KERM: Knowledge Enhanced Reasoning for Vision-and-Language Navigation

Xiangyang Li[1,2], Zihan Wang[1,2], Jiahao Yang[1,2], Yaowei Wang[3], Shuqiang Jiang[1,2,3]
[1]Key Lab of Intelligent Information Processing Laboratory of the Chinese Academy of Sciences (CAS),
Institute of Computing Technology, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
[3]Peng Cheng Laboratory, Shenzhen, 518055, China
lixiangyang@ict.ac.cn, {zihan.wang, jiahao.yang}@vipl.ict.ac.cn, wangyw@pcl.ac.cn, sqjiang@ict.ac.cn

Abstract

Vision-and-language navigation (VLN) is the task to enable an embodied agent to navigate to a remote location following the natural language instruction in real scenes. Most of the previous approaches utilize the entire features or object-centric features to represent navigable candidates. However, these representations are not efficient enough for an agent to perform actions to arrive the target location. As knowledge provides crucial information which is complementary to visible content, in this paper, we propose a Knowledge Enhanced Reasoning Model (KERM) to leverage knowledge to improve agent navigation ability. Specifically, we first retrieve facts (i.e., knowledge described by language descriptions) for the navigation views based on local regions from the constructed knowledge base. The retrieved facts range from properties of a single object (e.g., color, shape) to relationships between objects (e.g., action, spatial position), providing crucial information for VLN. We further present the KERM which contains the purification, fact-aware interaction, and instruction-guided aggregation modules to integrate visual, history, instruction, and fact features. The proposed KERM can automatically select and gather crucial and relevant cues, obtaining more accurate action prediction. Experimental results on the REVERIE, R2R, and SOON datasets demonstrate the effectiveness of the proposed method. The source code is available at https://github.com/XiangyangLi20/KERM.

1. Introduction

Vision-and-language navigation (VLN) [3, 12, 23, 24, 36, 38] is one of the most attractive embodied AI tasks, where agents should be able to understand natural language instructions, perceive visual content in dynamic 3D environments, and perform actions to navigate to the target location. Most previous methods [9, 15, 22, 31, 34] depend on sequential models (*e.g.,* LSTMs and Transformers) to continuously receive visual observations and align them with the instructions to predict actions at each step. More recently, transformer-based architectures [5, 7, 25] which make use of language instructions, current observations, and historical information have been widely used.

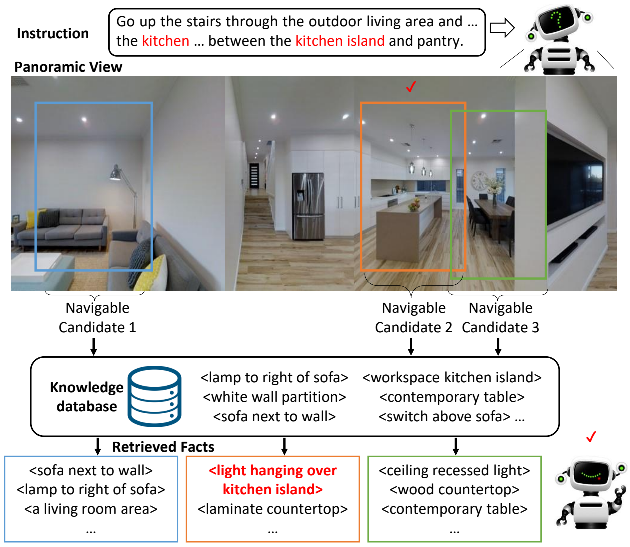

Figure 1. Illustration of knowledge related navigable candidates, which provides crucial information such as attributes and relationships between objects for VLN. Best viewed in color.

Most of the previous approaches utilize the entire features [5, 12, 13, 25] or object-centric features [1, 7, 10, 20] to represent navigable candidates. For example, Qi *et al.* [22] and Gao *et al.* [10] encode discrete images within each panorama with detected objects. Moudgil *et al.* [20] utilize both object-level and scene-level features to represent visual observations. However, these representations are not efficient enough for an agent to navigate to the target location. For example, as shown in Figure 1, there are three candidates. According to the instruction and the current location, candidate2 is the correct navigation. Based on the entire features of a candidate view, it is hard to select the correct one, as candidate2 and candidate3 belong to the same category (*i.e.,* "dining room"). Meanwhile, it is also hard to differentiate them from individual objects, as "lamp" and "light" are the common components for them.

As humans make inferences under their knowledge [11], it is important to incorporate knowledge related to navigable candidates for VLN tasks. First, knowledge provides crucial information which is complementary to visible content. In addition to visual information, high-level abstraction of the objects and relationships contained by knowledge provides essential information. Such information is indispensable to align the visual objects in the view image with the concepts mentioned in the instruction. As shown in Figure 1, with the knowledge related to candidate2 (*i.e.,* <light hanging over kitchen island>), the agent is able to navigate to the target location. Second, the knowledge improves the generalization ability of the agent. As the alignment between the instruction and the navigable candidate is learned in limited-seen environments, leveraging knowledge benefits the alignment in the unseen environment, as there is no specific regularity for target object arrangement. Third, knowledge increases the capability of VLN models. As rich conceptual information is injected into VLN models, the correlations among numerous concepts are learned. The learned correlations are able to benefit visual and language alignment, especially for tasks with high-level instructions.

In this work, we incorporate knowledge into the VLN task. To obtain knowledge for view images, facts (*i.e.,* knowledge described by language descriptions) are retrieved from the knowledge base constructed on the Visual Genome dataset [16]. The retrieved facts by CLIP [26] provide rich and complementary information for visual view images. And then, a knowledge enhanced reasoning model (KERM) which leverages knowledge for sufficient interaction and better alignment between vision and language information is proposed. Especially, the proposed KERM consists of a purification module, a fact-aware interaction module, and an instruction-guided aggregation module. The purification model aims to extract key information in the fact representations, the visual region representations, and the historical representations respectively guided by the instruction. The fact-aware interaction module allows visual and historical representations to obtain the interaction of the facts with cross-attention encoders. And the instruction-guided aggregation module extracts the most relevant components of the visual and historical representations according to the instruction for fusion.

We conduct the experiments on three VLN datasets, *i.e.,* the REVERIE [24], SOON [36], and R2R [3]. Our approach outperforms state-of-the-art methods on all splits of these datasets under most metrics. The further experimental analysis demonstrates the effectiveness of our method.

In summary, we make the following contributions:

- We incorporate region-centric knowledge to comprehensively depict navigation views in VLN tasks. For each navigable candidate, the retrieved facts (*i.e.,* knowledge described by language descriptions) are complementary to visible content.
- We propose the knowledge enhanced reasoning model (KERM) to inject fact features into the visual representations of navigation views for better action prediction.
- We conduct extensive experiments to validate the effectiveness of our method and show that it outperforms existing methods with a better generalization ability.

2. Related Work

Vision-and-Language Navigation. VLN [3, 6, 7, 14, 25, 31, 33, 34] has drawn significant research interests because of its practicality for potential applications (*e.g.,* healthcare robots and personal assistants). Given natural language instructions (*e.g.,* step-by-step instructions [3, 17], high-level instructions [24, 36], and dialog-based instructions [32]), VLN tasks require the agent to navigate to the goal location. Early methods [9, 34, 37] usually utilize recurrent neural networks (RNNs) to encode historical observations and actions, which are represented as a state vector. To make further advances, numerous approaches focus on incorporating more visual information. For example, Hong *et al.* [14] propose to learn the relationships among the scene, objects, and directional clues. In order to capture environment layouts, Wang *et al.* [33] employ topological maps to memorize the percepts during navigation, supporting long-term planning.

Inspired by the big success of vision-and-language pretraining (VLP) [26, 30], more recent approaches utilize transformer-based architectures. Hao *et al.* [13] propose PREVALENT which pretrains the navigation model under the self-learning paradigm. To learn general navigation-oriented textual representations, AirBERT [12] and HM3D-AutoVLN [6] create large-scale VLN data to improve the interaction between different modalities. VLNBERT [15] injects a recurrent function into cross-modal structures to maintain time-dependent state information. HAMT [5] concatenates language instructions, historical and current observations, and then feed them into a cross-modal transformer to derive multi-modal action prediction. DUET [7] combines local observations and the global topological map via graph transformers, facilitating action planning and cross-modal understanding. In this work, we leverage mined knowledge (*i.e.,* retrieved facts) of viewpoint images to improve the generalization ability and facilitate the alignment between vision and language for VLN.

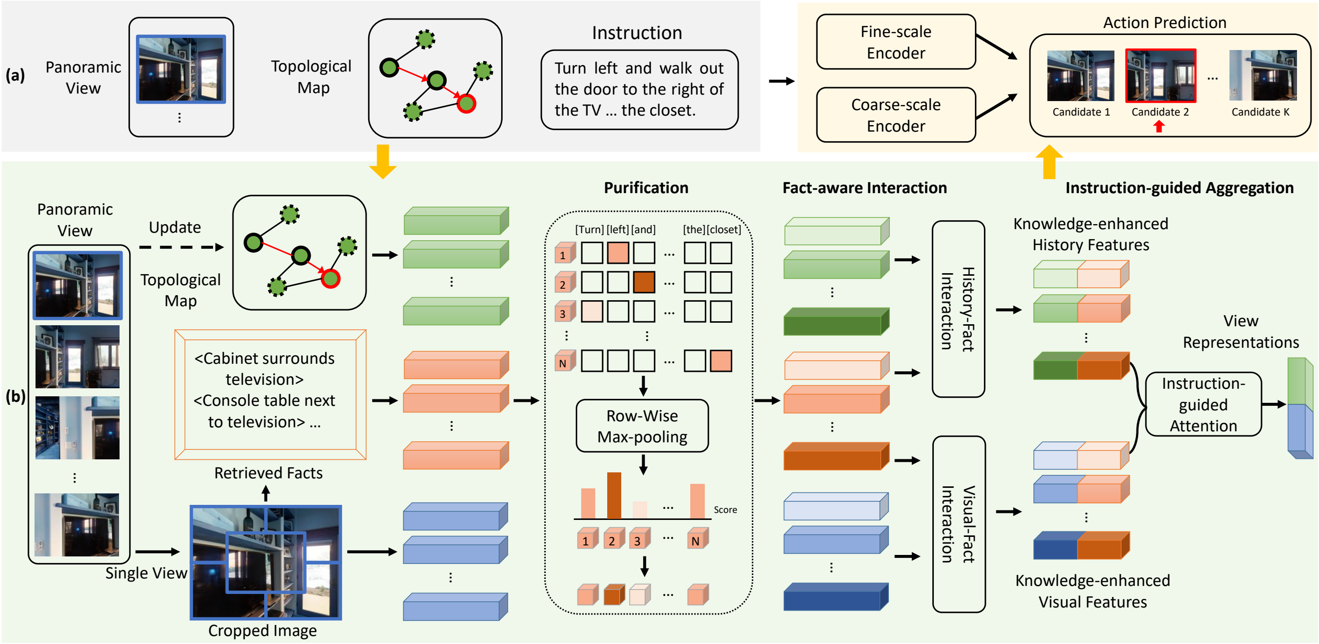

Figure 2. The overall pipeline. (a) The baseline method uses a dual-scale graph transformer to encode the panoramic view, the topological map, and the instruction for action prediction. (b) Our approach incorporates retrieved facts as additional input. The knowledge enhanced representations of each candidate view are obtained with the purification module, the fact-aware interaction module, and the instruction-guided aggregation module. Best viewed in color.

Vision-Language Reasoning with External Knowledge. There has been growing attention to incorporating external knowledge for multi-modal understanding and reasoning [10, 21, 28]. ConceptNet [29] and DBpedia [4] are widely used knowledge bases where concepts and relationships are represented with nodes and edges respectively. The structured knowledge can be represented with graph neural networks or graph transformers, enabling interaction within visual and linguistic features. Based on acquired knowledge, KE-GAN [21] utilizes the knowledge relation loss to re-optimize parsing results. CKR [10] leverages relationships among concepts according to ConceptNet to learn correlations among rooms and object entities. Another line of work employs unstructured knowledge which is retrieved from knowledge bases such as Wiktionary [19]. For example, K-LITE [28] enriches natural language supervision with knowledge descriptions, thus learning image representations that can understand both visual concepts and their knowledge. In this work, we leverage the knowledge that provides complementary and necessary information for the view images to benefit VLN tasks.

3. Method

We address VLN in discrete environments [3, 24, 36]. The environment is provided with a navigation connectivity graph $\mathcal{G}, = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ denotes navigable nodes and $\mathcal{E}$ denotes edges. Given natural language instructions, the agent needs to explore the environment to reach the target locations. $\hat{L} = \{\hat{l}_i\}_{i=1}^{M}$ denotes the word embeddings of the instruction containing M words. At step t, the agent observes a panoramic view of its current node V_t. The panoramic view contains 36 single views $\mathcal{S}_t = \{s_i\}_{i=1}^{36}$, and each single view is represented by a vector s_i accompanied with its orientation. The navigable views of V_t are a subset of $\mathcal{S}_t$. For tasks with object grounding [24, 36], N object features $\mathcal{O}_t = \{o_i\}_{i=1}^{N}$ for a panorama are extracted with annotated bounding boxes or object detectors [2].

3.1. Overview of Our Approach

The baseline follows the architecture of DUET [7], as illustrated in Figure 2(a). A topological map is constructed over time by adding newly observed information to the map and updating visual representations of its nodes. Specifically, $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t\}$ denotes the topological map at step t, $\mathcal{G}_t \in \mathcal{G}$ is the map of the environment after t navigation steps. As shown in Figure 2(a), there are three kinds of nodes in $\mathcal{V}_t$: visited nodes (*i.e.,* nodes with solid black border), the current node (*i.e.,* the node with solid red border), and navigable nodes (*i.e.,* nodes with dashed black border). Then at each step, all the features of the 36 single views for the current location, the topological map, and the instruction are fed into the dual-scale graph transformer to predict the next location in the map or the stop action.

Our model which leverages knowledge to obtain contextualized representation is illustrated in Figure 2(b). First, facts that provide complementary information to the visual features of each discrete view are retrieved. Then, the instruction, the topological map, the visual view features, and the corresponding facts are fed into the purification module, the fact-aware interaction module, and the instruction-guided aggregation module gradually to form knowledge enhanced representations. At last, the dual-scale graph transformer [7] is utilized to predict the action.

3.2. Fact Acquisition

To navigate to the target location, high-level abstraction of the objects and relationships by natural language provides essential information which is complementary to visual features. Such information is indispensable to model the relevance between the visual objects and concepts mentioned in instructions, facilitating the matching between candidate views and instructions. We first construct a knowledge base and then use the pretrained multi-modal model CLIP [26] to retrieve facts for each view.

Knowledge Base Construction. The knowledge base is the source of relevant facts that describe the visual region. In order to obtain the rich and varied descriptions, similar to [18], we parse the region descriptions from the Visual Genome [16] dataset to construct the knowledge base. Specifically, in the Visual Genome dataset, the parsed attribute annotations take the form of "attribute-object" pairs and the parsed relationship annotations take the form of "subject-predicate-object" triplets. We convert all of the "attribute-object" pairs and "subject-predicate-object" triplets to their synset canonical forms. After removing duplicates, we totally get 630K facts expressed by language descriptions that are used to build our knowledge base.

Fact Retrieval. Our main goal is to obtain facts for the view images when navigating in the environment. To this end, we crop each view image into five sub-regions (as shown in Figure 3) and retrieve facts for these sub-regions from the knowledge base. In order to retrieve relevant facts for the visual sub-region, we use the pretrained model CLIP [26]. It consists of an image encoder CLIP-I and a text encoder CLIP-T that encode image and text into a joint embedding space. We use the CLIP-T to encode all facts in the knowledge base as the search keys. The visual sub-regions are encoded by the CLIP-I as the queries. We then search in the knowledge base for the facts with the top-k highest cosine similarity scores. For each sub-region, we keep five facts with the top-5 highest cosine scores as the knowledge. Some examples of the top-5 results are shown in Figure 3.

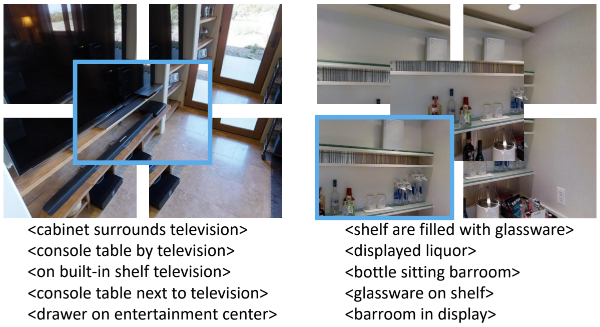

Figure 3. Five cropped sub-regions of the view image and the retrieved top-5 facts for the sub-region in the blue box.

3.3. Knowledge Enhanced Reasoning

At each step t, the visual features, the history features, the instruction features, and the fact features are fed into our proposed knowledge enhanced reasoning model (KERM) to support better action prediction, as shown in Figure 2(b). Specifically, for each single view n, the visual features $R_n = \{r_i\}_{i=1}^{5}$ are composed of 5 sub-regions where each sub-region r_i is encoded by CLIP-I. The fact features $E_n = \{e_i\}_{i=1}^{25}$ are the total fact representations, where each fact e_i is encoded by CLIP-T. To represent historical information and the layout of the explored environment, the history features H_n which are composed of all the node representations of the topological map $\mathcal{G}_t$ are also employed. As the same in [7], word embeddings $\hat{L}$ are fed into a multi-layer transformer to obtain the instruction features L.

3.3.1 Purification Module

Due to the complexity of the navigation environment, a large number of fact features within each view are not all needed by the agent to complete the current navigation task. The agent needs more critical, highly correlated knowledge with the current navigation instruction to understand and characterize the environment image. Therefore, we propose an instruction-aware fact purification module that aims to filter out less relevant information and capture critical information that is closely related to the current navigation task. Specifically, we first evaluate the relevance of each fact in the view to each token of navigation instruction by computing the fact-instruction relevance matrix A as:

$$A = \frac{(E_n W_1)(L W_2)^T}{\sqrt{d}} \tag{1}$$

where W_1 and W_2 are learned parameters and d is the dimension of features. After this, we compute the row-wise max-pooling on A to evaluate the relevance of each fact to the instruction as:

$$\alpha_n = max\{A_{n,i}\}_{i=1}^{M} \tag{2}$$

The fact features are purified by the guidance of instruction-aware attention as:

$$E_n^{'} = \alpha_n E_n \tag{3}$$

The purified visual region features $R_n^{'}$ and history features $H_n^{'}$ are obtained in the same way.

3.3.2 Fact-aware Interaction Module

After obtaining the purified features, we use a multi-layer cross-modal transformer for vision-fact interaction. Each transformer layer consists of a cross-attention layer and a self-attention layer. The cross-attention between visual features and fact features is calculated as:

$$\tilde{R_n} = CrossAttn(R_n^{'}, E_n^{'}) \tag{4}$$

And then, $\tilde{R_n}$ are fed into the self-attention layer to obtain the knowledge enhanced visual features $R_n^{''}$. The history-fact interaction is also conducted in this same way with another multi-layer cross-modal transformer, forming the knowledge enhanced history features $H_n^{''}$.

3.3.3 Instruction-guided Aggregation Module

At last, we use the instruction-guided attention to aggregate the vision-fact features into a contextual vector $\bar{r}_n$:

$$\eta_n = softmax(\frac{(R_n W_3)(L_0 W_4)^T}{\sqrt{d}}) \tag{5}$$

$$\bar{r}_n = \sum_{i=1}^{K} \eta_{n,i} R_{n,i}^{''} \tag{6}$$

where W_3 and W_4 are learned parameters and d is the dimension of features, K is the number of sub-regions (*i.e., K*=5), L_0 are the feature from the [CLS] token representing the entire instruction. With the same mechanism, we aggregate the history-fact features into $\bar{h}_n$. Then we use an FFN to fuse $\bar{r}_n$ and $\bar{h}_n$ into the knowledge enhanced representations of view n, which are fed into the dual-scale graph transformer [7] for action prediction.

3.4. Training and Inference

Pretraining. As demonstrated in [5, 7, 25], it is beneficial to pretrain transformer-based VLN models. We utilize the following four tasks to pretrain our KERM model.

1) Masked language modeling (MLM). We randomly mask out the words of the instruction with a probability of 15%, and then the masked words L_m are predicted with the corresponding contextual features.

2) Masked view classification (MVC). The MVC requires the model to predict the semantic labels of masked view images. We randomly mask out view images with a probability of 15%. Similar to [7], the target labels for view images are obtained by an image classification model [8] pretrained on ImageNet.

3) Single-step action prediction (SAP). Given a demonstration path P^* and its partial $P^*_{<t}$, the SAP loss in behavior cloning is defined as follows:

$$L_{SAP} = \sum_{t=1}^{T} -log\, p(a_t | L, P^*_{<t}) \tag{7}$$

where a_t is the expert action of $P^*_{<t}$.

4) Object grounding (OG). OG is used when object annotations are available:

$$L_{OG} = -log\, p(o^* | L, P_D) \tag{8}$$

where o^* is the ground-truth object at the destination location P_D. More details are presented in the supplementary materials.

Fine-tuning and Inference. For fine-tuning, we use the imitation learning method [27] and the SAP loss L_{SAP}. Different from the pretrain process using the demonstration path, the supervision of fine-tuning is from a pseudo-interactive demonstrator. It selects a navigable node as the next target node with the overall shortest distance from the current node to the destination. For tasks with object annotations, we also use the object grounding loss L_{OG}.

For inference, the model predicts an action at each step. If the action is not the stop action, the agent performs this action and moves to the predicted node, otherwise, the agent stops at the current node. The agent will be forced to stop if it exceeds the maximum action steps and then returns to the node with maximum stop probability as its final prediction. At the predicted location, the object with the highest prediction score is selected as the target.

4. Experiment

4.1. Datasets and Evaluation Metrics

Datasets. We evaluate our model on the REVERIE [24], SOON [36], and R2R [3] datasets. REVERIE contains high-level instructions and the instructions contain 21 words on average. The predefined object bounding boxes are provided for each panorama, and the agent should select the correct object bounding box from candidates at the end of the navigation path. The path length is between 4 and 7 steps. SOON also provides instructions that describe the target locations and target objects. The average length of instructions is 47 words. However, the object bounding boxes are not provided, while the agent needs to predict the center location of the target object. Similar to the settings in [7], we use object detectors [2] to obtain candidate object boxes. The path length of SOON is between 2 and 21 steps. R2R

Table 1. Comparison with state-of-the-art methods on the REVERIE dataset.

Methods	Val Seen						Val Unseen						Test Unseen					
	Navigation				Grounding		Navigation				Grounding		Navigation				Grounding	
	TL↓	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑	TL↓	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑	TL↓	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑
Seq2Seq [3]	12.88	35.70	29.59	24.01	18.97	14.96	11.07	8.07	4.20	2.84	2.16	1.63	10.89	6.88	3.99	3.09	2.00	1.58
RCM [34]	10.70	29.44	23.33	21.82	13.23	15.36	11.98	14.23	9.29	6.97	4.89	3.89	10.60	11.68	7.84	6.67	3.67	3.14
VLNBERT [15]	13.44	53.90	51.79	47.96	38.23	35.61	16.78	35.02	30.67	24.90	18.77	15.27	15.68	32.91	29.61	23.99	16.50	13.51
AirBERT [12]	15.16	49.98	47.01	42.34	32.75	30.01	18.71	34.51	27.89	21.88	18.23	14.18	17.91	34.20	30.28	23.61	16.83	13.28
HOP [25]	13.80	54.88	53.76	47.19	38.65	33.85	16.46	36.24	31.78	26.11	18.85	15.73	16.38	33.06	30.17	24.34	17.69	14.34
HAMT [5]	12.79	47.65	43.29	40.19	27.20	15.18	14.08	36.84	32.95	30.20	18.92	17.28	13.62	33.41	30.40	26.67	14.88	13.08
DUET [7]	13.86	73.68	71.75	63.94	57.41	51.14	22.11	51.07	46.98	33.73	32.15	23.03	21.30	56.91	52.51	36.06	31.88	22.06
KERM-pt (Ours)	14.25	**74.49**	**71.89**	**64.04**	**57.55**	**51.22**	22.47	**53.65**	**49.02**	**34.83**	**33.97**	**24.14**	18.38	**57.44**	52.26	**37.46**	**32.69**	**23.15**
KERM (Ours)	12.84	**79.20**	**76.88**	**70.45**	**61.00**	**56.07**	21.85	**55.21**	**50.44**	**35.38**	**34.51**	**24.45**	17.32	**57.58**	**52.43**	**39.21**	**32.39**	**23.64**

provides step-by-step instructions and is not required to predict object locations. The average length of instructions is 32 words and the average length of paths is 6 steps.

Evaluation Metrics. We utilize the standard evaluation metrics [3, 24] for VLN tasks to compare our method with previous approaches, including (1) Trajectory Length (TL): the agent's average path length in meters; (2) Navigation Error (NE): average distance in meters between the agent's final location and the target one; (3) Success Rate (SR): the proportion of successfully executed instructions with the NE less than 3 meters; (4) Oracle SR (OSR): SR given the oracle stop policy; (5) SPL: SR penalized by Path Length. For the REVERIE and SOON datasets that require object grounding, we adopt metrics including (1) Remote Grounding Success (RGS): proportion of successfully executed instructions; (2) RGSPL: RGS penalized by Path Length. Except for TL and NE, all metrics are higher the better.

4.2. Implementation Details

4.2.1 Model Architectures

We adopt the pretrained CLIP-ViT-B/16 [26] to retrieve facts for each view. For the fact-aware interaction module, the number of layers for the cross-modal transformer is set as 2, with a hidden size of 768. The parameters of this module are initialized with the pretrained LXMERT [30].

We use the ViT-B/16 [8] pretrained on ImageNet to extract object features on the REVERIE dataset as it provides bounding boxes. The BUTD object detector [2] is utilized on the SOON dataset to extract object bounding boxes. For the dual-scale graph transformer [7], the number of layers for the language encoder, panorama encoder, coarse-scale cross-modal encoder, and fine-scale cross-modal encoder are set as 9,2,4 and 4, respectively. And the parameters are also initialized with the pretrained LXMERT [30].

4.2.2 Training Details

For pretraining, we set the batch size as 16 using 4 NVIDIA RTX3090 GPUs. For the REVERIE dataset, we combine the original dataset with augmented data synthesized by DUET [7] to pretrain our model with 100k iterations. Then we fine-tune the pretrained model with the batch size of 2 for 20k iterations on 4 NVIDIA RTX3090 GPUs. For the SOON dataset, we only use the original data with automatically cleaned object bounding boxes, sharing the same settings in DUET [7]. We pretrain the model with 40k iterations. And then we fine-tune the pretrained model with the batch size of 1 for 20k iterations on 4 NVIDIA RTX3090 GPUs. For the R2R dataset, additional augmented R2R data in [13] is used in pretraining. We pretrain our model with 200k iterations. Then we fine-tune the pretrained model with the batch size of 2 for 20k iterations on 4 NVIDIA RTX3090 GPUs. For all the datasets, the best epoch is selected by SPL on the val unseen split.

4.3. Comparison to State-of-the-Art Methods

Table 1,2,3 illustrate the results of our method on the REVERIE, R2R and SOON datasets. Our approach achieves state-of-the-art performance under most metrics on all the seen/unseen splits of all three datasets, demonstrating the effectiveness of the proposed model. In particular, as shown in Table 1, our model outperforms the previous DUET [7] by 5.52% on SR, 6.51% on SPL, and 4.93% on RGSPL for val seen split of the REVERIE dataset. As shown in Table 2 and 3, our approach also shows performance gains on the R2R and SOON datasets, but not as pronounced on the REVERIE dataset. It is mainly because of the smaller size of data and more complex instructions, which make the generalization ability of our knowledge enhancement method not fully exploited.

In addition, as shown in Table 1, we show the results of our model with two training strategies. KERM denotes the full pipeline which is composed of pretraining and fine-tuning. For KERM, our model is first pretrained

Table 2. Comparison with state-of-the-art methods on the R2R dataset. † indicates reproduced results.

Methods	Val Seen				Val Unseen				Test Unseen			
	TL↓	NE↓	SR↑	SPL↑	TL↓	NE↓	SR↑	SPL↑	TL↓	NE↓	SR↑	SPL↑
Seq2Seq [3]	11.33	6.01	39	-	8.39	7.81	22	-	8.13	7.85	20	18
AuxRN [37]	-	3.33	70	67	-	5.28	55	50	-	5.15	55	51
PREVALENT [13]	10.32	3.67	69	65	10.19	4.71	58	53	10.51	5.30	54	51
EntityGraph [14]	10.13	3.47	67	65	9.99	4.73	57	53	10.29	4.75	55	52
VLNBERT [15]	11.13	2.90	72	68	12.01	3.93	63	57	12.35	4.09	63	57
AirBERT [12]	11.09	2.68	75	70	11.78	4.01	62	56	12.41	4.13	62	67
HOP [25]	11.26	2.72	75	70	12.27	3.80	64	57	12.68	3.83	64	59
HAMT [5]	-	-	-	-	11.46	2.29	66	61	12.27	3.93	65	60
DUET [7]	-	-	-	-	13.94	3.31	72	60	14.73	3.65	69	59
DUET†	12.30	2.28	78.84	72.89	13.94	3.31	71.52	60.42	14.74	3.65	69.25	58.68
KERM (Ours)	12.16	**2.19**	**79.73**	**73.79**	13.54	**3.22**	**71.95**	**60.91**	14.60	**3.61**	**69.73**	**59.25**

Table 3. Comparison with state-of-the-art methods on the val unseen split of the SOON dataset.

Method	TL↓	OSR↑	SR↑	SPL↑	RGSPL↑
GBE [36]	28.96	28.54	19.52	13.34	1.16
DUET [7]	36.20	50.91	36.28	22.58	3.75
KERM (Ours)	35.83	**51.62**	**38.05**	**23.16**	**4.04**

with augmented data, where the parameters are initialized with LXMERT [30]. And then, it is fine-tuned with standard data. KERM-pt denotes the pipeline which only fine-tunes our knowledge enhanced reasoning model with standard data, based on the baseline method (*i.e.,* the pretrained model in DUET [7]). Both models outperform the previous methods, further illustrating the effectiveness of the proposed knowledge enhancement method. Then, with the help of pretraining, KERM achieves better performance, reflecting that the generalization ability of our method can be further strengthened by pretraining with larger data.

4.4. Further Analysis

Ablation Study. As shown in Table 4, we evaluate the impact of the key components of our model on the val unseen split of REVERIE. First, the results in row 2-4 are better than row 1, which demonstrate that all these modules are beneficial for navigation. Then, the results in row 5 are better than row 2,3, validating the effect of the knowledge enhanced representations in the vision-fact interaction or the history-fact interaction. Finally, the results in row 5 are better than row 4, validating the effectiveness of purification operations. With the purification, interaction, and aggregation modules, our method obtains the best performance.

Fact vs. Object. In order to illustrate that region-based knowledge is superior to the object-centric information, we compare our method with the model utilizing object enhanced representations. Specifically, for each cropped region image, we use VinVL [35] to detect objects and take five objects with the highest scores as substitutes for facts. For fair comparison with retrieved facts, we prompt each object label by adding "an image of a" before the label to form a short sentence and then use CLIP-T to obtain object-centric features. The purification module and fact-aware interaction module are the same as the fact-based approach. As shown in Table 5, we compare the baseline method and the methods of adding object features and fact features on the val unseen split of REVERIE. We can see that the object features are beneficial to navigation, but they are still inferior to fact representations. This is consistent with our assumption that object tags are not sufficient to provide strong generalization performance compared to facts which have a much larger semantic feature space.

Table 4. Ablation study results on val unseen split of the REVERIE dataset. "Pur." denotes the purification module. "VF-Int." and "HF-Int." denote the vision-fact and history-fact interaction module respectively.

Pur.	VF-Int.	HF-Int.	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑
			51.07	46.98	33.73	32.15	23.03
✓		✓	53.78	49.33	35.01	33.80	23.89
✓	✓		53.53	48.96	34.70	33.63	23.58
	✓	✓	54.01	49.43	34.79	33.22	23.80
✓	✓	✓	**55.21**	**50.44**	**35.38**	**34.51**	**24.45**

Table 5. Comparison between fact and object.

	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑
Baseline	51.07	46.98	33.73	32.15	23.03
+ Object	54.27	49.73	33.95	33.91	23.40
+ Fact (KERM)	**55.21**	**50.44**	**35.38**	**34.51**	**24.45**

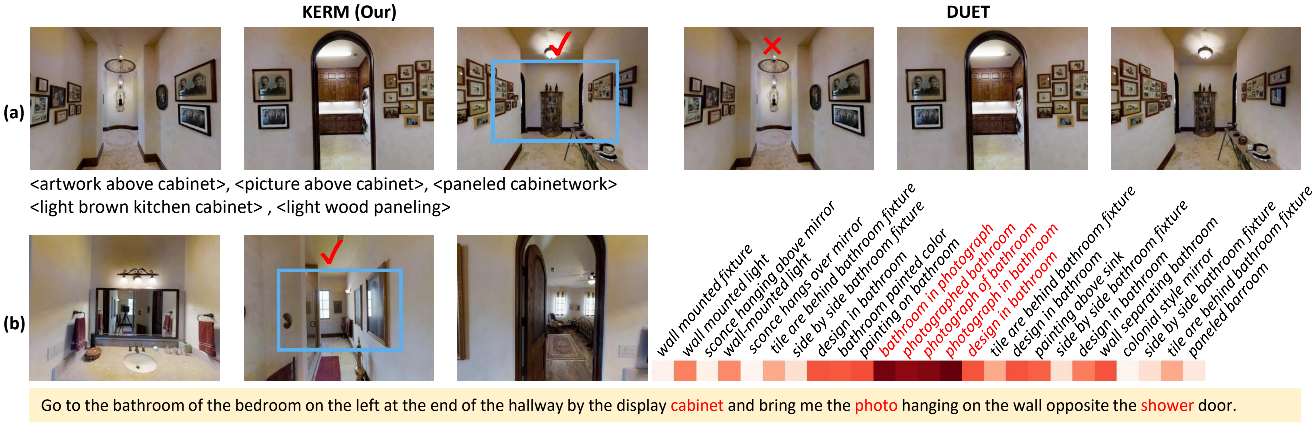
KERM (Our)
DUET
(a)
(b)
<artwork above cabinet>, <picture above cabinet>, <paneled cabinetwork>
<light brown kitchen cabinet> , <light wood paneling>
wall mounted fixture
wall mounted light
sconce hanging above mirror
wall-mounted light
sconce hangs over mirror
tile are behind bathroom fixture
side by side bathroom fixture
design in bathroom
bathroom painted color
painting on bathroom
bathroom in photograph
photographed bathroom
photograph of bathroom
photograph in bathroom
design in bathroom
tile are behind bathroom fixture
design in bathroom
painting above sink
side by side bathroom fixture
design in bathroom
wall separating bathroom
colonial style mirror
side by side bathroom fixture
tile are behind bathroom fixture
paneled barroom
Go to the bathroom of the bedroom on the left at the end of the hallway by the display cabinet and bring me the photo hanging on the wall opposite the shower door.

Figure 4. Visualization of navigation examples. The sentence within the yellow box is the navigation instruction for the agent. (a) shows a comparison where our KERM chooses the right location while the baseline model makes the wrong choice. (b) illustrates the weights for the 25 facts after the fact purification when the agent navigates to the right location. The facts for the sub-region within the blue box are annotated in red. The results show that our model can automatically select the relevant facts.

Table 6. The effect of the number of cropped sub-regions.

Number	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑
1	55.69	49.67	34.46	34.05	23.89
5	**55.21**	**50.44**	**35.38**	**34.51**	**24.45**
9	53.59	48.99	34.87	33.14	23.79

Number of Regions. As shown in Table 6, we explore the impact of the number of cropped sub-regions on navigation performance on the val unseen split of REVERIE. It is demonstrated that the strategy of five sub-regions yields the best results. It is mainly because fewer but larger sub-regions are difficult to retrieve more fine-grained facts, while more but smaller regions are too fragmented to contain all the complete parts that can retrieve accurate facts.

4.5. Qualitative Results

Examples in the val unseen split of REVERIE are illustrated in Figure 4. The sentence within the yellow box is the navigation instruction of these examples and the red color words are the landmarks (*e.g.,* "cabinet" and "shower door") or target objects (*e.g.,* "photo"). Figure 4(a) shows an example where our KERM chooses the correct navigation direction by capturing the facts associated with the "cabinet", as shown in the left bottom of Figure 4(a). The baseline model makes the wrong choice without the facts. The left of Figure 4(b) shows the cropped region in the blue box where target objects (*i.e.,* objects belong to "photo") are contained. In the right of Figure 4(b), a heat map is utilized to visualize the weight distribution after the purification process of the 25 facts mined from the target view image. Particularly, we use red color to highlight the five facts only mined from the region in the blue box. We can see that the facts annotated in red contain the words related to the target object (*i.e.,* "photo hanging on the wall opposite the shower door") in the navigation instruction, such as "photograph" and "bathroom". It is worth noting that these facts related to the target object have almost the highest purification weights, which demonstrates that our model can automatically select the relevant facts, thus obtain better performance.

5. Conclusion

In this paper, we propose KERM, a knowledge enhanced reasoning model to improve the generalization ability and performance for VLN. Our work utilizes a large number of external fact descriptions to build a knowledge base and introduces the CLIP model to retrieve the facts (*i.e.,* knowledge described by language descriptions) for view images. We further design a knowledge enhanced reasoning approach with the process of purification, interaction, and aggregation that automatically obtains contextual information for action prediction. We illustrate the good interpretability of KERM and provide case study in deep insights. Our approach achieves excellent improvement on many VLN tasks, demonstrating that leveraging knowledge is a promising direction in improving VLN and Embodied AI. For future work, we will improve our KERM with larger training data and employ it on VLN in continuous environments.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grants 62125207, 62102400, 62272436, U1936203, and U20B2052, in part by the National Postdoctoral Program for Innovative Talents under Grant BX20200338, and in part by the Peng Cheng Laboratory Research Project under Grant PCL2021A07.

References

[1] Dong An, Yuankai Qi, Yan Huang, Qi Wu, Liang Wang, and Tieniu Tan. Neighbor-view enhanced model for vision and language navigation. In *ACM MM*, pages 5101–5109, 2021. 1

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018. 3, 4.1, 4.2.1

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018. 1, 1, 2, 3, 4.1, 1, 4.1, 2

[4] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, 2007. 2

[5] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, pages 5834–5847, 2021. 1, 1, 2, 3.4, 1, 2

[6] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Learning from unlabeled 3d environments for vision-and-language navigation. In *ECCV*, pages 638–655, 2022. 2

[7] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, pages 16537–16547, 2022. 1, 1, 2, 3.1, 3.3, 3.3.3, 3.4, 4.1, 1, 4.2.1, 4.2.2, 4.3, 2, 3

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 3.4, 4.2.1

[9] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *NeurIPS*, 2018. 1, 2

[10] Chen Gao, Jinyu Chen, Si Liu, Luting Wang, Qiong Zhang, and Qi Wu. Room-and-object aware knowledge reasoning for remote embodied referring expression. In *CVPR*, pages 3064–3073, 2021. 1, 2

[11] Gerd Gigerenzer and Daniel G Goldstein. Reasoning the fast and frugal way: models of bounded rationality. *Psychological review*, 103(4):650, 1996. 1

[12] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *CVPR*, pages 1634–1643, 2021. 1, 1, 2, 1, 2

[13] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *CVPR*, pages 13137–13146, 2020. 1, 2, 4.2.2, 2

[14] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. In *NeurIPS*, pages 7685–7696, 2020. 2, 2

[15] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *CVPR*, pages 1643–1653, 2021. 1, 2, 1, 2

[16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017. 1, 3.2

[17] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, pages 4392–4412, 2020. 2

[18] Chia-Wen Kuo and Zsolt Kira. Beyond a pre-trained object detector: Cross-modal textual and visual context for image captioning. In *CVPR*, pages 17969–17979, 2022. 3.2

[19] Christian M Meyer and Iryna Gurevych. *Wiktionary: A new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography*. na, 2012. 2

[20] Abhinav Moudgil, Arjun Majumdar, Harsh Agrawal, Stefan Lee, and Dhruv Batra. SOAT: A scene-and object-aware transformer for vision-and-language navigation. In *NeurIPS*, pages 7357–7367, 2021. 1

[21] Mengshi Qi, Yunhong Wang, Jie Qin, and Annan Li. Kegan: Knowledge embedded generative adversarial networks for semi-supervised scene parsing. In *CVPR*, pages 5237–5246, 2019. 2

[22] Yuankai Qi, Zizheng Pan, Yicong Hong, Ming-Hsuan Yang, Anton van den Hengel, and Qi Wu. The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation. In *ICCV*, pages 1655–1664, 2021. 1, 1

[23] Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. Object-and-action aware model for visual language navigation. In *ECCV*, pages 303–317, 2020. 1

[24] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, pages 9982–9991, 2020. 1, 1, 2, 3, 4.1, 4.1

[25] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. HOP: History-and-order aware pre-training for vision-and-language navigation. In *CVPR*, pages 15418–15427, 2022. 1, 1, 2, 3.4, 1, 2

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 1, 2, 3.2, 3.2, 4.2.1

[27] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, pages 627–635, 2011. 3.4

[28] Sheng Shen, Chunyuan Li, Xiaowei Hu, Yujia Xie, Jianwei Yang, Pengchuan Zhang, Anna Rohrbach, Zhe Gan, Lijuan Wang, Lu Yuan, et al. K-lite: Learning transferable visual models with external knowledge. In *NeurIPS*, pages 15558–15573, 2022. 2

[29] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451, 2017. 2

[30] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, pages 5103–5114, 2019. 2, 4.2.1, 4.3

[31] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL*, pages 2610–2621, 2019. 1, 2

[32] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *CoRL*, pages 394–406, 2019. 2

[33] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *CVPR*, pages 8455–8464, 2021. 2

[34] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *CVPR*, pages 6629–6638, 2019. 1, 2, 1

[35] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *CVPR*, pages 5579–5588, 2021. 4.4

[36] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *CVPR*, pages 12689–12699, 2021. 1, 1, 2, 3, 4.1, 3

[37] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *CVPR*, pages 10012–10022, 2020. 2, 2

[38] Wanrong Zhu, Yuankai Qi, Pradyumna Narayana, Kazoo Sone, Sugato Basu, Xin Wang, Qi Wu, Miguel Eckstein, and William Yang Wang. Diagnosing vision-and-language navigation: What really matters. In *NAACL*, pages 5981–5993, 2022. 1