Knowledge-Driven Visual Target Navigation: Dual Graph Navigation

Shiyao Li[1] Ziyang Meng[1] Jiansong Pei[1] Jiahao Chen[2]
Bingcheng Dong[1] Guangsheng Li[1] Shenglan Liu[1⋆] Feilong Wang[1]

Abstract— **In unknown environments, navigating a robot by a given image to a specific location or instance is critical and challenging. The existing end-to-end approaches require simultaneous implicit learning of multiple subtasks, and modular approaches depend on metric information. Both approaches face high computational demands, often leading to difficulties in real-time updates and limited generalization, making them challenging to implement on resource-constrained devices. To address these challenges, we propose Dual Graph Navigation (DGN), a knowledge-driven, lightweight image instance navigation framework. DGN builds an External Knowledge Graph (EKG) from small-scale datasets to capture prior object correlations, efficiently guiding target exploration. During exploration, DGN builds an Internal Knowledge Graph (IKG) using an instance-aware module, which records explored objects based on reachability relationships rather than precise metric information. The IKG dynamically updates the EKG, enhancing the robot's adaptability to the current environment. Together, they realize topological perception and reduce computational overhead. Furthermore, unlike approaches characterized by over-dependence between components, DGN employs a plug-and-play modular design that allows independent training and flexible replacement of functional modules, effectively enhancing generalization performance while reducing training and deployment costs. Experiments illustrate that DGN generalizes well in different simulation environments (AI2-THOR, Habitat), achieving state-of-the-art performance on the ProcTHOR-10K dataset. It is compatible with three distinct real-world robot platforms, including edge computing devices without CUDA support. It exhibits a decision-making speed of 3.8 to 5.5 times over baseline methods. Further details can be found on the project page: https://dogplanningloyo.github.io/DGN/.**

I. INTRODUCTION

Navigating to a location that matches a target image or instance within an unknown environment is critical, requiring robots to possess both scene understanding and autonomous navigation capabilities [1–3]. However, the existing approaches are inevitable to be trained in specific datasets and rely on high-performance GPU workstations to ensure real-time inference [4, 5]. Due to high training costs and limited generalization, these approaches are challenging to deploy in real-world scenarios [6–9]. Therefore, developing efficient methods with strong generalization for resource-constrained platforms is crucial for the visual target navigation task [5, 10, 11].

Visual navigation approaches mainly fall into end-to-end [12–16] and modular approaches [17–19]. Their drawbacks in computational complexity and tightly coupled components restrict generalization in different environments. End-to-end reinforcement learning approaches learn the mapping relationships between observation and actions by continuous agent-environment interactions [20], but suffer from poor transferability [13, 21] and sparse rewards [22–25]. Language-driven end-to-end approaches utilize cross-modal models to assist navigation, which improve generalization [16, 26–28] but are challenging to deploy on resource-constrained devices [29] due to large parameter storage and high computational consumption. On the other hand, modular approaches assign separated subtasks to functional modules [30, 31] to improve sample efficiency and stability [32–34], yet often rely on metric-based information for environmental perception [13, 31].

To overcome these challenges, we propose a knowledge-driven, lightweight image instance navigation framework, Dual Graph Navigation (DGN), as illustrated in Fig. 1. DGN consists of an External Knowledge Graph (EKG) and an Internal Knowledge Graph (IKG), enabling topological perception of the environment. Initially constructed from small-scale datasets, the EKG captures object category correlations and provides navigation prompts based on the strongest correlations, avoiding the computational costs of full-map exploration or prediction. The IKG is constructed from an instance-aware module to represent semantic and reachability relationships (Sec. III-A.2). As a result, IKG enables memory-efficient real-time mapping without precise metrics, such as semantic occupancy maps. During exploration, the IKG continuously updates the EKG with reachability information, which offers a retraining-free solution in new environments. Additionally, DGN adopts a plug-and-play modular design that supports independent training and flexible replacement of functional modules. Unlike tightly coupled approaches, DGN can deploy appropriate models for navigation subtasks (e.g. instance-aware module and path-planning module) based on varying environmental conditions and hardware configurations. This contributes to stronger generalization while reducing deployment and training costs.

The main contributions of this paper are as follows:

- We present Dual Graph Navigation (DGN). This knowledge-driven lightweight image instance navigation framework utilizes correlations among object categories and reachability relationships instead of precise metric maps. Consequently, we avoid large-scale data training and significantly improve inference speed.
- We design a plug-and-play modular framework in DGN, which allows easy integration and replacement of func-

*This work was supported by National Natural Science Foundation of China No. 62376052

[1]Dalian University of Technology, Dalian 116024, China.

[2]Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China

⋆Corresponding author, Email:liusl@dlut.edu.cn

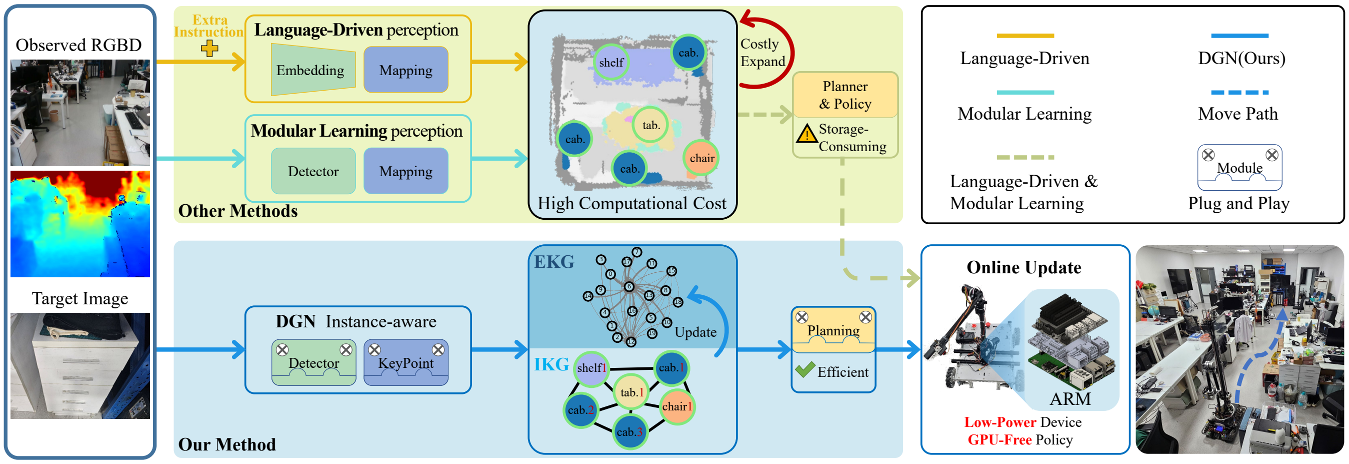

Fig. 1. Comparison between DGN and mainstream visual navigation methods. Mainstream visual navigation methods (Other Methods) accept RGBD input and partly require extra auxiliary language instruction, constructing semantic maps with tightly coupled perception modules, resulting in storage-consuming deployments. By solely using RGBD input, DGN (Our Method) constructs an Internal Knowledge Graph (IKG) from a plug-and-play instance-aware module to record semantic information and reachability relationships of objects. The IKG dynamically updates the External Knowledge Graph (EKG) of object category correlations. This empowers efficient navigation that is deployable on low-power, low-computation edge devices.

tional modules. This greatly enhances the ability of the system to adjust and expand in various situations and on different hardware platforms.

- We evaluate DGN in mainstream simulation environments, such as Habitat [35] and AI2-THOR [36]. DGN achieves state-of-the-art performance on the ProcTHOR-10K [37] dataset. Furthermore, real-world robots with resource constraints have successfully deployed the online-updating DGN, demonstrating faster decision-making than the baseline approaches.

II. Related Work

A. Visual Navigation

Visual navigation [12] is a long-standing robotic task where a robot navigates visually to locate a target or position based on a given image. Current methods are generally categorized into end-to-end and modular approaches. End-to-end approaches [12–14] directly map observations to actions but face challenges such as low sample efficiency and poor generalization [38]. To address these issues, TDVN [12] and ZER [13] optimize learning strategies for better training efficiency. Later works OVRL-V2 [14] and FGPrompt [22] further improve visual representation abilities to infer target locations. Recently, as language-driven models have performed well in image perception tasks, methods like ZSON [15] and CoW [16] improved navigation performance by utilizing CLIP [39] to obtain cross-modal information. Although the above end-to-end approaches are straightforward, the implicit learning of multiple subtasks brings about heavy computational overhead and poor model fit [32]. To break these limits, researchers proposed the modular approaches [17, 19], where the overall navigation task is divided into subtasks and distributed to specialized modules [40]. Classic modular methods like ANS [17] and Wu et al. [18] employ hierarchical planning to efficiently train separate modules, including environment perception, navigation planning and local policy. Latest methods IEVE [19] dynamically switch between different modules of exploration, verification and exploitation, significantly improving the decision-making ability of agents in complicated environments. These modular methods make reinforcement-learning-based decisions and rely on tightly coupled modules which cause performance bottlenecks and limit system optimization. In contrast, to overcome module selection limitations and enhance generalization, our method employs a plug-and-play modular framework that supports independent training and seamless module replacement.

B. Topological Perception

Environmental perception is essential to robotic navigation, underpinning decision-making and path planning. It is classified into implicit, metric, and topological approaches [41]. Implicit perception typically applies RNN, LSTM, etc., to represent navigation states with simple structures but with limited long-term memory [42]. Metric perception offers precise localization by adopting dense maps, yet it is susceptible to sensor noise and incurs significant computational costs during map construction and expansion [43]. In contrast, topological perception uses the graph to represent environmental features, enabling sparse yet robust representations with long-term exploration memory. Some topological methods simplify the metric perception. For example, in NTS [44] and Neural Planner [45], nodes are image features, and edges are rough geometric data. However, these methods employ handcrafted features and require metric consistency. To mitigate the impact of environmental errors on navigation performance, methods like SPTM [46] and VGM [47] shift focus from metric data to logical relationships, relieving the reliance on exact pose information. Nevertheless, these image-level node methods often struggle with distinguishing specific instances. To enhance navigation granularity, TSGM [41] introduces image nodes for

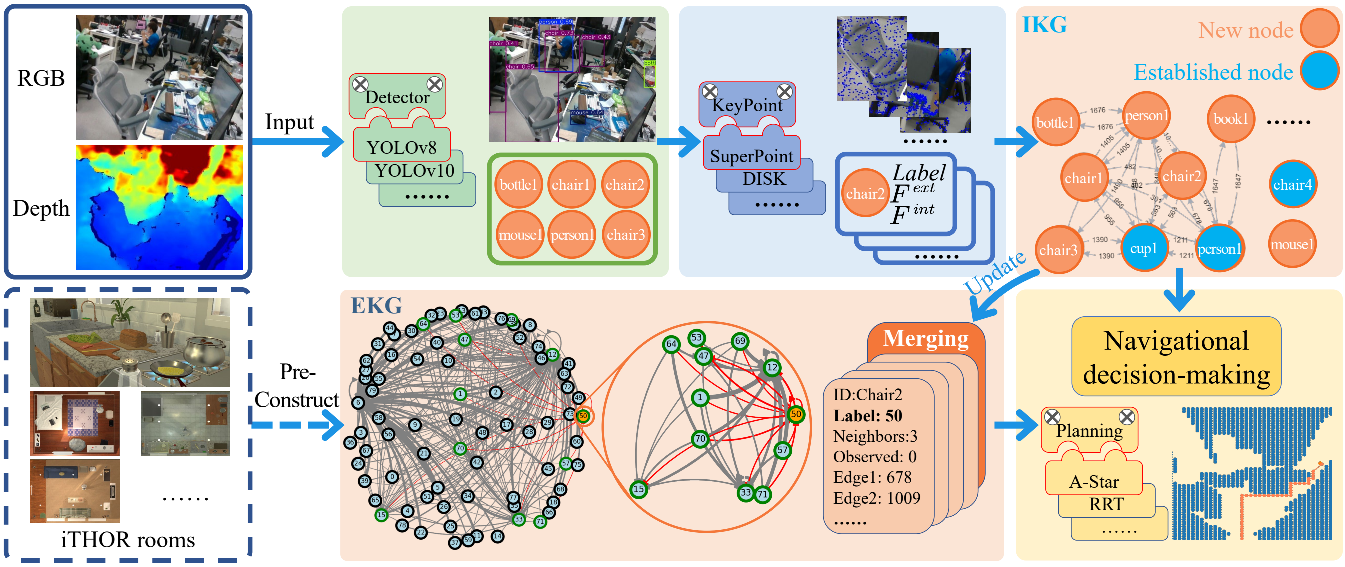

Fig. 2. DGN receives RGBD input and constructs the IKG, which generates nodes through target recognition and keypoint extraction. The IKG then online updates the EKG, which is pre-constructed by iTHOR. Together, the IKG and EKG determine the next navigation target, and the path planning module generates a path for the robot.

locations, object nodes for instances, and edges connecting objects in the current view with adjacent images. However, most topological methods focus on positional image data, neglecting semantic relationships between objects, which limits exploration efficiency. Our method solves these by introducing a dual-topology structure: the IKG for instance-level spatial representation and the EKG to guide target exploration based on object correlations.

III. METHOD

Our modular method employs dual topological graphs, the IKG and the EKG, to locate the navigation target, as illustrated in Fig. 2. We construct the IKG from the instance-aware module to detect objects, extract keypoints and obtain reachability relationships among objects. Consistently updated by IKG, the pre-constructed EKG records correlations among object categories and accordingly provides navigation prompts. Based on prompts from the EKG, the IKG determines the next navigation target. Subsequently, the path-planning module generates a collision-free path for the robot to move along. We employ a plug-and-play modular design, where the default models used for target recognition, keypoint extraction, and path planning are respectively YOLOv8 [48], SuperPoint [49], and A-star [50].

A. Environmental Perception

1) Instance-aware: The DGN adopts algorithm components of target recognition and keypoint extraction to extract the external features (F^{ext}) and internal features (F^{int}) of instances from the obtained RGBD data. The external feature of an instance is a category-wise aligned vector of its add-up distances from other instances in a category. To reduce noise inference, only the distance shorter than two times of the robot size S is counted. The set of target nodes recognized from the image is $\{v_1, v_2, ...v_i, ..., v_K\}$, where v_i represents the i-th instance target, and c_i represents its category index. K represents the number of instances identified in the image. The external feature $F^{ext}(v_i)$ of v_i is determined by

$$F^{ext}(v_i) = [f_{i1}, \ldots, f_{iN}]\,, F^{ext} \in \mathbb{R}^N, \tag{1}$$

$$f_{ij} = \sum_{k \in \mathcal{H}} d(v_i, v_k). \tag{2}$$

The set $\mathcal{H} = \{k \mid c_k = j, k \leq K, d(v_i, v_k) \leq 2S\}$. The function $d(v_i, v_k)$ represents the Euclidean distance between the observed objects v_i and v_k, while N denotes the number of recognizable categories. The internal feature $F^{int}(v_i)$ is a set of keypoints extracted from the target image to reduce background interference and computational overhead.

2) IKG Construction: IKG is defined as $\mathcal{G}_I = (\mathcal{V}_I, \mathcal{E}_I)$ for the representation of environmental map, where $\mathcal{V}_I$ is a node set of recognized instances, and $\mathcal{E}_I$ refers to a set of edges that reflect the reachability relationships between two instances. The weight of reachability relationship w_{ij} is calculated by weighting the path length and obstacle coverage between instances v_i and v_j, given by

$$w_{ij} = (1 - \lambda) \cdot d(v_i, v_j) + \lambda \cdot o(v_i, v_j), \tag{3}$$

where $o(v_i, v_j)$ is calculated by dividing the number of obstacles between v_i and v_j by the length of path, and $\lambda \in (0, 1)$ is a trade-off weight.

To compare the similarity between two nodes, they are first classified by label, and the Euclidean distance of $F^{ext}(v_i)$ is measured. Then, the matching degree of $F^{int}(v_i)$ is assessed using LightGlue[51]. This process determines whether the newly recognized instance node exists in the IKG. If it doesn't, the new node is added to the graph, connecting to every other reachable identified instance node through the reachability relationships, as seen in Fig. 2. Otherwise, it is utilized to update the corresponding node.

3) EKG Construction: Although the layouts of the indoor environment are related to region and culture, certain universal regular patterns can provide effective target exploration prompts [52, 53]. Therefore, we use the spatial layout of objects in 120 rooms from the iTHOR dataset to construct the EKG and capture prior commonsense knowledge. The EKG is defined as $\mathcal{G}_E = (\mathcal{V}_E, \mathcal{E}_E)$, where nodes $\mathcal{V}_E$ represent object categories and edges $\mathcal{E}_E$ reflect correlations probability between different object categories. The correlation probability $P(c_i|c_j)$ of finding an object in category c_i near an object in category c_j is calculated by

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

where $\psi\left(w_{ij}\right)$ represents the weight of all node relations with category c_i and c_j in IKG, and $i, j \in \{1, 2, \ldots, |\mathcal{V}_E|\}$. This knowledge graph is more refined and factual than the symmetric commonsense matrix proposed in [54] because it captures the directional and contextual nuances between categories of object. For example, books are typically found in a bookcase, however a book might also be on a desk or beside 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. Firstly, the node instances in the same category are merged and counted. Then, based on the number and weight of connections between merged nodes, the correlation probability in the EKG is added or updated to enhance the adaptability of the system in different environments, as shown in Fig. 2.

B. Navigation Decision-Making

The key to navigation target selection is to preferentially explore the areas where the target is most likely to appear rather than traversing all areas indiscriminately [55]. For instance, if the target identified from the target image is in category c_j, the EKG will provide exploration prompts by filtering the instance set $\{v_i \mid P(c_i|c_j) > \tau, v_i \in \mathcal{V}_I\}$, where τ represents a predefined correlation threshold. Later, the navigation priority $p(v_i)$ of these instances is evaluated by using the IKG, defined as:

$$p(v_i) = g_{target}(v_i) + g_{fre}(v_i) + g_{dis}(v_i) + P(c_i|c_j), \tag{5}$$

where the function $g_{target}(v_i)$ evaluates how well the instance v_i aligns with the final target goal. The term $g_{fre}(v_i)$ is defined as:

$$g_{fre}(v_i) = -\frac{\sigma(v_i)}{\max_{v_k \in \mathcal{V}_I} \sigma(v_k)}, \tag{6}$$

where $\sigma(v_i)$ denotes the number of times the instance v_i has been recognized in the IKG. This term reduces the likelihood of revisiting previously explored areas by lowering the priority of frequently encountered instances. The function $g_{dis}(v_i)$ represents the distance between the selected instance v_i and the robot, thereby encouraging exploration of farther areas. Finally, the instance v_i with the highest navigation priority $p(v_i)$ is selected as the next navigation target.

C. Path Planning

The path planning module generates a collision-free path 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. A new navigation decision is made only after the robot reaches an accessible point around the target, to improve decision efficiency.

IV. EXPERIMENTS AND RESULTS

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

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

A. Simulation Experiment

1) Datasets: Our DGN model is trained on single-room scenarios in iTHOR [36]. Subsequently, we evaluate its performance using the Gibson [56] dataset within the Habitat [35] simulator and the ProcTHOR-10K [37] dataset within AI2-THOR [36]. The physical simulation effects of these datasets behave differently, which is ideal for evaluating robot performance in varied environments [32, 57]. For the Gibson dataset, we maintain experimental settings consistent with those in [41]. For the ProcTHOR-10K dataset, 350 scenes are randomly selected, and at least 20 target locations in each scene are evaluated.

2) Experiment Details: In the visual target navigation task, the starting position of the robot is randomly set within the indoor environment of the floor where the target is located, and it needs to locate the specified position or instance in the image. 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 differs from the panoramic 360 degrees FoV sensors [18, 41, 44, 47] 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 action covering a distance of 0.25m or a rotation of 90 degrees. If the robot's action steps exceed 500 or it recognizes the target, it executes a stop command rather than being passively notified by the environment. A test is successful if the robot stops within 1m of the target. All simulations are conducted on a machine equipped with an Intel(R) Core(TM) i7-13700F CPU and a GeForce RTX 3090 Ti GPU.

TABLE I

Results of Comparative Study in Gibson.

Method	Perception	Framework	Pose-free	Camera	SR	SPL
ANS [17]	metric	modular	No	single	57.5	18.1
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 [41]	topological	end-to-end	Yes	panoramic	81.1	**67.2**
NTS [44]	topological	modular	No	panoramic	63.0	43.0
DGN (Ours)	topological	modular	Yes	single	**83.2**	65.3

TABLE II

Results of Comparative Study in ProcTHOR-10K.

Method	SR	SPL
Random Walking	12.2	12.2
TDVN [12]	18.9	2.6
TSGM [41]	35.7	33.7
DGN (Ours)	**62.3**	**40.8**

TABLE III

Results of Ablation Study in ProcTHOR-10K.

Method	SR	SPL
DGN w/o. EKG	16.8	12.9
DGN w/o. IKG	59.3	33.8
DGN	62.3	40.8
DGN w. GT SemSeg	78.5	51.5

3) Baseline: We compare the proposed method with a number of baselines that use various types of perception and frameworks. The considered baselines are as follows:

- **Random Walking:** The agent randomly samples an action from the action space A with a uniform distribution at each step.
- **ANS [17]**: This modular hierarchical method combines classic path planning with learned components.
- **TDVN [12]:** This baseline uses a deep siamese actor-critic network with shared convolutional networks to encode the current and target images. It 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 [41]:** This end-to-end method builds a semantic graph with image nodes and object nodes, that are integrated by a cross-graph mixer to guide navigation.
- **NTS [44]:** This is a modular method that constructs a topological map using semantic and geometric information, updating the environment in real time and efficiently navigating under global and local policies.

4) Result and Discussion: The quantitative results of the comparative study in the Gibson are shown in Table I. Since DGN utilizes semantic correlations among object categories as prompts for navigation, the robot can reach the target more efficiently in multiple rooms with different layouts, achieving a higher SR than other methods. With topological perception, DGN attains a better understanding of environments, which improves exploration efficiency. It exceeds other end-to-end methods in both SR and SPL, demonstrating superior exploration capability. Although the single-camera setup covers slightly less area at each time step than the panoramic method [41], DGN is more competitive in real-world deployments on various robot platforms.

We compare our method with baselines on the ProcTHOR-10K dataset in AI2-THOR to assess generalization across environments. Some methods are not included in the table due to the lack of open-source code or incompatibility of AI2-THOR. The baseline methods rely firmly on features like visual domain and their original environment layout. As a result, methods that perform well on the Gibson dataset show a sharp decline in ProcTHOR-10K. For example, TDVN [12] fails to adapt to environmental changes, often getting stuck and performing worse than Random Walking in SPL. In contrast, unlike the direct RGBD-based policy framework, our method uses abstractions of input images to reduce domain gaps across datasets [33]. Without fine-tuning, our method outperforms baselines in new environments, with a 26.56% gain in SR and a 7.15% improvement in SPL over TSGM [41]. This explains that our method better adapts to environmental changes and sustains strong generalization through online updates.

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

- **DGN w/o. EKG:** We replace the exploration prompts of EKG with random exploration prompts.
- **DGN w/o. IKG:** Navigation is based on the object most relevant to the target in the image without constructing the IKG and updating the EKG.
- **DGN w. GT SemSeg:** The ground-truth semantic sensor replaces the instance-aware module in DGN.

Table III highlights the important role of EKG and IKG in visual target navigation. DGN w/o. EKG robot tends to randomly select navigation directions, resulting in a significantly lower SR and SPL than DGN. This indicates that the relationships between targets are essential to effective navigation. Additionally, DGN w/o. IKG struggles to update its knowledge of different room layouts when exploring multiple rooms, leading to potential deadlock in complex environments and numerous ineffective movements. The online update mechanism of DGN addresses these issues,

TABLE IV

Comparative Study Results on Real Robots. The symbol (✓) indicates a successful run on the hardware platform, while the symbol (✗) marks failures.

Method	3090 Ti		Jetson NX		Raspberry Pi	
	Run-time(ms)	Update parameter	Run-time(ms)	Update parameter	Run-time(ms)	Update parameter
TSGM [41]	275.37	✓	✗	✗	✗	✗
TDVN [12]	187.84	✓	295.82	✗	443.04	✗
DGN(Ours)	**50.03**	✓	**97.43**	✓	**150.53**	✓

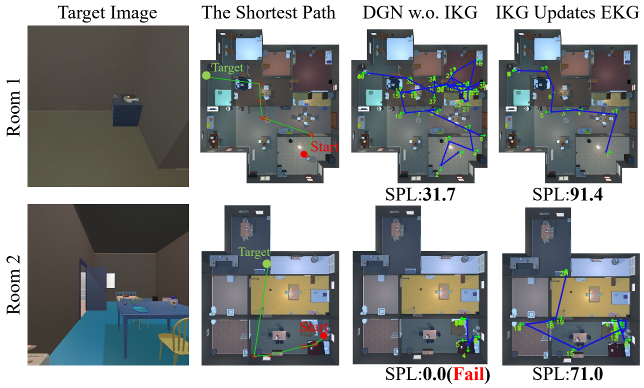

Fig. 3. Green lines indicate the shortest path, blue lines represent the actual path of the robot, and green numbers denote timesteps. Without IKG updating EKG online and recording explored objects, the robot may explore more rooms and be deadlocked in narrow spaces (row 2, column 3). With IKG online updates, the robot reaches the target more efficiently.

bridging the gap between prior knowledge and the current environment, thus improving target localization efficiency, as illustrated in Fig. 3.

Additionally, there is potential to enhance our model's performance. DGN w. GT SemSeg demonstrates approximately 16% higher SR and 11% higher SPL than our method, denoting the significant impact of instance-aware accuracy on overall performance. Integrating a more robust instance-aware module could be a promising avenue for further improvements on the model [19].

Table V shows the impact of replacing each module in DGN without fine-tuning. Compared to the default module (row 1), using lighter-weight recognition methods such as YOLOv10 [58] (row 2) speeds up perception but may lead to target loss due to reduced recognition accuracy. DISK [59] (row 3) captures more detailed structural information, albeit at the cost of increased computational load. The RRT [60] algorithm (row 4) trades off path quality for faster path planning. Our method supports flexible modular models replacement based on the needs of specific environments and hardware performance to gain complementary advantages.

B. Real-World Experiments

We deployed the DGN and the compared baselines in real-world tests across different hardware platforms to evaluate their performance and parameter updating capabilities. 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 (details in supplementary video). As shown in Table IV, our method is faster in computation and better in platform compatibility compared to other methods. The method in [41], 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 the perception module result in lower operational efficiency than our method. Unlike competing methods, our method excels on resource-constrained platforms by selecting appropriate models based on hardware performance and scenario requirements, enabling online updates of the prior knowledge of environments, so there is no need for complex retraining. The experiments demonstrate that DGN operates stably on low-power, computationally constrained devices without CUDA support, illustrating its adaptability.

TABLE V

Comparison of DGN Module Configuration Performance.

Detector	Keypoint	Planning	SR	SPL	Time(ms)
YOLOv8	SuperPoint	A-star	62.3	40.8	50.03
YOLOv10	SuperPoint	A-star	54.5	36.2	26.34
YOLOv8	DISK	A-star	60.1	36.6	73.44
YOLOv8	SuperPoint	RRT	59.0	35.5	34.23

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 the knowledge of explored areas and update the EKG online, enabling efficient topological perception of the environment. This framework employs a modular plug-and-play design, supporting the seamless replacement of target recognition, keypoint extraction, and path planning modules, thereby improving generalization across environments and supporting compatibility with various robot platforms. The experimental results show that with topological perception and modular design, the DGN can complete ImageNav tasks using only RGBD data on resource-constrained devices. Besides, potential improvements remain in instance recognition and motion control. How to enhance environmental perception, improve continuous navigation capabilities, and broaden the framework's applicability on more real-world robot platforms can be considered in future work.

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 *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 *IEEE International Conference on Robotics and Automation (ICRA)*, 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 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 *IEEE International conference on robotics and automation (ICRA)*, 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 *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 *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 *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 *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*, 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*, 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*, 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*, 2024.

[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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6672–6682.

[35] 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 *IEEE/CVF International conference on computer vision*, 2019, pp. 9339–9347.

[36] 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.

[37] 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.

[38] 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.

[39] 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*, 2021, pp. 8748–8763.

[40] 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 *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 916–10 925.

[41] 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*, 2023, pp. 393–402.

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

[43] 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.

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

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

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

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

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

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

[50] 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.

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

[52] 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*, 2020, pp. 513–529.

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

[54] 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.

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

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

[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, 2000, pp. 995–1001.