Knowledge-Driven Visual Target Navigation: Dual Graph Navigation

Albert Author[1] and Bernard D. Researcher[2]

***Abstract*— Visual target navigation is a critical task within the realm of embodied intelligence. The Existing end-to-end and modular approaches often encounter high computational demands, challenges in online updates, and limited generalization, which restrict their deployment on resource-constrained devices. To overcome these challenges, we introduce a knowledge-driven, lightweight image instance navigation framework, Dual Graph Navigation (DGN). The DGN constructs an External Knowledge Graph (EKG) using a limited dataset to capture prior correlation possibilities between objects, guiding the exploration process. During exploration, DGN builds an Internal Knowledge Graph (IKG) from an instance-aware module, recording explored regions and integrating with EKG to determine the next navigation target. The EKG is dynamically updated based on IKG, continually enhancing the system's adaptability to the current environment. Additionally, DGN's plug-and-play modular design supports independent training and flexible replacement of target recognition, keypoint extraction, and path planning algorithms, reducing training and deployment costs while improving adaptability across diverse environments. We deploy DGN on three types of real-world robot platforms (including edge without CUDA support) and simulation environments (AI2THOR, Habitat), and our experimental results demonstrate that it operates stably, achieving state-of-the-art performance on the ProcTHOR-10K dataset.**

I. INTRODUCTION

Visual navigation in unknown environments is a core challenge in the field of embodied intelligence, requiring robots to possess both scene understanding and autonomous navigation capabilities [1], [2]. Complex indoor environments impose strict requirements on a robot's real-time navigation abilities [3]. However, the existing approaches often need to be trained in specific environments and rely on high-performance GPU workstations to ensure real-time inference [4], [5]. High training costs and limited generalization capabilities of these approaches restrict their application in real-world scenarios. [6], [7], [8], [9]. Thus, it is crucial to develop algorithms that can operate efficiently on resource-constrained platforms while maintaining strong generalization capabilities [5], [10], [11].

Visual navigation approaches can be categorized into end-to-end [12], [13], [14], [15], [16] and modular approaches [17], [18], [19]. Both types of approaches typically have high computational complexity,making it difficult to deploy on resource-constrained devices. The end-to-end approaches based on reinforcement learning, map observations to actions through continuous interaction between the agent and the environment [20], but suffer from poor transferability [13], [21], sparse rewards, and high data dependency [22], [23], [24], [25]. The language-driven end-to-end navigation approaches use cross-modal models to aid navigation and improve generalization [16], [26], [27], [28]. But their large number of parameters results in high demands for computational and memory bandwidth, setting challenges for real-time operation on resource-constrained devices. [29]. Modular approaches, decompose tasks into specific subtasks realized by functional modules [30], [31], enabling high sample efficiency and stability [32], [33], [34], but they often depend on metric maps with semantic information, which are costly to construct and expand and require precise data and strict scene structure [13], [31].

To address the aforementioned issues, we propose a knowledge-driven lightweight image instance navigation framework, the Dual Graph Navigation (DGN), as illustrated in Fig. 1.The DGN leverages an External Knowledge Graph (EKG) and an Internal Knowledge Graph (IKG) to record prior knowledge and environmental structure, thereby reducing the dependence on precise metric information and extensive training. The framework adopts a modular design that supports the independent training and flexible replacement of functional modules, enhancing robustness and enabling efficient deployment on resource-constrained edge computing devices. Specifically,the DGN comprises three modules: environment perception guided by knowledge graphs, navigation decision-making, and path planning. In the environment perception module, the EKG is constructed from a small dataset to represent prior correlation probabilities, and offer exploration prompts. The IKG is built from instance-aware relationships formed during exploration (Sec. III-A.2), recording explored regions and prioritizing navigation targets while continuously updating the EKG throughout the navigation process. This knowledge-guided method with online updates, enhances navigation adaptability to the current environment. The decision-making module identifies potential targets from EKG, uses the IKG to assess priority, and determines the goal. Finally, the DGN uses the plug-and-play path planning module generate a trajectory, guiding the robot to the target.

The main contributions of this paper are as follows:

- We propose a knowledge-driven, lightweight visual instance navigation framework (DGN) that uses the EKG and IKG to record object category correlation and instance reachability, respectively. This method enables visual navigation without requiring large-scale data training or precise metric maps.
- The framework's modular plug-and-play design allows

*This work was not supported by any organization

[1]All authors are with the Department of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China. Emails: 201927018@mail.dlut.edu.cn

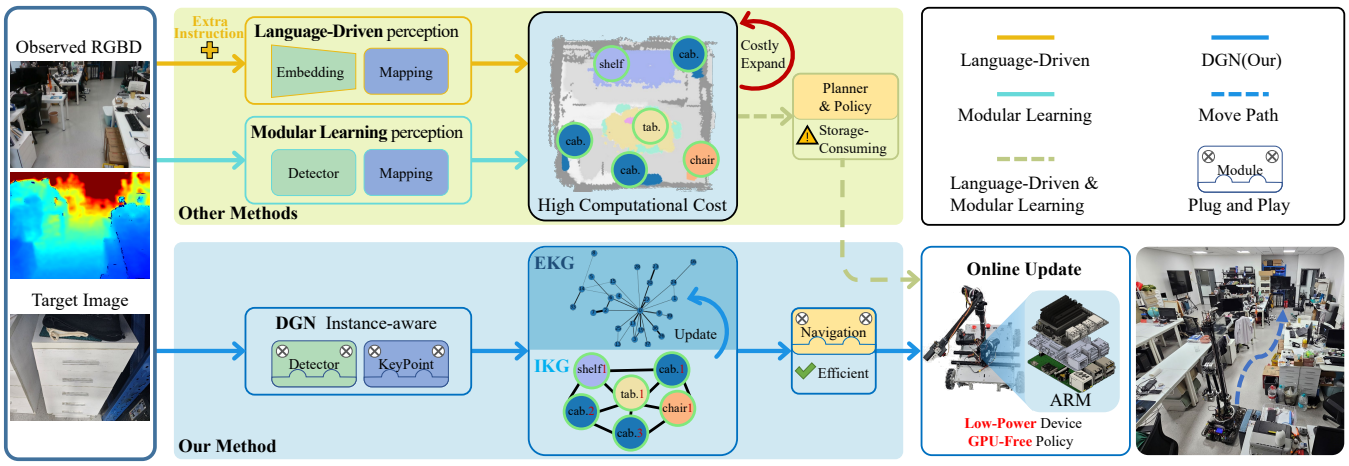

Fig. 1. Comparison between DGN and mainstream visual navigation methods. Mainstream visual navigation methods (Other methods) rely on RGBD input or driven by language, constructing semantic maps with tightly coupled perception modules, resulting in storage-consuming deployments. Our method (DGN) only utilizes RGBD input and constructs an Internal Knowledge Graph (IKG) using a plug-and-play instance-aware module, recording semantic information and topological relationships of instances while dynamically updating the External Knowledge Graph (EKG) with object category correlation. This enables efficient navigation that is deployable on low-power, low-computation edge devices.

for the flexible replacement of functional modules, enhancing the DGN's adaptability, scalability, and maintainability across diverse environments and embodiments.

- We evaluated the DGN in multiple mainstream simulation environments, achieving state-of-the-art performance on the ProcTHOR-10K [35] dataset. Additionally, the DGN has been successfully deployed on resource-constrained real-world robots, supporting online updates.

II. RELATED WORK

A. Visual Navigation

Visual navigation [12] is a long-standing robotic task where a robot navigate visually to locate a target or position based on a given image. Current methods are primarily divided into end-to-end and modular approaches. End-to-end methods [12], [13], [14] map observations directly to actions, offering a straightforward method but facing challenges such as low sample efficiency and poor generalization [36]. To address these issues, Target-driven RL [12] and ZER [13] optimize learning strategies for better training efficiency, while OVRL-V2 [14] and FGPrompt [22] enhance visual representation capabilities to infer target locations better. Recently, as language-driven models have performed well in image perception tasks, methods like ZSON [15] and CoW [16] have utilized CLIP [37] to obtain cross-modal information to enhance navigation performance. Despite the simplicity of end-to-end methods, they require massive-scale training and implicitly learn multiple subtasks, making the model difficult to fit and computationally burdened [32]. To break these limits, researchers comes up with modular approaches [17], [19], decomposing navigation into specialized sub-tasks addressed by distinct modules [38]. Classic modular methods like ANS [17] and Wu et al. [18] employ hierarchical planning, training separate modules for environment perception, navigation planning and local navigation to enhance enhancing learning capacity and efficiency. Apart from simple module replacement, recent methods like IEVE [19] propose a dynamic navigation method that actively switches between different modules of exploration, verification, and action utilization, improving the decision-making of agent ability in complex environments. these modular approaches train the overall navigation decision module based on reinforcement learning, leading to tight coupling between modules. They often rely on certain functional module models, which may lead to performance bottlenecks and narrow the system's optimization space. Our method supports independent training and the seamless replacement of different algorithms within modules, therefore extends the functionality of agents and overcome limitations in module selection.

B. Topological Perception

Environmental perception is essential for robotic navigation, supporting decision-making, and path planning. It can be divided into implicit, metric, and topological perception [39]. Implicit perception typically relies on RNN, LSTM, etc. to represent navigation states, which is simple structured and has a limited long-term memory. [40]. Metric perception provides precise localization and planning by adopting dense maps, though it is sensitive to sensor noise and costly to maintain [41]. Topological perception employs graph structures to represent features and relationships, allowing sparse representations while preserving long-term exploration memory, enhancing flexibility and robustness. Some topological approaches simplify metric perception, for example, in NTS [42] and SPTM [43] nodes are image features and edges are rough geometric data. Since these methods depend on handcrafted features and spatial information, some other approaches manage to reduce reliance on metric information, shifting the focus instead toward

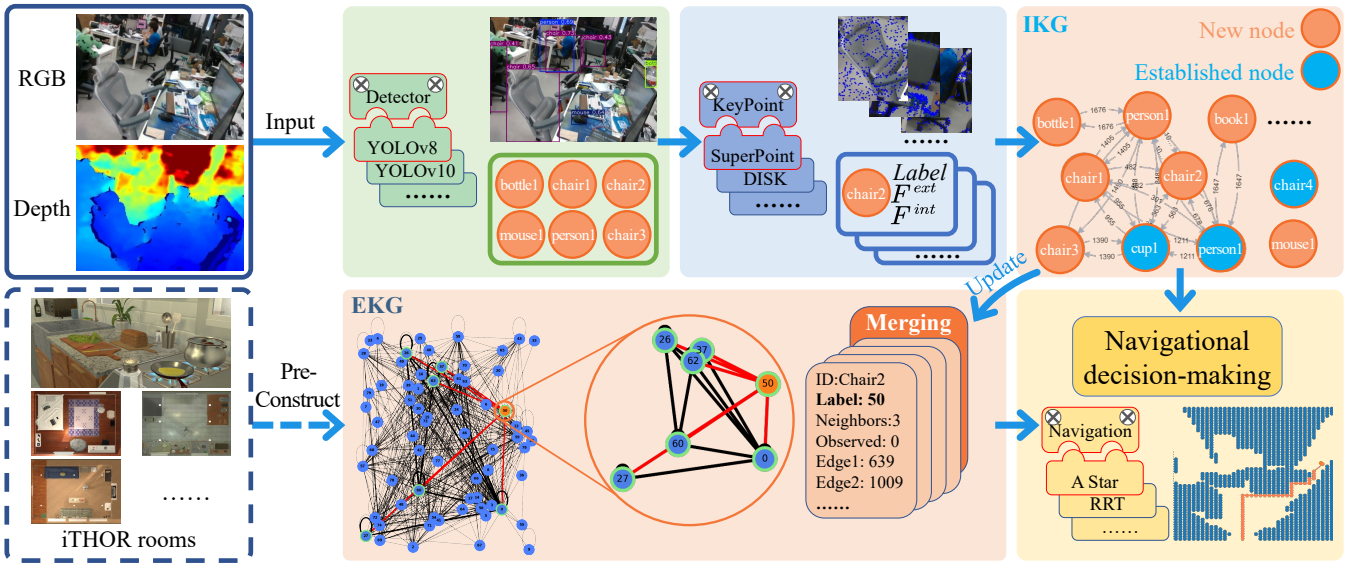

Fig. 2. The model receives RGBD input, with target recognition and keypoint extraction generating data to construct the IKG. The IKG then online updates the EKG built by iTHOR. Together, the IKG and EKG determine the next navigation point, and the path planning module outputs the motion path.

emphasizing logical relationships. Neural Planner [44] and VGM [45] construct graphs using predicted navigation positions and encoded visual features, minimizing dependence on pose data. However, these image-level node methods struggle to distinguish specific instances. To improve navigation granularity, TSGM [39] contains image nodes representing locations, object nodes with instances and edges linking adjacent images and targets in current image. Most topological methods focus on positional image data and overlook semantic relationships of objects, putting an upper limit on exploration efficiency. Our method addresses these limitations by using a dual-topology structure: The IKG for instance-level spatial representation and the EKG for exploration guidance considering object correlation.

III. Proposed Method

To enable resource-constrained indoor robots to perform image-based target navigation, we propose a lightweight, modular navigation framework. The overall architecture, is shown in Fig. 2. The DGN features a plug-and-play modular design, enhancing adaptability across diverse environments. Target recognition, keypoint extraction, and navigation modules can be replaced without fine-tuning. In our default setup, we employ YOLOv8 [46] for target recognition, SuperPoint [47] for keypoint extraction, and A-star [48] for path planning.

A. Environmental Perception

1) Instance Recognition: DGN adopts algorithm components of target recognition and keypoints extraction to extract the external (F^{ext}) and internal features (F^{int}) of target instances from the obtained RGBD data. The external features consist of the type and distance information of surrounding objects. The set of target nodes recognized from the image is $\{v_i\}$, where v_i represents the i-th instance target, and c_i represents its category. The external feature $F^{ext}(v_i)$ of v_i is given by

$$F_i^{ext} = [f_i[1], \ldots, f_i[n]] \tag{1}$$

$$f_i[j] = \sum_{c_k=j} d(v_i, v_k) \tag{2}$$

where $d(v_i, v_k)$ is the Euclidean distance between the observed objects v_i and v_k, and n is the number of recognizable categories. The internal feature $f^{int}(v_i)$ is composed of keypoint extraction of the target image, reducing background interference and computational overhead.

2) IKG Construction: We define the Internal Knowledge Graph representing the environmental map as $G_I = (V_I, E_I)$, where V_I is the set of target nodes, each corresponding to a recognized instance, and E_I is the set of edges representing the reachability relationships between instances. The reachability relationship weight w_{ij} is calculated by weighting the path length and obstacle coverage between instances v_i and v_j, given by

$$w_{ij} = \lambda_1 \cdot d(v_i, v_k) + \lambda_2 \cdot o(v_i, v_j) \tag{3}$$

where $o(v_i, v_j)$ is obtained by the ratio of the number of obstacles between v_i and v_j to the path length, and λ_1 and λ_2 are weights.

For comparing the similarity of two instance nodes, we first classify them by label, compare the similarity of $F^{ext}(v_i)$ in Euclidean distance, and then use LightGlue[49] to compare the matching degree of $F^{int}(v_i)$. This is used to determine whether the newly recognized instance node exists in the IKG. As shown in Fig. 2, if the new node does not exist in the current IKG, it is added to the graph memory using the reachability relationship. Otherwise, the information of the matched node is updated. This IKG construction based on reachability rather than precise spatial measurements avoids

graphing errors due to measurement errors, cuts maintenance costs and improves extension efficiency.

3) EKG Construction: Although the layouts of indoor environment are related to region and culture, certain universal regular patterns can provide effective exploration prompts[50], [51]. Therefore, we use the spatial layout of objects in 120 rooms from the iTHOR dataset to construct the External Knowledge Graph and acquire prior commonsense knowledge. The EKG is defined as $G_E = (V_E, E_E)$, where nodes V_E represent object categories, and edges E_E represent correlations between different object categories. The probability P$(c_j|c_i)$ of finding an object of category c_i near an object of category c_j is calculated by

$$P(c_i|c_j) = \frac{N\left(w_{ij}\right)}{\sum_{k=1}^{n} N\left(w_{ik}\right)}, \tag{4}$$

where $N\left(w_{ij}\right)$ represents the weight of all node relations with category c_i and c_j in IKG. This directed knowledge graph is finer and more factual than the symmetric commonsense matrix proposed in [52], because books are likely in a bookcase, but a book might also be on a desk or near a pen, i.e.$P\left(book|bookcase\right) > P\left(bookcase|book\right)$.

4) Online Update: To narrow the gap between prior knowledge and the current environment, the DGN dynamically updates prior knowledge with real-time observational data. First, the nodes in the IKG are merged and counted by category. Then, based on the number and weight of connections between merged nodes, the correlated relationships of different categories in the EKG are added or updated to enhance the adaptability of system in different environments, as shown in Fig. 2.

B. Navigation Decision-Making

The key to selecting a navigation target is to prioritize exploring areas where the target is likely to appear, rather than traversing all areas[53]. Therefore, for a target of category c_j, the EKG first provides exploration prompts, filtering out the instance set $\{v_i\}$ in the current view that exceeds the correlation threshold τ with the final target. Then, the navigation priority $p(v_i)$ of these instances is evaluated by the IKG,

$$p(v_i) = f_{target}\left(v_i\right) + f_{memory}(v_i) + f_{dis}(v_i) + P(c_i|c_j) \tag{5}$$

where f_{target} is the target feature function, determining whether the target is consistent with the final goal, f_{memory} is the normalized value of the number of times the instance has been observed, used to reduce the tendency to revisit explored areas, and f_{dis} is the distance between the selected target and the robot, encouraging the robot to explore farther areas. Finally, the node object v_i with the highest navigation priority is selected as the navigation target.

C. Path Planning

The path planning module generates a route from the current position to the target location. Unlike tightly coupled methods that generate only one action at a time, the DGN generates a complete action sequence to reach the next target. Only after the robot reaches a reachable point around the target, a new navigation decision is made, thereby improving decision efficiency.

IV. EXPERIMENTS AND RESULTS

In this section, we detail the results of DGN in both simulated and real-world tests. We compare our method with current baselines in simulation tests and conduct ablation studies to assess the performance of DGN. Additionally, we deploy the DGN on three robot platforms to verify its applicability in real-world scenarios.

Evaluation Metrics: To evaluate navigation performance, we used Success Rate (SR) and Success weighted by Path Length (SPL), as defined in [32]. Additionally, we compared the average time required for decision-making per step on the same device to measure the computational performance of the algorithm.

A. Simulation Experiment

1) Datasets: Our DGN model was trained on single-room scenarios in iTHOR[54]. Subsequently, we evaluated its performance using the Gibson[55] dataset within the Habitat[56] simulator and the ProcTHOR-10K[35] dataset in AI2THOR[54]. The physical simulation effects of these datasets differ, making them ideal for evaluating robot performance across varied environments[57]. For the Gibson dataset, we maintained experimental settings consistent with those in [39]. For the ProcTHOR-10K dataset, 350 scenes were randomly selected, with each evaluated using at least 20 randomly chosen target locations.

2) Experimental Detail: In the image goal navigation task, the starting position of robot is randomly set within the indoor environment of the floor where the target is located, and it needs to locate and reach the specified position in the image. We do not use GPS, IMU, or other sensors to obtain pose data. Instead, we rely solely on a single RGBD image sensor with a resolution of 600×600 and a field of view (FoV) of 90 degrees. This setup contrasts with the 360 degrees panoramic FoV sensors[18], [39], [42], [45] or pose sensors[12], [17] commonly used in ImageNav tasks[12]. Although the use of these sensors simplifies localization, they are difficult to implement on many robot platforms and significantly increase computational costs[13]. At each timestep, the robot takes an action within the action space $A = \{MoveAhead, MoveLeft, MoveRight, RotateLeft, RotateRight, Done\}$, with each move covering a distance of 0.25m and a rotation of 90 degrees. If the robot's number of action steps exceeds 500 or it actively recognizes the target, it will execute a stop command, rather than being passively notified by the environment. A test is considered successful if, when the robot stops, the distance to the target is within 1m. All simulations were conducted on a machine equipped with an Intel(R) Core(TM) i7-13700F CPU and a GeForce RTX 3090 Ti GPU.

3) Baseline: To assess the navigation performance of our model on evaluation datasets, we considered the following baselines:

TABLE I
Results of Comparative Study in Gibson.

Method	Perception	Method	No-Pose	Camera	SR	SPL
TDVN[12]	Implicit	end-to-end	No	Single	49.3	45.3
OVRL-V2[14]	Implicit	end-to-end	No	Single	82.0	58.7
TSGM[39]	Graph	end-to-end	Yes	Panoramic	81.1	**67.2**
NTS[42]	Graph	modular	No	Panoramic	63.0	43.0
DGN (Ours)	Graph	modular	Yes	Single	**83.2**	65.3

TABLE II
Results of Comparative Study in ProcTHOR-10K.

Method	SR	SPL
Random	12.2	12.2
TDVN[12]	18.85	2.57
TSGM[39]	35.71	33.66
DGN (Ours)	**62.27**	**40.81**

TABLE III
Results of Ablation Study in ProcTHOR-10K.

Method	**SR**	**SPL**
Random	12.2	12.2
DGN w/o EKG	16.81	12.9
DGN w/o IKG	59.32	33.8
DGN	62.27	40.81
DGN w/ GT SemSeg	78.45	51.54

- **Randomly Walking:** The agent performs a uniformly random action from the action space A at each timestep.
- **Target-driven RL[12]:** This baseline uses a deep siamese actor-critic network with shared convolutional networks to encode the current and target images. It handles the explored regions implicitly and is trained end-to-end via reinforcement learning.
- **OVRL-V2[14]:** This end-to-end method demonstrates state-of-the-art performance in ImageGoal Navigation. It employs ViT + LSTM for implicit environment perception and is trained using DD-PPO[21].
- **TSGM[39]:** This end-to-end method updates graph memory through a Cross Graph Mixer and uses a Memory Attention Module for online memory updates in topological environments.
- **NTS[42]:** This is a modular method based on topology awareness. It encodes images using ResNet18 and updates topological perception through a graph construction module, selecting sub-goals by a global policy and taking actions by a local policy.

4) Result and Discussion: The quantitative results of the comparative study in Gibson are shown in Table I. DGN outperforms other methods in terms of SR, indicating its strong generalization ability to adapt to different room layouts and effectively reach the target.Additionally, DGN with topological perception is higher in SPL compared to frontier implicit perception end-to end methods. [14], suggesting that topological perception provides a better understanding of the environment, aiding the robot in exploring unknown areas. However, the single-camera-relied DGN may be slightly less perceptive in a single timestep compared to method using panoramic cameras [39], but this sensor configuration has advantages when deployed to different robot platforms and real-world environments.

Our method was compared with baseline methods in the ProcTHOR-10K dataset in Ai2THOR. Some methods were not included in the table due to no access to open source code or their inability to run in Ai2THOR. As shown in Table II, the DGN outperforms other baseline methods in new environments without fine-tuning. Notably, methods that performed well in the Gibson dataset showed significantly reduced performance in ProcTHOR-10K. This is because the baseline methods rely firmly on features like image characteristics and layout specific to the original simulation environment, making it difficult to adapt to new environments. The SPL of TDVN [12] even falls below random exploration, as it struggles to adapt to complex room changes and tends to get stuck. In contrast, our method's loosely coupled modular design allows it to flexibly adapt to environmental changes and maintain high generalization performance through online updates. The DGN outperforms competing baselines by 26.56% in SR and 7.15% in SPL, indicating that using semantic results as input for navigation decision-making effectively avoids significant shifts between training domain and evaluation domain, which are common in end-to-end approaches.

5) Ablation Study: To understand the role and importance of each module in DGN, we conducted the following ablations on the ProcTHOR dataset:

- **DGN w/o EKG:** We replaced the EKG-provided object correlation-based exploration prompts with random exploration prompts.
- **DGN w/o IKG:** We navigated based on the object with the highest relevance to the target in the image without constructing the IKG and updating the EKG.
- **DGN w/ GT SemSeg:** We replace the instance perception module in our framework with the ground-truth semantic sensor in the simulator.

Table III illustrates the critical role of EKG and IKG in visual navigation tasks. DGN w/o EKG shows only a marginal improvement in SPL over the Random baseline, suggesting that exploration prompts of the EKG effectively guide the robot towards the target, minimizing aimless wandering and enhancing navigation performance. Although DGN w/o IKG achieves significantly higher SR and SPL than Random, its

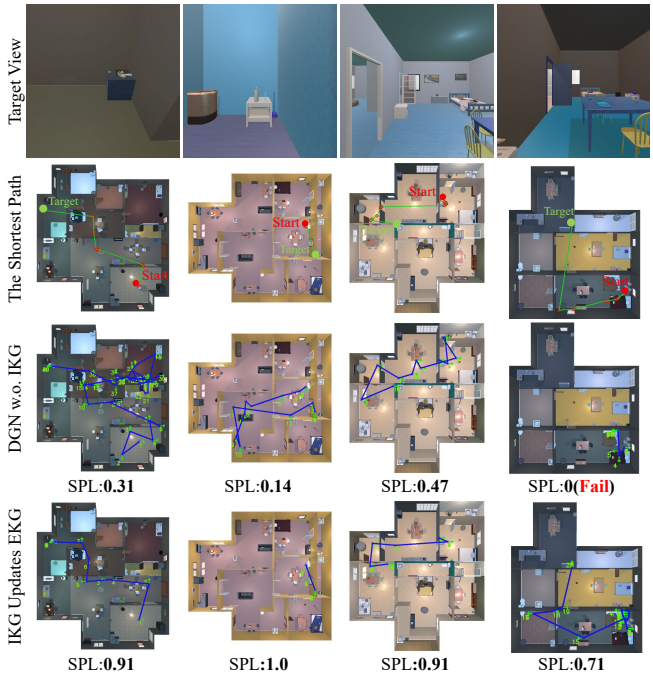

Fig. 3. Green lines indicate the shortest path, blue lines represent the actual path of robot, and green numbers denote timesteps. Without IKG updating EKG online, the robot may explore more rooms and be deadlocked in narrow spaces (row 3, column 4). With online updates, the robot reaches the target more efficiently.

performance remains below that of the DGN, due to its inability to record and interpret explored areas. Furthermore, as depicted in Fig. 3, in various multi-room environments, the online update mechanism of the DGN mechanism effectively bridges the gap between prior knowledge and the current environment, enabling the robot to locate the target more efficiently.

Additionally, there are still chances for our proposed model to progress in performance. Compared to our method, the DGN w/ GT SemSeg gains approximately 16% higher SR and 11% higher SPL, mean the substantial impact of instance-aware accuracy on overall performance. Since the modular plug-and-play design of our method allows for seamless integration of different algorithmic modules, integrating some robust instance perception algorithms coud be a promising solution as algorithm performance advance.

Table IV presents the evaluation of DGN performance following the replacement of each module without fine-tuning. Compared to the default module (row 1), using lighter-weight recognition methods such as YOLOv10 [58] (row 2) speed up the perception but may result in target loss due to reduced recognition accuracy. DISK [59] (row 3) captures more detailed structural information, though it increases computational burden. The use of the RRT [60] algorithm (row 4) trades off path quality for path planning speed. Our method supports flexible modular algorithm replacement based on the specific needs of environments and embodiments, achieving complementary advantages.

TABLE IV

Comparison of DGN Module Configuration Performance.

Detector	Keypoint	Navigation	SR	SPL	Time(ms)
YOLOv8	SuperPoint	A-star	62.27	40.81	47.55
YOLOv10	SuperPoint	A-star	54.52	36.23	26.34
YOLOv8	DISK	A-star	58.97	36.59	73.44
YOLOv8	SuperPoint	RRT	60.07	35.46	34.23

TABLE V

Comparative Study Results on Real Robots.

Device	TSGM [39]	TDVN [12]	**DGN(Ours)**
Run on 3090ti(ms)	275.37	187.84	**52.18**
Updates on 3090Ti	True	True	True
Run on Jetson NX(ms)	Failse	295.82	**97.43**
Update on Jetson NX	Failse	Failse	True
Run on Raspberry Pi(ms)	Failse	433.04	**150.53**
Updates on Raspberry Pi	Failse	Failse	True

B. Real-World Experiments

We deployed the DGN and comparison baselines in real-world tests across different hardware platforms to evaluate their performance and parameter updating capabilities. To assess algorithm robustness, we conducted tests on a robot platform equipped with a Realsense D455 camera, using three distinct hardware configurations: an x64 device with an Intel Core i7-13700F CPU and GeForce RTX 3090 Ti GPU, a Jetson NX, and a Raspberry Pi 5. Our method demonstrates strong adaptability to cameras with varying heights and configurations (see supplementary video for details). As shown in Table V, our method is faster in computation and better in platform compatibility compared to other approaches. The method in [39], with its large number of parameters set and high input data demands, is challenging to deploy effectively on resource-constrained devices. Although the method in [12] is simpler, its limits on perception module results in lower operational efficiency than our method. Unlike competing methods, our method excels on resource-constrained platforms by selecting appropriate algorithms based on hardware performance and scenario requirements, enabling online updates of the prior knowledge of environments, so there is no need for complex retraining. Experimental results show that DGN operates stably even on low-cost, low-power devices without CUDA support, verifying excellent adaptability.

V. Conclusion and Further Work

In this paper, we propose a knowledge-driven dual topological navigation framework that utilizes the EKG to provide navigation prompts and the IKG to record and update the knowledge of explored areas. This framework employs a modular plug-and-play design, supporting zero-fine-tuning replacement of target recognition, keypoint extraction and path planning algorithms, enhancing environmental adaptability and being compatible with three robot platforms. The experiments show that with topological perception and modular design, the DGN can complete ImageNav tasks

using only RGBD data on resource-constrained devices. However, there is still room for optimization in instance recognition and smooth motion control. In the future, we will develop more robust environmental perception and continuous navigation modules and expand adaptability to more real-world robot platforms.

REFERENCES

[1] Y. D. Yasuda, L. E. G. Martins, and F. A. Cappabianco, "Autonomous visual navigation for mobile robots: A systematic literature review," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–34, 2020.

[2] W. Li, X. Song, Y. Bai, S. Zhang, and S. Jiang, "Ion: Instance-level object navigation," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 4343–4352.

[3] C. Pérez-DArpino, C. Liu, P. Goebel, R. Martín-Martín, and S. Savarese, "Robot navigation in constrained pedestrian environments using reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1140–1146.

[4] J. Ichnowski, K. Chen, K. Dharmarajan, S. Adebola, M. Danielczuk, V. Mayoral-Vilches, N. Jha, H. Zhan, E. LLontop, D. Xu, *et al.*, "Fogros2: An adaptive platform for cloud and fog robotics using ros 2," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5493–5500.

[5] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto, "Ok-robot: What really matters in integrating open-knowledge models for robotics," *arXiv preprint arXiv:2401.12202*, 2024.

[6] C. Li, F. Xia, R. Martín-Martín, M. Lingelbach, S. Srivastava, B. Shen, K. Vainio, C. Gokmen, G. Dharan, T. Jain, *et al.*, "igibson 2.0: Object-centric simulation for robot learning of everyday household tasks," *arXiv preprint arXiv:2108.03272*, 2021.

[7] K. Chen, R. Hoque, K. Dharmarajan, E. LLontopl, S. Adebola, J. Ichnowski, J. Kubiatowicz, and K. Goldberg, "Fogros2-sgc: A ros2 cloud robotics platform for secure global connectivity," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1–8.

[8] T. Kim, S. Lim, G. Shin, G. Sim, and D. Yun, "An open-source low-cost mobile robot system with an rgb-d camera and efficient real-time navigation algorithm," *IEEE Access*, vol. 10, pp. 127 871–127 881, 2022.

[9] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra, "Sim2real predictivity: Does evaluation in simulation predict real-world performance?" *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6670–6677, 2020.

[10] S. Mittal, "A survey on optimized implementation of deep learning models on the nvidia jetson platform," *Journal of Systems Architecture*, vol. 97, pp. 428–442, 2019.

[11] J. Ichnowski, W. Lee, V. Murta, S. Paradis, R. Alterovitz, J. E. Gonzalez, I. Stoica, and K. Goldberg, "Fog robotics algorithms for distributed motion planning using lambda serverless computing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4232–4238.

[12] 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 *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.

[13] Z. Al-Halah, S. K. Ramakrishnan, and K. Grauman, "Zero experience required: Plug & play modular transfer learning for semantic visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 031–17 041.

[14] K. Yadav, A. Majumdar, R. Ramrakhya, N. Yokoyama, A. Baevski, Z. Kira, O. Maksymets, and D. Batra, "Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav," *arXiv preprint arXiv:2303.07798*, 2023.

[15] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 340–32 352, 2022.

[16] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 171–23 181.

[17] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," *arXiv preprint arXiv:2004.05155*, 2020.

[18] Q. Wu, J. Wang, J. Liang, X. Gong, and D. Manocha, "Image-goal navigation in complex environments via modular learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6902–6909, 2022.

[19] X. Lei, M. Wang, W. Zhou, L. Li, and H. Li, "Instance-aware exploration-verification-exploitation for instance imagegoal navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 329–16 339.

[20] K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, 2021.

[21] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," *arXiv preprint arXiv:1911.00357*, 2019.

[22] X. Sun, P. Chen, J. Fan, J. Chen, T. Li, and M. Tan, "Fgprompt: fine-grained goal prompting for image-goal navigation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[23] X. Ye and Y. Yang, "Efficient robotic object search via hiem: Hierarchical policy learning with intrinsic-extrinsic modeling," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4425–4432, 2021.

[24] J. Ye, D. Batra, A. Das, and E. Wijmans, "Auxiliary tasks and exploration enable objectgoal navigation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 117–16 126.

[25] Y. Shi, J. Liu, and X. Zheng, "Lfenav: Llm-based frontiers exploration for visual semantic navigation," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2024, pp. 375–388.

[26] D. Shah, B. Osiński, S. Levine, *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on robot learning*. PMLR, 2023, pp. 492–504.

[27] D. Shah, M. R. Equi, B. Osiński, F. Xia, B. Ichter, and S. Levine, "Navigation with large language models: Semantic guesswork as a heuristic for planning," in *Conference on Robot Learning*. PMLR, 2023, pp. 2683–2699.

[28] U. Kallakuri, B. Prakash, A. N. Mazumder, H.-A. Rashid, N. R. Waytowich, and T. Mohsenin, "Atlas: Adaptive landmark acquisition using llm-guided navigation," in *First Vision and Language for Autonomous Driving and Robotics Workshop*.

[29] J. Wei, S. Cao, T. Cao, L. Ma, L. Wang, Y. Zhang, and M. Yang, "T-mac: Cpu renaissance via table lookup for low-bit llm deployment on edge," *arXiv preprint arXiv:2407.00088*, 2024.

[30] L. Mezghan, S. Sukhbaatar, T. Lavril, O. Maksymets, D. Batra, P. Bojanowski, and K. Alahari, "Memory-augmented reinforcement learning for image-goal navigation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3316–3323.

[31] S. Rudra, S. Goel, A. Santara, C. Gentile, L. Perron, F. Xia, V. Sindhwani, C. Parada, and G. Aggarwal, "A contextual bandit approach for learning to plan in environments with probabilistic goal configurations," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5645–5652.

[32] J. Sun, J. Wu, Z. Ji, and Y.-K. Lai, "A survey of object goal navigation," *IEEE Transactions on Automation Science and Engineering*, 2024.

[33] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, "Navigating to objects in the real world," *Science Robotics*, vol. 8, no. 79, p. eadf6991, 2023.

[34] J. Zhang, L. Dai, F. Meng, Q. Fan, X. Chen, K. Xu, and H. Wang, "3d-aware object goal navigation via simultaneous exploration and identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6672–6682.

[35] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi, " procthor: Large-scale embodied ai using procedural generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5982–5994, 2022.

[36] A. Aubret, L. Matignon, and S. Hassas, "An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey," *Entropy*, vol. 25, no. 2, p. 327, 2023.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[38] J. Krantz, T. Gervet, K. Yadav, A. Wang, C. Paxton, R. Mottaghi, D. Batra, J. Malik, S. Lee, and D. S. Chaplot, "Navigating to objects specified by images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 916–10 925.

[39] N. Kim, O. Kwon, H. Yoo, Y. Choi, J. Park, and S. Oh, "Topological semantic graph memory for image-goal navigation," in *Conference on Robot Learning*. PMLR, 2023, pp. 393–402.

[40] H. Li, Z. Wang, X. Yang, Y. Yang, S. Mei, and Z. Zhang, "Memonav: Working memory model for visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 913–17 922.

[41] F. Wang, C. Zhang, W. Zhang, C. Fang, Y. Xia, Y. Liu, and H. Dong, "Object-based reliable visual navigation for mobile robot," *Sensors*, vol. 22, no. 6, p. 2387, 2022.

[42] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological slam for visual navigation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 875–12 884.

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

[44] E. Beeching, J. Dibangoye, O. Simonin, and C. Wolf, "Learning to plan with uncertain topological maps," in *European Conference on Computer Vision*. Springer, 2020, pp. 473–490.

[45] O. Kwon, N. Kim, Y. Choi, H. Yoo, J. Park, and S. Oh, "Visual graph memory with unsupervised representation for visual navigation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 890–15 899.

[46] R. Varghese and M. Sambath, "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. IEEE, 2024, pp. 1–6.

[47] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[48] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia engineering*, vol. 96, pp. 59–69, 2014.

[49] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 627–17 638.

[50] M. Narasimhan, E. Wijmans, X. Chen, T. Darrell, D. Batra, D. Parikh, and A. Singh, "Seeing the un-scene: Learning amodal semantic maps for room navigation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 513–529.

[51] A. J. Zhai and S. Wang, "Peanut: Predicting and navigating to unseen targets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 926–10 935.

[52] Y. Li, Y. Ma, X. Huo, and X. Wu, "Remote object navigation for service robots using hierarchical knowledge graph in human-centered environments," *Intelligent Service Robotics*, vol. 15, no. 4, pp. 459–473, 2022.

[53] B. Yu, H. Kasaei, and M. Cao, "Frontier semantic exploration for visual target navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4099–4105.

[54] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, *et al.*, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

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

[56] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9339–9347.

[57] A. Eftekhar, K.-H. Zeng, J. Duan, A. Farhadi, A. Kembhavi, and R. Krishna, "Selective visual representations improve convergence and generalization for embodied ai," *arXiv preprint arXiv:2311.04193*, 2023.

[58] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-time end-to-end object detection," *arXiv preprint arXiv:2405.14458*, 2024.

[59] M. Tyszkiewicz, P. Fua, and E. Trulls, "Disk: Learning local features with policy gradient," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 254–14 265, 2020.

[60] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.