





Cognitive Navigation for Intelligent Mobile Robots: A Learning-Based Approach With Topological Memory Configuration

Qiming Liu , Xinru Cui , Zhe Liu , and Hesheng Wang , *Senior Member, IEEE*

Abstract—Autonomous navigation for intelligent mobile robots has gained significant attention, with a focus on enabling robots to generate reliable policies based on maintenance of spatial memory. In this paper, we propose a learning-based visual navigation pipeline that uses topological maps as memory configurations. We introduce a unique online topology construction approach that fuses odometry pose estimation and perceptual similarity estimation. This tackles the issues of topological node redundancy and incorrect edge connections, which stem from the distribution gap between the spatial and perceptual domains. Furthermore, we propose a differentiable graph extraction structure, the topology multi-factor transformer (TMFT). This structure utilizes graph neural networks to integrate global memory and incorporates a multi-factor attention mechanism to underscore elements closely related to relevant target cues for policy generation. Results from photorealistic simulations on image-goal navigation tasks highlight the superior navigation performance of our proposed pipeline compared to existing memory structures. Comprehensive validation through behavior visualization, interpretability tests, and real-world deployment further underscore the adaptability and efficacy of our method.

Index Terms—Graph neural networks (GNNs), spatial memory, topological map, visual navigation.

I. INTRODUCTION

IN recent years, autonomous robot navigation has emerged as a prominent research field. The primary focus is on empowering robots with task-oriented mobility awareness in unseen environments, leveraging their onboard, first-person perception [1]–[6]. Early learning-based navigation pipeline

Manuscript received July 16, 2023; revised December 4, 2023 and January 15, 2024; accepted February 14, 2024. This work was supported in part by the National Natural Science Foundation of China (62225309, 62073222, U21A20480, 62361166632). Recommended by Associate Editor Honghai Liu. (Corresponding author: Hesheng Wang.)

Citation: Q. Liu, X. Cui, Z. Liu, and H. Wang, “Cognitive navigation for intelligent mobile robots: A learning-based approach with topological memory configuration,” *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 9, pp. 1933–1943, Sept. 2024.

Q. Liu and X. Cui are with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: qimingliu@sjtu.edu.cn; cxr0726@sjtu.edu.cn).

Z. Liu is with the MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: liuzhesjtu@sjtu.edu.cn).

H. Wang is with the Department of Automation, Key Laboratory of System Control and Information Processing of Ministry of Education, Key Laboratory of Marine Intelligent Equipment and System of Ministry of Education, Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wanghesheng@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2024.124332

typically employs neural networks to map perception to decision-making [1], [2]. However, this approach results in a robot that relies solely on instant sensory inputs for decision-making, which can be short-sighted and challenging for adapting long-term tasks [7]. To address this issue, the introduction of memory mechanism has been proposed [3], [5], [8], [9]. Memory essentially incorporates episodic historical observations, enabling the robot to utilize a broader spectrum of temporal and spatial data for more comprehensive decision-making.

Topological memory is extensively utilized in a multitude of learning-based navigation systems [10]–[15]. In contrast to traditional scene representation structures that provide a dense and global description of the entire scene [6], [8], [16], [17], the topological map abstracts physical spaces into a network of discrete nodes. Each node encapsulates the observational features of its specific location, and the edges illustrate the accessibility between nodes [18]. This abstract representation can potentially decrease computational and storage costs by concentrating solely on essential landmark features [11], [19], [20]. Moreover, the interconnection of landmarks, as denoted by the edges, naturally signifies the spatial correlation of the scene’s structure and features, providing an advantage for global planning and effective memory utilization.

In order to construct a topological representation of the scene, some methods have relied on pre-existing environmental knowledge [10], [11], [21], [22]. More recent studies have leveraged perceptual similarity to inform the online construction of topological maps [12], [13], [23]–[27], operating under the assumption that dissimilar observations imply distinct topological nodes. Yet this approach does not account for the discrepancy between perceptual resemblance and physical location; visually similar scenes might be geographically separate, and different perspectives of the same location can appear perceptually distinct. Such inconsistencies can lead to superfluous nodes and inaccurate mappings in topological map generation [23], [24]. To address this challenge, this paper introduces a new mapping methodology that integrates perceptual similarity assessments with pose estimations, capturing both perceptual and spatial proximities for robust and precise topological map construction.

Building upon the generated topological memory, another important issue is how to utilize information from the memory structure to facilitate better decision-making. While the majority of existing research utilizes topological maps primar-

ily for path planning and task decomposition [10], [14], [19], [21], this approach does not exploit the full breadth of data available in the topological structure, and compromises path efficiency due to the sparsity of the map [12], [13], [15], [22], [28]. In response, we develop a differentiable graph extraction framework, the topology multi-factor transformer (TMFT), which facilitates adaptive extraction of memory features in response to the evolving task and environmental conditions.

In summary, this paper introduces a learning-based visual navigation framework that employs topological structures for spatial memory. Our approach offers substantial enhancements in both the construction and utilization of topological memory compared to existing models. We present a two-factor verification technique to construct more precise and coherent topological memory by integrating relation assessment in both perceptual and spatial domains. Furthermore, we develop a neural-based pipeline TMFT to extract topological memory for the generation of navigational policies. The TMFT harnesses global topological data and selectively utilizes memory features that align with the robot's present observations and task objectives. Our primary contributions include:

- 1) The introduction of an online topological memory construction approach that synergizes neural odometry with perceptual similarity measures to bridge the gap between spatial proximity and perceptual resemblance. This technique provides a more robust and precise topological map even without panoramic perception, and it shows superior performance in complex environments characterized by scene similarity and changes in viewpoint.

- 2) The creation of a neural-based topological memory extraction method TMFT for navigation decision-making. This framework employs graph neural networks to integrate global memory and introduces a multi-factor attention mechanism for targeted extraction of memory information, thus improving the robot's task awareness and selective attention to pertinent memory indicators.

- 3) Empirical validation of our method's enhanced navigational performance within the photorealistic Gibson environment for image-goal tasks. We confirm the precision of our topological map generation and demonstrate adaptive memory integration with environmental and task-related cues through visualization and interpretability studies. Additionally, we test our system in real-world settings to ascertain its practical applicability.

II. RELATED WORK

A. Memory in Autonomous Navigation

To facilitate robot navigation in unseen environments, traditional geometry-based approaches utilize simultaneous localization and mapping (SLAM) techniques [18], [29], [30] to simultaneously model the environment and track the robot's location. The spatial maps created in real-time underpin trajectory planning algorithms that orchestrate the robot's movements [4], [31], [32]. One of the strengths of this methodology is its capacity to produce dependable and consistent trajectories, largely owing to the incorporation of episodic obser-

vations within the spatial map. Recent learning-based navigation strategies have even emphasized the role of historical memory, given their tendency to make reactive decisions based on present perceptions [1], [2], [7]. Earlier models attempted to encapsulate historical data within the parameters of neural networks [3], [5] or external buffers [33], but these solutions often fall short of the robust storage capabilities and the clear interpretability. Consequently, contemporary research has begun to merge structured scene maps [6], [8], [16], [17] from geometry-based methodologies with learning-based navigation frameworks, exploring the construction and utilization of these structured memory representations in novel environments.

Our research is centered on employing topological maps as a form of memory representation to bolster the efficacy of learning-based navigation systems. Unlike representations such as grid maps or point clouds that require global or dense coverage of the entire scene, topological memory [10]–[15] potentially offers better efficiency [11], [19], [20] by representing the scene discretely through topological nodes. Furthermore, topological memory representations can naturally indicate the connectivity between different regions of the scene. Nevertheless, generating and leveraging topological maps in complex settings—where similar scenes or frequent changes in viewpoint occur—present significant challenges [24]. This paper endeavors to tackle these obstacles by establishing reliable methods for constructing and effectively utilizing topological maps for navigation.

B. Generation of Topological Maps

Previous studies often presupposed prior knowledge of the environment [10], [11], [21], [22], creating topological maps offline using predetermined rules [34] or keyframe-based methods [11]. Such techniques, however, are ill-suited for navigating unseen areas. To address this, more recent work has shifted towards developing methods for online, incremental construction of topological maps. These methods typically generate discrete topological nodes and interconnect them by analyzing sequences of observations. Some approaches attempt to ascertain the spatial relationships between nodes, employing strategies like siamese networks to differentiate between similar and dissimilar samples [26], or by categorizing the distance intervals between observations [27]. To enhance the interpretability and robustness of similarity prediction, alternative research has introduced contrastive learning and metric losses to produce continuous measures of similarity [12], [13], [23]–[25]. These measures are often used in conjunction with thresholding techniques to decide when to introduce new nodes within the topological map.

A key limitation of these methods is their reliance on perceptual data to infer spatial relationships—A correlation that is not always consistent. Exceptions such as similar scenes and viewpoint rotation highlight a fundamental gap between perceptual similarity and spatial distances. Although some studies recognize this issue, their solutions, such as adopting panoramic vision [19], [23], [24], only partially address the problem without fully resolving the perceptual-spatial disconnect. In response to these challenges, our paper proposes inte-

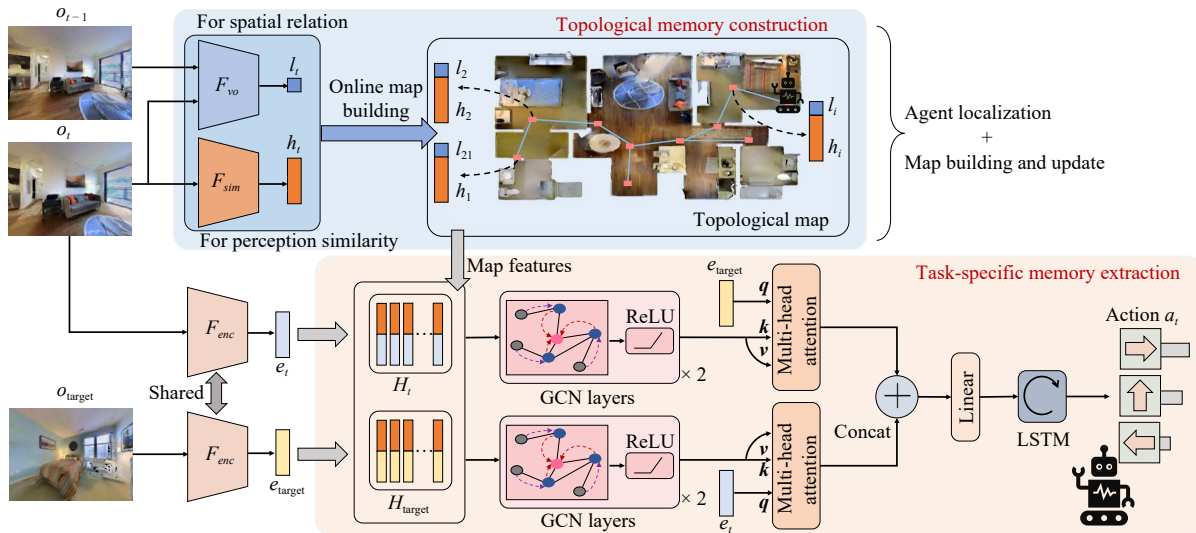


Fig. 1. Overall system architecture. To construct reliable topological memory, we introduce a two-factor validation that integrates perceptual similarity estimation and odometry pose estimation simultaneously. This strategy mitigates the distribution disparity between the spatial and perceptual domains during map construction. To utilize memory cues in the topological map, we encode the cognitive features of the topological map using graph convolutional networks. Subsequently, adaptive attention is employed to extract task and environment-specific memory information, which guides navigation decisions.

grating neural odometry with similarity assessments to independently evaluate distances in both spatial and perceptual domains. This dual-pronged approach lays the groundwork for a more robust and dependable logic in topological map construction that is effective even with limited viewpoint data.

C. Spatial Memory Extraction From Topological Maps

Topological maps inherently showcase the positions of drivable areas and their interconnectivity, facilitating a straightforward navigation strategy [10], [14], [19]–[21]. This method begins by localizing the robot’s current and target observations within the topological map. Next, a pathfinding algorithm, such as Dijkstra [35], computes the route from the robot’s location to the target. The derived waypoints are then dispatched to the lower-level control system for execution. While this technique capitalizes on the strengths of topological mapping, it encounters difficulties in creating smooth and efficient trajectories, attributed to the inherent sparsity of topological data. Moreover, the approach does not fully harness the global perceptual insights present in the topological map, as it primarily serves localization purposes and overlooks the information outside the computed route.

Recent research is exploring the integration of neural networks to implicitly leverage topological memories for navigation. For instance, works such as [22], [36], [37] employ graph neural networks (GNNs) to facilitate end-to-end action generation. Reference [13] advances this concept by using a Transformer to encode and decode features within the graph. Reference [28] introduces hierarchical topology to provide a detailed representation of expansive environments. Reference [34] enhances the interpretability of edges by grounding node positions in real-world coordinates, supporting high-level decision-making. While the aforementioned methods implicitly utilize historical data, our paper delves into the nuanced challenge of adaptive memory extraction. We propose a system that enables the agent to selectively concentrate on the

historical information most pertinent to the current task state and objectives, thereby optimizing decision-making efficacy.

III. METHODOLOGY

A. Overall System Architecture

We propose a novel approach for visual navigation that employs an online constructed topological map. To ensure robust and reliable topology construction in complex scenarios, we introduce a two-factor verification logic that combines perception similarity estimation with neural odometry. This bridges the distribution gap between the spatial and perceptual domains. To extract task-specific information from the continually updated memory structure, we introduce the TMFT. TMFT effectively utilizes the inherent spatial structural information of the topological map. Capitalizing on the topological map’s compact spatial maintenance, our design enhances long-term task comprehension and improves navigation performance. The overall system structure is illustrated in Fig. 1 and involves two key steps:

1) *Topological Memory Construction*: This step uses perceptual similarity estimation and spatial distance estimation to perform online, incremental construction and updates of the topological map. The integration of neural odometry allows a two-factor verification of the map in both perceptual and spatial domains, which can effectively tackle the issues related to redundant nodes and incorrect edge connections, and enhance the reliability of memory construction.

2) *Task-Specific Memory Extraction*: To fully exploit the spatial memory maintained in the topological map, we propose the TMFT to perform cross decoding on the memory structure, incorporating both current and target observations. This process enables the reliable extraction of specific environmental spatial structural information, and enhances the understanding of navigation tasks from a broader perspective.

In this paper, we consider the image-goal navigation task

where a robot needs to reach the corresponding position of a given target image using an onboard camera. At each time step t , the robot receives a first-person perspective RGBD image observation o_t , and outputs an action, $a_t \in \mathcal{A}$, where the discrete action space $\mathcal{A} = \{\text{go straight, turn left, turn right}\}$. The navigation is considered successful when the robot reaches the target point within 1 m.

B. Topological Memory Construction

The existing methods for constructing topological maps often compare similarities between various observations within a scene, mapping perceptual relationships to spatial locations. This can be problematic, particularly in complex environments or with frequent rotation motions. Our paper introduces a more precise method, integrating perceptual similarity estimation with spatial estimation via neural odometry. We believe this technique has three main benefits. First, it addresses different aspects of localization—similarity estimation evaluates perceptual interrelationships, while odometry focuses on spatial distances. Second, it enhances the reliability and fault tolerance of memory construction, as both perceptual similarity and odometry provide estimative values, not requiring high precision. Lastly, our method eliminates the need for a panoramic view, utilizing a more practical and cost-effective limited field-of-view camera.

We implement an image encoder F_{sim} to encode observations at given moments. We use those encoded vectors from different observations to quantify the similarity of image pairs. To correct any localization errors, a neural odometry F_{vo} is employed to estimate the pose transformation from adjacent frame observations. The robot's global pose is progressively accumulated using the ego-motion estimations. In our implementation, we utilize a ResNet-18 structure for F_{sim} [38]. For F_{vo} , we adopt the strategy from [39], which includes processes like image pre-processing, concatenation of adjacent frames, convolution, and the output of egomotion. We use $V = \{v_1, v_2, \dots, v_{N_t}\}$ to represent the set of topological nodes, where N_t is the current number of nodes in the map. Each node v_i stores the encoded vector h_i instead of the raw image observation to reduce storage requirements. This approach allows us to calculate perceptual similarity by leveraging the previously stored visual features $H = \{h_1, h_2, \dots, h_{N_t}\}$, thus eliminating the need for additional computations. Additionally, each node retains the estimated global pose p_i , acquired through neural odometry. The detailed topological memory construction procedure can be described in Algorithm 1.

1) *Agent Localization*: At each time step t , we assume the current observation o_t is at an unlocalized virtual node v_t with encoded observation h_t , and global pose l_t . The similarity sequence $S = \{s_i | s_i = \cos(h_i, h_t), i = 1, 2, \dots, N_t\}$ and the spatial distance sequence $D = \{d_i | d_i = |l_i - l_t|, i = 1, 2, \dots, N_t\}$ are computed. If there exists a node $v_k \in V$ whose similarity s_k with the current observation is greater than the threshold s_{th} , we localize v_t at v_k in the perceptual domain. If there are multiple nodes whose similarity scores are greater than s_{th} , we select the node with the highest similarity score. Additionally, a neu-

ral odometry is employed to aid in the localization process and offer secondary verification. For the obtained node v_k mentioned above, if the predicted spatial distance d_k between v_k and v_t exceeds threshold d_{th} , the robot is more inclined to rely on the location information from odometry. This suggests that the two visually similar nodes are spatially distant, so the robot should not be localized at v_k . In this case, v_t is estimated to be at $v_{k'}$, where $k' = \text{argmin}(D)$.

Algorithm 1 Topological Memory Construction

Data:

node of online updating topological map $V = \{v_1, v_2, \dots, v_{N_t}\}$
node embedding $H = \{h_1, h_2, \dots, h_{N_t}\}$
location estimated by odometry $L = \{l_1, l_2, \dots, l_{N_t}\}$
agent's current location and last location on topological map v_t, v_n
agent's current observation embedding h_t

```

1: for  $t$  in maxstep  $T$  do
2:    $S \leftarrow \{s_i | s_i = \cos(h_i, h_t), i = 1, 2, \dots, N_t\}$ 
3:    $D \leftarrow \{d_i | d_i = |l_i - l_t|, i = 1, 2, \dots, N_t\}$ 
4:    $s_k \leftarrow \max(S), d_{k'} \leftarrow \min(D)$ 
5:   if  $s_k \geq s_{th}$  then
6:     LOCALIZATION
7:     if  $d_k \leq d_{th}$  then
8:        $v_t \leftarrow v_k, h_k \leftarrow h_t$ 
9:       add edge  $(v_t, v_n)$  on map
10:    else
11:       $v_t \leftarrow v_{k'}, h_{k'} \leftarrow h_t$ 
12:    end if
13:  else
14:    GRAPH UPDATE
15:    if  $\text{card}(\delta(v_t, B)) < M$  then
16:      add  $v_t$  to  $V$ 
17:      add edge  $(v_t, v_n)$  on map
18:    end if
19:  end if
20: end for
21: return

```

After successful localization, the perception feature of the localized node is substituted with the current feature h_t , and an edge is established between the localized node and the last localized node.

2) *Map Building and Update*: If the current virtual node v_t cannot be localized in the existing graph ($\max(S) < s_{th}$), v_t is created as a new node and connected to v_n . Afterwards, we utilize global position information obtained from odometry to regularize the graph and achieve more accurate graph construction: a) Nodes that are far apart in spatial distance are not connected, if the spatial distance d_{ij} between nodes v_i and v_j meets the condition that $d_{ij} = |l_i - l_j| > d_{th}$; b) To ensure a rational distribution of nodes, it is imperative to limit the number of nodes within a defined spatial range. Supposing that v_t is a newly generated node, if the number of nodes within the neighborhood $\delta(v_t, B)$ centered at v_t with a radius of B , is greater than threshold M , v_t replaces the earliest generated node in $\delta(v_t, B)$.

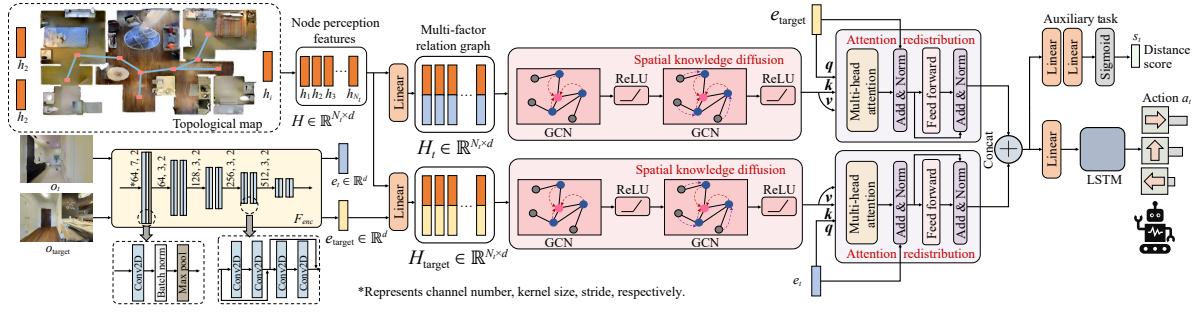


Fig. 2. Detailed structure of topological memory extraction. First, the current and target observations are encoded using F_{enc} . The encoded vectors are individually fused with the feature vectors H stored in the topological nodes. The TMFT consists of following two processes. Spatial knowledge diffusion: Feed the fused representations into GCNs to implicitly learn the global memory cues from the topological map. Attention redistribution: Reallocate attention to the nodes using cross-attention on both the current perception and target clues, which provides crucial cognitive features for generating navigation actions.

C. Task-Specific Memory Extraction

In order to extract task-related information from a scene’s topology for subsequent navigation control, we design a memory extraction mechanism called topology multi-factor transformer (TMFT). This mechanism, as depicted in Fig. 2, comprises two crucial processes: spatial knowledge diffusion and attention redistribution. Spatial knowledge diffusion is designed to implicitly learn the global spatial distribution of the entire scene. It takes into account the current observation, target information, and cognitive clues within the topological map. Following this, attention redistribution is employed. This process extracts key cognitive features from the merged multi-factor relation graph. These extracted features are then used to generate navigation actions.

1) *Spatial Knowledge Diffusion*: This process aggregates information from neighboring topological nodes, potentially expanding the agent’s perception field while also integrating visual cues for subsequent information extraction. We implement F_{enc} to first encode the current observation o_t and the target observation o_{target} into feature embedding e_t and e_{target} . Subsequently, we separately concatenate e_t and e_{target} with encoded vectors $H = \{h_1, h_2, \dots, h_{N_t}\}$ stored in the topological map, which are then passed through a linear layer FC to obtain \mathcal{H}_t and $\mathcal{H}_{target} \in \mathbb{R}^{N_t \times d}$. We feed the two vectors into a multi-layer GCN [40] with L layers for feature extraction, yielding multi-factor memory M_t and $M_{target} \in \mathbb{R}^{N_t \times d}$, respectively. Within each graph convolutional layer, the node feature matrix \mathcal{H}_t or $\mathcal{H}_{target} \in \mathbb{R}^{N_t \times d}$ of the topological graph undergoes a feature projection using the parameter matrix $W \in \mathbb{R}^{d \times d}$, and the projected matrix is then aggregated with the features of neighboring nodes. The graph convolutional process can be summarized as follows:

$$\mathcal{H}_{\mathcal{E}}^{(0)} = \text{FC}([H, e_{\mathcal{E}}]) \quad (1)$$

$$\mathcal{H}_{\mathcal{E}}^{(K)} = \text{ReLU}\left(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} \mathcal{H}_{\mathcal{E}}^{(K-1)} W^{(K-1)}\right) \quad (2)$$

$$\mathcal{M} = \mathcal{H}_{\mathcal{E}}^{(L)} \quad (3)$$

where K represents the K -th layer, \mathcal{E} represents subscript t or target, and correspondingly, \mathcal{M} represents M_t or M_{target} . \hat{A} means the adjacency matrix with self-loops and \hat{D} corresponds to its degree matrix.

2) *Attention Redistribution*: Graph convolution aggregates information from neighboring nodes, allowing the agent to broaden its perception field based on topological memory. We anticipate that this will enable the agent to prioritize attention towards task-related cues, such as the target and surrounding structural information, thus enhancing spatial awareness and task reasoning abilities. To accomplish this, our TMFT structure reallocates attention to the feature vector graphs M_t and M_{target} . This method captures relationships between distant nodes in the topological graph, effectively addressing the over-smoothing issue commonly found in GNNs [41].

We use e_t as query and M_{target} as key and value to apply the multi-head attention [42] and generate the feature vector $x_t \in \mathbb{R}^d$

$$Q_i = e_t W_i^q, \quad K_i = M_{target} W_i^k, \quad V_i = M_{target} W_i^v \quad (4)$$

$$head_i = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d}}\right) V_i \quad (5)$$

$$x_t = \text{Concat}(head_1, head_2, \dots, head_{NH}) W^O \quad (6)$$

where W_i^q , W_i^k , and $W_i^v \in \mathbb{R}^{d \times d}$ are parameter matrices for the i -th attention head, and $W^O \in \mathbb{R}^{(NH \times d) \times d}$ is the output parameter matrix. The ‘‘Concat’’ here refers to concatenating NH attention heads $head_i \in \mathbb{R}^d$ along the feature channel dimension for future fusion. Similarly, we can obtain $x_{target} \in \mathbb{R}^d$ by reallocating attention to M_t using e_{target} .

We then apply residual connections and layer normalization to x_t and x_{target} , and then feeding them into a fully connected layer to obtain y_t and y_{target}

$$\hat{x}_t = \text{LN}(x_t + e_t) \quad (7)$$

$$y_t = \text{LN}(\text{ReLU}(\text{FC}(\hat{x}_t)) + \hat{x}_t) \quad (8)$$

where LN represents layer normalization, and FC denotes a fully connected layer. It should be noted that the generation process of y_t and y_{target} is identical, with their attention structures being the same but their parameters being independently trained. Subsequently, y_t and y_{target} are concatenated to form the overall cognitive feature, which is finally utilized by stack of MLPs and LSTM layers to learn the action policy. We note that TMFT does not contain any trainable parameters associated with the number of nodes N_t , and thus it is capable of handling topological memories of changing structures.

To ensure the interpretability of TMFT operations, we introduce an auxiliary task to predict the distance score (refer to (11)) between target and current positions. The context vectors y_t and y_{target} are concatenated and passed through two linear layers to output the distance score s_t . Precisely predicting such distance score facilitates the TMFT in acquiring a deeper understanding of task-specific cognitive information, ultimately bolstering the robot's comprehension of navigation tasks.

D. Training

We utilize a two-phase optimization approach to train the network modules. We first train the visual encoder F_{sim} and visual odometry F_{vo} for memory construction. Then we freeze the parameters of F_{sim} and F_{vo} and train the memory extraction module TMFT as well as the navigation policy.

To train F_{sim} , we sample 5k random observations per training scenario. Similarity labels for each observation pair are set based on geometric rules: A label of 1 is given if the orientation between observations is less than 45° and distance less than 1.5 m, otherwise the label is -1 . The cosine similarity loss between observation pairs is minimized during the F_{sim} training. F_{vo} is trained using two consecutive frame images to estimate pose transformation. The dataset for training F_{vo} is collected following the method in [39]. The L2 loss is used to minimize the difference between the transformation estimated by F_{vo} and the ground truth transformation.

To train memory extraction module as well as the navigation policy, we employ imitation learning (IL) where cross-entropy loss is implemented to minimize the log-likelihood between the network's output actions and the expert actions. The loss function for IL is

$$\mathcal{L}_{IL} = \mathbb{E} \left[- \sum_{t=0}^{t=T} a_t^* \log(p(a_t|o_t)) \right] \quad (9)$$

where a_t^* represents the expert action of the robot at the time step t , and T denotes the length of the trajectory. The training set contains 4 k episodes with 300 k state-action pairs from 30 highly realistic indoor environments.

The auxiliary task of predicting distance scores can be regarded as a regression problem. We employ the widely used L2 loss in regression tasks for its ability to facilitate smooth convergence

$$\mathcal{L}_{aux} = \mathbb{E} \left[\sum_{t=0}^{t=T} \|s_t^* - s_t\|^2 \right] \quad (10)$$

where s_t^* represents the ground truth distance score between the current position and the target point

$$s_t^* = \max(1 - \text{dis}/d_{\text{max}}, 0) \quad (11)$$

where $d_{\text{max}} = 3$ m and dis is the actual distance from the current position to the target. This suggests that s_t^* represents the normalized distance score, which is non-zero only when the robot is within a 3m range of the target. s_t^* increases as the robot approaches the target. The auxiliary task is trained simultaneously with IL. The overall loss is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{IL} + \mathcal{L}_{aux}. \quad (12)$$

IV. IMPLEMENTATION

1) *Task Settings*: We conduct image-goal navigation using Habitat simulator [43] in the photorealistic Gibson dataset [44]. The objective of image-goal navigation is to reach the position of a given target image within a maximum timestep N_m while avoiding obstacles. The robot carries a monocular RGBD camera in the body front with 144×192 resolution and 90° horizontal field of view. The hyperparameters are set as follows. $N_m = 300$, $s_{th} = 0.4$, $d_{th} = 1.0$, $M = 3$, $B = 1.2$. N_m is chosen based on the size of the scenes, while the selection of s_{th} , d_{th} , M and B relies on our sensitivity test and implementation experience to achieve an appropriate node density for the topological map.

2) *Evaluation*: During testing, we employed a separate set of 7 environments, distinct from those used for training. The test dataset consists of 630 episodes, categorized into three levels of difficulty based on the distance between the starting point and the target: a) *Easy*: 1.5 m–3.0 m; b) *Medium*: 3.0 m–5.0 m; c) *Hard*: 5.0 m–10.0 m. We use the following three standard metrics to evaluate the navigation performance.

i) Success rate (SR) \uparrow : The ratio between the successful and the total numbers of episodes.

ii) Success weighted by path length (SPL) \uparrow : Estimates the path efficiency along with the success rate

$$SPL = \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{l_i}{\max(p_i, l_i)} S_i \quad (13)$$

where N_e is the number of episodes, l_i is the optimal path length between current and target positions in episode i , p_i is the actual path length executed by the agent in episode i , and S_i indicates whether the i -th episode is successful.

iii) Distance to success (DTS) \downarrow : measures the closest distance to the target, which is averaged between different episodes

$$DTS = \frac{1}{N_e} \sum_{i=1}^{N_e} \max(\|x_i - g_i\|_2 - d, 0) \quad (14)$$

where $\|x_i - g_i\|_2$ is the L2 of closest distance to goal in i -th episode and d is the success threshold distance (1 m).

3) *Baseline and Ablation Models*: We introduce the following baseline or ablation models to compare their navigation performances: a) *Reactive* [2]: The robot navigates in the environment without any memory, i.e., the robot extracts perception features and then directly generates actions using MLP; b) *Nav A3C* [3]: An internal-memory-based method that maintains memory using LSTM; c) *ANS* [17]: Establishes a metric spatial map and employs hierarchical policies by selecting waypoints to navigate. d) *Multi-store memory (MSM)* [33]: Combines short-term working memory and long-term episodic memory to generate actions; e) *VGM* [13]: Uses visual input to construct a topological map for environment cognition, and generate actions using GNNs; f) *NRNS* [20]: Learns to self-localize and navigate in the environment from passive videos, where topological map is utilized for scene representation. g) *Ours w/o TMFT*: The ablation of our model, which directly utilizes raw topological features for map feature fusing and extraction.

TABLE I
PERFORMANCE METRICS OF DIFFERENT BASELINE OR ABLATION MODELS UNDER TESTING ENVIRONMENTS WITH DIFFERENT DIFFICULTIES

	Easy			Medium			Hard			Total		
	SR \uparrow	SPL \uparrow	DTS \downarrow	SR \uparrow	SPL \uparrow	DTS \downarrow	SR \uparrow	SPL \uparrow	DTS \downarrow	SR \uparrow	SPL \uparrow	DTS \downarrow
Reactive [2]	50.00%	0.332	0.527	21.74%	0.129	1.677	10.58%	0.062	4.087	27.44%	0.174	2.097
Nav A3C [3]	64.71%	0.311	0.325	41.88%	0.193	0.915	12.44%	0.047	3.023	39.68%	0.184	1.421
MSM [33]	64.00%	0.412	0.364	51.82%	0.341	0.753	32.24%	0.176	2.084	49.35%	0.310	1.067
ANS [17]	65.74%	0.390	0.235	47.67%	0.324	1.010	24.91%	0.191	2.662	46.11%	0.302	1.302
VGM [13]	66.97%	0.464	0.260	54.69%	0.373	0.677	42.16%	0.287	1.629	54.61%	0.375	0.855
NRNS [20]	52.07%	0.364	0.718	36.90%	0.276	1.385	20.00%	0.125	2.570	36.32%	0.255	1.558
Ours w/o TMFT	78.70%	0.476	0.165	66.30%	0.364	0.420	44.32	0.254	1.254	63.11%	0.365	0.613
Ours	87.04%	0.528	0.105	67.72%	0.351	0.486	50.27%	0.289	1.299	68.35%	0.389	0.630

V. RESULTS

In this section, we evaluate the navigation performance of various models. Furthermore, to demonstrate the effectiveness of our topology-based navigation pipeline, we conduct ablation and interpretability experiments. Lastly, we deploy our model on a physical robot to verify its adaptability.

A. Navigation Performance Evaluation

Table I illustrates the performance metrics of different models in various difficulty settings. Overall, our proposed framework demonstrates a significant improvement in the robot’s image-goal navigation performance.

- In comparison to the Reactive model lacking memory ability and the Nav-A3C model using an implicit memory approach, our Full model demonstrates a remarkable increase in overall success rates by 149.1% and 72.3%, respectively. This indicates the necessity of employing an explicit memory structure in navigation tasks. Furthermore, when compared to the MSM model that utilizes a memory pool, our Full model exhibits a notable improvement of 38.5% in success rate. In contrast to the metric map-based model (ANS), our Full model also improves success rate by 48.23%. This suggests that, in comparison to other structured and unstructured memory representations, topological maps offer more substantial enhancements in navigation performance. We contend that the interpretable structures and compact nature of topological maps reduce the need to maintain redundant environmental information, thereby facilitating a more dependable construction and extraction process.

- Compared to the NRNS and VGM models that also use topological maps for environmental structure representation, our Full model improves success rates by 88.2% and 25.2% respectively, and also boosts the SPL by 52.6% and 3.8% respectively. Unlike NRNS, we utilize a transformer architecture in the memory retrieval phase to reallocate attention on graph nodes, which captures longer-range dependencies. Relative to VGM, our TMFT structure enables cross-decoding of the topological map, improving task-specific memory utilization. These improvements may also attribute to the more reliable memory construction facilitated by our two-factor memory construction strategy.

- The Full model outperforms the ablation model that lacks

TMFT, showcasing superior navigation skills. This implies that TMFT potentially heightens the robot’s spatial awareness and task reasoning capabilities by enabling a better grasp of global information.

Behavioral Logic in Typical Scenarios: To better illustrate the behavioral logic of the proposed model, we visualize the robot’s trajectory in a typical testing scenario, as depicted in Fig. 3. It is evident that our method enables the robot to construct an understanding of the environment during the whole navigation process. Initially, the robot, with limited environmental knowledge, explores unseen areas while avoiding collisions. At time step t_1 , it realizes that the target is not in the bedroom and moves to explore the living room at time step t_3 . Its partial understanding of the environment helps it avoid repeated paths and search for the target along new traversable

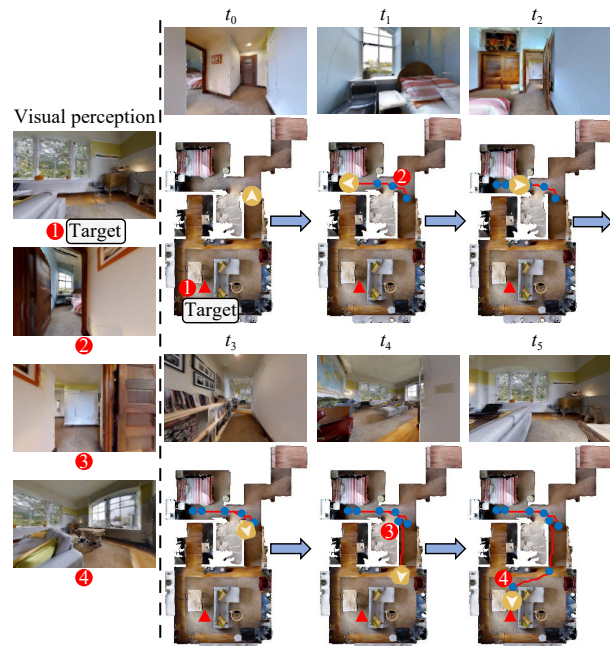


Fig. 3. Behavior visualization of our model at different cognitive stages in a typical navigation episode. The blue circles represent constructed topological nodes, and the red triangle indicates the target location. The model effectively utilizes memory information to search for targets in unseen environments while avoiding repetitive paths.

routes. At time step t_5 , upon identifying an image similar to the target, the robot moves decisively towards that area, showcasing its robust task reasoning abilities.

B. Visualization and Interpretability Experiments

1) *Sensitivity to Node Density*: To assess the impact of different node densities on navigation performance, we select five distinct s_{th} values and evaluate the agent's performance across various s_{th} . The sensitivity test results are depicted in Fig. 4. Overall, as s_{th} increases, the navigation performance of the model decreases. We speculate that this effect arises from the presence of redundant environmental information in dense topological maps, which can disrupt the robot's decision-making process. It also demonstrates that our model does not require an excessive representation of the environment. Instead, it effectively extracts task-specific information from sparse topological maps, leading to favorable navigation performance. Moreover, even when s_{th} is large, the decrease in model performance remains below 15%, and the model still outperforms most of the baseline models, showing a certain degree of robustness.

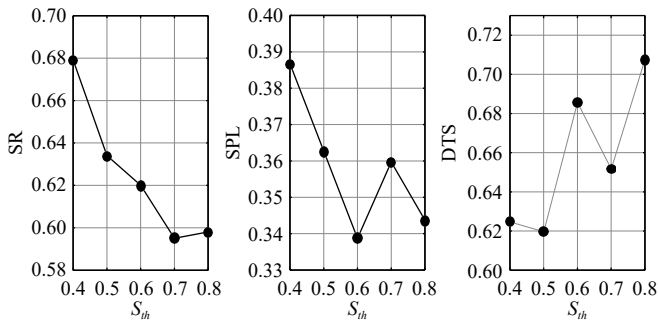


Fig. 4. Sensitivity test of navigation performance to the topological node density. The parameter s_{th} is used to control the density of nodes in the topological map, where a higher s_{th} value indicates a denser node distribution.

2) *Comparison of Topological Map Construction*: To better demonstrate the effectiveness of topological map construction in unseen environments, we showcase visualizations of topological maps produced by the Full model, the Full model without odometry (which relies solely on perception similarity for topology construction), and the VGM model (which also uses only perception similarity for topology construction, but with panoramic views). The mapping results are depicted in Fig. 5. Contrasting with the model that excludes odometry, our Full model prevents incorrect links between distant nodes by integrating spatial domain data from odometry. When compared to the VGM, our Full model eliminates superfluous nodes, ensures a balanced node distribution, and negates the need for panoramic observations.

To further illustrate the performance of our topology construction method in complex situations, we provide an example from the Quantico environment. As illustrated in Fig. 5, **A** and **B** represent two visually similar but distinct locations, while **C** and **D** represent different observation directions at the same location. The results suggest that the incorporation of a two-factor verification logic can lead to more reliable topol-

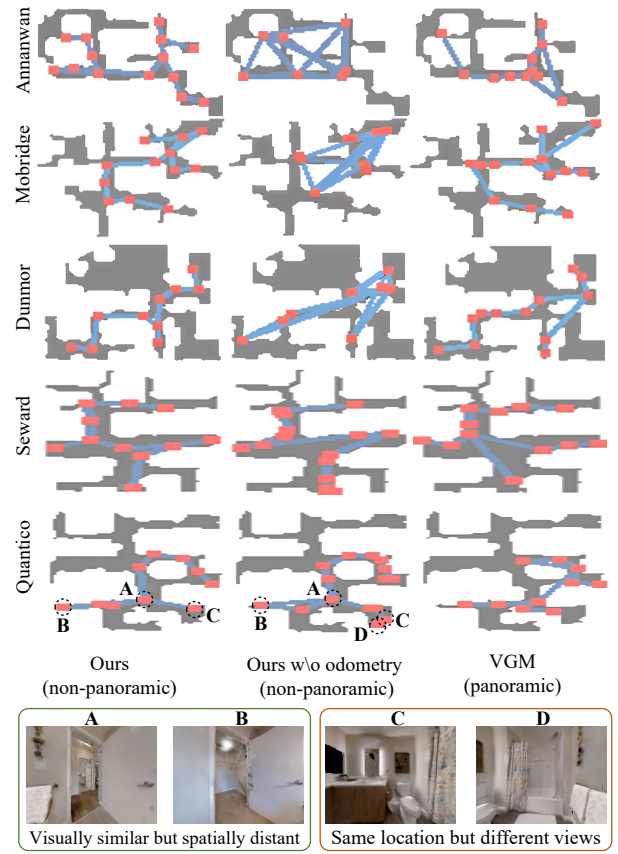


Fig. 5. Visualization of topological map construction. We visualize the mapping performance of different models in unseen environments with the same exploration trajectory. We demonstrate an example on how our model handles complex topology construction situations in Quantico. **A** and **B** represent two visually similar but distinct locations, while **C** and **D** represent different observation directions at the same location. Their corresponding perceptions are illustrated at the bottom.

ogy construction in challenging scenarios such as similar scenes and viewpoint rotation.

3) *Node Attention Visualization*: In Fig. 6, we visualize the process of topological map construction and attention allocation in a single episode. The attention scores for y_{target} are typically assigned to nodes near the robot, while the attention scores for y_i are usually allocated to nodes farther away from the robot. It is worth noting that when the robot enters the wrong room, y_i and y_{target} are more inclined to be allocated to the visited nodes that are more likely to reach the target. For example, at time step t_2 , when the robot enters the kitchen and finds the observation different from the target, the attention scores are predominantly allocated to the nodes closer to the target. Consequently, at time step t_3 , the robot exits the kitchen, returns to the previously visited nodes, and continues to search for the target. We believe that the TMFT structure effectively facilitates the robot's exploration of possible target areas in the topological map, guiding the robot's target search in unseen environments.

C. Real-World Experiment

We employ our Full model on an embodied agent in a real-world scenario to validate its adaptability and transferability.

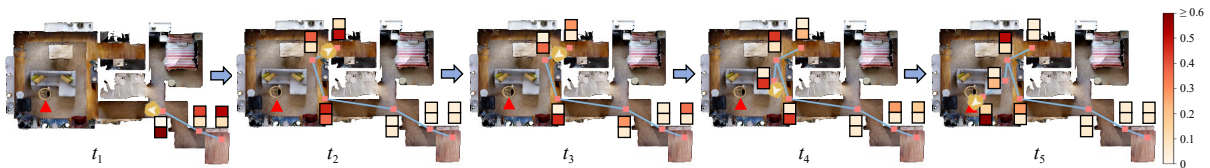


Fig. 6. Visualization of the node attention in a typical episode. The pink rectangles correspond to the nodes within the topology graph and the red triangle represents the target point. Adjacent to each node, there are two rectangular boxes indicating the attention scores assigned to the node by the attention redistribution module. The upper box represents the attention score for y_t , while the lower box represents the attention score for y_{target} . Darker color indicates more attention on that node.

The experimental site consists of a square area measuring $5\text{ m} \times 5\text{ m}$ with scattered obstacles. A turtlebot robot equipped with an RGBD camera is utilized to perform image-goal navigation tasks. To mitigate the impact of perceptual disparities between simulation and real-world on navigation performance, we employ transfer learning. Initially, a human expert guides the robot manually towards the target to collect data. Subsequently, we use the collected data to retrain the Full model using the strategy in Section III-D, enabling it to adapt to real-world environments.

We present two typical test episodes in Fig. 7. The experimental results demonstrate the effectiveness of our method in enabling the robot to create a coherent and standardized topological map of the environment during real-world navigation. This, in turn, facilitates the extraction of task-specific information from the topological map and promotes the generation of robust navigation strategies. The aforementioned results obtained in real-world scenarios align with the conclusions derived from the simulations, thereby validating the potential applicability of our system in real-world settings.

pipeline that capitalizes on topological memory. Initially, we introduce a two-factor verification technique for topological map construction by integrating both pose estimation and perceptual similarity assessment. Second, we present a neural-based memory extraction structure, TMFT. This structure enables the robot to concentrate on more critical spatial memory based on task progress and target cues. Our design allows for more reliable memory construction in complex environments, and the adaptive memory retrieval approach aligns better with human intuition. To validate the effectiveness of our proposed system, we conduct image-goal navigation experiments, visualization, and interpretability tests, which demonstrate superior performance. We also implement our system in real-world environments. Looking ahead, our future work will focus on developing a new paradigm for topological map generation which moves away from explicit similarity comparison pipeline. Additionally, we are developing hierarchical topology structures to facilitate the preliminary exploration behavior of robots in unseen scenarios.

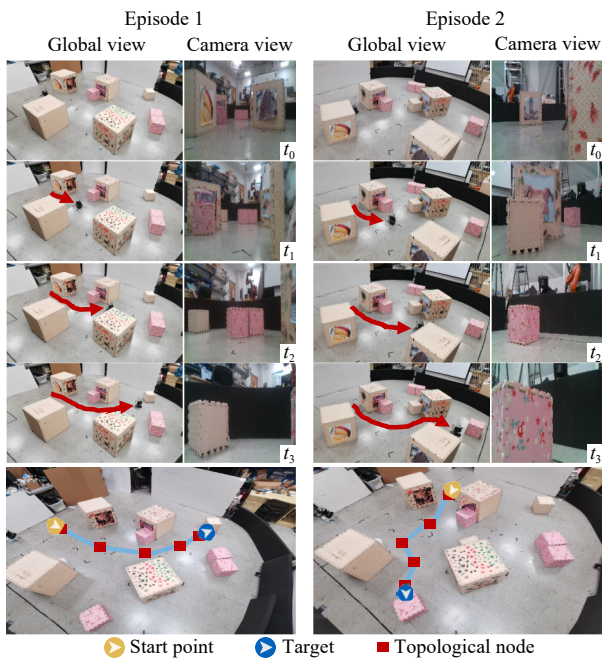


Fig. 7. Real word experiment results. We record the global view and camera view during the image-goal navigation process. The last row displays the topological map.

VI. CONCLUSION

In this paper, we propose a learning-based navigation

REFERENCES

- [1] B. Li, Z. Huang, T. Chen, T. Dai, Y. Zang, W. Xie, B. Tian, and K. Cai, "MSN: Mapless short-range navigation based on time critical deep reinforcement learning," *IEEE Trans. Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8628–8637, 2023.
- [2] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2017, pp. 3357–3364.
- [3] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, "Learning to navigate in complex environments," in *Proc. Int. Conf. Learning Representations*, 2017, pp. 1–11.
- [4] L. Jiang, H. Huang, and Z. Ding, "Path planning for intelligent robots based on deep Q-learning with experience replay and heuristic knowledge," *IEEE/CAA J. Autom. Sinica*, vol. 7, pp. 1179–1189, 2020.
- [5] A. Singla, S. Padakandla, and S. Bhatnagar, "Memory-based deep reinforcement learning for obstacle avoidance in UAV with limited environment knowledge," *IEEE Trans. Intelligent Transportation Systems*, vol. 22, no. 1, pp. 107–118, 2021.
- [6] G. Georgakis, K. Schmeckpeper, K. Wanchoo, S. Dan, E. Mitsakaki, D. Roth, and K. Daniilidis, "Cross-modal map learning for vision and language navigation," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2022, pp. 15439–15449.
- [7] Z. Gao, J. Qin, S. Wang, and Y. Wang, "Boundary GAP based reactive navigation in unknown environments," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 2, pp. 468–477, 2021.
- [8] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 7272–7281.
- [9] T. Wang, X. Xu, F. Shen, and Y. Yang, "A cognitive memory-

- augmented network for visual anomaly detection,” *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 7, pp. 1296–1307, 2021.
- [10] N. Savinov, A. Dosovitskiy, and V. Koltun, “Semi-parametric topological memory for navigation,” in *Proc. Int. Conf. Learning Representations*, 2018, pp. 1–12.
- [11] X. Meng, N. Ratliff, Y. Xiang, and D. Fox, “Scaling local control to large-scale topological navigation,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2020, pp. 672–678.
- [12] L. Mezghan, S. Sukhbaatar, T. Lavril, O. Maksymets, D. Batra, P. Bojanowski, and K. Alahari, “Memory-augmented reinforcement learning for image-goal navigation,” in *Proc. IEEE/RSS Int. Conf. Intelligent Robots and Systems*, 2022, pp. 3316–3323.
- [13] O. Kwon, N. Kim, Y. Choi, H. Yoo, J. Park, and S. Oh, “Visual graph memory with unsupervised representation for visual navigation,” in *Proc. IEEE/CVF Int. Conf. Computer Vision*, 2021, pp. 15870–15879.
- [14] R. R. Wiyatno, A. Xu, and L. Paull, “Lifelong topological visual navigation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9271–9278, 2022.
- [15] N. Kim, O. Kwon, H. Yoo, Y. Choi, J. Park, and S. Oh, “Topological semantic graph memory for image-goal navigation,” in *Proc. 6th Annual Conf. Robot Learning*, 2023, pp. 393–402.
- [16] Y. Wu, Y. Wu, A. Tamar, S. Russell, G. Gkioxari, and Y. Tian, “Bayesian relational memory for semantic visual navigation,” in *Proc. IEEE/CVF Int. Conf. Computer Vision*, 2019, pp. 2769–2779.
- [17] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, “Learning to explore using active neural slam,” in *Proc. Int. Conf. Learning Representations*, 2020, pp. 1–13.
- [18] H. Choset and K. Nagatani, “Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization,” *IEEE Trans. Robotics and Automation*, vol. 17, pp. 125–137, 2001.
- [19] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, “Neural topological slam for visual navigation,” in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2020, pp. 12872–12881.
- [20] M. Hahn, D. S. Chaplot, S. Tulsiani, M. Mukadam, J. M. Rehg, and A. Gupta, “No RL, no simulation: Learning to navigate without navigating,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 26661–26673, 2021.
- [21] Y. Hu, B. Subagdja, A.-H. Tan, and Q. Yin, “Vision-based topological mapping and navigation with self-organizing neural networks,” *IEEE Trans. Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7101–7113, 2022.
- [22] D. Li, Q. Zhang, and D. Zhao, “Graph attention memory for visual navigation,” in *Proc. 4th Int. Conf. Data-Driven Optimization of Complex Systems*, 2022, pp. 1–7.
- [23] A. Taniguchi, F. Sasaki, and R. Yamashina, “Pose invariant topological memory for visual navigation,” in *Proc. IEEE/CVF Int. Conf. Computer Vision*, 2021, pp. 15364–15373.
- [24] A. Taniguchi, F. Sasaki, M. Muroi, and R. Yamashina, “Planning on topological map using omnidirectional images and spherical CNNs,” *Advanced Robotics*, vol. 36, no. 3, pp. 153–166, 2022.
- [25] K. Liu, T. Kurutach, C. Tung, P. Abbeel, and A. Tamar, “Hallucinative topological memory for zero-shot visual planning,” in *Proc. 37th Int. Conf. Machine Learning*, 2020, vol. 119, pp. 6259–6270.
- [26] B. Eysenbach, R. R. Salakhutdinov, and S. Levine, “Search on the replay buffer: Bridging planning and reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 32, pp. 15246–15257, 2019.
- [27] Z.-H. Yin and W.-J. Li, “Toma: Topological map abstraction for reinforcement learning,” arXiv preprint arXiv: 2005.06061, 2020.
- [28] Z. Ravichandran, L. Peng, N. Hughes, J. D. Griffith, and L. Carlone, “Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks,” in *Proc. Int. Conf. Robotics and Automation*, 2022, pp. 9272–9279.
- [29] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Trans. Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [30] F. Blochlinger, M. Fehr, M. Dymczyk, T. Schneider, and R. Siegwart, “Topomap: Topological mapping and navigation based on visual SLAM maps,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2018, pp. 3818–3825.
- [31] N. Ganganath, C.-T. Cheng, T. Fernando, H. H. C. Iu, and C. K. Tse, “Shortest path planning for energy-constrained mobile platforms navigating on uneven terrains,” *IEEE Trans. Industrial Informatics*, vol. 14, no. 9, pp. 4264–4272, 2018.
- [32] B. K. Patle, S.-L. Chen, A. Singh, and S. K. Kashyap, “Optimal trajectory planning of the industrial robot using hybrid s-curve-PSO approach,” *Robotic Intelligence and Automation*, vol. 43, pp. 153–174, 2023.
- [33] H. Sang, R. Jiang, Z. Wang, Y. Zhou, and B. He, “A novel neural multistore memory network for autonomous visual navigation in unknown environment,” *IEEE Robotics and Autom. Letters*, vol. 7, no. 2, pp. 2039–2046, 2022.
- [34] K. Chen, J. P. de Vicente, G. Sepulveda, F. Xia, A. Soto, M. Vazquez, and S. Savarese, “A behavioral approach to visual navigation with graph localization networks,” in *Proc. Robotics: Science and Systems*, 2019, pp. 1–10.
- [35] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.
- [36] F. Zhu, X. Liang, Y. Zhu, Q. Yu, X. Chang, and X. Liang, “SOON: Scenario oriented object navigation with graph-based exploration,” in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2021, pp. 12684–12694.
- [37] E. Beeching, J. Dibangoye, O. Simonin, and C. Wolf, “Learning to plan with uncertain topological maps,” in *Proc. European Conf. Computer Vision*, 2020, pp. 473–490.
- [38] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [39] X. Zhao, H. Agrawal, D. Batra, and A. Schwing, “The surprising effectiveness of visual odometry techniques for embodied pointgoal navigation,” in *Proc. IEEE/CVF Int. Conf. Computer Vision*, 2021, pp. 16107–16116.
- [40] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learning Representations*, 2017, pp. 1–10.
- [41] Y. Rong, Y. Bian, T. Xu, W. Xie, Y. WEI, W. Huang, and J. Huang, “Self-supervised graph transformer on large-scale molecular data,” in *Advances in Neural Information Processing Systems*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. San Francisco, USA: Curran Associates, Inc., 2020, pp. 12559–12571.
- [42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.
- [43] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, “Habitat: A platform for embodied AI research,” in *Proc. IEEE/CVF Int. Conf. Computer Vision*, 2019, pp. 9338–93469.
- [44] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, “Gibson ENV: Real-world perception for embodied agents,” in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2018, pp. 9068–9079.



Qiming Liu received the B.Eng. degree in automation from Shanghai Jiao Tong University in 2020. He is currently a Ph.D. candidate in control science and engineering at Shanghai Jiao Tong University. His current research interests include robot navigation, embodied AI, and reinforcement learning.



Xinru Cui is currently a graduate student in automation at Shanghai Jiao Tong University. His current research interests include robot learning, embodied AI, and intelligent navigation.

research interests include multi-robot cooperation and autonomous driving system.



Hesheng Wang (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from the Harbin Institute of Technology in 2002, and the M.Phil. and Ph.D. degrees in automation and computer-aided engineering from The Chinese University of Hong Kong, Hong Kong, China, in 2004 and 2007, respectively. He is currently a Professor with the Department of Automation, Shanghai Jiao Tong University. His current research interests include visual servoing, service robot, computer vision, and

autonomous driving.



Zhe Liu received the Ph.D. degree in control technology and control engineering from Shanghai Jiao Tong University in 2016. From 2017 to 2020, he was a Post-Doctoral Fellow with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China. From 2020 to 2022, he was a Research Associate with the Department of Computer Science and Technology, University of Cambridge, UK. Currently he is an Associate Professor with the MoE Key Laboratory of Artificial Intelligence, Shanghai Jiao Tong University. His current

Dr. Wang is an Associate Editor of *IEEE Transactions on Automation Science and Engineering*, *IEEE Robotics and Automation Letters*, *Robotic Intelligence and Automation* and the *International Journal of Humanoid Robotics*, a Technical Editor of the *IEEE/ASME Transactions on Mechatronics*, an Editor of Conference Editorial Board (CEB) of IEEE Robotics and Automation Society. He served as an Associate Editor of the *IEEE Transactions on Robotics* from 2015 to 2019. He was the General Chair of IEEE ROBIO 2022 and IEEE RCAR 2016, and the Program Chair of the IEEE ROBIO 2014 and IEEE/ASME AIM 2019. He will be the General Chair of IEEE/RSJ IROS 2025.