Aligning Knowledge Graph with Visual Perception for Object-goal Navigation

Nuo Xu, Wen Wang, Rong Yang, Mengjie Qin, Zheyuan Lin, Wei Song*, Chunlong Zhang*, Jason Gu, Chao Li

Abstract— Object-goal navigation is a challenging task that requires guiding an agent to specific objects based on first-person visual observations. The ability of agent to comprehend its surroundings plays a crucial role in achieving successful object finding. However, existing knowledge-graph-based navigators often rely on discrete categorical one-hot vectors and vote counting strategy to construct graph representation of the scenes, which results in misalignment with visual images. To provide more accurate and coherent scene descriptions and address this misalignment issue, we propose the Aligning Knowledge Graph with Visual Perception (AKGVP) method for object-goal navigation. Technically, our approach introduces continuous modeling of the hierarchical scene architecture and leverages visual-language pre-training to align natural language description with visual perception. The integration of a continuous knowledge graph architecture and multimodal feature alignment empowers the navigator with a remarkable zero-shot navigation capability. We extensively evaluate our method using the AI2-THOR simulator and conduct a series of experiments to demonstrate the effectiveness and efficiency of our navigator. Code available: https://github.com/nuoxu/AKGVP.
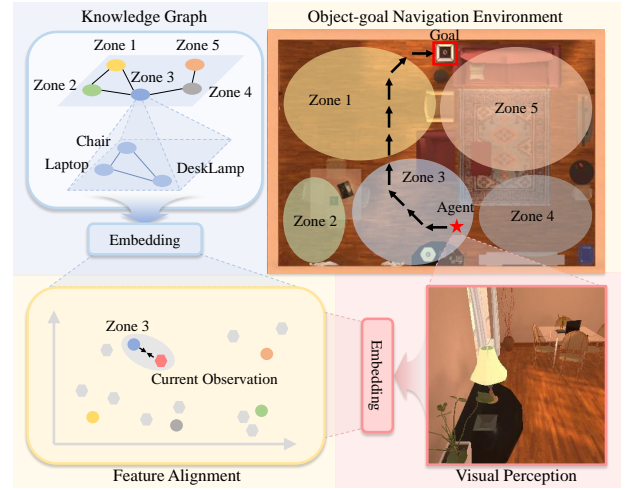
Fig. 1. The core idea of our AKGVP method. In order to effectively describe the same environment, we can leverage two modalities of data: knowledge graphs derived from natural language descriptions and observation images characterized by visual descriptions. Our primary objective is to align these two modalities within a shared feature space, facilitated by visual-language pre-training. Ultimately, these modalities are fused for decision-making.

I. INTRODUCTION

Object-goal navigation [1], [2] presents a formidable challenge as agents endeavor to navigate unfamiliar and dynamic environments based on visual observations to locate specific objects. However, traditional approaches encounter difficulties in locally visible settings due to the absence of spatial layout information, impeding efficient navigation. Achieving successful object-goal navigation necessitates surpassing robust perception and enabling agents to possess a comprehensive understanding of their surroundings, empowering them to make intelligent decisions to accomplish their goals. This calls for a seamless integration of perception, language, and action, where agents adapt their navigational strategies based on real-time visual input, language instructions, and sensory feedback. Successful visual navigation hinges upon mapping visual observations to actions and teaching machines to perceive objects, infer attributes, and reason about relationships. Agents must exhibit enhanced abilities to reason, plan, and execute goal-directed actions that remain adaptable to dynamic environmental changes. Consequently, learning informative visual representations and modeling Semantic Maps [3] or Knowledge Graphs

Nuo Xu, Wen Wang, Rong Yang, Mengjie Qin, Zheyuan Lin, Wei Song, Chunlong Zhang, Jason Gu and Chao Li are with ZhejiangLab, Hangzhou 311100, China (e-mail:{nuo.xu, wangwen, yang_rong, qmj, linzhy, weisong, zcl1515, jgu, lichao}@zhejianglab.com). Jason Gu is also with Department of Electrical and Computer Engineering, Dalhousie University, Halifax, Canada. Wei Song and Chunlong Zhang are co-corresponding authors (*).

[4] of the scenes become indispensable for augmenting performance in unfamiliar environments.

Object-goal navigators encompass three primary categories: end-to-end methods, knowledge-graph-based methods, and semantic-map-based methods. In the end-to-end approach, agents gather observations (including RGB images, depth, and pose) from the environment at each time step. Utilizing these inputs and object categories, agents employ various temporal models, such as RNNs [5], LSTMs [6], [7], [8], [9], [10], memory-based models [11], or even without explicit temporal modeling [12], [13], [14], to determine the subsequent actions. The end-to-end approach exclusively relies on inputs like RGB images for direct action generation. Nevertheless, it often struggles with generalizing in complex scenarios. To overcome this challenge, researchers have introduced additional modeling techniques, such as knowledge graphs [15], [16], [17], [18] and semantic maps [19], [20], [21], [22], [23]. Knowledge graphs provide robust representation capabilities, capturing entity correlations and contextual information. Graph structures integrate vast amounts of structured and semi-structured data. In contrast, semantic maps enable spatial perception, describing entity positions, directions, and geometry. They offer real-time updates, aiding rapid adaptation in dynamic environments.

Nevertheless, the existing knowledge-graph-based navigators suffer from a significant drawback, primarily stemming from their extensive dependence on discrete categorical vectors and vote-counting strategy for constructing graph representation of the scenes. Unfortunately, this approach frequently leads to a misalignment between these categorical representations and the actual visual images of the scene. Relying solely on decision-making losses, such as reinforcement learning, proves challenging in effectively aligning the two modalities. Addressing this limitation is of paramount importance in order to enhance the performance and accuracy of object-goal navigators in real-world scenarios. With this in mind, our research endeavors to push the boundaries of knowledge-graph-based navigators by focusing on improving their capabilities and overall performance.

To address the aforementioned challenges, we propose the Aligning Knowledge Graph with Visual Perception (AKGVP) for object-goal navigation. As depicted in Fig. 1, our approach introduces a continuous modeling framework that captures the hierarchical knowledge-graph-based scene architecture, leading to more accurate and coherent scene descriptions. Additionally, we leverage visual-language pre-training techniques to align scene language description with visual perception, bridging the gap between semantic graph and visual understanding. This alignment enhances the comprehension of instructions and scenes for agent within the context of navigation tasks, enabling a more refined representation of the visual environment. Moreover, the integration of a continuous knowledge graph architecture and multimodal feature alignment empowers the navigator with a remarkable zero-shot navigation capability [24], [10], [23]. In this work, we extensively evaluate the effectiveness and efficiency of our proposed AKGVP method using the AI2-THOR [25] simulator, a widely adopted platform for embodied navigation research. Through a series of rigorous experiments, we demonstrate the superior performance of AKGVP in achieving object-finding tasks. The contributions of this research are as follows:

- A novel continuous knowledge-graph-based scene modeling framework is introduced that captures the intricate hierarchical scene architecture, resulting in more accurate and coherent scene descriptions.
- Moreover, one visual-language pre-training technique is leverage advanced to effectively align scene language descriptions with visual perception, proposing our Aligning Knowledge Graph and Visual Perception (AKGVP).
- In this study, we extensively evaluate the effectiveness and efficiency of our proposed AKGVP using the AI2-THOR simulator. Through rigorous experiments, the superior performance of AKGVP is demonstrated.

II. METHODOLOGY

In the Object-goal Navigation (ObjNav) task, the agent is situated within the 3D indoor environments Q and aims to efficiently approach predefined goal objects G in the fewest steps. During each episode, the agent is randomly initialized at position $l = \{x, z, \theta_1, \theta_2\}$ within the environment $q \in Q$.

Here, (x, z) represents the plane coordinates, while θ_1 and θ_2 correspond to the yaw and pitch angles, respectively. At each timestamp t, the agent learns a policy function denoted as $\pi(a_t|s_t, g)$, which predicts the appropriate action $a_t \in \mathcal{A}$ based on the current state (first-person observation) s_t and the goal object $g \in G$. Here, $\mathcal{A} = \{MoveAhead, RotateLeft, RotateRight, LookDown, LookUp, Done\}$. Notably, the agent independently determines the completion of an action, relying on its own judgment rather than environmental cues. The movement of agent occurs within a discretized scene space with intervals of 0.5m. The rotation and the vertical tilt angle remain fixed at 45 and 30 degrees, respectively. An episode is deemed successful when the agent selects the Done action while the goal object is visible within a threshold distance of 1.5m, indicating proximity. Failure occurs if the agent does not meet this criterion.

A. Overview

As illustrated in Fig. 2, the AKGVP framework consists of three indispensable components: an aligned encoder, a high-level controller, and a low-level controller. The aligned encoder assumes a critical role in separately encoding images and natural language, facilitating the alignment of features through multimodal pre-training. Harnessing knowledge graph modeling, the high-level controller adeptly plans sub-goals for the navigator, effectively guiding the movement of agent across diverse zones. In parallel, the low-level controller leverages the fused multimodal information to make well-informed action decisions, enabling the agent to interact with the environment and exert precise control over its movements. Fig. 2 offers a visual depiction of the intricate interplay between these components within the AKGVP framework. In the subsequent sections, we will delve into the detailed exposition of each of these three components.

B. MultiModal Feature Alignment

In this section, we present the aligned encoder, a pivotal component responsible for vision-language alignment and feature extraction.

Multimodal Pre-training. Multimodal pre-training, exemplified by the pioneering CLIP [26] model, represents a technique wherein a model is trained on diverse modalities, such as images and natural language, fostering the acquisition of rich representations that capture intermodal correlations. This pre-training paradigm is designed to empower the model with the ability to comprehend and reason about the intricate relationships existing between visual and textual information. Through pre-training, CLIP learns to establish associations between images and text via the juxtaposition of positive pairs (matching image-text pairs) against negative pairs (non-matching pairs). This process effectively encourages the model to align similar images with their corresponding textual descriptions within a shared feature space.

Aligned Encoder. In fact, CLIP (ResNet-50 [27]) is applied in the Aligned Encoder module within the AKGVP model, comprising two components: the visual encoder and the language encoder. To preserve the spatial position information
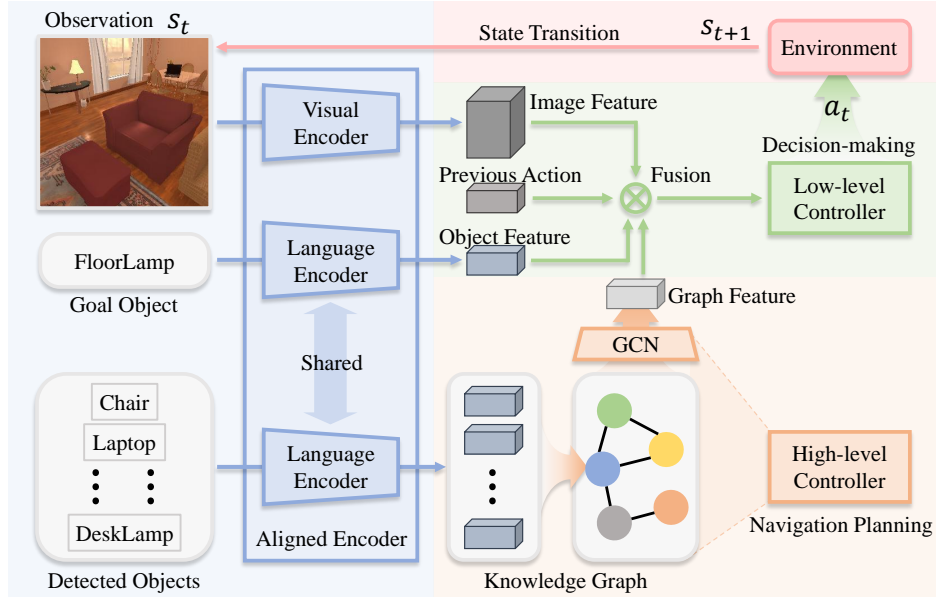
Fig. 2. Pipeline of our AKGVP method. AKGVP is composed of three essential components: an aligned encoder, a high-level controller, and a low-level controller. The encoder plays a crucial role by separately encoding images and natural language, facilitating feature alignment through multimodal pre-training. The high-level controller leverages knowledge graph modeling to effectively plan sub-goals for the navigator, directing the movement of agent across different zones. On the other hand, the low-level controller utilizes the fused multimodal information to make informed action decisions, enabling the agent to interact with the environment and control its movements adeptly.

in the image, the last fully connected layer of the visual encoder is removed, while the language encoder remains intact. During training, the parameters of the aligned encoder are consistently frozen, requiring no updates. As depicted in Fig. 2, the Image Feature f_{img} is represented as a $7 \times 7 \times 2048$ tensor, while the Object Feature f_{obj} is a continuous 1024-dimensional vector. These architectural choices contribute to the efficient encoding and representation of visual and textual information in AKGVP. In contrast, conventional knowledge-graph-based navigators [15], [16], [17], [18] frequently employ discrete one-hot vectors to represent objects in the scene, lacking alignment with visual perception. The application of CLIP presents compelling advantages, including the cultivation of robust representations, improved generalization capabilities, and zero-shot ability. These advantages will collectively augment the capacity of the model to navigate towards the desired goal object in a more cognizant and efficient manner, in the realm of object-goal navigation.

C. Knowledge Graph Modeling

The high-level controller leverages Graph Convolutional Networks (GCNs) [28] to extract prior information from the knowledge graph. This extracted high-level semantic knowledge serves as a valuable guide for the agent to navigate and explore unknown environments more effectively, using the power of the knowledge graph.

Spatial Location Clustering. In order to construct a comprehensive knowledge graph that captures the object distribution within the training environment $q \in Q$, the agent meticulously explores all positions $l = \{x, z, \theta_1, \theta_2\}$. At each position (x, z), the agent diligently detects the presence of each goal object $g_k \in G$ within its observations by Faster RCNN [29], meticulously recording this crucial information

using $f_{(x,z)}$. The detection results from various viewing angles (θ_1, θ_2) will be averaged when sharing the same coordinates (x, z).

$$f_{(x,z)} = \frac{1}{|\theta_1||\theta_2| \sum_k \alpha_k} \sum_{\theta_1, \theta_2, k} \alpha_k f_{obj,k}. \quad (1)$$

where α_k signifies the presence of the goal object g_k within the observation field of view. It takes the value of 1 if the object is visible and 0 otherwise. $f_{obj,k}$ denotes the Object Feature associated with the goal object g_k. Additionally, $|\theta_1|$ and $|\theta_2|$ represent the total choices of yaw and pitch angles, respectively. It is worth noting that scenes with similar characteristics often exhibit a recurring pattern of typical regions adorned with similar object types and layouts. For instance, a bedroom typically encompasses an area comprising a bed, a chest, and pillows, forming a distinctive arrangement. In these analogous scenes, we employ K-Means clustering techniques to group the recorded information $f_{(x,z)}$ corresponding to each spatial coordinate (x, z) across each room. This process effectively divides the room into distinct zones $\{Z_m | (x_i, z_i) \in Z_m, m = 1, \cdots, M\}$, reflecting the underlying variations in object distribution and facilitating more granular analysis of the environment.

Graph Definition. We denote our knowledge graph as $\Omega = (V, E)$, where V represents the nodes and E represents the edges connecting the nodes. In our specific case, e.g., a certain room, each node $v_m \in V$ corresponds to each zone Z_m with its cluster center $\delta(v_m)$. Meanwhile, each edge $e(v_m, v_n) \in E$ captures the adjacent probability between two zones (Z_m, Z_n).

$$\delta(v_m) = f_{Z_m} = \frac{\sum_{(x_i, z_i) \in Z_m} f_{(x_i, z_i)}}{|Z_m|}. \quad (2)$$

$$e(v_m, v_n) = \frac{\sum_{(x_i, z_i) \in Z_m} \sum_{(x_j, z_j) \in Z_n} d_{ij}}{|Z_m||Z_n|}, \qquad (3)$$
$$d_{ij} = \mathbb{I}[|x_i - x_j| + |y_i - y_j| \leq \epsilon].$$

where $|Z_m|$ is the element number, $\mathbb{I}[\cdot]$ is an indicative function, and ϵ is a hyper-parameter threshold. If the condition inside the brackets is satisfied, $\mathbb{I}[\cdot]$ equals 1; otherwise, it equals 0. Given the fixed number of regions M, the knowledge graph of each room exhibits a similar structure. To align the knowledge graphs across different rooms within the same kind of scene (*e.g.*, living room, kitchen, bedroom, and bathroom), we employ the Kuhn-Munkres algorithm for pairwise matching. Subsequently, we merge the nodes and edges by averaging their values, ensuring a cohesive representation across the diverse rooms within the scene. It is important to note that traditional knowledge-graph-based navigators [15], [16], [17], [18] construct nodes using discrete one-hot vectors and vote counting strategy, which fails to fully exploit the information contained in the natural language modality. As a result, these models exhibit limited generalization ability. In contrast, our knowledge graph employs continuous feature representations, enabling more powerful zero-shot generalization.

Graph Adaptation. Building precise representations of every specific scene is challenging, especially in new, unfamiliar environments with varying layouts. To address this, we dynamically update the node features of the initial knowledge graph based on real-time first-person observations. This process enables the initial knowledge graph with general prior information to evolve and adapt to the current environment.

$$\delta(V^t) = \lambda I_{Z_C} f_{(x_t, z_t)}^T + (I - \lambda I_{Z_C} I_{Z_C}^T) \delta(V^{t-1}). \qquad (4)$$

where matrix $V^0 \in \mathbb{R}^{M \times N}$ encapsulates the attributes of all nodes V in the knowledge graph $\Omega = (V, E)$ at time stamp $t = 0$, $I_{Z_C} \in \mathbb{R}^{M \times 1}$ describes the node index of the zone Z_C where the agent is currently located, and N is the feature length, which is 1024. The proximity between the current observation feature $f_{(x_t, z_t)}$ and the cluster centers can be computed, enabling the identification of the closest cluster center, which serves as the current region Z_C. Note that, the observed feature $f_{(x_t, z_t)}$ here is not required to traverse all views (θ_1, θ_2) of the same location (x, z). In addition, λ represents a trainable parameter that governs the global impact of the currently observed feature $f_{(x_t, z_t)}$ on the knowledge graph. It is crucial to emphasize that these updates are episode-specific, as each episode within the scenes of the same category starts with the identical initial knowledge graph.

D. Navigation Policy

The hierarchical navigation policy comprises high-level and low-level controllers. The high-level controller harnesses the power of the knowledge graph to effectively plan sub-goals for guiding the movement of agent across diverse regions. In parallel, the low-level controller leverages the integrated multimodal input to exhibit motion control.

High-level Controller. This paper defines the target zone Z_T as the region in the initial knowledge graph with the highest probability of containing the target object g. The objective of the agent is to plan a path that starts from the current zone Z_C and reaches the target zone Z_T with the maximum connection probability, which is calculated as the product of all edges along the path $\prod_t e_t$. By doing so, real-time planning of sub-goal zones Z_S becomes possible. Meanwhile, we leverage the power of GCN, a specialized neural network designed to process graph-structured data. GCN provides an effective framework for modeling and comprehending intricate relationships within knowledge graphs. The GCN output generates node-level representations while preserving the number of nodes and the length of node features. In Fig. 2, the output Graph Feature $f_{gra} \in \mathbb{R}^{1024}$ of the high-level controller refers to a specific node feature extracted from the GCN output, corresponding to the sub-goal zone. With the utilization of GCN, our approach enables the agent to make informed decisions based on the understanding of complex relationships within the knowledge graph. By incorporating GCN into our framework, we enhance the ability of agent to plan paths towards sub-goals in a more efficient and effective manner.

Low-level Controller. Similar to the traditional knowledge-graph-based navigators, AKGVP incorporates LSTM to model the low-level controller and learn the policy $\pi(a_t | s_t, g)$. The network takes a concatenation of Image Feature f_{img}, Object Feature f_{obj}, Graph Feature f_{gra}, and Previous Action f_{act} as input. Among them, f_{act} is the historical action of the last timestamp decision by agent, represented by a 6-dimensional one-hot vector. To optimize the performance of the low-level controller, we employ the A3C [30] algorithm within the framework of reinforcement learning. This algorithm enables effective training and policy optimization, empowering the agent to make informed decisions and navigate efficiently within the environment.

III. EXPERIMENTS

A. Experimental Setup

Datasets. We evaluate the efficacy of our approach using AI2-THOR [25] simulator, renowned for its provision of highly realistic 3D indoor scenes with near-photographic observations. The AI2-THOR dataset encompasses a diverse collection of 120 scenes across four categories: living room, kitchen, bedroom, and bathroom, featuring a wide range of 22 kinds of goal objects {AlarmClock, Book, Bowl, CellPhone, Chair, CoffeeMachine, DeskLamp, FloorLamp, Fridge, GarbageCan, Kettle, Laptop, LightSwitch, Microwave, Pan, Plate, Pot, RemoteControl, Sink, StoveBurner, Television, Toaster}. Each scene includes a minimum of 4 goal objects. Following the conventions set by prior research [9], [11], [17], [18], we allocate 20 rooms for training, 5 for validation, and 5 for testing purposes, ensuring a consistent and fair experimental setup. Furthermore, to validate the zero-shot capability of the navigators, we deliberately excluded the six goal objects {Bowl, DeskLamp, Laptop, LightSwitch, Plate, StoveBurner} from the training set. Consequently, the test set exclusively focuses on evaluating the performance with respect to these six specific goal

objects. This allows us to assess the generalization ability of navigators in dealing with unseen objects.

Settings. In the framework setting, a reward system is introduced aimed at minimizing the trajectory length to the goal. Specifically, when the agent successfully reaches the goal object within a specified number of steps, it receives a significant positive reward of 5. Conversely, for each step taken towards the goal without reaching it, a small negative reward of -0.01 is imposed. For AI2-THOR-related settings, please refer to Sec. II. Adam optimizer is adopted to update the network parameters with a learning rate of 10^{-4}. To train our models, a total of 6 million episodes is conducted, ensuring that each episode involves randomly selecting the starting position and goal for the agent.

Evaluations. We conducted a rigorous evaluation by conducting experiments in triplicate, and the reported results are presented as mean $\pm$ standard deviation. To assess the performance of our approach, we selected three evaluation metrics: Success Rate (SR), Success weighted by Path Length (SPL), and Distance to Goal (DTS). In the tables below, an upward arrow ($\uparrow$) indicates that a higher value is desirable, while a downward arrow ($\downarrow$) indicates that a lower value is preferred. SR evaluates the ability of agent to successfully locate the target object, while SPL takes into account both success rate and path length. DTS quantifies the distance between the agent and the goal at the end of each episode.

B. Performance Analysis

To provide comprehensive evidence of the superiority of AKGVP, we conduct both qualitative and quantitative analyses, comparing it with the State-Of-The-Art (SOTA) navigators, *e.g.*, HOZ [17] and L-sTDE [18]. The quantitative results are presented in Tab. I and Tab. II, while the qualitative results are depicted in Fig. 3. In Tab. I and Tab. II, AKGVP-Base refers to the algorithm described earlier, while AKGVP-CI, based on the AKGVP-Base version, incorporates the L-sTDE [18] method by using the causal inference algorithm to eliminate the prediction bias brought by the experience, *e.g.*, prior knowledge from the knowledge graph. These analyses offer a thorough evaluation and highlight the advantages of AKGVP in ObjNav tasks.

General Navigation Performance. As depicted in Tab. I, our proposed AKGVP method showcases outstanding performance in the regular ObjNav task when compared to SOTA navigators. Specifically, AKGVP-CI achieves the highest success rate (SR) of 76.78% and the shortest distance to the goal (DTS) of 0.35m, surpassing the L-sTDE method by 2.59% and 0.08m, respectively. Moreover, AKGVP-Base demonstrates the highest success weighted by path length (SPL) with a value of 40.66%. It is worth noting that the SPL value of AKGVP-CI might not be optimal due to the agent being in closer proximity to the target when the navigation stops, resulting in additional steps. For evaluation purposes, an agent is deemed successful without discrimination if it is within a distance of 1.5m from the goal object. These results in Tab. I collectively underline the superior performance of our method, showcasing higher success rates and more effi-

TABLE I
GENERAL NAVIGATION PERFORMANCE COMPARISON.

Navigator	SR↑(%)	SPL↑(%)	DTS↓(m)
Random	4.35 ± 1.69	2.39 ± 1.37	1.41 ± 0.01
Baseline [12]	58.76 ± 0.18	34.48 ± 0.43	0.68 ± 0.01
SP [16]	62.19 ± 0.67	37.60 ± 0.35	0.61 ± 0.02
SAVN [7]	63.27 ± 0.11	38.20 ± 0.04	0.56 ± 0.01
EOTP [8]	65.61 ± 0.25	38.93 ± 0.10	0.55 ± 0.01
ORG [11]	66.53 ± 0.29	39.00 ± 0.34	0.54 ± 0.01
ORG+TPN [11]	68.60 ± 0.29	39.40 ± 0.17	0.54 ± 0.01
VTNet [9]	70.10 ± 1.00	39.60 ± 0.10	0.52 ± 0.01
HOZ [17]	70.48 ± 0.54	39.84 ± 0.21	0.48 ± 0.02
L-sTDE [18]	74.19 ± 0.60	40.30 ± 0.27	0.43 ± 0.01
Ours AKGVP (Base)	73.63 ± 0.56	$\mathbf{40.66}\pm0.22$	0.44 ± 0.02
Ours AKGVP (CI)	$\mathbf{76.78}\pm0.51$	39.63 ± 0.32	$\mathbf{0.35}\pm0.01$

TABLE II
ZERO-SHOT NAVIGATION PERFORMANCE COMPARISON.

Navigator	SR↑(%)	SPL↑(%)	DTS↓(m)
Baseline [12]	19.34 ± 0.28	8.81 ± 0.41	1.20 ± 0.01
EmbCLIP [10]	44.14 ± 0.54	15.76 ± 0.21	0.79 ± 0.01
HOZ [17]	47.54 ± 0.53	15.05 ± 0.35	0.76 ± 0.01
L-sTDE [18]	54.75 ± 0.60	16.64 ± 0.31	0.62 ± 0.02
Ours AKGVP (Base)	51.80 ± 0.51	17.47 ± 0.28	0.69 ± 0.01
Ours AKGVP (CI)	$\mathbf{69.51}\pm0.47$	$\mathbf{28.86}\pm0.25$	$\mathbf{0.41}\pm0.02$

cient path planning in general ObjNav task, when compared to the baselines and other state-of-the-art approaches.

Zero-shot Navigation Performance. In comparison to the results presented in Tab. I, the zero-shot navigation performance demonstrated in Tab. II is particularly remarkable. The AKGVP-CI method demonstrates robust generalization capabilities even when encountering unknown objects. While there is a slight decline in the performance of AKGVP-CI compared to the general navigation scenario, with a decrease in SR by 7.27%, an increase in SPL by 10.77%, and a slight increase in DTS by 0.06m, the Baseline and SOTA navigators experience more significant setbacks with deteriorated performance indicators. Additionally, the zero-shot navigator EmbCLIP, which solely relies on the CLIP image encoder without utilizing the CLIP language encoder and high-level controller, also underperforms. These results highlight the importance of the continuous knowledge graph architecture, as well as multimodal feature alignment, in achieving superior performance in zero-shot navigation tasks.

Visualization and Qualitative Study. As depicted in Fig. 3, AKGVP-CI exhibits robust generalization capabilities in both general navigation and zero-shot navigation tasks. While the state-of-the-art navigator L-sTDE excels in general navigation, it frequently encounters confusion in zero-shot navigation, *e.g.*, premature stops, incorrect target identification, and progress blocked by obstacles. Furthermore, AKGVP-CI demonstrates more precise action decision-making, ensuring smoother and more efficient navigation trajectories.

C. Ablation Study

In this section, we conduct three types of ablation experiments to investigate the impact of different pre-training settings (Tab. III) and model components (Tab. IV) on the
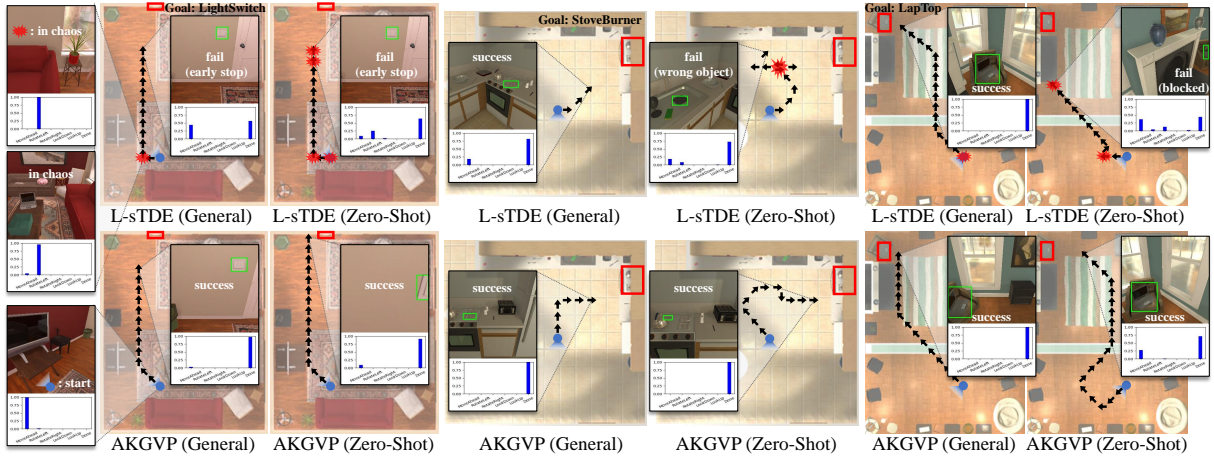
Fig. 3. Qualitative results (zoom in for detailed viewing). The visualization of navigation results for the four navigators in three rooms is presented from left to right, along with the corresponding observations from the final frame of navigation. The red explosion icon denotes instances where the agent becomes disoriented and exhibits erratic behavior, such as spinning in circles, getting stuck by obstacles, or repetitive small-scale rotations. The blue circle represents the starting position, which is consistent across all four navigators. The histogram provides a visual representation of the action probabilities, highlighting the likelihood of the agent selecting each of the six actions. For additional instances of the comparison between these two methods, please refer to the accompanying video.

TABLE III

ABLATION STUDY ON PRE-TRAINING SETTINGS.

Pre-training	Frozen Parameters	SR↑(%)	SPL↑(%)	DTS↓(m)
ImageNet [31]	✓	73.66	37.21	0.46
ImageNet [31]	✗	74.12	38.31	0.43
CLIP [26]	✓	**76.78**	39.63	**0.35**
CLIP [26]	✗	75.86	**40.97**	0.40

TABLE IV

ABLATION STUDY ON MODEL COMPONENT DECOMPOSITION.

f_{img}	f_{obj}	f_{gra}	f_{act}	SR↑(%)	SPL↑(%)	DTS↓(m)
✗	✓	✗	✗	9.47	1.45	1.22
✓	✗	✗	✗	16.80	2.81	1.37
✓	✓	✗	✗	69.57	37.58	0.48
✓	✓	✓	✗	73.63	38.82	0.38
✓	✓	✓	✓	**76.78**	**39.63**	**0.35**

overall performance of our model. The objective is to provide insights into the key factors influencing performance.

Form of Pre-training. The purpose of this ablation experiment is to assess the effectiveness of visual language pre-training. While keeping the model structure unchanged, we replace the visual encoder in AKGVP with the ResNet-50 [27] pre-trained on the ImageNet [31] dataset, and remove the language encoder, thereby reverting to the one-hot encoding approach by the traditional navigator. The results are recorded in Tab. III. It is evident that the algorithm without multimodal pre-training alignment exhibits an average decrease of 2.28% in SR, 2.54% in SPL, and an increase of 0.06m in DTS compared to the aligned algorithm. This indicates the importance and effectiveness of multimodal pre-training for achieving superior navigation performance.

Frozen Parameters. To assess the impact of frozen parameters on navigation performance, an experiment is conducted that investigated whether to freeze the parameters of the pre-training model in Tab. III. When the pre-trained model lacks multimodal alignment, unfreezing the parameters leads to improved navigation performance. Specifically, the SR value increases by 0.46%, the SPL value increases by 1.10%, and

the DTS decreases by 0.03m. In contrast, navigators that have undergone multimodal alignment show no significant performance gains when unfreezing the encoder parameters. This observation indicates that the contribution of unfreezing the parameters to training the encoder is limited.

Model Components. To investigate the impact of the four inputs (Image Feature f_{img}, Object Feature f_{obj}, Graph Feature f_{gra}, Previous Action f_{act}) on the decision-making process of the low-level controller, we conducted a series of ablation experiments by dissecting the model structure, as presented in Tab. IV. The results reveal that f_{img} and f_{obj} exert the most significant influence, with the removal of either input resulting in a substantial decline in navigator performance. On the other hand, the inclusion of sub-goal information f_{gra} provided by the high-level controller and historical action information f_{act} enhances path planning capabilities, leading to slight improvements in navigation performance.

IV. CONCLUSION

The contributions of this research lie in the development of the Aligning Knowledge Graph with Visual Perception (AKGVP) method for the ObjNav task, which addresses the misalignment between discrete scene features and first-person visual observations through continuous knowledge graph modeling and visual-language pre-training. Furthermore, we provide comprehensive evaluations of AKGVP, showcasing its superior performance and efficiency in both general and zero-shot object-goal navigation tasks. By aligning language description with visual perception, AKGVP holds promise for advancing the field of embodied intelligence, enabling more accurate and effective navigation in dynamic environments.

References

[1] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020.

[2] B. Li, J. Han, Y. Cheng, C. Tan, P. Qi, J. Zhang, and X. Li, "Object goal navigation in eobodied ai: A survey," in *Proceedings of the International Conference on Video, Signal and Image Processing*, 2022, pp. 87–92.

[3] X. Han, S. Li, X. Wang, and W. Zhou, "Semantic mapping for mobile robots in indoor scenes: a survey," *Information*, vol. 12, no. 2, p. 92, 2021.

[4] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, 2021.

[5] 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.

[6] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *International Conference on Robotics and Automation*. IEEE, 2019, pp. 8846–8852.

[7] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6750–6759.

[8] B. Mayo, T. Hazan, and A. Tal, "Visual navigation with spatial attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 898–16 907.

[9] H. Du, X. Yu, and L. Zheng, "Vtnet: Visual transformer network for object goal navigation," in *International Conference on Learning Representations*, 2021, pp. 1–12.

[10] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 829–14 838.

[11] H. Du, X. Yu, and L. Zheng, "Learning object relation graph and tentative policy for visual navigation," in *European Conference on Computer Vision*, 2020, pp. 19–34.

[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 *International Conference on Robotics and Automation*. IEEE, 2017, pp. 3357–3364.

[13] W. B. Shen, D. Xu, Y. Zhu, L. J. Guibas, L. Fei-Fei, and S. Savarese, "Situational fusion of visual representation for visual navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2881–2890.

[14] O. Maksymets, V. Cartillier, A. Gokaslan, E. Wijmans, W. Galuba, S. Lee, and D. Batra, "Thda: Treasure hunt data augmentation for semantic navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 374–15 383.

[15] Y. Wu, Y. Wu, A. Tamar, S. Russell, G. Gkioxari, and Y. Tian, "Bayesian relational memory for semantic visual navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2769–2779.

[16] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," in *International Conference on Learning Representations*, 2019, pp. 1–11.

[17] S. Zhang, X. Song, Y. Bai, W. Li, Y. Chu, and S. Jiang, "Hierarchical object-to-zone graph for object navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 130–15 140.

[18] S. Zhang, X. Song, W. Li, Y. Bai, X. Yu, and S. Jiang, "Layout-based causal inference for object navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 792–10 802.

[19] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4247–4258, 2020.

[20] Y. Liang, B. Chen, and S. Song, "Sscnav: Confidence-aware semantic scene completion for visual semantic navigation," in *International Conference on Robotics and Automation*. IEEE, 2021, pp. 13 194–13 200.

[21] S. Wani, S. Patel, U. Jain, A. Chang, and M. Savva, "Multion: Benchmarking semantic map memory using multi-object navigation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9700–9712, 2020.

[22] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal navigation with interaction-free learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 890–18 900.

[23] 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.

[24] F. Pourpanah, M. Abdar, Y. Luo, X. Zhou, R. Wang, C. P. Lim, X.-Z. Wang, and Q. J. Wu, "A review of generalized zero-shot learning methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[25] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An Interactive 3D Environment for Visual AI," *arXiv*, 2017.

[26] 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.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[28] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[30] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1928–1937.

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.