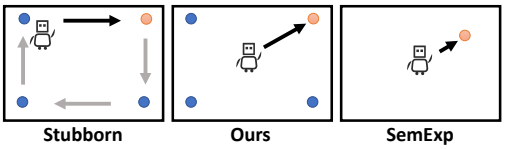

Figure 4: Illustration of exploration policy. (Left) Heuristic direction selection Luo et al. (2022); (Middle) Learning-based direction selection (Ours); (Right) Learning-based dense global goal prediction Chaplot et al. (2020b)

Verification Stage. During navigation, the agent has to decide where and whether to stop when observing objects whose predicted semantic label is the same as the target object. Most works consider tackling this problem by simply setting a hard confidence threshold on the semantic prediction Chaplot et al. (2020b). In this case, if the agent observes a point with a higher predicted probability than the given threshold, the agent will switch its policy to a simple reaching policy that allows it to directly rush to the location of that point. However, this strategy has two major limitations: 1) it relies on a single-point prediction and thus is not robust to wrong semantic predictions with high confidence; 2) different object categories behave differently under a semantic predictor, making the confidence threshold hard to decide. These limitations can lead to numerous mistakes, and what's worse, the agent won't be able to recover from the mistake once it starts marching to the wrong goal.

To tackle these issues, we propose a verification stage that leverages the predicted confidence threshold and spatial semantic consistency. Specifically, we define a policy $a_t = \pi_v(x_t; \theta_v)$ which takes in the online fused 3D points $\mathcal{M}_{3D}$ and target category ID as observation and outputs a threshold-indicating action $s_t \sim \{0, 1..., 10\}$. The actual threshold s_t can be obtained by:

$$\tau_t = \tau_{low} + \frac{(s_t - 5)}{5} \cdot (1 - \tau_{low}), \tag{3}$$

where the τ_{low} is set to 0.75 in our implementation with $\tau_t \in [0.5, 1]$. The predicted threshold is then used to filter out the low confidence points. And for each remaining point, we search its nearest points along the octree $\mathcal{O}$ and label the points with at latest one near point share the same category as the potential goal points. Finally, we choose the points with at least 4 nearest potential points as the final target goal. Note that, the agent will consistently perform semantic point fusion and on-the-fly update the target goal, which can dismiss the mistake when reaching a wrong category object.

Local Planner. Following existing works, we use the Fast Marching Method Sethian (1999) to compute the shortest path from the robot location to the given direction or target goal. The local planner then takes deterministic actions to drive the agent along this shortest path. This could be further improved by changing to a learning model, like Chaplot et al. (2020a); Wijmans et al. (2019).

Rewards. For the exploration policy, we share a similar reward design as Ye et al. (2021); Batra et al. (2020b). The agent receives a sparse success reward $r_{success} = 2.5$, a slack reward $r_{slack} = 10^{-2}$ and an exploration reward $r_{explore}$. The exploration reward is a dense reward, defined by the number of new inserted point n_p^{new} as $r_{explore} = n_p^{new} \times 10^{-3}$. The slack reward and exploration reward encourages the agent to take the most effective direction to the unobserved area. And for the verification policy, we remove the exploration reward.

4 EXPERIMENT

4.1 EXPERIMENT SETUP.

We perform experiments on the Matterport3D (MP3D) Chang et al. (2017) and Gibson Xia et al. (2018) datasets with the Habitat simulator Savva et al. (2019a). Both Gibson and MP3D contain photorealistic 3D reconstructions of real-world environments. For Gibson, we use 25 train /5 val scenes from the Gibson tiny split. And we follow the same setting as in Chaplot et al. (2020b); Ramakrishnan et al. (2022) where we consider 6 goal categories, including *chair*, *couch*, *potted plant*, *bed*, *toilet* and *TV*. For MP3D, we use the standard split of 61 train /11 val scenes with Habitat ObjectNav dataset Savva et al. (2019b), which consists of 21 goal categories (the full list can be found in the appendix). Note that, the depth map and odometry are noise-free from simulation (follow the definition of Batra et al. (2020b)). Estimation of the pose from noisy sensor readings is out of the scope of this work and can be addressed, if necessary, by incorporating off-the-shelf odometry Zhao et al. (2021).

Implementation Details. On MP3D, we use a pre-trained semantic model RedNet Jiang et al. (2018) as Ramakrishnan et al. (2022); Ye et al. (2021). On Gibson, we leverage a Mask R-CNN He et al. (2020), which is trained with COCO dataset Lin et al. (2014). For each frame, we randomly sample 512 points for point-based fusion and use a sliding window to maintain the latest update points(4096) for the learning policy. Moreover, we use PointNet Qi et al. (2017) to obtain the feature of 3D points and a fully convolutional network for the 2D top-down map. During training, we sample actions every 25 steps and use Proximal Policy Optimization (PPO) Schulman et al. (2017) for both exploration and verification tasks. More details can be found in the appendix.

Evaluation Metrics. Following existing works Batra et al. (2020a); Ramakrishnan et al. (2022), we adopt the following evaluation metrics: 1) Success rate: the percentage of successful episodes 2) SPL: success weighted by path length. It measures the efficiency of the agent over oracle path length. 3) Soft SPL: a softer version of SPL measure the progress towards the goal (even with 0 success). 3) DTS: geodesic distance (in m) to the success at the end of the episode. We usually use SPL first to measure the agent's performance, as did in Habitat ChallengeYadav et al. (2022).

Baselines. We consider mainstream baselines in the ObjectNav task. For end-to-end RL methods, we cover DD-PPO Wijmans et al. (2019), Red-Rabiit Ye et al. (2021), THDA Maksymets et al. (2021), and Habiat-web Ramrakhya et al. (2022). For modular based methods, we cover FBE Robotics (1997), ANS Chaplot et al. (2020a), L2M Georgakis et al. (2022), SemExp Chaplot et al. (2020b), Stubborn Luo et al. (2022) and PONI Ramakrishnan et al. (2022). Note that, some works use additional data to improve the performance, *e.g.* Habitat-web leverages human demonstration trajectories, and THDA utilizes data augmentation. It is challenging to compare all the methods fairly. Therefore, we are particularly interested in the three most relevant baselines: SemExp, Stubborn, and PONI. These three methods, performing as a strong baseline, share the same semantic predictor Jiang et al. (2018) as our method.

4.2 RESULTS

Comparison on MP3D and Gibson. We evaluate our approach on MP3D (val) and Gibson (val) in contrast with other baselines, including end-to-end RL(rows 1 - 4) and modular-based methods(rows 5 - 10). Our approach is grouped into modular-based methods. The results are demonstrated in Table.1. On the MP3D dataset, our method is significantly better than all existing baselines in SPL and DTS, and achieves the best success rate among all modular-based methods. Considering three particular methods: SemExp, Stubborn, and PONI, which share the same 2D semantic predictor as ours, we outperform these three on all metrics, clearly performing the superiority of our method among modular-based methods. Moreover, compared to the end-to-end RL-based methods, like Habitat-web Yadav et al. (2022) trained with extra human demonstration, our method still achieves more efficient navigation with 10% higher SPL and competitive success rate. Still, the performance of our method on the success rate could be further improved with a more accurate 2D semantic predictor Liu et al. (2021) and training data Zhou et al. (2018). A qualitative visualization can be found Figure.5. Here, our method online updates the semantic prediction and successfully dismisses the wrong target goal. For more episode qualitative results, please refer to the appendix.

On the Gibson dataset, our method achieves comparable performance to other baselines. However, due to the different 2D semantic predictors for methods, it is unfair to compare the final performance. Here, we provide the results only as a reference.

Table 1: ObjectNav validation results on MP3D and Gibson. Baselines are adapted from Ramakrishnan et al. (2022), Chaplot et al. (2020b). Note that, the L2M, SemExp and Stubborn do not report the results on MP3D validation. We therefore faithfully provide the results, denoted with *, by evaluating the pre-trainedmo model from their open available code.

Method	Gibson (val)			Mappterport 3D (val)		
	SPL(%)↑	Succ.(%)↑	DTS(m)↓	SPL(%)↑	Succ.(%)↑	DTS(m)↓
DD-PPO Wijmans et al. (2019)	10.7	15.0	3.24	1.8	8.0	6.90
Red-Rabbit Ye et al. (2021)	-	-	-	7.9	34.6	-
THAD Maksymets et al. (2021)	-	-	-	11.1	28.4	5.58
Habitat-Web Ramrakhya et al. (2022)	-	-	-	10.2	**35.4**	-
FBE Robotics (1997)	28.3	64.3	1.78	7.2	22.7	6.70
ANS Chaplot et al. (2020a)	34.9	67.1	1.66	9.2	27.3	5.80
L2M Georgakis et al. (2022)	-	-	-	11.0	32.1	5.12
SemExp* Chaplot et al. (2020b)	39.6	71.7	1.39	10.9	28.3	6.06
Stubborn* Luo et al. (2022)	-	-	-	13.5	31.2	5.01
PONI Ramakrishnan et al. (2022)	41.0	**73.6**	1.25	12.1	31.8	5.10
Ours	**41.7**	72.5	**1.21**	**14.8**	32.6	**4.04**

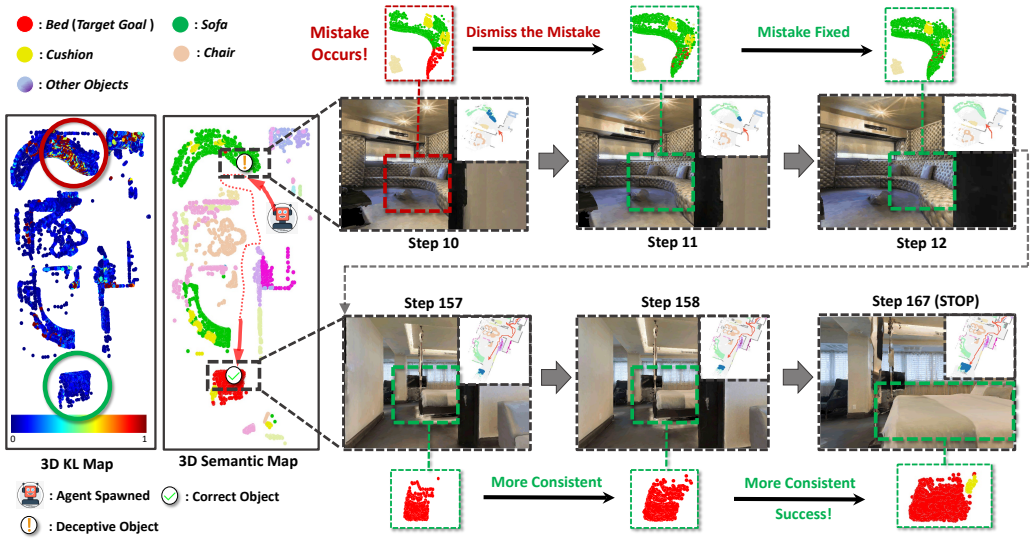

Figure 5: Qualitative results of navigation using the proposed method. We visualize an episode on Matterport3D (val), along with a top-down semantic map and KL map. The robot is expected to find a bed. At step 10, the robot obtain a partial observation of a sofa, which is mistakenly recognized as a bed. And during the approach to the sofa, the multi-view observation improve the semantic prediction and successfully dismiss the mistake. Then the robot continue to explore the environment until step=157, when the robot obtain new partial observation of an object which could be the bed. As the robots moving to the target, the points are more complete and the semantic prediction is tend to be bed. And for KL map, we can find that the false target goal (red circle) has large inconsistency contrasted to the correct object(green circle).

Comparison on Exploration Policy. We conduct an experiment on MP3D to validate the efficiency of our exploration policy. To remove the effect of the 2D semantic predictor and stop strategy, all competitors share the same semantic predictor and the verification policy proposed in Sec3.3. The results is reported in Table 2. Our exploration outperforms the mainstream existing methods Ramakrishnan et al. (2022); Luo et al. (2022), including learning-based and heuristic strategies. Moreover, we find that learn-

Table 2: Comparison on exploration policy; G. denotes Global Goal; D. denotes Direction.

Method	SPL(%)	Succ.(%)	DTS(m)
Learn Continuous G.	11.2	29.2	5.063
Learn Grid G.	13.0	31.5	4.944
Learn 8 D.	13.6	31.7	4.692
Heuristic. 4 D.	14.4	32.1	5.024
Learn 4 D. (Ours)	**14.8**	**32.6**	**4.036**

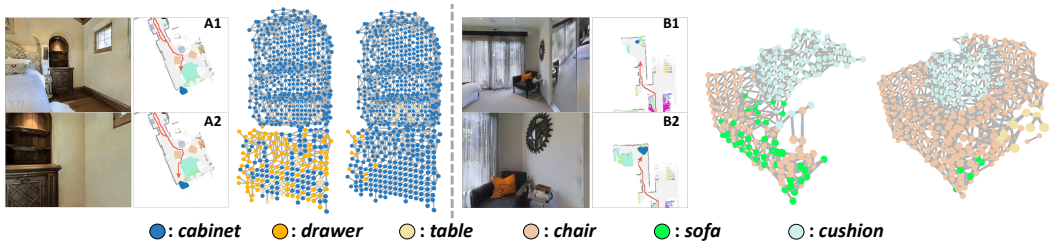

Figure 6: Visualization of the results of online semantic point fusion. The fusion algorithm can give more accurate and complete points to guide the agent.

ing 4 directions demonstrates better performance than 8. This proves that the 4 corner directions can most efficiently drive the agent to explore the environment while benefiting from smaller action space.

Comparison on Verification Policy. Another critical challenge in OjectNav is how to properly stop the robot. Therefore, We evaluate our verification policy on MP3D along with other stop strategies, including a 2D map-based policy adopted in Chaplot et al. (2020b); Ramakrishnan et al. (2022) and 3D points-based methods proposed by our approach. The results are shown in Table. 3. We observe a performance improvement (rows 1 - 4) by leveraging the proposed semantic point fusion algorithm. It can be concluded that the multi-view observations provide more accurate semantic prediction, which effectively reduces false positive prediction (see examples in Figure.6). Moreover, instead of setting a hard confidence threshold to verify the objects, our method demonstrates better performance benefiting from both dynamic threshold prediction and spatial semantic consistency.

Ablation Study. We also perform an ablation study to verify the effectiveness of different components of our method. The results are demonstrated in Table.4. From rows 1-2, we find that only using the 3D points for exploration does not outperform the 2D map. The reason is that the sampled 3D points suffer from a shorter perception field than a 2D map. The cooperation of the 2D top-down map and 3D points (row 4) shows significant improvement by incorporating extensive scene perception (in 2D) and fine-grained object perception (in 3D). Moreover, rows (3-4) and (4-5) proved the effectiveness of leveraging consistency and verification policy, respectively.

Table 3: Comparison of verification policy.

Method	Type Repr.	Thre.	SPL(%)	Succ.(%)	DTS(m)
Deterministic	2D	0.75	13.0	29.7	5.168
	2D	0.85	12.8	30.1	5.151
	3D	0.75	13.7	32.3	4.179
	3D	0.85	13.5	31.5	4.386
Learning (Ours)	3D	-	**14.8**	**32.6**	**4.036**

Table 4: Ablation study of proposed method.

2D map	3D points Pos.	KL div.	V. Policy	SPL(%)	Succ.(%)	DTS(m)
✓				13.1	31.4	4.971
	✓	✓		12.9	30.8	4.931
✓	✓			13.5	32.4	4.810
✓	✓	✓		13.7	32.3	4.179
✓	✓	✓	✓	**14.8**	**32.6**	**4.036**

Analysis of Computation Cost. Our online semantic point fusion algorithm is extremely memory efficient, which requires about $0.5GB$ for one scene, and can perform online fusion at a rate of 15 FPS. Moreover, our two-stage reinforcement learning framework requires only 48 GPU hours on MP3D to achieve the SOTA performance among all modular-based methods. This is comparable to supervised learning modular-based methods Ramakrishnan et al. (2022) and significantly faster (5-28x) than existing reinforcement learning based methods Chaplot et al. (2020b); Ye et al. (2021).

5 Conclusion

We present a novel two-stage goal navigation framework for the object goal navigation task powered by a semantic point fusion algorithm. By fusing the 3D points from multi-view observation, our method can on-the-fly update the semantic prediction and spatial consistency, enabling a comprehensive 3D scene understanding. Furthermore, to make training 3D-aware agents more efficient, we disentangle the goal navigation task into two complementary sub-tasks, exploration and verification, with each learning in different discrete action space. Finally, the results clearly demonstrate the superiority of our 3D-aware navigation method. In the future, we would like to exploit the 3D-aware agent in other embodied AI tasks, *e.g.* mobile manipulation.

Ethics Statement. We believe this work has a broader impact by enabling more intelligent autonomous robots with 3D scene understanding to navigate in unseen environments. The proposed 3D-aware robots could be further used in future embodied AI tasks, such as housekeeping and nursing for disabled populations. However, our method also has technical limitations, which can yield social consequences. First, the data for training our agent mostly comes from North America and Europe. It could lead to safety risks when adopted in out-of-distribution environments. Second, our method relies on robust odometry for accurate pose tracking. This could be intractable under challenging environments, such as low-lighting conditions or fast camera shaking. These issues could be resolved by involving more diverse data and using multi-modal odometry.

Reproducibility Statement. The source code is attached in the supplementary, and the hyperparameter can be directly applied. We further provide a README to guide the installation, training, and evaluation.

REFERENCES

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*, abs/2006.13171, 2020a.

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. In *arXiv:2006.13171*, 2020b.

Tommaso Campari, Paolo Eccher, Luciano Serafini, and Lamberto Ballan. Exploiting scene-specific features for object goal navigation. In *European Conference on Computer Vision*, pp. 406–421. Springer, 2020.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020a.

Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020b.

Devendra Singh Chaplot, Murtaza Dalal, Saurabh Gupta, Jitendra Malik, and Russ R Salakhutdinov. Seal: Self-supervised embodied active learning using exploration and 3d consistency. *Advances in Neural Information Processing Systems*, 34:13086–13098, 2021.

Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *ECCV*, 2018.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Georgios Georgakis, Yimeng Li, and Jana Kosecka. Simultaneous mapping and target driven navigation. *ArXiv*, abs/1911.07980, 2019.

Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. Learning to map for active semantic goal navigation. In *International Conference on Learning Representations (ICLR)*, 2022.

Margarita Grinvald, Fadri Furrer, Tonci Novkovic, Jen Jen Chung, Cesar Cadena, Roland Siegwart, and Juan Nieto. Volumetric instance-aware semantic mapping and 3d object discovery. *IEEE Robotics and Automation Letters*, 4(3):3037–3044, 2019.

Saurabh Gupta, Varun Tolani, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. *International Journal of Computer Vision*, 128:1311–1330, 2017.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397, 2020.

Shi-Sheng Huang, Ze-Yu Ma, Tai-Jiang Mu, Hongbo Fu, and Shi-Min Hu. Supervoxel convolution for online 3d semantic segmentation. *ACM Transactions on Graphics (TOG)*, 40(3):1–15, 2021.

Hosagrahar V Jagadish, Beng Chin Ooi, Kian-Lee Tan, Cui Yu, and Rui Zhang. idistance: An adaptive b+-tree based indexing method for nearest neighbor search. *ACM Transactions on Database Systems (TODS)*, 30(2):364–397, 2005.

Jindong Jiang, Lunan Zheng, Fei Luo, and Zhijun Zhang. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. *arXiv preprint arXiv:1806.01054*, 2018.

Ilya Kostrikov. Pytorch implementations of reinforcement learning algorithms. `https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail`, 2018.

Cheng Lin, Tingxiang Fan, Wenping Wang, and Matthias Nießner. Modeling 3d shapes by reinforcement learning. In *ECCV*, 2020.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Leyao Liu, Tian Zheng, Yun-Jou Lin, Kai Ni, and Lu Fang. Ins-conv: Incremental sparse convolution for online 3d segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18975–18984, 2022.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. *ArXiv*, abs/2111.09883, 2021.

Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A strong baseline for indoor object navigation. *arXiv preprint arXiv:2203.07359*, 2022.

Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. Thda: Treasure hunt data augmentation for semantic navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15374–15383, 2021.

John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pp. 4628–4635. IEEE, 2017.

Arsalan Mousavian, Alexander Toshev, Marek Fišer, Jana Košecká, Ayzaan Wahid, and James Davidson. Visual representations for semantic target driven navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8846–8852. IEEE, 2019.

Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4205–4212. IEEE, 2019.

Alexey Nekrasov, Jonas Schult, Or Litany, B. Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. *2021 International Conference on 3D Vision (3DV)*, pp. 116–125, 2021.

Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. *ArXiv*, abs/1702.08360, 2018.

C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017.

Yiding Qiu, Anwesan Pal, and Henrik I Christensen. Learning hierarchical relationships for object-goal navigation. *arXiv preprint arXiv:2003.06749*, 2020.

Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18890–18900, 2022.

Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5173–5183, 2022.

Ieee Robotics. Proceedings 1997 ieee international symposium on computational intelligence in robotics and automation cira'97 - towards new computational principles for robotics and automation, july 10-11, 1997, monterey, california, usa. In *CIRA*, 1997.

Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. *arXiv preprint arXiv:2002.06289*, 2020.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019a.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9339–9347, 2019b.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

James A. Sethian. Fast marching methods. *SIAM Rev.*, 41:199–235, 1999.

Thang Vu, Kookhoi Kim, Tung Minh Luu, Xuan Thanh Nguyen, and Chang-Dong Yoo. Softgroup for 3d instance segmentation on point clouds. *ArXiv*, abs/2203.01509, 2022.

Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3961–3970, 2020.

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019.

Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7515–7525, 2021.

Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9068–9079, 2018.

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.

Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018.

Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectgoal navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16117–16126, 2021.

Jiazhao Zhang, Chenyang Zhu, Lintao Zheng, and Kai Xu. Fusion-aware point convolution for online semantic 3d scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4534–4543, 2020.

Xiaoming Zhao, Harsh Agrawal, Dhruv Batra, and Alexander G. Schwing. The surprising effectiveness of visual odometry techniques for embodied pointgoal navigation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16107–16116, 2021.

Lintao Zheng, Chenyang Zhu, Jiazhao Zhang, Hang Zhao, Hui Huang, Matthias Nießner, and Kai Xu. Active scene understanding via online semantic reconstruction. *Computer Graphics Forum*, 38, 2019.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2018.

Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12689–12699, 2021.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Kumar Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3357–3364, 2017.

A APPENDIX

A.1 MODEL DETAILS

Inputs. Three types of inputs are used in our policy network: 1) The sampled 3D points (4096) from the 3D coordinate map. Due to the progressively increased size of the 3D coordinate map, it is not practical to leverage all the fused points. We instead only consider the latest observing points, which are recently updated. Specifically, we use a sliding window to add the new observed/updated points and remove non-observed points. These sampled points significantly reduce the memory resource requirements but still offer significant spatial and temporal information. 2) A 2D top-down map is also constructed to model the scene layout and semantics. We follow Chaplot et al. (2020a) to construct the 2D map at a coarse level (20cm) and project the fused 3D point semantic to obtain the semantic channels. 3) Extra information, including the number of steps, the discretized orientation of the agent, and the target category ID. Using extra information can give an explicit environment state to benefit the agent, which has been proved in Chaplot et al. (2020a;b).

Exploration and Verification Policy. Our exploration policy takes the sampled points, 2D top-down map, and extra information as inputs and predicts a discrete direction to navigate the robot (Fig.7). Specifically, the policy uses a PointNet Qi et al. (2017) to encode the 3D points information (xyz, semantics, and kL divergence) to obtain a global feature (256D). The 2D top-down map will be passed to a fully convolutional network and been flatten to a feature vector (256D). And the extra information is embedded into a feature vector (24D). Note that, the processing of the 2D top-down map and extra information share the same idea as Chaplot et al. (2020b). Finally, the three feature vectors are concatenated, followed by an MLP. The network pipeline is based on the PPO implementation Kostrikov (2018). The network architecture of the verification policy is almost the same as the exploration policy, by removing the 2D top-down map branch from the input (Fig.8).

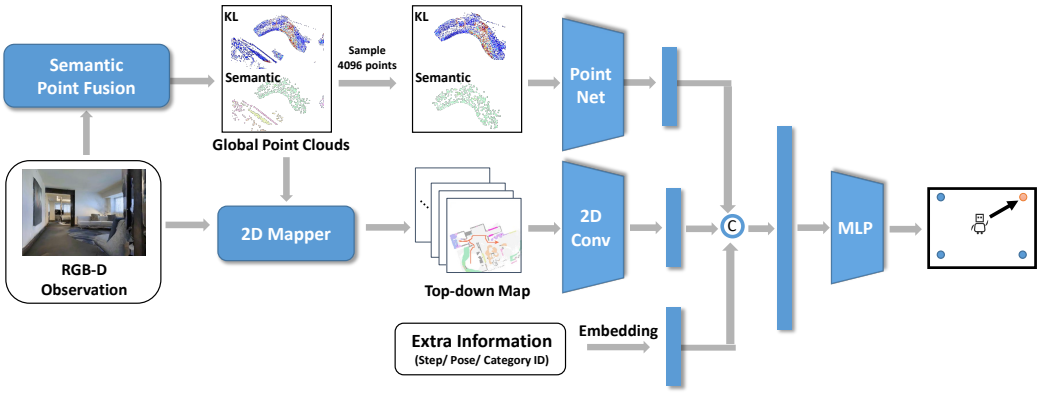

Figure 7: Network architecture of our exploration policy

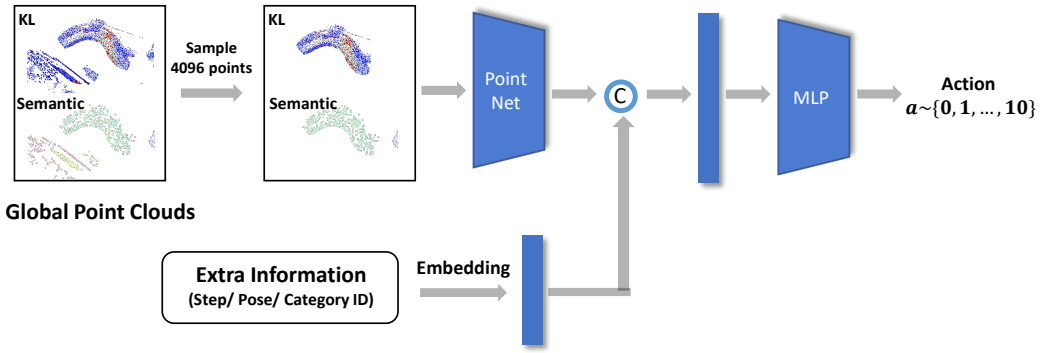

Figure 8: Network architecture of our verification policy

A.2 IMPLEMENTATION DETAILS

Online Semantic Point Fusion. For each frame, we randomly sample 512 points and use a pre-trained 2D semantic predictor to obtain the corresponding semantic prediction. If a point could find any close point with a distance smaller than 4cm, it would be considered a new point and be added to the 3D coordinate map. Otherwise, its semantic prediction will be fused to the closest point in 3D. Also, for each point, we only connect the points with distance $\in [4cm, 15cm]$.

Goal Changing Strategy. During the evaluation, to make the comparison fair for other direction selection strategies Luo et al. (2022), we leverage the same direction-changing mechanism, which updates the goal only when it gets trapped or reaches a step-threshold.

A.3 EXPERIMENT DETAILS

Here, we provide additional descriptions of the experiments to support the main paper.

MP3D ObjectNav Dataset. The MP3D ObjectNav dataset from the Habitat challenge consists of 21 object categories: 'chair', 'table', 'picture', 'cabinet', 'cushion', 'sofa', 'bed', 'chest of drawers', 'plant', 'sink', 'toilet', 'stool', 'towel', 'tv monitor', 'shower', 'bathtub', 'counter', 'fireplace', 'gym equipment', 'seating', and 'clothes'. The train / val splits consist of 263,2422 / 2,195 episodes from 61 / 11 MP3D scenes.

Comparision on Exploration Policy. *Learn Continuous G.*: learn to predict a continuous global goal Batra et al. (2020b); *Learn Grid G.*: Learn to predict a dense discrete global goal Georgakis et al. (2022); *Heuristic 4 D.*: A heuristic direction selection strategy Luo et al. (2022). *Learn 4/8 D.*: learn to predict a direction to navigate the robots. The eight directions of "Learn 8 D." are equally distributed in the 2D plane.

A.4 ADDITIONAL EXPERIMENTS

Comparison with L2M. L2M Georgakis et al. (2022) provides a self-made dataset which consist of 781 episodes from 10 MP3D (val) scenes. Following the setting of the L2M dataset, we evaluate our method on L2M Dataset, and the results are reported in Table.5.

Table 5: Comparison on L2M dataset. The results of L2M and SemExp are quoted form Georgakis et al. (2022)

Method	SPL(%)	SoftSPL(%)	Succ.(%)	DTS(m)
SemExp Chaplot et al. (2020b)	17.9	-	30.1	4.782
L2M Georgakis et al. (2022)	17.0	22.1	39.1	3.373
Our	**21.2**	**30.5**	**40.2**	**3.278**

A.5 LIMITATIONS

We have elaborated the advantages of our approach in Sec.3 and Sec.4 of the main paper. Yet there are some limitations we would like to acknowledge:

First, there exists a noticeable gap in the number of points among different categories. Large objects (e.g., sofa) may have much more points (up to 100x) than small objects (e.g., plants). Therefore, the large object may contribute more to the final prediction, leading to a performance drop in searching of small objects. One possible solution to alleviate this problem is to define a center key-point Vu et al. (2022) to weigh the unbalanced points.

Second, during training, our agent on-the-fly aggregates numerous scene observations. It can be quite a waste to discard these scene contexts right after the end of each episode. The training efficiency could be further improved if we can transfer the RL learning task into a supervised learning task, as did in Ramakrishnan et al. (2022)(PONI).

A.6 ADDITIONAL VISUALIZATIONS

Finally, we provide some additional visualizations of selected episodes in Fig.9 and Fig.10. For each episode, we display first-person observation(row1), predicted 2D Semantic Map(row2) and structured 3D points(row3) of some specific steps in sequence, which either demonstrate the agent dismissed mistakes encountered halfway or it became more and more confident when proceeding towards the true target goal.

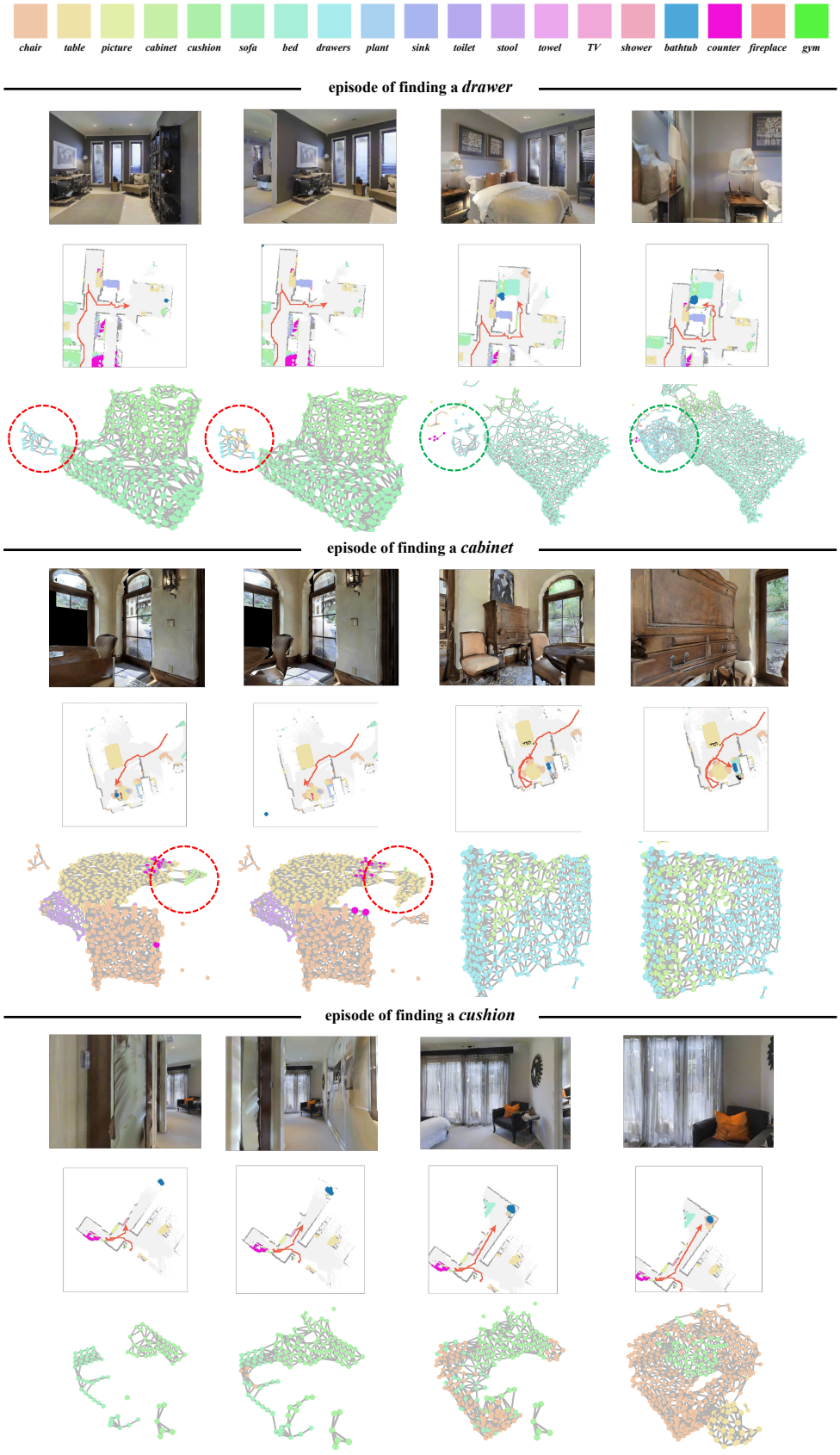

Figure 9: EPS Page1.

Figure 10: EPS Page2.