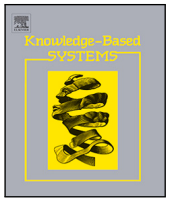
Knowledge-Based Systems 304 (2024) 112446

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Semantic Environment Atlas for Object-Goal Navigation

Nuri Kim, Jeongho Park, Mineui Hong, Songhwai Oh *

Department of Electrical and Computer Engineering and ASRI, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 08826, Republic of Korea

ARTICLE INFO

Keywords:
Semantic navigation
Embodied agents
Autonomous navigation

ABSTRACT

In this paper, we introduce the Semantic Environment Atlas (SEA), a novel mapping approach designed to enhance visual navigation capabilities of embodied agents. The SEA utilizes semantic graph maps that intricately delineate the relationships between places and objects, thereby enriching the navigational context. These maps are constructed from image observations and capture visual landmarks as sparsely encoded nodes within the environment. The SEA integrates multiple semantic maps from various environments, retaining a memory of place-object relationships, which proves invaluable for tasks such as visual localization and navigation. We developed navigation frameworks that effectively leverage the SEA, and we evaluated these frameworks through visual localization and object-goal navigation tasks. Our SEA-based localization framework significantly outperforms existing methods, accurately identifying locations from single query images. Experimental results in Habitat Savva et al. (2019)scenarios show that our method not only achieves a success rate of 39.0%—an improvement of 12.4% over the current state-of-the-art—but also maintains robustness under noisy odometry and actuation conditions, all while keeping computational costs low.

1. Introduction

Embodied AI technologies, which are becoming increasingly ubiquitous in modern life, are proving integral to various applications, including delivery robots, household chore robots, and self-driving cars. The pivotal success factor in this field has been the development of intelligent agents that use RGB sensors to interpret semantic knowledge, particularly through learning-based methods such as reinforcement learning (RL) [1–22]. These methods, while powerful, introduce a significant challenge: high computational costs.

Addressing this challenge, this paper introduces the Semantic Environment Atlas (SEA), a novel map type. The SEA is specifically designed to tackle visual localization and navigation tasks in a computationally efficient manner. The SEA sets itself apart with three distinctive characteristics that collectively enable successful visual navigation.

The first distinctive characteristic of the SEA is its *robust* navigation performance against sensor noise. Sensor noise is a common problem in navigation tasks, which tends to accumulate during sequential decision-making processes. Traditional approaches have tried to mitigate this issue through loop closure, but such solutions are challenging for deep learning-based methods that lack state-space-based noise filtering. Consequently, current navigation methods [14,15,20,23] often assume a noiseless pose sensor—an unrealistic premise in real-world scenarios. In contrast, our method leverages semantic knowledge, enabling it to navigate robustly even with noisy sensors.

The second distinctive property of the SEA is its ability to *localize* the current position using semantic knowledge. This capability addresses a key challenge: predicting an object's position with a partially observed map. While recent work [16,24,25] has integrated graph-based priors into the metric map to counter this issue, our method takes a step further by incorporating additional semantic knowledge, such as the relationships between objects and places, thereby bolstering localization performance.

The third and final property of the SEA is its *adaptability*. Unlike recent methods [14,15] which do not update upon environmental changes, the SEA is designed to self-update based on these changes. This adaptive quality permits navigation agents to adjust their destinations and explore alternative target locations if the initial object search is unsuccessful.

The SEA is constructed using semantic graph maps, which incorporate both place-object and place-place relationships. An agent uses the place-object relationship to pinpoint the target location where an object is most likely to be found. To reach this target, the agent leverages place relationships to determine the optimal semantic path. For local navigation, the agent identifies subgoal candidates based on current object observations and chooses the subgoal with the highest reachability to the target. Our method's reliance on semantic path planning eliminates the need for a global pose sensor, thus enhancing

* Corresponding author.
E-mail addresses: nuri.kim@rllab.snu.ac.kr (N. Kim), jeongho.park@rllab.snu.ac.kr (J. Park), mineui.hong@rllab.snu.ac.kr (M. Hong), songhwai@snu.ac.kr (S. Oh).

https://doi.org/10.1016/j.knosys.2024.112446
Received 19 May 2024; Received in revised form 15 August 2024; Accepted 28 August 2024
Available online 5 September 2024

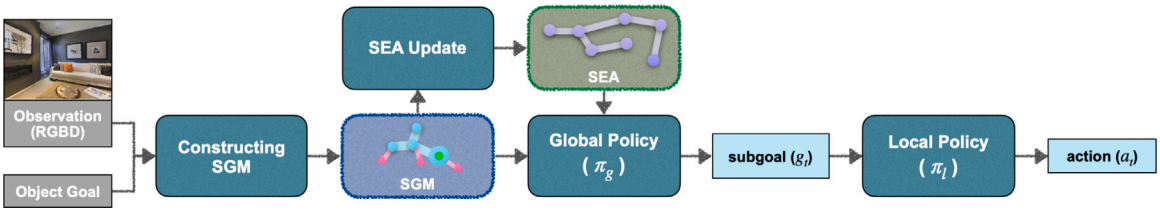


Fig. 1. Overview of semantic environmental atlas (SEA). The semantic graph map (SGM) is updated using visual observations. Then, the place relationship and place-object connections across environments are updated using multiple semantic graph maps collected from different environments. A global policy samples a subgoal g_t, which is reachable and most likely to be near to the target place. A local policy generates navigational actions to reach the subgoal.

robustness against noisy odometry sensors. Additionally, we implement relation updates in new environments since the semantic structure can vary significantly from one environment to another. These updates allow the place graph to adapt, ensuring the most effective semantic path is selected in new settings.

The proposed method is evaluated using MP3D [26] for object goal navigation. The experimental study demonstrated that our navigation framework, by leveraging the SEA, achieved a success rate of 39.0%. This result marks a substantial 12.4% improvement over the current state-of-the-art.

2. Related work

Visual navigation without any map. As a policy network, a recurrent neural network (RNN) is a simple method to make an implicit semantic prior [19,21,27]. DDPPO [27] has a vanilla RL policy with a CNN backbone followed by an LSTM as a policy function. RedRabbit [21] augments DDPPO with multiple auxiliary tasks, such as predicting agent dynamics, environment states, and map coverage with ObjectNav. Treasure Hunt Data Augmentation (THDA) [19] improves the RL reward and model inputs, which result in better generalization to new scenes. Since an RNN has a difficulty of backpropagating a long sequence, an RNN can be replaced with an explicit structure [14–16, 20,22–24,28].

Visual navigation with a metric map. Spatial metric map-based RL methods [14,15,20,23] propose independent modules for semantic mapping, high-level semantic exploration, and low-level navigation. The semantic exploration module is learned through RL, yet it is more sample-efficient and generalizes better than end-to-end RL. Active Neural SLAM (ANS) [14] has a hierarchical structure to explore an environment: global and local policies. The global policy constructs a top-down 2D map and estimates a global goal. Given the global goal from the global policy module, a local policy module plans a path to the goal using a simple local navigation algorithm. Semantic exploration [15] is a study that extends ANS. The metric map does not only represent obstacles but draws a semantic map and uses it for navigation to improve performance. This method implicitly learns semantic information for navigation. PONI [20] reduced computational costs in visual navigation by proposing non-interactive learning. Additionally, it improved the navigation performance by learning the encoder by calculating the probability that there is a space or an object beyond the frontier boundary of the current map and then moving to the boundary where the object is likely placed. However, since this method is trained using a top-down map, it is greatly affected by the pose sensor.

Visual navigation with a graph map. Our work proposes a method to collect semantic priors and use it for navigation. Several works have employed semantic priors into a graph to enhance semantic reasoning in visual navigation [16,22,24]. Wu et al. [22] tackle the room navigation task using room relationship, while it does not consider the relationship between a room and an object. Zhang et al. [24] divide a room into several zones to find an object and find the reachability between these zones for navigation. However, the connection between an object and a zone is ambiguous. For example, a bed can exist in any zone in a bedroom. Campari et al. [16] improve performance by building an abstract model in addition to the existing metric map-based methods. Here, the abstract model comprises nodes composed of images and objects, and the connection between nodes is an action taken to navigate between two places. However, actions for moving from one place to another could differ depending on the structure of houses. For example, in one house, the bedroom may be to the right of the living room, and in another, the bedroom may be to the left. Therefore, the structure of the environment is hard to be expressed with the abstract model.

The proposed method collects relationships between place clusters and objects using a sequence of observations and uses them in a new environment for navigation.

3. Proposed method

3.1. Problem statement

In a given unknown environment, an agent is tasked with traveling to an object specified by its category name (e.g., chair) (see Fig. 1). At the start of each episode ($t = 0$), the agent is placed at a random navigable position within the environment. The agent is equipped with a 640×480 RGB-D sensor (s_t^d) and a 512×128 panoramic RGB sensor (s_t^p), along with the goal category (O_{goal}) for the current time step. The panoramic RGB sensor (s_t^p) is specifically used for constructing the semantic graph map. It is important to note that a pose sensor is employed only in the local policy, and global pose sensor readings are not used in this work. The agent can perform actions $a_t \sim \mathcal{A}$, where $\mathcal{A}$ includes moving forward (0.4 m), turning left (30°), turning right (30°), and stopping. To complete the task, the agent must press the stop button once it is within $d_s = 1.0$ m of the target. The episode concludes either when the agent stops or when the time budget of $T = 500$ steps is exceeded.

3.2. Semantic graph map

Inspired by the topological graph map approach outlined in [17], we construct semantic graph maps, E_t, for navigation in unknown environments, as depicted in Fig. 2. At each time point t, the semantic graph map includes three types of nodes—place nodes ($\mathcal{V}_{\text{place}}$), image nodes ($\mathcal{V}_{\text{im}}$), and object nodes ($\mathcal{V}_{\text{ob}}$)—and corresponding edges: $\mathcal{E}_{\text{im}}$, $\mathcal{E}_{\text{io}}$, and $\mathcal{E}_{\text{pi}}$. Each place node, represented as P_i, connects to image nodes with an affinity matrix $\mathbf{A}_{\text{im}} \in \mathbb{R}^{N_i \times N_i}$ indicating the relational strengths. The object nodes, x_i where $x_i \in \mathbb{R}^{1 \times D_o}$, are similarly linked to image nodes with an affinity matrix $\mathbf{A}_{\text{io}} \in \mathbb{R}^{N_i \times N_o}$. The edges $\mathcal{E}_{\text{pi}}$ connect place and image nodes with an affinity matrix $\mathbf{A}_{\text{pi}} \in \mathbb{R}^{N_p \times N_i}$, illustrating the relationships between places and images. The affinity matrices are computed using a multi-layer perceptron (MLP) network, which processes the features of nodes to output a scalar similarity value. The semantic graph map is constructed incrementally as the agent navigates, with the graph at time t being a subset of the graph at time $t+1$. This dynamic mapping allows the agent to reason about and navigate through the relationships among objects, images, and places towards the designated goal.

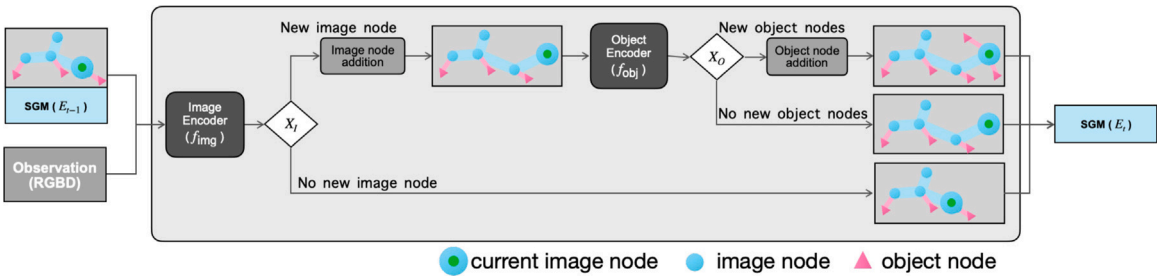


Fig. 2. Construction of semantic graph map. By integrating the current observation and the previous semantic graph map (SGM; E_{t-1}), the graph map is updated. If it is discovered that the current location differs from the previous location, an image node is added to the graph. Similarly, object nodes are added to the graph when previously undetected objects are detected.

Place graph. In visual navigation, accurately identifying semantic places, such as living rooms and bedrooms, is crucial. To address this challenge, room navigation methods [22,29] employ a place recognition algorithm. However, some places can be ambiguous and difficult to classify distinctly. To overcome this issue, a clustering method [30] is applied to train a place encoder, f_{place}, which groups similar features across similar places. This encoder takes as input image features, object features, and object categories to extract place information, defined as $v_t = f_{\text{place}}(s_t^p, o_i^{\text{f}}, o_i^{\text{cat}})$. Here, o_i represents objects detected from a panoramic RGB sensor s_t^p; o_i^{f} is a feature vector of o_i; and o_i^{cat} is the object's category. Using a panoramic RGB sensor is advantageous because the recognition of the place is invariant to camera rotation. To bring images with similar semantic meanings, such as those from a bedroom, closer together in the metric space, a contrastive loss is employed. The loss function for training the place encoder with a batch of B images is formulated as follows:

$$\mathcal{L}_{Place} = \sum_{i=1}^{B} -\log \frac{\exp(v_i \cdot v_i'/\zeta)}{\sum_{j=0}^{r} \exp(v_i \cdot v_j'/\zeta)}, \tag{1}$$

where v_i is the query embedding and v_i' are positive place embeddings for location i, sampled from the same place, and v_j' includes one positive embedding and r negative embeddings from different places, with ζ acting as a temperature hyper-parameter. Positive samples are drawn using ground truth place information which includes labels for ambiguous places such as 'other room'. To handle this, we utilize eight specific room labels, described in Section 4.2. We modify the contrastive loss by replacing the positive sample with images generated by randomly rotating the query image, a method we denote as $\mathcal{L}_{\text{Near}}$. This approach aims to bring images from nearby locations closer together in the metric space. Subsequently, we apply the K-means clustering algorithm to cluster features, resulting in a set $\mathbb{P} = \{P_1, \ldots, P_{N_p}\}$. These clustering results are then used as the ground truth for the clustering loss. Metric learning involves iteratively combining all losses, $\mathcal{L}_{met} = \mathcal{L}_{\text{place}} + \mathcal{L}_{\text{Near}} + \mathcal{L}_{\text{Cluster}}$, with the K-means clustering process.

Image graph. An image encoder [18], represented as $i_t = f_{\text{img}}(s_t^p)$, is crucial for assessing image similarity to determine the novelty of nodes in the semantic graph map. When the agent moves to a new location, it evaluates the similarity between the current and previous image nodes using a cosine similarity function, $\text{sim}(\cdot,\cdot)$. If this similarity, $\text{sim}(i_{t-1}, i_t)$, drops below the threshold $\theta_{\text{im}} = 0.8$, the system checks if the observed image node already exists in the graph. If not, indicating no similar existing image nodes, the node is considered new (i_t) and connected to the previous image node (i_{t-1}). Conversely, if a similar node is found, it is updated with the new image and also linked to i_{t-1}. Image graph construction allows the system to capture and represent the spatial relationships between different places and objects within an environment. This spatial representation is crucial for tasks that require an understanding of the layout of an environment, such as navigation and path planning. The adjacency matrix of image nodes, $\mathbf{A}_{\text{im}} \in \mathbb{R}^{N_i \times N_i}$, is a binary matrix that records these connections. Additionally, when a new image node is detected, it and the place cluster associated with the image node are linked, setting $\mathbf{A}_{\text{pi}}[i, j] = 1$ for the ith place and jth image, reinforcing the semantic links.

Object graph. Objects are encoded using an object encoder, $x_i = f_{\text{obj}}(s_t^p, o_i^{\text{f}}, o_i^{\text{cat}})$, which uses contrastive learning to recognize objects as the same even when viewed from different perspectives [17]. Here, o_i represents an object detected from s_t^p using MaskRCNN [31]. The graph update module assesses whether these objects are already present in the graph. If the similarity between the detected object and an existing object in the graph, $\text{sim}(x_i, x_j)$ is greater than $\theta_{\text{o}} = 0.8$ and their categories match ($o_i^{\text{cat}} = o_j^{\text{cat}}$), the object is considered the same. If the detected object is not in the graph, it is added as a new object node and linked to the current image node (i_t), with $\mathbf{A}_{\text{io}}[i, j] = 1$ indicating the connection between the ith image node and the jth object node. Conversely, if the object is already in the graph but the detected object has a higher detection score, the existing object node is updated to reflect the new detection.

3.3. Localization

In this study, we test the localization function, F_{loc}, to demonstrate how semantic knowledge can enhance localization accuracy. The input to this function includes the current semantic graph map, $\mathcal{G}_t$, along with source and target images. We employ Graph Neural Networks (GNNs) to encode the graph data effectively and Transformer [32] decoder networks to extract relevant localization information. Specifically, F_{loc} utilizes these images to identify corresponding nodes within the graph. It then estimates the distance between these nodes, thereby facilitating precise localization based on semantic relationships captured in the graph.

3.4. Semantic Environment Atlas

We propose a Semantic Environment Atlas (SEA) that synthesizes semantic graph maps collected from various environments into a unified structure, rather than merely compiling individual maps. This integration facilitates a deeper and more comprehensive understanding of the environments and their interrelationships. The SEA, denoted as $S_t = \{\mathbf{\Gamma}, \mathbf{R}\}$, comprises two key components: a place reachability matrix ($\mathbf{\Gamma}$) and a place-object connection matrix ($\mathbf{R}$). The place reachability matrix ($\mathbf{\Gamma}$) defines the accessibility between different places, indicating possible paths and their navigability. Meanwhile, the place-object connection matrix ($\mathbf{R}$) details the associations between various places and the objects found within them, providing crucial contextual information that enhances navigational decisions and spatial reasoning.

Place-place relationship. A node within the place graph represents the centroid of a place cluster, and connections between nodes signify the reachability between places. This reachability is derived from semantic graph maps collected across various training environments. If two connected image nodes, i_a and i_b, belong to different place clusters, P_a and P_b, it is inferred that the clusters are reachable. A

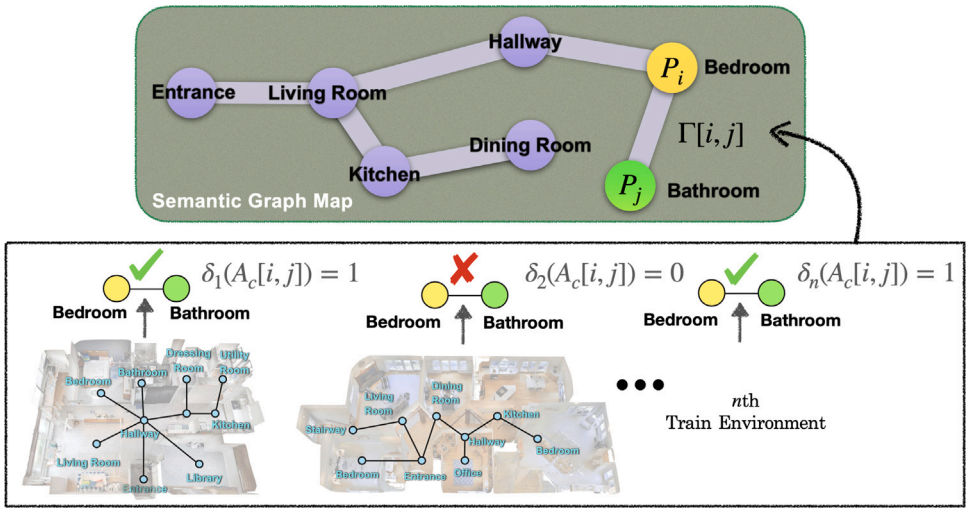


Fig. 3. Formation of Semantic Graph Map from Training Environments. The figure illustrates the process of forming a semantic graph map from multiple training environments. In each training environment, it checks to see whether there is a pair of place clusters. If there a connection between the pair of places, the reachability is set to one; otherwise, zero.

connection between clusters in any scene sets the cluster reachability to one; otherwise, it is zero. This procedure is repeated across all training scenes, and the average value is taken as the final measure of reachability between places. The formation of the semantic graph map from episodic graphs in training environments can be shown in Fig. 3. The top section of the figure shows a semantic graph map where nodes represent different rooms (e.g., Entrance, Living Room, Kitchen) and edges represent the connectivity between them. For example, the edge $\Gamma[i,j]$ connects nodes P_i (Bedroom) and P_j (Bathroom), indicating a valid connection in the graph. In the bottom section of the figure, n floor plan layouts are depicted, each demonstrating the connectivity between rooms in various training environments. In the first floor plan, the connection between the bedroom and bathroom is correctly identified ($\delta_1(A_c[i,j]) = 1$), as indicated by the green check mark. In the second training environment, the connectivity between bedroom and bathroom is not recognized ($\delta_2(A_c[i,j]) = 0$), marked by the red cross. When considering N training environments, the reachability ($\boldsymbol{\Gamma}$) between place cluster i and j is calculated as follows:

$$\boldsymbol{\Gamma}[i,j] = \frac{\sum_{n=1}^{N} \delta_n(\mathbf{A}_c[i,j])}{\sum_{n=1}^{N} \delta_n(P_i)\delta_n(P_j)^T}, \tag{2}$$

where $\mathbf{A}_c = \mathbf{A}_{pi}\mathbf{A}_{im}\mathbf{A}_{pi}^T$ and $\delta_n(\mathbf{A}_c[i,j])$ indicates the existence of a connection between place cluster i and j. The function $\delta_n(P_i)$ denotes the presence of the place cluster P_i in the nth scene.

It is important to note that the same places are not connected, thus $\mathbf{A}_c[i,j] = 0$ when $i = j$. Furthermore, to normalize the reachability, we calculate it based on the number of environments in which each cluster appears, rather than the total number of environments. Reachability is set to zero if a cluster does not appear in any scene.

Place-object relationship. The relationship between object nodes and place clusters is established by connecting object nodes to image nodes within a graph, and then linking these image nodes to place graph nodes. This linkage facilitates the computation of the probability distribution for place and object categories. Specifically, for the nth training environment, we calculate $\mathbf{A}_{po}^n = \mathbf{A}_{pi}^n\mathbf{A}_{io}^n\mathbf{A}_{oc}^n$, where $\mathbf{A}_{oc} \in \mathbb{R}^{N_o \times N_c}$ maps each object node to its corresponding object category and N_c is the number of object categories.

By aggregating all semantic graph maps from the training environments, the relational connection between each place cluster and the object categories is defined as $\mathbf{R} = \sum_{n=1}^{N} \mathbf{A}_{po}^n$, where $\mathbf{R} \in \mathbb{R}^{N_p \times N_c}$ represents the number of connections between place clusters and object categories.

Given a set of object categories $\mathbb{O} = \{O_1, \ldots, O_{N_c}\}$, the probabilities of encountering a specific place cluster i given an object category j, and conversely, the probability of encountering an object category j given a place cluster i, are computed as follows:

$$p(P_i|O_j) = \frac{\mathbf{R}[i,j]}{\sum_{k=1}^{N_p} \mathbf{R}[k,j]}, p(O_j|P_i) = \frac{\mathbf{R}[i,j]}{\sum_{c=1}^{N_c} \mathbf{R}[i,c]}, \tag{3}$$

where $\mathbf{R}[i,j]$ indicates the number of connections between place cluster i and object category j. These probability distributions are illustrated in Section 2 of the supplementary material.

Updating relations. Our experimental setup is designed to test the navigation agent's ability to plan its path using common sense, akin to human navigation, in a new environment without a pre-existing map. To adapt effectively to these unfamiliar settings, the SEA updates the graph in a Bayesian manner [22]. As illustrated in Fig. 4, when a new place cluster or object-place connection is discovered during the construction of the semantic graph map, the prior probability is adjusted based on the new observations to calculate the posterior distribution. Given that these probabilities are determined by counting occurrences, the impact of new connections is generally minor. Thus, the update rate is set at 0.1 of the maximum count value ($\max(\mathbf{R}[i,j])$). The graph is continuously updated at every step. If the target object is not detected in the expected target place, the probability associated with the target object being in that place decreases. As the connection between the target object and the place cluster weakens, the next most connected place cluster is identified and explored.

3.5. Global policy

The global policy (π_g) utilizes the RGB-D image from the directional camera (s_t^d) to determine subgoals (g_t) through semantic path planning, which leverages the place relationships and place-object relationships within the SEA. For example, to navigate from a bathroom to a kitchen, the calculated path might be bathroom → bedroom → living room → kitchen. Rather than attempting to locate the kitchen directly from the bathroom, the agent first navigates to intermediary nodes such as the bedroom and the living room, thereby systematically discovering the optimal path.

Current place. To enable the semantic path planning, the agent first determines its current location by encoding RGB images and object information using the place encoder (as detailed in Section 3.2). The extracted place feature, derived from the encoded RGB image, object

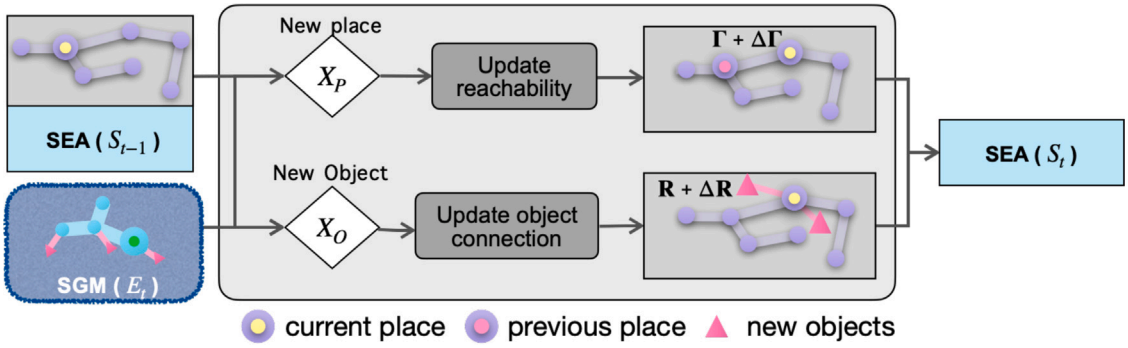


Fig. 4. Adaptive SEA update procedure.

features, and object category, is then compared to place clusters (P) using cosine similarity to identify the nearest place cluster, thereby locating the agent within the environment.

Target place. The target place is chosen based on its association with the target object. The place cluster with the highest probability of containing the target object is selected as the target destination. This selection process is mathematically formulated as follows:

$$k_t^* = \text{argmax}_k p_t(P_k|O_{\text{goal}}), \tag{4}$$

where P_k represents the potential target place, and $p_t(P_k|O_{\text{goal}})$ denotes the probability of place cluster k given the target object O_{goal}, as defined in .

Subgoal place. Using conventional graph-based planning methods, a navigation agent can identify an optimal trajectory to the target place. However, the optimal subgoal may not always be near the current location. To address this, subgoal candidates are selected among the visible places identified using detected objects from the directional sensor.

To streamline the selection process and reduce computational complexity, the probability of each place, $p_t(\mathbb{P}|o_t)$, is approximated using the importance of the object category. The object importance is defined as the inverse of the entropy of the object distribution conditioned on places, given as $1/\mathbb{E}_{\mathbb{P}}[-\log p_t(\mathbb{O}|\mathbb{P})]$. Objects associated with a single place cluster have high importance, whereas those common to multiple clusters exhibit lower importance.

Among the observed objects, the category deemed most important, denoted o, is used to determine the place cluster with the highest probability, calculated as $\text{argmax}_{\mathbb{P}} p(\mathbb{P}|o)$. To facilitate this, the directional image with a field of view (FOV) of 120° is narrowed by 40° to focus on objects directly ahead (o_{f*}), to the left (o_{l*}), and to the right (o_{r*}). These selected objects help estimate the subgoal candidates: $\mathbb{P}_s = \{P_f, P_l, P_r\}$, where $P_x = \text{argmax}_{\mathbb{P}} p_t(\mathbb{P}|o_{x*})$ for x representing the front, left, and right directions, respectively.

A subgoal place (g_t) with the highest reachability to the target place cluster is then chosen from these subgoal places. The selection is based on the following formula:

$$g_t = \text{argmax}_{P_{\tau_0} \in \mathbb{P}_s} \prod_{i=1}^{m-1} \mathbf{\Gamma}_{P_{\tau_{i-1}} P_{\tau_i}} \cdot \mathbf{\Gamma}_{P_{\tau_{m-1}} P_{k_t^*}}, \tag{5}$$

where $\{P_{\tau_0}, \ldots, P_{\tau_{m-1}}, P_{k_t^*}\}$ represents the optimal semantic path from the subgoal to the target place. If a subgoal is beyond a reasonable straight-line distance, it is considered unreachable and is excluded, similar to the NRNS method [25]. The remaining candidate that is both reachable and closest to the semantic subgoal is selected as the final subgoal. If all potential subgoals belong to the same place cluster, or if no detected objects aid the decision, a subgoal is randomly chosen among them. This mechanism encourages broader exploration by the agent, preventing it from being confined to a specific area.

Furthermore, the semantic path is derived from a shortest path calculation by setting the edge weight in the place graph between ith place and jth place to $-\log(\mathbf{\Gamma}_{P_i P_j})$:

$$\mathrm{T}^* = \text{argmin}_\tau \exp \sum_{i=1}^{m} -\log(\mathbf{\Gamma}_{P_{\tau_{i-1}} P_{\tau_i}}), \tag{6}$$

where $\mathrm{T}^* = \{\tau_0, \ldots, \tau_m\}$ is a set of indices representing the optimal semantic path, starting from P_{τ_0} and ending at P_{τ_m}, the goal place. This trajectory, T^*, represents the most probable path, effectively bridging the start and the target locations, optimizing the agent's navigation strategy.

3.6. Local policy

The local policy (π_l) processes directional RGB-D sensor data (s_t^d) along with local pose sensor readings to navigate the agent towards the designated subgoal g_t. It employs the fast marching method (FMM) [33] to compute the shortest path from the agent's current location to the subgoal. This computation makes use of the obstacle channel, which is derived from the top-down map created from the depth component of the RGB-D input. Upon determining the shortest path, the local policy executes a series of deterministic actions to guide the agent along this path. This strategy of navigation has been validated in previous research, demonstrating its effectiveness in various scenarios [14,15, 20].

4. Experiments

4.1. Baselines

Non-interactive baselines. `BC`: A baseline for behavior cloning was trained using an RNN-based policy that takes RGB-D, agent pose, and goal object category as inputs.

End-to-end RL baselines. `DD-PPO` [27]: Standard end-to-end RL with distributed training over several nodes is proposed. `Red-Rabbit` [21]: Auxiliary tasks that improve sampling efficiency and generalization to previously unseen domains are provided. `THDA` [19]: RL reward and model inputs are improved, which results in better generalization to new scenes.

Metric map-based baselines. `FBE` [23]: A traditional frontier-based exploration method is adapted to object goal navigation using a detector to detect the target. It triggers a stop when the target is reached using the metric map. `ANS` [14]: A spatial metric map-based RL policy trained for exploration is adapted to the object goal navigation using the same heuristic as `FBE`[23] for goal detection and stopping. `PONI` [20]: Non-interactive training is used to navigate and only trained potential fields are used to determine the next subgoal.

Graph map-based baselines. **ANS + SI** [16]: An abstract model is attached to **ANS** [14]. The agent incrementally extends the abstract model and reuses the learned model from previous episodes. **SemExp + SI** [16]: An abstract model is attached to semantic exploration (**SemExp** [15]) using the same abstract model strategy as **ANS + SI**. For **DD-PPO**, **Red-Rabbit**, **THDA**, and **PONI**, publicly available MP3D results on the Habitat ObjectNav leaderboard are used. For **ANS**, pre-trained models released by the authors are evaluated. For **ANS + SI** and **SemExp + SI**, official results from the published paper are used.

4.2. Experimental settings

Datasets. We utilized the Habitat simulator [34] to conduct experiments using the Matterport3D (MP3D) [26] datasets, which feature photorealistic 3D reconstructions of the real world. The standard 61 train/11 val splits for the ObjectNav configuration, as described in Section 3.1, were employed. It should be noted that only the local policy depends on the depth and pose, making the proposed method considerably more practical for use in the real world with noisy pose and depth sensors. The Habitat ObjectNav dataset [34] was used for MP3D experiments, with 21 goal categories (provided in the supplementary Section 1). We utilized 2195 episodes for the test.

Evaluation metrics. All methods were evaluated using the success rate (**Success**), success weighted by path length (**SPL**) [35], and distance to success (**DTS**). **Success** is determined by calculating the ratio of successful test episodes to the total number of test episodes. **SPL** takes into account both the Success and path length. When there are M episodes, $\textbf{SPL} = \frac{1}{M}\sum_{i=1}^{M} Y_i \frac{l_i}{\max(p_i, l_i)}$, where l_i is the length of the shortest path from goal to target, p_i is the length of the path taken by the agent, and Y_i is the binary indicator of Success for ith episode. Finally, **DTS** is the L_2 distance (measured in m) between the agent and the success threshold ($1.0m$) of the goal object at the end of the episode, as described in [35].

Implementation details. To construct SEA, we examined ten episodes from each train scene, for a total of 610 episodes. The maximum number of time steps was set to 500, and the environment was explored randomly. For object detection, we trained a MaskRCNN [31] model to identify 40 object categories in MP3D environments. Our method does not construct a metric map, thus a different stopping criterion was used compared to metric map-based methods. If an object is detected from a distance and exceeds a target object detection score threshold, the agent approaches the object and checks whether it is the target object. If the object detection score is lower after the encounter, it is assumed that it is not the target. The object feature with the highest detection score along the approaching path is stored in the checked object list. If a detected target object is highly similar to the checked objects, it is not rechecked, as it has already been searched. Additional details can be found in Section 1 of the supplementary material.

4.3. Results

SEA outperforms navigation baselines. Our SEA method sets a new benchmark by surpassing all previous state-of-the-art baseline methods on the MP3D dataset's validation split, as detailed in Table 1. This achievement encompasses end-to-end reinforcement learning (RL), metric map-based baselines, and graph map-based methods, particularly in terms of Success and DTS metrics. Remarkably, SEA demonstrates a staggering 926.3% improvement in Success compared to the behavior cloning model. Furthermore, SEA outperforms the PONI[20] by 22.64% in Success and by 13.20% in SPL, thanks to its development of more efficient pathways. While PONI[20] is highly reactive and excels in exploration with its frontier-based path generation, it falls short in exploitation; in contrast, SEA uses topological maps for long-term planning, allowing it to create more effective routes through better exploitation. When compared to the SemExp + SI[16] model, which

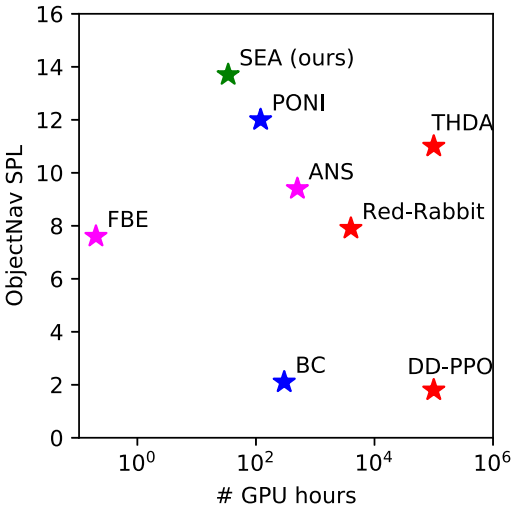


Fig. 5. Training cost.

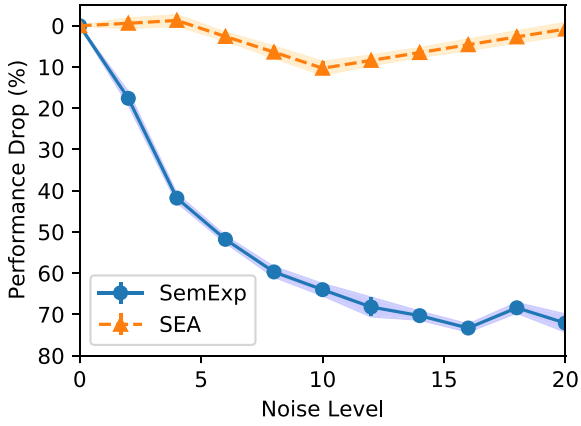


Fig. 6. Ablation study on pose sensor noises.

combines an abstract model with a semantic exploration approach, SEA increases Success by 12.4%. This is particularly notable given that SEA does not use global pose information and is trained without interactive reinforcement learning. SEA outperforms SemExp[16] in terms of Success but has slightly lower SPL (by 1.4 percentage points) due to its adaptability. This adaptability allows SEA to eventually locate the object, even if it initially follows inefficient routes. In contrast, SemExp[16] relies on a highly accurate metric map and a perfect pose sensor, making it very efficient at following the optimal path. However, if SemExp[16] is led astray onto an incorrect path, it lacks the mechanism to update and correct itself, making it difficult to locate the object. SEA's ability to adapt and correct its course enables higher success rates, even though this sometimes means taking longer and less efficient paths, resulting in lower SPL performance.

SEA has low computational requirements. The place encoder can be trained within a day using a single GPU. Constructing SEA takes about 10 h with a single GPU. During the inference, a GPU with 3000 MB memory is enough to run the trained encoder and detector. Our SEA has the lowest cost, three times less than the non-interactive SoTA baseline [20], as demonstrated in Fig. 5.

SEA is robust to pose noises. SEA demonstrates enhanced robustness to pose sensor noise by utilizing place reachability for long-term planning instead of building a metric map. As shown in Fig. 6, SEA experiences a modest performance drop in Success, with a 10% decrease at a noise level of 10 and only a 0.77% reduction at noise level 20, despite

Table 1
Habitat ObjectNav results on MP3D. We report the results from the top-performing methods.

Method	Pose noise	MP3D (val)		
		Success ↑	SPL ↑	DTS ↓
BC	✗	3.8	2.1	7.5
DDPPO [27]	✗	8.0	1.8	6.9
Red-Rabbit [21]	✗	34.6	7.9	–
THDA [19]	✗	28.4	11.0	5.6
FBE [23]	✗	22.7	7.2	6.7
ANS [14]	✗	27.3	9.2	5.8
PONI [20]	✗	31.8	12.1	5.1
ANS + SI [16]	✗	27.9	13.1	6.1
SemExp + SI [16]	✗	34.7	**15.1**	5.8
SEA (ours)	✔	**39.0**	13.7	**5.0**
SEA w/o Update	✔	33.3	13.6	5.7

Table 2
Localization results.

Method	Input	Acc@0.5 m ↑	Acc@1 m ↑
NRNS [25]	I	10.5	66.5
VGM [18]	I	36.9	62.7
TSGM [17]	I+O	38.9	65.1
SEA	I+O+P	**40.4**	**73.1**

Table 3
Impact of place info.

SEA ablations		MP3D (val)		
Subgoal	Stop	Success ↑	SPL ↑	DTS ↓
✗	✗	29.2	11.9	6.1
✔	✗	36.4	14.2	5.3
✔	✔	39.0	13.7	5.0

significant pose sensor interference. Here, the noise levels are indicative of real-world scenarios, with noise level 1 mimicking common robotic system disturbances and higher levels representing more severe interference. In contrast, the SemExp model shows a marked decline in efficiency—42% at noise level 4 and 64% at noise level 10, further deteriorating with higher noise levels. This emphasizes SEA's ability to maintain efficiency through consistent replanning, leveraging its global policy model effectively, even when local paths are obscured by noise. These results highlight SEA's superior adaptability over traditional metric map-based approaches that rely on precise pose sensors.

Semantic information helps to improve localization. The accumulated graph data serves as a form of memory that integrates various pieces of semantic information. To determine whether this accumulated semantic information is indeed beneficial, we employ a trained network with attached probes to evaluate its utility for localization purposes. The localization probe network comprises Graph Neural Networks (GNNs) and transformer decoder networks. These networks are trained with the ground truth location coordinates (x, y). After training, the performance is assessed on a test set by calculating the distance between the predicted and actual locations.

Upon analyzing the localization results Table 2, it is evident that our SEA method, utilizing inputs images (I), objects (O), and places (P), exhibits superior performance in both Acc@0.5 m and Acc@1 m metrics with scores of 40.4 and 73.1, respectively. The Acc@1 m metric signifies the average accuracy of distance calculations within a 1-m range. If the calculated distance is accurate within this range, an accuracy score of 1 is assigned, while inaccuracies are denoted as 0. When the performance enhancement of SEA is calculated in terms of percentage increase, we observe substantial improvements over other methods. Specifically, compared to NRNS [25], SEA demonstrates an extraordinary increase of approximately 285% in Acc@0.5 m. In comparison to VGM [18], SEA shows a noticeable improvement as well. The TSGM [17] method, with inputs I and O, ranks second to SEA, yet SEA still surpasses it in terms of accuracy. To summarize, these results not only indicate the dominance of SEA in localization accuracy, particularly within a 1-m range, but also emphasize the considerable performance enhancement achieved by incorporating semantic knowledge.

Relation update is effective on adapting to unseen environment. The effectiveness of adapting to the unseen environment through relation updates is demonstrated in the results of an ablation study presented in the second row from the bottom of Table 1. As new place connections or place-object connections arise during testing, the probability distribution is updated at each step of the current episode. This acquired connection information is utilized solely within the episode and is not stored for future use. The experiment yielded a 17.1% increase in success rate compared to the case where no episodic update was applied. Notably, SPL only improved by 0.7%. This is likely due to the episodic update altering the posterior distribution. The agent initially navigates to the location where the target object is most likely to be found and checks for its presence. If the object is not discovered in the initial place cluster, the agent may become stuck. However, the relation update weakens the connection between the target place cluster and the target object, allowing the agent to replan and reach the next most connected place cluster. As the agent successfully finds the object by moving to the next place cluster, SPL decreases as the path length of the successful path becomes longer.

Place cluster is useful for planning. We evaluated the impact of place-specific information in the planning process using a model that randomly designates subgoals. In this context, "Subgoal" refers to the place-based subgoal selection method, which strategically enhances reachability based on place connections. As delineated in Table 3, the implementation of this subgoal selection method resulted in quantifiable improvements. Specifically, success rates increased by 24.7%, and the SPL metric concurrently rose by 19.3%. The term "Stop" denotes the place-based stop mechanism, a model variant that facilitates termination before reaching the predetermined target place cluster. The examination of this mechanism revealed distinct effects. The success rate decreased by 7.1%. In conclusion, the use of place clustering proves to be advantageous for the planning process.

SEA can choose different trajectories based on different goals. This experiment demonstrates that the path is contingent upon the chosen goal, with the starting point fixed and goal categories varied. In Fig. 7, the yellow star marks the starting point, and the red region indicates the goal boundary (1 m radius), solely for visualization. The figure shows trajectories for six object goals: fireplace, cabinet, chair, chest of drawers, bathtub, and bed. Each trajectory adapts to its respective target. Detailed analysis is available in the supplementary material.

5. Visualization

Construction of SGMs. The process of constructing a SGM is depicted in Fig. 8. A SGM integrates place graphs, image graphs, and object graphs. In this representation, image nodes are depicted as circles. Their colors correspond to the respective place clusters. Object nodes are depicted as triangles. The colors of these nodes indicate object categories, matching the colors of the bounding boxes in the panoramic RGB image. For clarity, connections between image nodes and object nodes have been omitted in the visualization. It is important to note that top-down maps and the positions of image and object nodes are utilized solely for

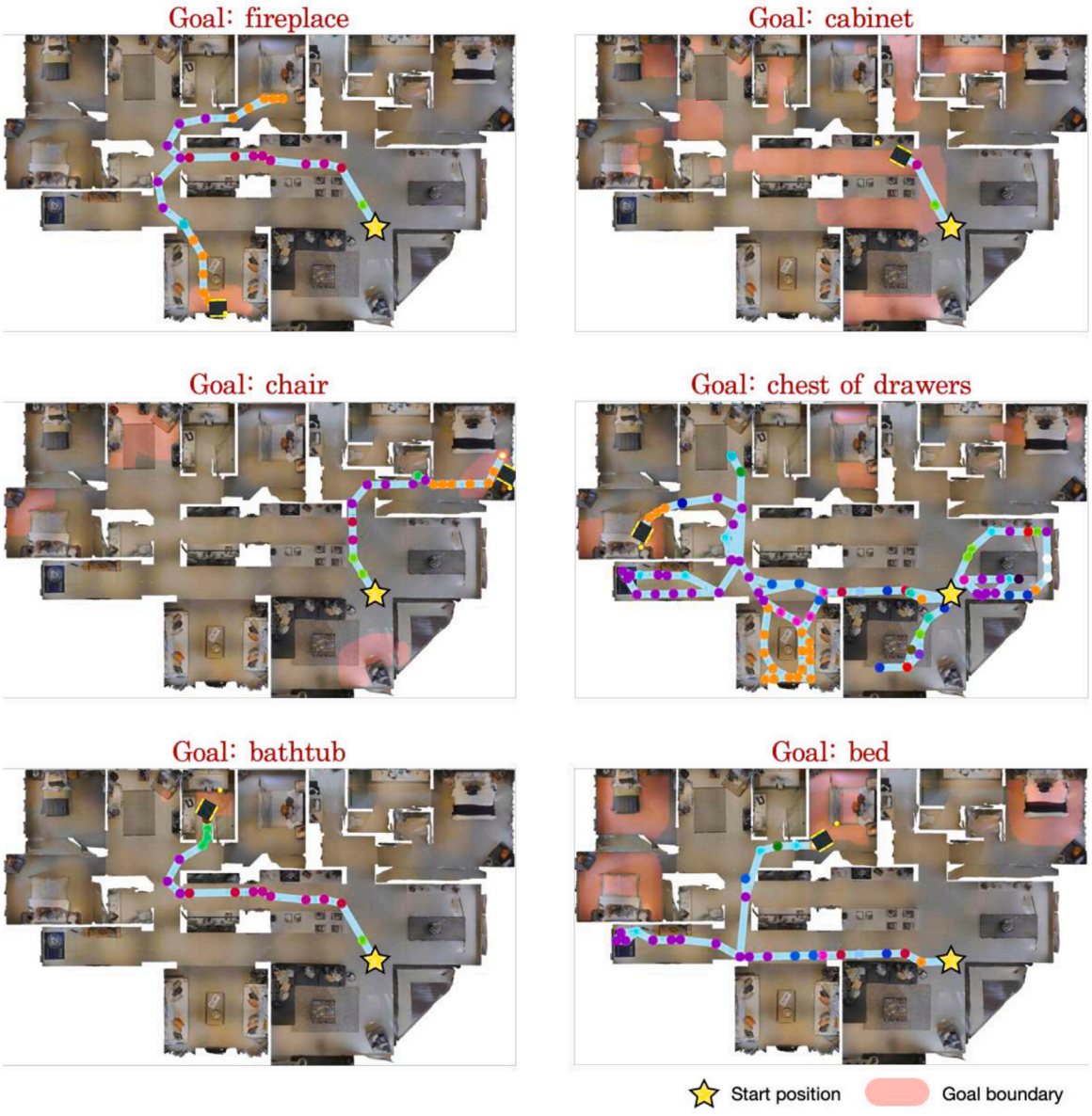


Fig. 7. Learning semantic relationships. The figure illustrates how SEA can identify efficient paths by leveraging semantic relationships. It shows the trajectories for six target objects: a fireplace, a cabinet, a chair, a chest of drawers, a bathtub, and a bed. These trajectories demonstrate various pathways that are adapted to each specific target object. The starting point is marked by a yellow star, while the goal boundary is represented by a red region, indicating that the goal is within a 1-m radius.

visualization purposes and are not used as input data. Additionally, a supplementary video is available that demonstrates the construction of semantic graph maps, showcasing one episode across 20 training scenes.

Example visualization of episodes. We provide an example visualization of an episode in Fig. 9. This shows how semantic prior graphs are used in the global policy to perform ObjectNav, where the goal is identified as 'bed'. The navigation process begins with the agent perceiving the bedroom (P_{12}) to be on the left side and the kitchen to be at the front and right side. Based on this initial perception, the agent decides to move left, anticipating that the target location might be there. Upon reaching the subgoal, the agent searches for a bed but does not find one. Consequently, the agent exits the current location and re-evaluates the subgoal, noticing a door on the left and inferring that the target location is likely on the left side. While proceeding towards the subgoal, the agent eventually encounters the target object, the bed. The agent then formulates a local plan to reach the bed. Finally, after confirming that the location matches the intended destination, the bedroom (P_{12}), the agent presses the stop button. Detailed information and additional examples are provided in the supplementary material.

6. Conclusions and future work

6.1. Conclusions

We present SEA, a method for learning semantic relationships between places and objects in unknown environments with low computational cost. Our approach identifies the object's location and navigates using place connections. By adapting to the unseen environment through relation updates, SEA achieves state-of-the-art results for ObjectNav in MP3D. Unlike other methods are not robust to noise pose sensor, SEA is robustly navigate environment with noisy settings. Our method is the first to not require a global metric map for ObjectNav in large environments like MP3D. We show that incorporating semantic relationships improves localization and navigation tasks.

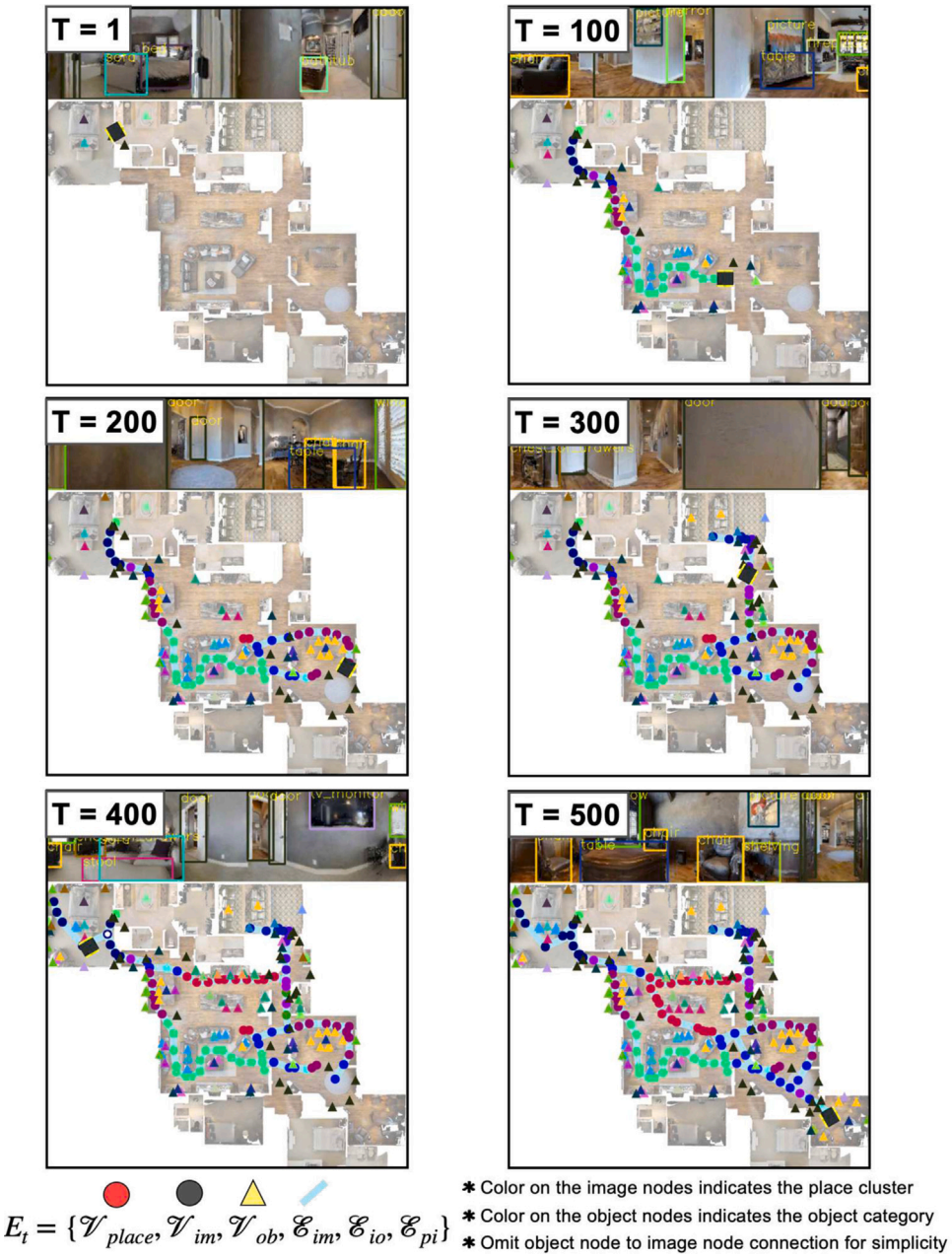


Fig. 8. Example of constructing an episodic graph. While randomly investigating the surroundings, the agent accumulates an episodic graph. A place graph, an image graph, and an object graph comprise an episodic graph. Color on the image node indicates the place cluster, circles indicate image nodes, and triangles represent object nodes.

6.2. Future work

In addition to the current approach, we propose future directions and still open challenges:

1. **Incorporating language features for 3D objects**: Our proposed method relies solely on image features to define objects. Using language features for 3D objects could lead to more general features that can improve object search and correlation graph connections in the metric space.

2. **Agents in dynamic environments**: Recognizing changes in the environment, such as a cup being moved from the kitchen to the living room, can aid in task-solving. If an agent maintains memory in the form of a graph, it can adapt much better to dynamic environments compared to using a metric map.

3. **Recognizing physical laws in the environment**: To manipulate objects or perform meaningful control tasks, it is necessary to understand the physical laws governing the environment. For example, avoiding small wooden blocks on the floor or considering the center of mass when picking up a tool.

4. **Interactive intelligence**: Our method should not only rely on its own intelligence but also interact with humans or other robots in the environment to update the topological map. Further advancements in these areas can lead to more robust and effective navigation and localization in complex environments.

We believe that our proposed method is a step towards achieving this goal, and we look forward to future developments and improvements. Additional details can be found in the supplementary material.

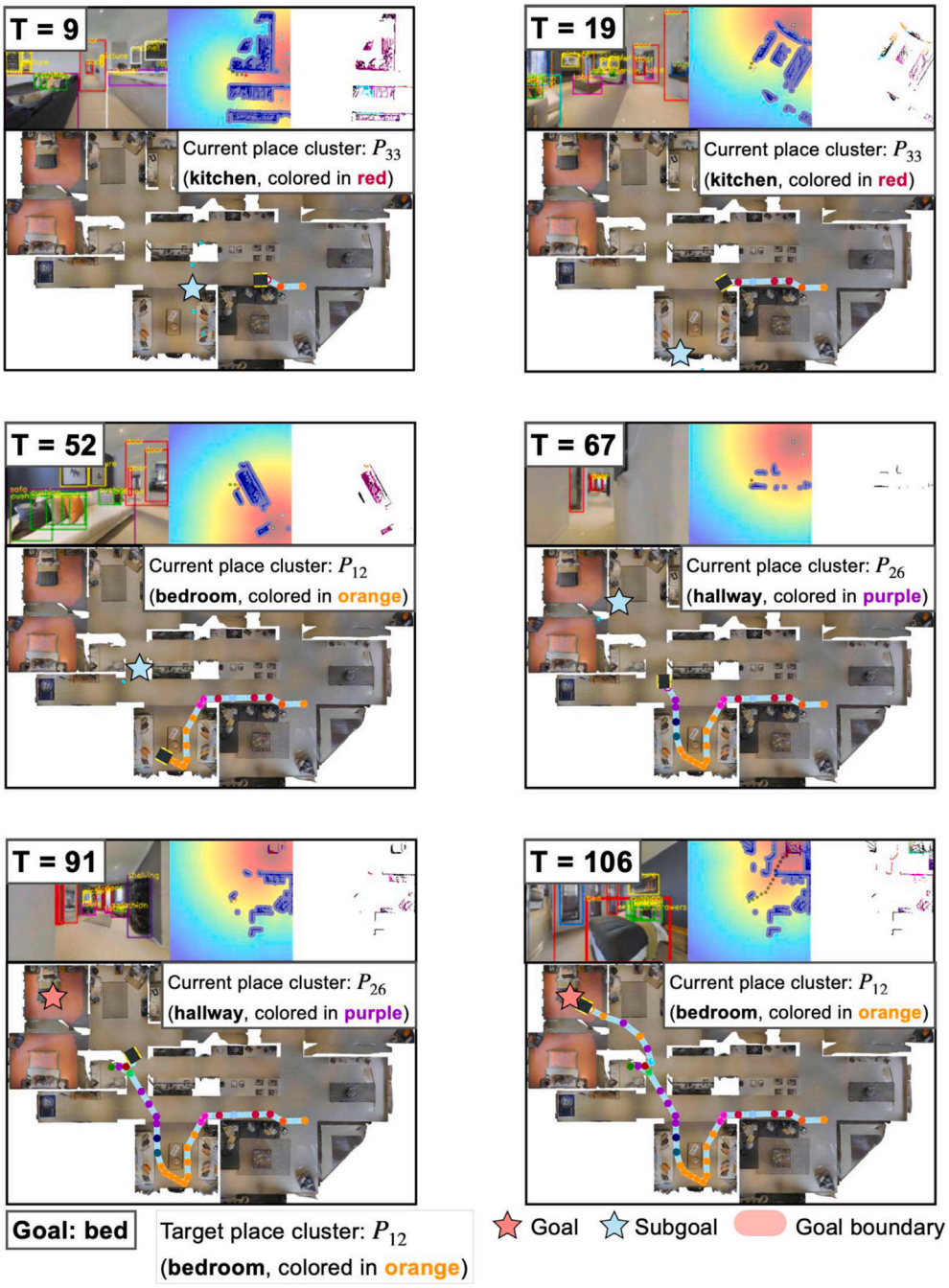


Fig. 9. Qualitative examples of navigation using SEA. When given the goal object as a bed, the agent formulates a plan to navigate to the bedroom (P_{12}). To reach the bedroom (P_{12}), the agent predicts the subgoal place cluster using the categories of the initially visible objects.

CRediT authorship contribution statement

Nuri Kim: Writing – original draft, Visualization, Methodology, Conceptualization. **Jeongho Park:** Writing – review & editing. **Mineui Hong:** Methodology. **Songhwai Oh:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01190, [SW Star Lab] Robot Learning: Efficient, Safe, and Socially-Acceptable Machine Learning).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.knosys.2024.112446.

References

[1] P. Mirowski, M. Grimes, M. Malinowski, K.M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, A. Zisserman, R. Hadsell, et al., Learning to navigate in cities without a map, in: Neural Information Processing Systems (NeurIPS), 2018.

[2] K. Chen, J.P. de Vicente, G. Sepulveda, F. Xia, A. Soto, M. Vázquez, S. Savarese, A Behavioral Approach to Visual Navigation with Graph Localization Networks, Robot. Sci. Syst. (RSS) (2019).

[3] J. Yang, Z. Ren, M. Xu, X. Chen, D. Crandall, D. Parikh, D. Batra, Embodied visual recognition, in: IEEE International Conference on Computer Vision, ICCV, 2019.

[4] T. Chen, S. Gupta, A. Gupta, Learning exploration policies for navigation, in: International Conference on Learning Representations, ICLR, 2019.

[5] A. Kumar, S. Gupta, D. Fouhey, S. Levine, J. Malik, Visual memory for robust path following, in: Advances in Neural Information Processing Systems (NeurIPS), 2018.

[6] K. Fang, A. Toshev, L. Fei-Fei, S. Savarese, Scene memory transformer for embodied agents in long-horizon tasks, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019.

[7] N. Savinov, A. Dosovitskiy, V. Koltun, Semi-parametric topological memory for navigation, in: International Conference on Learning Representations, ICLR, 2018.

[8] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, J. Malik, Cognitive mapping and planning for visual navigation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017.

[9] E. Parisotto, R. Salakhutdinov, Neural map: Structured memory for deep reinforcement learning, in: International Conference on Learning Representations, ICLR, 2018.

[10] G. Avraham, Y. Zuo, T. Dharmasiri, T. Drummond, Empnet: Neural localisation and mapping using embedded memory points, in: IEEE International Conference on Computer Vision, ICCV, 2019.

[11] Y. Lv, N. Xie, Y. Shi, Z. Wang, H.T. Shen, Improving target-driven visual navigation with attention on 3D spatial relationships, 2020, arXiv preprint arXiv:2005.02153.

[12] H. Du, X. Yu, L. Zheng, Learning object relation graph and tentative policy for visual navigation, in: European Conference on Computer Vision, ECCV, 2020.

[13] Y. Qiu, A. Pal, H.I. Christensen, Target driven visual navigation exploiting object relationships, 2020, arXiv preprint arXiv:2003.06749.

[14] D.S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, R. Salakhutdinov, Learning to explore using active neural SLAM, in: International Conference on Learning Representations, ICLR, 2020.

[15] D.S. Chaplot, D. Gandhi, A. Gupta, R. Salakhutdinov, Object goal navigation using goal-oriented semantic exploration, 2020, arXiv preprint arXiv:2007.00643.

[16] T. Campari, L. Lamanna, P. Traverso, L. Serafini, L. Ballan, Online learning of reusable abstract models for object goal navigation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2022.

[17] N. Kim, O. Kwon, H. Yoo, Y. Choi, J. Park, S. Oh, Topological semantic graph memory for image goal navigation, in: Conference on Robot Learning, CoRL, 2022.

[18] O. Kwon, N. Kim, Y. Choi, H. Yoo, J. Park, S. Oh, Visual graph memory with unsupervised representation for visual navigation, in: IEEE International Conference on Computer Vision, ICCV, 2021.

[19] O. Maksymets, V. Cartillier, A. Gokaslan, E. Wijmans, W. Galuba, S. Lee, D. Batra, THDA: Treasure hunt data augmentation for semantic navigation, in: IEEE International Conference on Computer Vision, ICCV, 2021.

[20] S.K. Ramakrishnan, D.S. Chaplot, Z. Al-Halah, J. Malik, K. Grauman, PONI: Potential functions for ObjectGoal navigation with interaction-free learning, 2022, arXiv preprint arXiv:2201.10029.

[21] J. Ye, D. Batra, A. Das, E. Wijmans, Auxiliary tasks and exploration enable objectgoal navigation, in: IEEE International Conference on Computer Vision, ICCV, 2021.

[22] Y. Wu, Y. Wu, A. Tamar, S. Russell, G. Gkioxari, Y. Tian, Bayesian relational memory for semantic visual navigation, in: IEEE International Conference on Computer Vision, ICCV, 2019.

[23] B. Yamauchi, A frontier-based approach for autonomous exploration, in: IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA, 1997.

[24] S. Zhang, X. Song, Y. Bai, W. Li, Y. Chu, S. Jiang, Hierarchical object-to-zone graph for object navigation, in: IEEE International Conference on Computer Vision, ICCV, 2021.

[25] M. Hahn, D.S. Chaplot, S. Tulsiani, M. Mukadam, J.M. Rehg, A. Gupta, No RL, no simulation: Learning to navigate without navigating, in: Neural Information Processing Systems (NeurIPS), 2021.

[26] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, Y. Zhang, Matterport3D: Learning from RGB-D data in indoor environments, in: International Conference on 3D Vision, 3DV, 2017.

[27] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, D. Batra, DD-PPO: Learning near-perfect PointGoal navigators from 2.5 billion frames, in: arXiv Preprint , 2019, arXiv:1911.00357.

[28] Z. Ravichandran, L. Peng, N. Hughes, D. Griffith, L. Carlone, Hierarchical representations and explicit memory: Learning effective navigation policies on 3D scene graphs using graph neural networks, in: IEEE International Conference on Robotics and Automation, ICRA, 2019.

[29] M. Narasimhan, E. Wijmans, X. Chen, T. Darrell, D. Batra, D. Parikh, A. Singh, Seeing the un-scene: Learning amodal semantic maps for room navigation, in: European Conference on Computer Vision, ECCV, 2020.

[30] J. Li, P. Zhou, C. Xiong, S.C. Hoi, Prototypical contrastive learning of unsupervised representations, in: International Conference on Learning Representations, ICLR, 2021.

[31] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: IEEE International Conference on Computer Vision, ICCV, 2017.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Neural Information Processing Systems (NeurIPS), 2017.

[33] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Natl. Acad. Sci. 93 (4) (1996) 1591–1595.

[34] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, D. Batra, Habitat: A platform for embodied AI research, in: IEEE International Conference on Computer Vision, ICCV, 2019.

[35] P. Anderson, A. Chang, D.S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, A.R. Zamir, On evaluation of embodied navigation agents, in: arXiv Preprint , 2018, arXiv:1807.06757.