Fine-Grained Self-Localization from Coarse Egocentric Topological Maps

Daiki Iwata, Kanji Tanaka, Mitsuki Yoshida, Ryogo Yamamoto, Yuudai Morishita and Tomoe Hiroki
University of Fukui, 3-9-1 Bunkyo, Fukui City, Fukui 910-0017, Japan

Keywords: Active Topological Navigation, Ego-Centric Topological Maps, Incremental Planner Retraining.

Abstract: Topological maps are increasingly favored in robotics for their cognitive relevance, compact storage, and ease of transferability to human users. While these maps provide scalable solutions for navigation and action planning, they present challenges for tasks requiring fine-grained self-localization, such as object goal navigation. This paper investigates the action planning problem of active self-localization from a novel perspective: can an action planner be trained to achieve fine-grained self-localization using coarse topological maps? Our approach acknowledges the inherent limitations of topological maps; overly coarse maps lack essential information for action planning, while excessively high-resolution maps diminish the need for an action planner. To address these challenges, we propose the use of egocentric topological maps to capture fine scene variations. This representation enhances self-localization accuracy by integrating an output probability map as a place-specific score vector into the action planner as a fixed-length state vector. By leveraging sensor data and action feedback, our system optimizes self-localization performance. For the experiments, the de facto standard particle filter-based sequential self-localization framework was slightly modified to enable the transformation of ranking results from a graph convolutional network (GCN)-based topological map classifier into real-valued vector state inputs by utilizing bag-of-place-words and reciprocal rank embeddings. Experimental validation of our method was conducted in the Habitat workspace, demonstrating the potential for effective action planning using coarse maps.

1 INTRODUCTION

Topological maps are widely utilized in robotics due to their higher cognitive relevance compared to geometric maps, compact storage requirements, and ease of transferability to human users. Numerous researchers have investigated methods for creating topological maps and applying them to navigation and action planning. These maps offer lightweight, scalable solutions that are simpler than geometric maps and require significantly less storage. A topological map typically consists of coarsely quantized region nodes and a set of edges representing relationship between these regions, providing a concise representation of the workspace. This region-based representation is robust against minor errors in self-localization, and as long as the estimation remains within the same region node, the impact on navigation performance is minimal (Ulrich and Nourbakhsh, 2000; Ranganathan and Dellaert, 2008; Lui and Jarvis, 2010). However, this error tolerance poses challenges for tasks requiring fine-grained self-localization, such as safe driving. In fact, much of the past research on topological navigation has relied on the combined use of accurate metric maps or assumed the availability of infrastructure such as Global Positioning Systems (GPS). Little progress has been made in achieving fine-grained self-localization using only coarse topological maps, with (Chaplot et al., 2020) being a notable exception, though its active vision does not primarily focus on active localization.

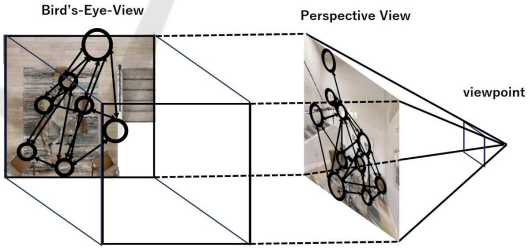

Figure 1: Topological navigation using ego-centric topological maps. Left: Conventional world-centric map. Right: The proposed ego-centric map.

In this paper, we focus on the action planning problem active self-localization from the novel per-

Iwata, D., Tanaka, K., Yoshida, M., Yamamoto, R., Morishita, Y. and Hiroki, T.
Fine-Grained Self-Localization from Coarse Egocentric Topological Maps.
DOI: 10.5220/0013098000003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 2: VISAPP*, pages 810-819
ISBN: 978-989-758-728-3; ISSN: 2184-4321

spective of topological map-based self-localization, specifically exploring the research question: "Can an action planner be trained to achieve fine-grained self-localization using coarse topological maps?" This approach is not universally applicable, as the effective range of the given topological map has its limits: On one hand, overly coarse topological maps fail to provide useful information for the action planner. Conversely, when the resolution is already fine, self-localization becomes accurate enough to make action planning trivial.

Thus, we aim to develop an excellent action planner while also investigating its limitations. Regarding the development, our key insight is to use egocentric topological maps—rather than traditional world-coordinate-based maps—to capture fine scene variations and achieve fine-grained self-localization (Fig. 1). Then, the output probability map in the form of a place-specific score vector is integrated into the action planner as a fixed-length state vector. By leveraging sensor data and action feedback, the system improves self-localization accuracy and converges toward an optimal estimation.

As a technical contribution, we present a novel self-localization framework that employs a classifier as the front-end and a sequential state estimator as the back-end. This approach enables the integration of graph convolutional network (GCN) classifiers, typically used for topological map recognition, with the de facto standard particle filter for sequential self-localization into a unified pipeline. This integration is achieved by utilizing bag-of-place-words and reciprocal rank vectors as intermediate representations. The experimental investigation has been validated in the Habitat workspace (Szot et al., 2021).

The contributions of this paper are summarized as follows: (1) We formalize the active localization problem based on a novel egocentric topological map that does not require pre-computation and maintenance of world-centric maps. (2) This approach enables fully incremental real-time active localization, allowing localization, planning, and planner training to be completed within the real-time budget of each viewpoint. (3) By utilizing coarse, region-based topological maps, we achieve fine-grained self-localization beyond the region level, demonstrating state-of-the-art self-localization performance as validated through experiments.

2 RELATED WORK

Topological navigation is a behavior adopted by various animal species, including humans (Leonard and Durrant-Whyte, 1991)(Thrun et al., 2002). A topological map models the environment as a graph, where only characteristic scene parts are encoded; thus, it provides a much more compact representation than metric maps. This is in contrast to geometric map models, such as grid maps, where raw data and geometric features (lines, edges, etc.) are used to represent the environment as a set of coordinates of objects or obstacles. Furthermore, topological maps are one of the most effective means of dealing with uncertainties in visual robot navigation (Brooks, 1985), and various frameworks have been proposed, including geometric features (Stankiewicz and Kalia, 2007)(Tapus and Siegwart, 2008)(Nüchter and Hertzberg, 2008)(Tapus and Siegwart, 2008), appearance features (Lui and Jarvis, 2010)(Lowe, 1999)(Ulrich and Nourbakhsh, 2000)(Mikolajczyk et al., 2005), visual pedestrian localization (Zha and Yilmaz, 2021), and matching techniques (Li and Olson, 2012)(Cummins and Newman, 2008)(Ranganathan and Dellaert, 2008)(Aguilar et al., 2009)(Neira and Tardós, 2001). However, most rely on globally consistent world-centric maps; thus, they are not applicable to egocentric maps. Recently, impressive methods for active localization have been proposed for cases using grid data, such as images (Chaplot et al., 2018); however, active localization for non-grid data, such as topological maps, is still largely unexplored. To the best of our knowledge, this is the first study to develop a fully incremental, real-time active localization framework that does not rely on any globally consistent world-centric models to be pre-computed or maintained.

The localization applications considered in this study pertain primarily to semantic localization, a recently emerging domain-invariant localization application (Schönberger et al., 2018). In (Schönberger et al., 2018), 3D point clouds and semantic features were used for highly robust and accurate semantic localization, and novel deep neural networks were employed to embed the geometric and semantic features. In contrast, we explored a purely monocular localization problem that did not rely on 3D models/measurements. In (Yu et al., 2018), the semantic region edges provided by semantic segmentation were used as features. In contrast, we do not rely on the availability of precise semantic segmentation; instead, we use only the coarse semantic, size, and location attributes of the scene parts. In (Gawel et al., 2018), a semantic graph was employed as a scene model to achieve accurate localization via graph matching in outdoor scenes. However, this method assumes perfect semantic segmentation. Furthermore, they rely on costly graph matching, and their considerable computational burden may limit their scal-

ability. In (Guo et al., 2021a), semantic histograms were extracted from a semantic graph map to achieve a highly efficient topological localization. However, this method assumes the availability of discriminative scene graphs and may encounter difficulties in semantically poor domains (also known as bucolic environments (Benbihi et al., 2020)). In contrast, our active localization approach relies only on very simple semantic and spatial features, and therefore robust against segmentation noise and has good generalization performance. Importantly, egocentric topological maps do not require the management of maps in a world-centered coordinate system, making them naturally compatible with map-less navigation (e.g., object goal navigation) (Chaplot et al., 2020).

3 APPROACH

3.1 System Overview

Active localization typically comprises two main modules: passive localization and action planning. Passive localization is responsible for estimating the robot's state (e.g., viewpoint) given the latest egomotion and perceptual measurements. The action planner is responsible for determining the optimal next-best-view action, given the latest state estimate, by simulating possible future robot-environment interactions. These two submodules are described as follows:

We formulate passive localization as a place classification problem to classify a given egocentric topological map into predefined place classes. Note that this is one of the most scalable formulations of the self-localization problem (Lowry et al., 2015), among other formulations such as image retrieval, multiple hypothesis tracking, geometric matching, and viewpoint regression. For example, in (Weyand et al., 2016), a planet-scale place classification problem was considered using adaptive partitioning of a large-scale workspace (i.e., planet) into place classes. For simplicity, in this study, the grid-based place partitioning in (Kim et al., 2019) is adopted, as it allows the incremental addition/deletion of place classes and is thus more suitable for autonomous robotics applications. As the input modality for the robot, (Kim et al., 2019) assumes the use of 3D LiDAR, whereas we use an RGB camera. Nevertheless, the movement of the robot on a two-dimensional plane in a top-down view coordinate system is common to both, and thus the 2D grid partitioning from (Kim et al., 2019) can be directly applied to our grid partitioning.

Action planning is formulated as a discrete time-discounted Markov decision process (MDP) (Sutton and Barto, 1998). A discrete time-discounted MDP is a general formulation consisting of a set of states S, a set of actions A, a state-transition distribution P, a reward function R, and a discount rate γ. In our particular scenario, the state $s \in S$ was estimated using a passive localization module. An action consists of turns with a rotation angle r and forward movements by a distance f. A reward was provided for successful localization at the end of each training episode. Specifically, an episode consists of L=4 repetitions of sense-plan-action cycles, and the agent receives at the final viewpoint in each episode, a reward value of r=1 if the top-1 ranked place class is consistent with the ground truth or r=-1 otherwise.

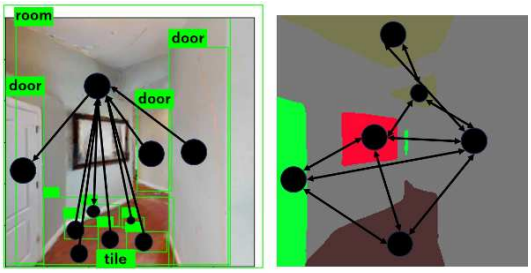

Figure 2: Scene parsing. Left: SGB (Tang et al., 2020). Right: Ours.

3.2 Active Localization with Ego-Centric Topological Maps

Similarity-preserving mapping from an image to a scene description is an important requirement for active localization. For our egocentric topological map, graphs with similar node/edge attributes must be reproduced from similar viewpoints.

This issue is most relevant to scene-graph generation (Tang et al., 2020), which aims to parse an image into a scene graph. However, most of these existing approaches are optimized for scene understanding and related applications and not for localization applications. In fact, our preliminary experiments revealed that their performance in localization applications is often extremely limited.

Figure 2 shows the results of our evaluation of the state-of-the-art scene parsing in (Tang et al., 2020) and the results of our three-step approach. Although the former method can precisely describe the causal relationships between parts, it is often not invariant across different viewpoints. By contrast, our approach is designed to increase invariance at the expense of distinctiveness.

We focused on invariance rather than translation accuracy to cross-view domain changes, and we fol-

lowed a conservative three-step heuristic method (Zhu et al., 2022), including (1) image segmentation into part regions (i.e., nodes), (2) part region descriptions, and (3) inter-part relationship inferences (i.e., edges), as detailed below.

The part segmentation step segments an input image of size 256×256 pixels into subregions using the semantic segmentation model in (Zhou et al., 2017). This model consisted of a ResNet module (Cao et al., 2010) and a pyramid pooling module (Zhao et al., 2017) trained on the ADE20K dataset.

The part description step describes each part of a region using a combination of semantic and spatial descriptors. Then, we further categorize the semantic labels output using the semantic segmentation method in (Zhou et al., 2017) into 10 coarser meta categories, including "wall, " "floor, " "ceiling, " "bed, " "door, " "table, " "sofa, " "refrigerator", "TV, " and "Other." Regions smaller than 100 pixels in area were considered dummy objects and were not used as graph nodes. For the spatial descriptor, the spatial attributes of a part region are compactly represented by a "size/location" category (Cao et al., 2010). First, each part was categorized into one of three categories with respect to the "size" category. A size category is determined according to the area of the bounding box, including "small (0)" $S < S_o$, "medium (1)" $S_o \leq S < 6S_o$, and "large (2)" $6S_o \leq S$." S_o is a constant corresponding to 1/16 of the image area, set based on the simple idea of dividing the image into a 4x4 grid. Then, the bounding box center location was discretized using a grid of 3×3=9 cells, and we used the cell ID ($\in [0,8]$) as the location category. Note that the above attributes are all human-interpretable semantic categories and do not introduce complex appearances or spatial attributes, such as real-valued descriptors. Finally, a node feature is defined as a one-hot vector of dimensions $10 \times 3 \times 9 = 270$ in the combined space of the semantic, size, and location categories.

The edge connection step connects node pairs that are spatially close to each other with edges. Specifically, a part pair was considered to be in spatial proximity if the bounding boxes overlapped. A training set of ego-centric topological maps was then fed into the training set of a graph neural network. For the network architecture, a graph convolutional neural network (GCN) in a deep graph library (Wang et al., 2019) was employed. The number of layers of the GCN was set to two. This GCN is specifically used to classify input, place-specific ego-centric topological maps into several prototype place classes. This classification task essentially follows the classical prototype method in the field of computer vision. However, in our application, an explicit set of prototype classes is not manually provided, so the robot must define them in an unsupervised manner. The simple way to define this is to perform unsupervised clustering of the training ego-centric topological maps, sampled from the target workspace, into K groups, treating each cluster as a prototype class. Following this simple idea, we define K prototype place classes. As a result, the classification output from methods like GCN is typically represented as class-specific rank vectors. From the perspective of information fusion, it is common to express this as a class-specific reciprocal rank vector. This can be considered as a score-based bag-of-place-words representation, where the score values in this case are reciprocal rank values. Specifically, in our implementation, we generate place prototypes by dividing the workspace into K coarse grid cells in an overhead coordinate system. **Figure 3** shows the test view sequence and the classification results of the graph neural network. It can be observed that prototype places with similar scores are included for spatially adjacent viewpoints, and they exhibit high similarity. We exploit this fact to compress the infinitely growing ego-centric topological maps into a graph neural network.

It can be seen that prototypes with similar score values are included for spatially adjacent viewpoints, and have high similarity. We exploit this fact to compress an infinitely increasing egocentric topological map into a graph neural network. Nevertheless, it can be also seen that there are subtle differences in the relative strengths of the scores between the different views. We utilize such subtle differences as cues to discriminate between different scenes.

Another issue is that the outputs of the graph neural network are usually not calibrated as a proper probability function. Here, we propose a graph neural model as the ranking function. This is motivated by the fact that it is common to use a neural network as a classifier or ranking function rather than a probability regressor, and there is considerable experimental evidence for its effectiveness (Krizhevsky et al., 2012). Specifically, we interpret the class-specific probability map output from the graph neural network as a class-specific reciprocal rank (RR) vector derived from the field of multimodal information fusion (Cormack et al., 2009). The RR vector is a class-specific score vector and can be used as a state vector of the given input graph. Note that the time cost for transforming a class-specific probability vector into a reciprocal rank vector is on the order of the number of place classes and is very low.

A particle filter-based sequential passive localization method was employed to update the belief of

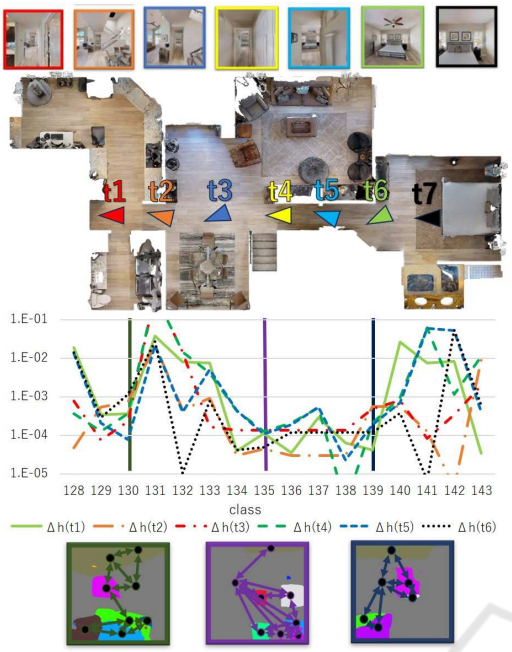

Figure 3: Seven spatially adjacent input images along the robot trajectory and their bag-of-words representation: The graph shows the time difference of bag-of-words histogram $h[t](t = 1,2,3,4,5,6,7)$ of viewpoint sequence of length 7. $\Delta h[t]$=$h[t+1] - h[t]$(t=1,2,3,4,5,6). Among the elements of $\Delta h[t]$, three visual words with small absolute values of time difference are chosen, and their prototype ego-centric topological maps are shown in the figure.

the robot's state (i.e., 3dof pose) incrementally in real time, given the latest observations and actions at each viewpoint. We slightly modified the standard implementation of the particle filter localization (Dellaert et al., 1999) for our application. First, because the measurement is a class-specific reciprocal rank vector and not a likelihood vector, the weight of each particle was updated using the reciprocal rank fusion rule (Cormack et al., 2009) rather than the Bayesian rule. Next, the classification results were obtained for each class by max pooling the weights of the particles belonging to that class, regardless of their bearing attributes. The number of particles was set to 10,000 for every Habitat workspace. The detailed algorithm of reciprocak rank-based particle filter (RRPF) is provided in Algorithm 1.

3.3 Incremental Training of Action Planner

In this section, the integration of incremental planner training into an active localization pipeline is described. First, we extended the BoW concept such that the output of the graph neural network can be interpreted as a BoW descriptor. Subsequently, we reformulated the planner training task as nearest-neighbor-based Q-learning in (Sutton and Barto, 1998), and further extended it to an incremental training scheme by introducing an incremental BoW-based nearest-neighbor engine. As a result, we obtain a novel fully-incremental framework that is able to complete not only visual recognition and action planning, but also planner training can also be completed within each viewpoint's real-time budget.

BoW is a popular scene descriptor in robotics. It describes a given input scene using an unordered collection of visual words $\{(w_i, s_i)\}_{i=1}^{N}$. A vocabulary function f, typically a k-means dictionary (Sivic and Zisserman, 2003), should be pretrained to map an ego-centric topological map to visual words $\{w_i\}_{i=1}^{N}$ with its importance score s_i representing how much each word w_i contributes to the scene representation. However, applying a BoW descriptor to structured scene models like ego-centric topological maps is not a trivial task, as the BoW descriptors always ignore the relationship between scene parts. Typical vocabularies such as the k-means dictionary (Cummins and Newman, 2008) assume one-to-one mapping from a scene part to a visual word and are thus not applicable to structured data. Here, we propose to reuse the graph neural network as the vocabulary. Specifically, we viewed a collection of N place classes $\{w_i\}_{i=1}^{N}$, with class-specific reciprocal rank scores $\{s_i\}_{i=1}^{N}$ provided by the graph neural network as a collection of visual words. Note that the resulting BoW descriptor is now a fixed-length vector and transferable to many other machine learning frameworks.

Q-learning is a standard RL framework for reinforcement learning (Sutton and Barto, 1998). It aims to learn an optimal state-action map through robot-environment interactions with delayed rewards. The naive implementation of the state-action-value function requires unacceptable spatial costs, particularly when the state space becomes high dimensional. To address this issue, researchers have developed fast approximation variants for Q-function. Nearest neighbor Q-learning (NNQL) (Shah and Xie, 2018) is a recent example of such a variant. It approximates the state-action value function using the nearest neighbor search. Recall that the Q-function is updated in the following formula (Sutton and Barto, 1998): $Q(s_t,a) \leftarrow Q(s_t,a) + \alpha\,[r_{t+1} + \gamma \max_{p\in A} Q(s_{t+1},p)$ - $Q(s_t,a)]$. In this updated formula, the number of times that the Q function is referenced is once for calculating $Q(s_t,a)$ and $|A|$ times for calculating $Q(s_{t+1},p)$. Therefore, the nearest-neighbor search must be performed $(|A|+1)$ times for each viewpoint.

We replaced the nearest neighbor search in NNQL

Algorithm 1: Reciprocal Rank-based Particle Filter Algorithm.

Initialization:
    Randomly generate particles from a uniform distribution (e.g., 10,000 particles).
    Initialize each particle's pose as $(\text{location}, \text{orientation})$, and set the initial score to 0.
Motion Model Application:
    Update the origin pose based on the action index.
    Apply the same transformation to each particle to generate new pose hypotheses.
    Convert rotation angles to radians.
    Compute the new positions using trigonometric functions.
Observation Model Application:
    For each particle, determine the class ID within the environment based on the particle's new pose (location and orientation).
    Update the particle's score based on the observations.
Score Update:
    For each class, update the score using the Reciprocal Rank Fusion (RRF) formula:

$$\text{score} \mathrel{+}= \frac{1}{\text{RANK}+1}$$

    Where RANK is the ranking position of the class.
    Reflect the updated scores in the reciprocal rank vector.
Resampling:
    Generate a new set of particles based on the updated scores.

with BoW retrieval. Specifically, each database element is represented by a triplet consisting of state s, action a, and value q. Then, the Q-value for a given state-action pair (s,a) is stored in an inverted index, which is built independently for each possible action a, using each word w that makes up the state s as an index. The optimal next-best-view action a^* for some state s is chosen in the following steps. First, the database for each action a was retrieved using s as a query, yielding a shortlist of the most relevant $k = 4$ database items. The value of each state-action pair (s,a) was then computed by averaging the k Q-values. As mentioned above, the Q-value is obtained for each candidate of the state-action pair (s,a). Note that by building a temporary hash table that maps score values to items given a search result, the shortlist length and cost for finding the top-k nearest neighbor items can be made independent of the database size and very small, respectively.

4 EXPERIMENTS

In this section, we describe the experiments we performed and report and analyze the results. In summary, we evaluated active localization frameworks in a variety of challenging and crowded indoor environments and found that the proposed method with the simplest ego-centric topological maps already significantly outperformed state-of-the-art techniques for semantic localization.

Figure 4: Experimental environments.

Experiments were performed using the 3D photorealistic simulator Habitat-Sim (Szot et al., 2021). Five workspaces, "00800-TEEsavR23oF," "00801-HaxA7YrQdEC," "00802-wcojb4TFT35," "00806-tQ5s4ShP627," and "00808-y9hTuugGdiq," from the Habitat-Matterport3D Research Dataset (HM3D) was imported into Habitat-Sim. The robot workspace is partitioned by a grid-based partitioning method with spatial resolution of 2 [m] and 30 [deg]. As a results, the above workspaces are partitioned into 576, 648, 720, 336, and 576 place classes, respectively. A bird's eye view of the robot workspaces are shown in **Fig. 4**.

The localization performance was evaluated using top-1 accuracy. Recall that the particle filter is employed to extend the single-view GCN-based place classification to sequential localization (III-C). The top-1 accuracy was calculated by evaluating whether the top-1 classes of the class-specific rank values output by the particle filter were consistent with the ground-truth class for each test sample.

The number of epochs was set to 5. The batch size was 32. The learning rate was 0.001. For each dataset,

the GCN classifier was trained using a training set consisting of ego-centric topological maps with class labels as supervision. In reinforcement learning, the planner is trained using 10,000 training episodes by default. The number of sense-plan-action cycles per episode was L=4. At the final viewpoint in each episode, the reward function returns a reward of +1 if the class top-1 ranked by the particle filter is consistent with the ground truth; otherwise it returns a reward of -1. The hyperparameters for the NNQL training were set as follows: The number of iterations is 10,000. The learning rate α is 0.1. The discount factor γ is 0.9. During action planning, the action with the highest Q value is usually selected, but actions are randomly selected until the 25th episode, and thereafter, actions are determined by the ε-greedy algorithm, where $\varepsilon = 1/(0.1 * ([episodeID] + 1) + 1)$.

The proposed method was compared with the baseline and ablation methods. To date, active localization using first-person-view scene graphs like ego-centric topological maps has not been explored. To address this, the baseline method was built by replacing an essential module of the proposed framework, the GCN with ego-centric topological map, with a state-of-the-art semantic histogram embedding in (Guo et al., 2021b). In the semantic histogram method, each graph node votes to generate a histogram of length D^3, where $D = 10$ denotes the number of semantic labels. The histogram bin ID is determined by concatenating the length three sequence of semantic labels from three graph nodes: the node of interest, an adjacent node (a child), and the child's adjacent node (a grandchild). Our own implementation of the Python code was used. Two ablation methods, single-view localization and passive multi-view localization frameworks, were compared with the proposed active localization (i.e., active multi-view frameworks). The passive multi-view framework differs from the proposed framework in that it does not perform action planning but determines actions randomly. The single-view framework terminated the localization task from the first viewpoint for each episode.

The performance results are summarized in **Table 1**. As expected, the proposed active localization framework clearly outperformed the two ablation methods in all Habitat workspaces. The proposed method is competitive and outperforms state-of-the-art passive localization, although it uses a simple topological map as the input modality. Furthermore, the proposed method exhibits a more stable active localization performance than the baseline semantic histogram framework. This may be because the combination of graph neural networks and ego-centric topological maps captures better contextual information from simple semantic scene graphs.

Table 1: Performance results.

		800	801	802	806	808
GCN	active (Ours.)	67.8	69.2	68.3	68.8	61.6
	passive	58.7	57.0	51.5	62.3	52.4
	single-view	50.6	50.2	39.1	50.9	38.4
sem. histo. (Guo et al., 2021b)	active	N/A	40.4	39.3	54.3	N/A
	passive	N/A	34.4	32.7	48.6	N/A
	single-view	N/A	26.0	19.0	31.6	N/A

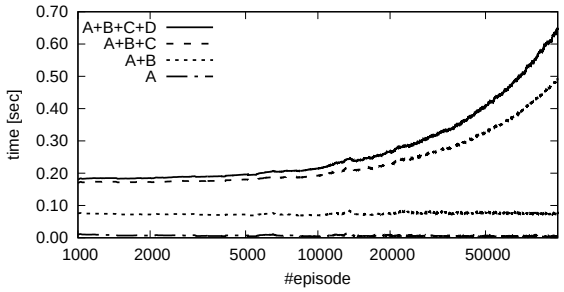

Figure 5: Time cost per sense-plan-acition cycle. A: Preprocessing. B: Particle filter. C: Action planning. D: Planner retraining.

Figure 5 shows timing performance. performance (CPU: Core i7-11700K, programming language: C++) As can be seen, the computational cost of the proposed method is sub-constant and real-time, at least up to 10000 episodes. As expected, fully-incremental and real-time processing is achieved.

Figure 6 shows examples of success and failure. As shown in the figure, the robot's movement toward viewpoints with a high concentration of natural landmark objects often improves the localization performance. For example, from the first viewpoint, the robot was facing a wall and could not observe any valid landmarks, but from the next viewpoint, by changing the direction of travel, it was able to detect a door, improving the self-localization accuracy using this landmark object. A typical example of a failure is shown in **Fig. 6**. In this case, the majority of viewpoints in the episode faced nondescript objects, such as walls and windows. Notably, the recognition success rate tended to decrease when the viewpoint was too close to the object and the field of view was narrowed.

One of the novelties of the proposed framework is that it allows not only visual recognition and action planning but also planner training to be completed within the real-time budget of each viewpoint. We conducted additional experiments to demonstrate this performance. In this additional experiment, planner training is performed while the robot performs long-term navigation using a random environment exploration algorithm. At that time, as in the previous experiments, we will repeat an episode consist-

(a) success examples.

(b) failure examples.

Figure 6: Examples of L repetitions of sense-plan-action cycles.

ing of $L = 4$ sense-plan-action cycles. Also, at the beginning of the episode, the particle filter is initialized. Note that, unlike the previous experiments, the final robot pose of one i-th episode becomes the initial robot pose of the next $(i+1)$-th episode. After developing the first version of the long-term exploration algorithm, it was observed that the robot frequently gets stuck in a narrow depression formed by an obstacle in the workspace, and wasted many training episodes. Therefore, we modified the exploration algorithm so as to reduce the chance of getting stuck. Specifically, we modified the action set to include more translation actions among the nine actions, by replacing some rotate actions with translation actions. The modified action set consists of the following pairing of rotate r [deg] and forward f [m]: $(r, f) \in \{$ (0, 0.5), (0, 1.5), (30, 0.3), (-35, 0.3), (80, 1), (-85, 1), (140, 0), (-145, 0), (180, 0)$\}$. We trained over 10,000 episodes and evaluated over 1000 episodes, using the workspace "00801-HaxA7YrQdEC". The top-1 accuracy result was 69.2. By modifying the action set, we were able to explore the map evenly, which resulted in high results. A closer look at the results shows that when the test was performed only with coordinates that the robot had experienced, the result was 78.2, and in all other cases it was 65.1.

The total distance traveled by the robot during this training was 4598.3 meters. This time, we have fine-tuned the action set to get good results on the current workspace, so it is not clear whether this method generalizes to other environments and it is a subject for future research. A future challenge is to develop a general-purpose action set that can be generalized to various environments. Another challenge is to develop a method that allows the robot to successfully pass through narrow passages. Ensemble learning is a promising direction for further improving performance (Islam et al., 2003).

In conclusion, the proposed method with fully incremental real-time planner training outperforms state-of-the-art approaches despite the fact that it uses simple semantic features.

5 CONCLUSIONS

In this paper, we proposed a practical solution for trainable active localization using topological maps. The key idea of the proposed method is to employ a novel ego-centric topological map rather than requiring precomputation and maintenance of a world-centric map. The collection of ego-centric maps, which increases incrementally and unlimitedly in proportion to the robot's travel distance, is compressed to a fixed size using a graph neural network, and then transferred to a novel incremental action planner and planner training module. As a result, fully-incremental real-time active localization was achieved, allowing localization, planning, and planner training to be completed within the real-time budget of each viewpoint. We verified the scalability, incrementality, real-time nature, and robustness of our method through training scenarios involving many intermittent navigations and unprecedented long-distance navigations.

REFERENCES

Aguilar, W., Frauel, Y., Escolano, F., Martínez-Pérez, M. E., Espinosa-Romero, A., and Lozano, M. A. (2009). A robust graph transformation matching for non-rigid registration. *Image Vis. Comput.*, 27(7):897–910.

Benbihi, A., Arravechia, S., Geist, M., and Pradalier, C. (2020). Image-based place recognition on bucolic environment across seasons from semantic edge description. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*, pages 3032–3038. IEEE.

Brooks, R. A. (1985). Visual map making for a mobile robot. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, Missouri, USA, March 25-28, 1985*, pages 824–829. IEEE.

Cao, Y., Wang, C., Li, Z., Zhang, L., and Zhang, L. (2010). Spatial-bag-of-features. In *The Twenty-Third IEEE*

Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010, pages 3352–3359. IEEE Computer Society.

Chaplot, D. S., Parisotto, E., and Salakhutdinov, R. (2018). Active neural localization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Chaplot, D. S., Salakhutdinov, R., Gupta, A., and Gupta, S. (2020). Neural topological slam for visual navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12875–12884.

Cormack, G. V., Clarke, C. L. A., and Büttcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In Allan, J., Aslam, J. A., Sanderson, M., Zhai, C., and Zobel, J., editors, *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 758–759. ACM.

Cummins, M. J. and Newman, P. M. (2008). FAB-MAP: probabilistic localization and mapping in the space of appearance. *Int. J. Robotics Res.*, 27(6):647–665.

Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte carlo localization for mobile robots. In *1999 IEEE International Conference on Robotics and Automation, Marriott Hotel, Renaissance Center, Detroit, Michigan, USA, May 10-15, 1999, Proceedings*, pages 1322–1328. IEEE Robotics and Automation Society.

Gawel, A., Don, C. D., Siegwart, R., Nieto, J. I., and Cadena, C. (2018). X-view: Graph-based semantic multiview localization. *IEEE Robotics Autom. Lett.*, 3(3):1687–1694.

Guo, X., Hu, J., Chen, J., Deng, F., and Lam, T. L. (2021a). Semantic histogram based graph matching for real-time multi-robot global localization in large scale environment. *IEEE Robotics Autom. Lett.*, 6(4):8349–8356.

Guo, X., Hu, J., Chen, J., Deng, F., and Lam, T. L. (2021b). Semantic histogram based graph matching for real-time multi-robot global localization in large scale environment. *IEEE Robotics Autom. Lett.*, 6(4):8349–8356.

Islam, M. M., Yao, X., and Murase, K. (2003). A constructive algorithm for training cooperative neural network ensembles. *IEEE Transactions on neural networks*, 14(4):820–834.

Kim, G., Park, B., and Kim, A. (2019). 1-day learning, 1-year localization: Long-term lidar localization using scan context image. *IEEE Robotics Autom. Lett.*, 4(2):1948–1955.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Bartlett, P. L., Pereira, F. C. N., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114.

Leonard, J. J. and Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Trans. Robotics Autom.*, 7(3):376–382.

Li, Y. and Olson, E. B. (2012). IPJC: the incremental posterior joint compatibility test for fast feature cloud matching. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 3467–3474. IEEE.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999*, pages 1150–1157. IEEE Computer Society.

Lowry, S., Sünderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., and Milford, M. J. (2015). Visual place recognition: A survey. *ieee transactions on robotics*, 32(1):1–19.

Lui, W. L. D. and Jarvis, R. A. (2010). A pure vision-based approach to topological SLAM. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 18-22, 2010, Taipei, Taiwan*, pages 3784–3791. IEEE.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. (2005). A comparison of affine region detectors. *Int. J. Comput. Vis.*, 65(1-2):43–72.

Neira, J. and Tardós, J. D. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robotics Autom.*, 17(6):890–897.

Nüchter, A. and Hertzberg, J. (2008). Towards semantic maps for mobile robots. *Robotics Auton. Syst.*, 56(11):915–926.

Ranganathan, A. and Dellaert, F. (2008). Automatic landmark detection for topological mapping using bayesian surprise. Technical report, Georgia Institute of Technology.

Schönberger, J. L., Pollefeys, M., Geiger, A., and Sattler, T. (2018). Semantic visual localization. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6896–6906. Computer Vision Foundation / IEEE Computer Society.

Shah, D. and Xie, Q. (2018). Q-learning with nearest neighbors. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3115–3125.

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, pages 1470–1477. IEEE Computer Society.

Stankiewicz, B. J. and Kalia, A. A. (2007). Acquistion of structural versus object landmark knowledge. *Journal*

of Experimental Psychology: Human Perception and Performance, 33(2):378.

Sutton, R. S. and Barto, A. G. (1998). Reinforcement learning: An introduction. *IEEE Trans. Neural Networks*, 9(5):1054–1054.

Szot, A., Clegg, A., Undersander, E., Wijmans, E., Zhao, Y., Turner, J., Maestre, N., Mukadam, M., Chaplot, D. S., Maksymets, O., Gokaslan, A., Vondrus, V., Dharur, S., Meier, F., Galuba, W., Chang, A. X., Kira, Z., Koltun, V., Malik, J., Savva, M., and Batra, D. (2021). Habitat 2.0: Training home assistants to rearrange their habitat. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 251–266.

Tang, K., Niu, Y., Huang, J., Shi, J., and Zhang, H. (2020). Unbiased scene graph generation from biased training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3713–3722. Computer Vision Foundation / IEEE.

Tapus, A. and Siegwart, R. (2008). Topological SLAM. In Bessière, P., Laugier, C., and Siegwart, R., editors, *Probabilistic Reasoning and Decision Making in Sensory-Motor Systems*, volume 46 of *Springer Tracts in Advanced Robotics*, pages 99–127.

Thrun, S. et al. (2002). Robotic mapping: A survey.

Ulrich, I. and Nourbakhsh, I. R. (2000). Appearance-based place recognition for topological localization. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000, April 24-28, 2000, San Francisco, CA, USA*, pages 1023–1029. IEEE.

Wang, M., Yu, L., Zheng, D., Gan, Q., Gai, Y., Ye, Z., Li, M., Zhou, J., Huang, Q., Ma, C., Huang, Z., Guo, Q., Zhang, H., Lin, H., Zhao, J., Li, J., Smola, A. J., and Zhang, Z. (2019). Deep graph library: Towards efficient and scalable deep learning on graphs. *CoRR*, abs/1909.01315.

Weyand, T., Kostrikov, I., and Philbin, J. (2016). Planet - photo geolocation with convolutional neural networks. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, volume 9912 of *Lecture Notes in Computer Science*, pages 37–55. Springer.

Yu, X., Chaturvedi, S., Feng, C., Taguchi, Y., Lee, T., Fernandes, C., and Ramalingam, S. (2018). VLASE: vehicle localization by aggregating semantic edges. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018*, pages 3196–3203. IEEE.

Zha, B. and Yilmaz, A. (2021). Map-based temporally consistent geolocalization through learning motion trajectories. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2296–2303. IEEE.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6230–6239. IEEE Computer Society.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. (2017). Scene parsing through ADE20K dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5122–5130. IEEE Computer Society.

Zhu, G., Zhang, L., Jiang, Y., Dang, Y., Hou, H., Shen, P., Feng, M., Zhao, X., Miao, Q., Shah, S. A. A., and Bennamoun, M. (2022). Scene graph generation: A comprehensive survey. *CoRR*, abs/2201.00443.