Collaborative Map-Based and Route-Based Policy Learning for Continuous Vision-and-Language Navigation

Jiewen Hou, *Graduate Student Member, IEEE*, Meina Kan, *Member, IEEE*, Lixuan Zhang, *Graduate Student Member, IEEE*, Hao Liang, *Graduate Student Member, IEEE*, Shiguang Shan, *Fellow, IEEE*, and Xilin Chen, *Fellow, IEEE*

***Abstract*—Vision-and-Language Navigation in Continuous Environments (VLN-CE) requires an agent to follow language instructions to reach a target in unseen, 3D environments. A powerful VLN-CE agent requires two crucial abilities during cross-modal planning: spatial reasoning to explore towards the target location in partially observable environments, and procedural alignment to match navigation routes with language instructions for sequential guidance. Existing cross-modal planning policies are divided into two types, each focusing on one of these capabilities. The map-based policy promotes spatial reasoning by grounding instructions on graph representations, while the route-based policy favors procedural alignment by aligning sequential observations with language instructions. However, these policies are typically studied independently, making agents that rely on a single crucial ability unable to plan effectively in complex scenes. Inspired by human navigation, we propose a collaborative policy learning framework that integrates the advantages of both policies for cross-modal planning. This framework includes three processes: Spatio-Procedural Topological Mapping, which constructs a multiplex graph to support the learning of map-based and route-based policies; Dual-Stream Encoding, which performs cross-modal encoding for both policies in parallel; and Hierarchical Policy Integration, which fuses both policies via feature-level collaboration and logit-level fusion. Extensive experiments on the VLN-CE datasets confirm the effectiveness of our framework.**

***Index Terms*—Vision-based navigation, ai-enabled robotics, representation learning.**

I. Introduction

VISION-AND-LANGUAGE navigation (VLN) tasks [1] require an agent to interpret natural language instructions and navigate to a target location using visual observations. For real-world applications, Krantz et al. [2] introduce VLN in continuous environments (VLN-CE), allowing the agent to traverse within a 3D mesh freely with low-level actions. In order to make informed action decisions in unseen, complex environments, a VLN agent must master two crucial abilities for cross-modal planning: (1) spatial reasoning to explore towards the target location in partially observable environments by grounding instructions; and (2) procedural alignment between the agent's route and language instructions to proceed towards the target under sequential guidance. As shown in Fig. 1, when executing the instruction, "*Leave the bedroom and take a right in the hallway. Enter the bathroom on your left.*", an agent requires spatial reasoning to find landmarks outside its current view, such as determining which exit from the bedroom leads to the hallway. It also needs procedural alignment to distinguish the correct target from multiple similar options, such as identifying which of several bathrooms is the one specified by the instruction.

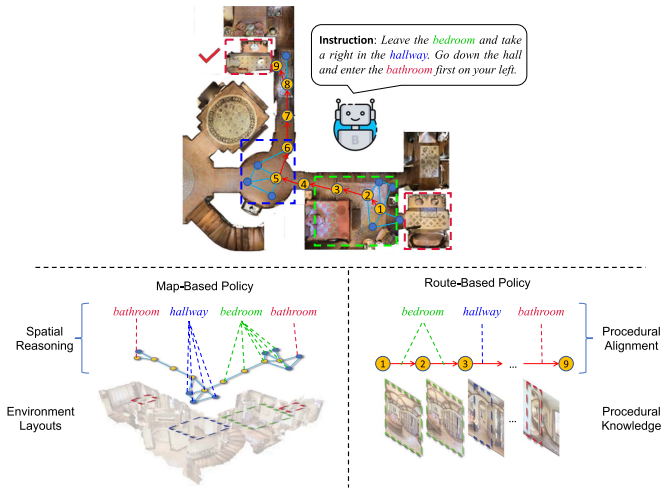

Fig. 1. A powerful VLN agent requires two crucial abilities: 1) Spatial Reasoning: Grounding the landmarks (e.g., *bedroom*, *hallway*, *bathroom*) from the language instruction into the environment for effective exploration towards the target; 2) Procedural Alignment: Ensuring the agent's route adheres to the procedural sequence described in the language instruction (*bedroom* → *hallway* → *bathroom*).

Existing planning policies infer the next action through cross-modal encoding between structural memories and language instructions. Map-based policies promote spatial reasoning by grounding instructions on graph representations (e.g., metric maps [3], [4] or topological maps [5], [6]). However, these

Received 30 September 2025; accepted 19 January 2026. Date of publication 9 February 2026; date of current version 25 February 2026. This article was recommended for publication by Associate Editor S. Cascianelli and Editor M. Vincze upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grant 62495084, Grant 62461160331, and Grant 62495082. *(Corresponding author: Meina Kan.)*

The authors are with the State Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, Chinaand also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: houjiewen24z@ict.ac.cn; kanmeina@ict.ac.cn; lixuan.zhang@vipl.ict.ac.cn; lianghao21s@ict.ac.cn; sgshan@ict.ac.cn; xlchen@ict.ac.cn).

Code is available at: https://github.com/VIPL-EPP/CoMaR.

Digital Object Identifier 10.1109/LRA.2026.3662659

policies lack procedural guidance, such as grounding the bathroom description to the wrong bathroom beside the initial location in Fig. 1. Route-based policies facilitate procedural alignment by aligning procedural knowledge (e.g., egocentric observation sequence [7], [8]) with language instructions. These policies are less effective at abstracting the environment layouts, such as the bedroom and hallway, when relying solely on sequential observations. These two policies have been studied separately, yielding agents that are specialized in a single ability. Consequently, these agents struggle to perform effective cross-modal planning in unseen environments.

However, humans rely on two forms of structural knowledge [9] for navigation: One is survey knowledge, a cognitive map to abstract the environment layouts, enabling high-level reasoning (e.g., shortcut discovery and detour planning). The other is procedural knowledge, which sequentially connects landmarks with associated movements from egocentric views. Inspired by the dual-knowledge system of human navigation, we propose a Collaborative Map-based and Route-based policy learning framework (CoMaR), aiming to empower the VLN agent with robust spatial reasoning and procedural alignment, through three core processes: 1) **Spatio-Procedural Topological Mapping**: During navigation, we online update a multiplex graph, which includes a spatial graph to capture the layouts of the explored environment and a procedural graph to capture procedural knowledge along the agent's route. 2) **Dual-Stream Encoding**: A graph transformer encodes the environment layouts, and performs instruction grounding for the map-based policy. Concurrently, a temporal transformer encodes procedural knowledge of traversed routes, and assesses consistency with the language instructions for the route-based policy. 3) **Hierarchical Policy Integration**: The map-based and route-based policies are integrated at both the feature and logit levels for cross-modal planning. We validate the effectiveness of our proposed framework through extensive experiments on the VLN-CE datasets [2] (R2R-CE and RxR-CE), conducted on the Habitat platform.

In summary, the main contributions of this work are threefold: 1) We propose a collaborative policy learning framework CoMaR for VLN, equipping the agent with more robust spatial reasoning and precise procedural alignment capabilities. 2) We introduce three key innovative designs that enable deep collaboration between map-based and route-based policies: a multiplex graph for spatio-procedural topological mapping, dual-stream cross-modal encoding, and hierarchical policy integration at the feature and logit levels. 3) Extensive experiments on two standard VLN-CE benchmarks demonstrate that CoMaR achieves state-of-the-art performance, validating the efficacy of our approach.

II. Related Work

VLN in Continuous Environments: Most of the early work in VLN was established over discretized simulated scenes (e.g., R2R [1] and RxR [10]), where agents teleport between nodes on a pre-defined navigation graph. Despite the efficiency of this approach, directly transferring models trained in discrete spaces to real-world robot applications is impractical. To facilitate real-world deployment, the more realistic paradigm of VLN in continuous environments (VLN-CE [2]) was proposed. In VLN-CE tasks, such as R2R-CE and RxR-CE, agents can navigate freely to any unobstructed space in the simulator using low-level actions. Current VLN-CE methods can be categorized into two groups: topology graph-based approaches, such as HNR [6] and ETPNav [5], which perform cross-modal reasoning over an online topological graph to predict a navigable location, and execute low-level actions with a controller module; and video-based approaches built on pretrained vision-language models (VLMs), including NaVid [7], Uni-NaVid [11], and NAVILA [12], which infer low-level actions end-to-end by modeling interaction between egocentric observation sequence and language instructions. Despite differences in designs, these methods can be divided into two categories based on their memory structures for cross-modal planning: the map-based policy, which relies on topological graphs, and the route-based policy, which relies on sequences of egocentric observations.

Map-Based Policy for VLN: In human navigation, the brain constructs a cognitive map which stores knowledge of the spatial relationships between goals, landmarks, and other salient points in space. Such capabilities primarily engage entorhinal/hippocampal circuits in human brains [9]. To achieve better spatial reasoning ability, many VLN methods adopt spatial maps—such as topological maps [5], metric maps [13], and grid maps [14]—to abstract the environment along the traversed routes. These different map types provide the agent with spatial awareness of environment layouts at various scales. A map-based policy is then learned, which performs spatial reasoning by encoding these environment layouts and modeling their cross-modal interaction with the language instructions. This policy has two key advantages: it encourages the agent to explore its surroundings to resolve ambiguities in language instructions, and it facilitates backtracking to correct navigation errors [15].

Route-Based Policy for VLN: In addition to map-based policies, the human brain utilizes a route-based planning policy. This policy is built upon procedural knowledge, which models the direct connection between navigation behavior and landmarks and has been linked by many studies to striatal and parietal circuits [9]. The sequential structure of first-person video captures procedural knowledge for navigation. Many VLN methods guide the agent's next action by processing this sequence of egocentric observations and aligning it with language instructions. State-of-the-art route-based approaches [7], [11], [12] apply VLMs to model interactions between language instructions and egocentric videos. The sequential structure of these videos enables precise procedural alignment between the language instructions and the navigation route.

Although humans naturally combine both strategies, VLN research has studied map-based and route-based policies separately, leaving their collaboration for cross-modal planning underexplored. This letter addresses this gap by proposing a novel collaborative learning framework, CoMaR, that integrates both approaches.

III. Methodology

A. Navigation Setups

This work addresses the task of VLN in continuous environments, where an agent is required to follow a specific route described by a language instruction to reach the target location. In particular, we follow a practical setting VLN-CE [2], where the agent navigates on a 3D mesh of an environment with low-level actions (e.g., `FORWARD` (0.25 m), `ROTATE LEFT/RIGHT` (15°), and `STOP`). For each navigation episode, the agent is initialized at a starting location and given natural

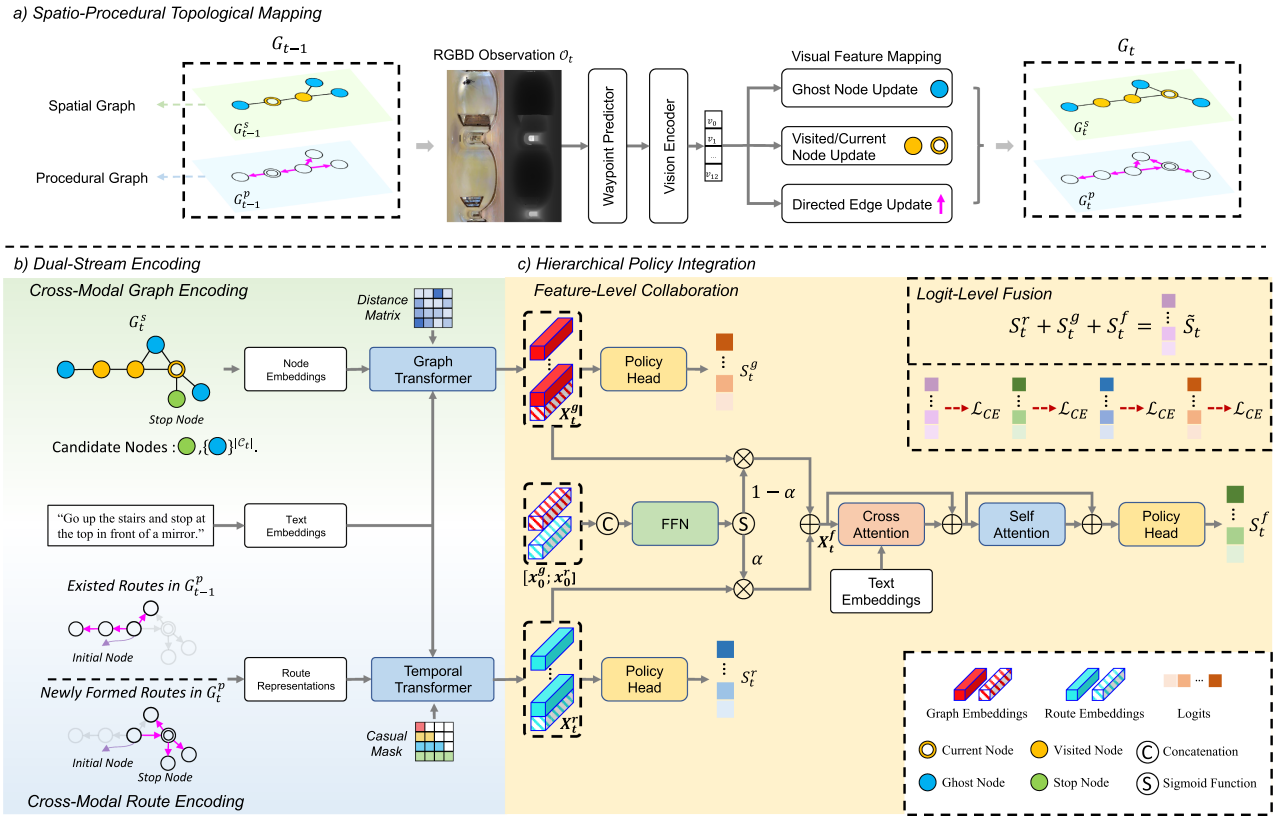

Fig. 2. Our proposed framework CoMaR consists of three main stages: a) Spatio-Procedural Topological Mapping, which constructs a multiplex graph G_t, including a spatial graph G_t^s and a procedural graph G_t^p; b) Dual-Stream Encoding, which performs cross-modal encoding by separately processing each graph layer with the language instructions; c) Hierarchical Policy Integration, which integrates the map-based and route-based policies via feature-level collaboration and logit-level fusion to generate the final decision logits $\tilde{S}_t$.

language instructions $\mathcal{W} = \{w_1, w_2, \ldots, w_L\}$ with encoded text embeddings $\boldsymbol{W}$. The agent then needs to explore the environment and reach the target. At time step t, the agent perceives its surroundings through panoramic vision, consisting of 12 RGB images $\mathcal{R}_t = \{r_{t,i}\}_{i=1}^{12}$ and 12 depth images $\mathcal{D}_t = \{d_{t,i}\}_{i=1}^{12}$, captured at 30-degree intervals around its current position. We assume access to the agent's pose (e.g., position and heading) at each step, and our framework can also be applied under monocular RGB-D sensing configurations.

We follow the setup of topology graph-based approaches [5], [6]. Specifically, at each decision step t, an online topological map is constructed with a pre-trained waypoint predictor [5], which predicts waypoints by identifying potential navigable locations in the agent's surroundings. These waypoints are then integrated as nodes into the online topological map through topological mapping. During cross-modal planning, the agent selects a ghost node (i.e., an unvisited node) on the updated map as a subgoal, which is reached by a series of low-level actions with a controller module [5].

To promote cross-modal planning, we propose CoMaR, a collaborative policy learning framework inspired by human navigation, which supports long-horizon subgoal selection by jointly leveraging map-based and route-based policies. CoMaR consists of three components: (i) Spatio-Procedural Topological Mapping, which maintains a dual-layer graph that captures both environment layout and procedural knowledge; (ii) Dual-Stream Encoding, which performs cross-modal encoding for the two policies in parallel; and (iii) Hierarchical Policy Integration, which fuses the two policies at both the feature and logit levels.

B. Spatio-Procedural Topological Mapping

Spatial reasoning relies on structured memory of environmental layouts. Existing map-based policies [5], [6] use topological mapping to abstract the navigable and visited locations into an undirected spatial graph $G^s = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of edges. Each node $n_i \in \mathcal{N}$ represents a reachable location, storing its global 3D coordinates, $\mathbf{p}_i \in \mathbb{R}^3$, and panoramic-pooled visual features derived from the vision encoder (12-view average pooling for visited nodes; observable-view pooling for ghost nodes [5]). Each edge $e_{i,j} \in \mathcal{E}$ denotes navigability between adjacent nodes and stores their Euclidean distance, $\mathbf{d}_{i,j} \in \mathbb{R}$. However, G^s is insufficient for procedural alignment: **1)** undirected edges overlook visit order along the route (e.g., A$\rightarrow$ B$\rightarrow$ C vs. C$\rightarrow$ B$\rightarrow$ A yield the same spatial graph); **2)** The node features are direction-agnostic due to panoramic pooling, making them weak procedural cues for route representations and ineffective for procedural alignment.

Therefore, as shown in Fig. 2(a), we propose spatio-procedural topological mapping. Specifically, at each time step t, we online update a multiplex graph $G_t = \{G_t^s, G_t^p\}$, where $G_t^s = \{\mathcal{N}_t, \mathcal{E}_t\}$ is the undirected spatial graph and $G_t^p = \{\mathcal{N}_t, \mathcal{E}_t'\}$ denotes a directed procedural graph with a directed

edge set $\mathcal{E}'_t$. The procedural graph addresses the aforementioned limitations of the spatial graph as follows:

- For limitation **1**), the procedural graph includes a directed edge set $\mathcal{E}'_t$ that indicates the partial order of visits between adjacent nodes connected by $(e_{i,j} \in \mathcal{E}_t)$. The direction of an edge is determined by two rules. For an edge between two visited nodes, it points from the one visited earlier to the one visited more recently. For an edge between a visited node and a ghost node, it points from the visited node towards the ghost node.
- To tackle limitation **2**), we leverage the contextual visual embeddings $V_t = \{v_i\}_{i=1}^{12}$ output by the vision encoder [5] to update the visual features of directed edges without panoramic pooling. Specifically, each directed edge feature is the embedding of the egocentric view from the source node towards the target node. Such visual features serve as a direction-specific visual proxy for route representations to support procedural alignment.

We follow the mapping process of ETPNav [5] to update two graph layers simultaneously. The whole update process at time step t is as follows:

$$G_t = \text{Topo-Mapping}(G_{t-1}, \mathcal{O}_t), \tag{1}$$

where $G_t = \{G_t^s, G_t^p\}$, $\mathcal{O}_t = \{\mathcal{R}_t, \mathcal{D}_t\}$ denotes the RGB-D observations at the time step t. To represent a STOP action, we add a "stop" node and connect it to the current node.

C. Dual-Stream Encoding

As shown in Fig. 2(b), after updating the multiplex graph, we perform parallel cross-modal encodings using G_t: instruction grounding for the map-based policy and procedural alignment for the route-based policy. This yields graph and route embeddings ($\boldsymbol{X_t^g}$, $\boldsymbol{X_t^r}$) for candidate nodes, which are then evaluated by the corresponding policies for subgoal selection. To avoid unnecessary revisits [5], candidate nodes are restricted to ghost nodes (denoted as the set $\mathcal{C}_t \subset \mathcal{N}_t$) and the stop node.

1) Cross-Modal Graph Encoding: We use a graph transformer to encode the graph topology and performs instruction grounding based on the encoded instructions $\boldsymbol{W}$ and the node embeddings $\boldsymbol{X_t^n}$ derived from G_t^s. This generates a stack of graph embeddings $\boldsymbol{X_t^g}$ for the map-based policy.

Node Embeddings $\boldsymbol{X_t^n}$: For each node $n_i \in \mathcal{N}_t$ in G_t^s, the visual feature is augmented with a pose encoding and a navigation step encoding. The pose encoding embeds the relative pose of each node with respect to the agent's current position, while the navigation step encoding embeds the latest visited timestep for visited nodes and 0 for ghost nodes. This generates a stack of node embeddings $\boldsymbol{X_t^n}$.

Graph Transformer: The node embeddings and encoded instructions are fed into a graph transformer for cross-modal encoding. The graph transformer is a multi-layer transformer similar to LXMERT [16], containing cross-attention layers and self-attention layers. The cross-attention layer takes the node embeddings $\boldsymbol{X_t^n}$ as queries and the encoded instructions $\boldsymbol{W}$ as keys and values, thereby grounding the language instructions onto the node embeddings. The self-attention layer utilizes graph-aware self-attention (GASA) [5] to encode the topological structure of the spatial graph by using a distance matrix to model the interaction of node embeddings. The distance matrix is constructed from the all-pairs shortest distances obtained from the graph edges $\mathcal{E}_t$. After encoding, the graph embeddings for the current node and the visited nodes are masked to avoid revisits. The entire graph encoding process can be summarized as follows with the graph transformer:

$$\boldsymbol{X_t^g} = \text{Transformer}(\boldsymbol{X_t^n}, \boldsymbol{W}), \tag{2}$$

where $\boldsymbol{X_t^g} = [\boldsymbol{x_0^g}, \boldsymbol{x_1^g}, \ldots, \boldsymbol{x_{|\mathcal{C}_t|}^g}]$, $\boldsymbol{x_0^g}$ denotes the graph embedding of the stop node; $\boldsymbol{x_i^g}, (i > 0)$ denotes the graph embeddings of the ghost nodes in G_t.

2) Cross-Modal Route Encoding: We use a temporal transformer to perform procedural alignment between encoded instructions $\boldsymbol{W}$ and route representations $\boldsymbol{X^e}$ derived from the procedural graph G_t^p. This process generates a stack of route embeddings $\boldsymbol{X_t^r}$ for the route-based policy.

Route Representation $\boldsymbol{X^e}$: The representation for a single route of length l is constructed in several steps. First, we form a stack of its directed edge features along the route, denoted as $[\boldsymbol{e'_1}, \ldots, \boldsymbol{e'_l}]$, $e'_i \in \mathcal{E}'_t$ in G_t^p, capturing the procedural knowledge from the initial node to a specific candidate node. This sequence is then augmented with special tokens: we prepend a learnable "start" token to the beginning of each route. To support the STOP action, any directed edge pointing to the stop node is represented by a learnable "stop" token. To encode the sequential order, we add positional embeddings to the feature sequence, yielding the route representation $\boldsymbol{X^e}$.

Temporal Transformer: The Temporal Transformer includes a multi-layer transformer decoder for cross-modal encoding, and a single transformer encoder layer for comparison of derived route embeddings. Specifically, the route representation $\boldsymbol{X^e}$ is fed into the multi-layer transformer decoder to align with the encoded instructions $\boldsymbol{W}$:

$$\boldsymbol{X^o} = \text{TransformerDecoder}(\boldsymbol{X^e}, \boldsymbol{W}). \tag{3}$$

As demonstrated in Vision-and-Language Pretraining studies [17], the last token of the encoded sequence $\boldsymbol{X^o}$ denotes the route embedding $\boldsymbol{x^o}$, which assesses the alignment between a specific route and language instructions. At each time step t, we only compute the newly formed routes in G_t^p that were not present in G_{t-1}^p. Following the assessment approach [18], a merged causal mask is applied to generate these new route embeddings simultaneously. These newly formed route embeddings are stored in the corresponding candidate nodes in the procedural graph G_t^p. We take an average of the route embeddings if a candidate node possesses embeddings from multiple routes. After alignment, we apply a single transformer encoder layer to refine the route embeddings $\tilde{\boldsymbol{X}}_t^o = [\boldsymbol{x_0^o}, \boldsymbol{x_1^o}, \ldots, \boldsymbol{x_{|\mathcal{C}_t|}^o}]$, which are stored in the candidate nodes of G_t^p:

$$\boldsymbol{X_t^r} = \text{TransformerLayer}(\tilde{\boldsymbol{X}}_t^o), \tag{4}$$

where $\boldsymbol{X_t^r} = [\boldsymbol{x_0^r}, \boldsymbol{x_1^r}, \ldots, \boldsymbol{x_{|\mathcal{C}_t|}^r}]$, $\boldsymbol{x_0^r}$ denotes the final route embedding of the stop node, and $\boldsymbol{x_i^r}(i > 0)$ denotes the final route embeddings of ghost nodes.

D. Hierarchical Policy Integration

As shown in Fig. 2(c), based on the graph and route embeddings ($\boldsymbol{X_t^g}$, $\boldsymbol{X_t^r}$) of the candidate nodes, we integrate the map-based and route-based policies at the feature and logit levels to generate the final decision logits. The agent chooses a candidate node based on the decision logits, and uses a low-level controller [5] to reach it.

1) Feature-Level Collaboration: During feature-level collaboration, we use a gating mechanism to adaptively fuse the graph and route embeddings ($\boldsymbol{X_t^g}, \boldsymbol{X_t^r}$), and apply a cross-modal transformer layer to model interactions with the encoded instructions $\boldsymbol{W}$. This yields fused embeddings $\tilde{\boldsymbol{X}}_t^f$ for a collaborative policy.

Specifically, we generate the gating weight α from the cross-modal embeddings of the stop nodes ($\boldsymbol{x_0^g}, \boldsymbol{x_0^r}$), which summarize the agent's current navigation state. We compute α using a feed-forward network (FFN) followed by a sigmoid function, and use it to linearly fuse the graph and route embeddings to obtain the fused embeddings $\boldsymbol{X_t^f}$:

$$\alpha = \text{Sigmoid}(\text{FFN}([\boldsymbol{x_0^g}; \boldsymbol{x_0^r}])),$$
$$\boldsymbol{X_t^f} = \alpha \times \boldsymbol{X_t^r} + (1-\alpha) \times \boldsymbol{X_t^g}. \tag{5}$$

The fused embeddings $\boldsymbol{X_t^f}$ are fed into a cross-modal transformer layer to model interaction with encoded instructions $\boldsymbol{W}$. This process includes a cross-attention layer to model cross-modal interaction and a self-attention layer to compare the embeddings among the candidate nodes:

$$\tilde{\boldsymbol{X}}_t^f = \text{TransformerLayer}(\boldsymbol{X_t^f}, \boldsymbol{W}), \tag{6}$$

where $\boldsymbol{X}_t^f = [x_0^f, \tilde{x}_1^f \ldots, x_{|\mathcal{C}_t|}^f]$ denotes the final fused embeddings after cross-modal modeling.

2) Logit-Level Fusion: After performing feature-level integration, a navigation policy head NP($\cdot$) generates logits $S_t = [s_0, s_1, \ldots, s_{|\mathcal{C}_t|}]$ for candidate nodes in G_t. The policy head NP($\cdot$) is a feed forward network that outputs a score reflecting the probability preference for selecting each node:

$$s_i = \text{NP}(\boldsymbol{x_i}), i = 0, 1, \ldots, |\mathcal{C}_t|. \tag{7}$$

Here, $\boldsymbol{x_i}$ denotes embeddings of candidate nodes after cross-modal encoding. In logit-level fusion, we generate three distinct policies: map-based policy from the graph embeddings $\boldsymbol{X_t^g}$, route-based policy from the route embeddings $\boldsymbol{X_t^r}$, and collaborative policy from the fused embeddings $\tilde{\boldsymbol{X}}_t^f$. To fully leverage the advantages of these policies, their logits are combined via additive operation:

$$S_t^r = \text{NP}_r(\boldsymbol{X_t^r});\ S_t^g = \text{NP}_g(\boldsymbol{X_t^g});\ S_t^f = \text{NP}_f(\tilde{\boldsymbol{X}}_t^f);$$
$$\tilde{S}_t = S_t^r + S_t^g + S_t^f, \tag{8}$$

where $\tilde{S}_t = [\tilde{s}_0, \tilde{s}_1, \ldots, \tilde{s}_{|\mathcal{C}_t|}]$ is the final logits. The agent selects the next subgoal in a greedy manner based on $\tilde{S}_t$. If $\tilde{s}_0$ is selected, the agent will cease movement at its current position, resulting in the termination of the current episode.

The training objective of CoMaR is to minimize the cross-entropy loss $\mathcal{L}_{CE}$ between the predicted policy logits and the navigation supervision $\mathcal{A}_t^*$, which selects the optimal node from the set of candidate nodes. For R2R-CE, the optimal node is the candidate node that minimizes the distance to the target location [1]. For RxR-CE, which does not assume a shortest-path-to-goal, the optimal node is determined using a path fidelity strategy [10]. We jointly optimize CoMaR by applying a cross-entropy loss $\mathcal{L}_{CE}$ to the logits from all policies:

$$\mathcal{L} = \sum_t \mathcal{L}_{\text{CE}}(S_t^g, \mathcal{A}_t^*) + \mathcal{L}_{\text{CE}}(S_t^p, \mathcal{A}_t^*) + \mathcal{L}_{\text{CE}}(S_t^f, \mathcal{A}_t^*) + \mathcal{L}_{\text{CE}}(\tilde{S}_t, \mathcal{A}_t^*). \tag{9}$$

IV. Experiments

In this section, we first describe our experimental setup, including the datasets, evaluation metrics, and implementation details. Afterwards, we evaluate CoMaR through comparative experiments, ablation studies, qualitative analyses, and real-world evaluations.

A. Experimental Setup

We evaluate our framework on two standard benchmarks for Vision-and-Language Navigation in Continuous Environments (VLN-CE): R2R-CE [1], [2] and RxR-CE [2], [10].

Dataset: R2R-CE [1] adapts Matterport3D [23] to Habitat [24], comprising 5,611 shortest-path trajectories, each annotated with step-by-step English instructions (avg. 32 words). The agent uses 15° turns, 90° FOV, and a 0.10 m chassis radius, allowing obstacle sliding. RxR-CE [10] is a larger multilingual benchmark with longer instructions (avg. 120 words) and includes longer trajectories that are not constrained to shortest paths. Its agent setting is also more challenging (30° turns, 79° FOV, 0.18 m radius) and disallows obstacle sliding, increasing collision risk.

Evaluation Metrics: Following standard practice [25], [26], [27], we evaluate our framework's performance using the following metrics: Trajectory Length (TL), Navigation Error (NE), Success Rate (SR), Oracle Success Rate (OSR), Success weighted by Path Length (SPL), Normalized Dynamic Time Warping (NDTW), and Success weighted by nDTW (SDTW).

Implementation Details: We follow the standard panoramic VLN-CE setup [28] for evaluation in simulation. First, to demonstrate that CoMaR generalizes across datasets, we use ETPNav [5] as the map-based policy and evaluate CoMaR on R2R-CE and RxR-CE. Second, to show that CoMaR is compatible with different map-based policies, we also evaluate CoMaR using the state-of-the-art g3D-LF model [22] as the map-based policy on R2R-CE. We further conduct ablation studies and qualitative analyses using ETPNav. All simulation experiments were implemented in PyTorch [29] and run on 8 NVIDIA RTX 4090 GPUs. For real-world evaluations, all deployed models are trained solely on R2R-CE simulator data, without any real-world fine-tuning. We apply CoMaR under the monocular sensing setup with the g3D-LF backbone [22], using a Unitree Go2 quadruped and running the cross-modal planning pipeline on a cloud server with a single NVIDIA RTX 4090 GPU.

B. Comparison to State-of-The-Art Methods

Tables I and II present the comparison results of our method against other VLN-CE approaches on the R2R-CE and RxR-CE datasets. The results on the R2R-CE dataset (in Table I) show that CoMaR achieves at least a 2% improvement across all metrics over the widely-used baseline ETPNav [5] on the Val-Seen, Val-Unseen, and Test-Unseen splits. This demonstrates the generalization capability of our method in both seen and unseen environments. Furthermore, when applied to the current state-of-the-art method, g3D-LF [22]—a map-based policy that incorporates look-ahead exploration—CoMaR yields further significant performance gains. Specifically, CoMaR improves upon g3D-LF by 5% on both SR and SPL on the Val-Seen split, and by 4% and 3% on SR and SPL respectively on the

TABLE I
COMPARISON WITH STATE-OF-THE-ART METHODS ON THE R2R-CE DATASET

Methods	Val Seen					Val Unseen					Test Unseen				
	TL	NE↓	OSR↑	SR↑	SPL↑	TL	NE↓	OSR↑	SR↑	SPL↑	TL	NE↓	OSR↑	SR↑	SPL↑
SASRA [19]	8.89	7.17	-	36.0	34.0	7.89	8.32	-	24.0	22.0	-	-	-	-	-
CM2 [13]	12.05	6.10	50.7	42.9	34.8	11.54	7.02	41.5	34.3	27.6	13.90	7.70	39	31	24
WS-MGMAP [20]	10.12	5.65	51.7	46.9	43.4	10.00	6.28	47.6	38.9	34.3	12.30	7.11	45	35	28
GridMM [14]	12.69	4.21	69	59	51	13.36	5.11	61	49	41	13.31	5.64	56	46	39
BEVBert [3]	-	-	-	-	-	-	4.57	67	59	50	-	4.70	67	59	50
HNR [6]	11.79	3.67	76	69	61	12.64	4.42	67	61	51	13.03	4.81	67	58	50
ENP-ETPNav [21]	11.82	3.90	73	68	59	11.45	4.69	65	58	50	12.71	5.08	64	56	48
ETPNav [5]	11.78	3.95	72	66	59	11.99	4.71	65	57	49	12.87	5.12	63	55	48
$\text{CoMaR}_{\text{Ours}}$	11.61	3.72	**76**	**69**	**62**	12.49	4.51	67	60	51	13.53	4.91	65	58	50
g3D-LF [22]	12.08	4.01	72	63	55	-	4.53	68	61	52	-	4.53	68	58	51
$\text{CoMaR}_{\text{Ours}}$	11.87	**3.47**	75	68	60	**-**	**4.15**	**70**	**64**	**54**	**-**	**4.37**	**69**	**62**	**54**

TABLE II
COMPARISON WITH STATE-OF-THE-ART METHODS ON THE RXR-CE DATASET

Methods	Val Unseen				
	NE ↓	SR↑	SPL↑	NDTW↑	SDTW↑
LAW-Pano [30]	11.04	10.0	9.0	-	-
Seq2Seq [1]	12.1	13.93	11.96	30.86	11.01
CWP-CMA [28]	8.76	26.59	22.16	47.05	-
CWP-BERT [28]	8.98	27.08	22.65	46.71	-
AO-Planner [31]	7.06	43.3	30.5	50.1	-
Reborn [32]	5.98	48.60	42.05	63.35	41.82
HNR [6]	5.51	56.39	46.73	63.56	**47.24**
ETPNav [5]	5.64	54.79	44.89	61.90	45.33
$\text{CoMaR}_{\text{Ours}}$	**5.22**	**57.39**	**47.04**	**63.96**	47.03

TABLE III
ABLATION STUDY OF ROUTE REPRESENTATION DESIGNS

Methods		R2R-CE Val Unseen			
		NE ↓	OSR ↑	SR↑	SPL↑
Ablation on Route-Based Policy					
Route Policy	*Node Feature*	5.08	**63.02**	52.58	42.92
	Edge Feature	**4.97**	62.80	**55.68**	**48.39**
Ablation on CoMaR					
ETPNav		4.72	64.43	57.20	49.17
CoMaR	*Node Feature*	4.78	64.21	56.66	48.36
	Edge Feature	**4.51**	**67.37**	**59.65**	**51.36**

TABLE IV
ABLATION STUDY OF DUAL-STREAM ENCODING AND HIERARCHICAL POLICY INTEGRATION

DSE		HPI		R2R-CE Val Unseen			
Map	Route	Feature	Logit	NE ↓	OSR ↑	SR↑	SPL↑
✓				4.72	64.43	57.20	49.17
	✓			4.97	62.80	55.68	48.39
✓	✓	✓		**4.47**	67.10	59.54	50.20
✓	✓		✓	4.57	65.95	58.72	51.21
✓	✓	✓	✓	4.51	**67.37**	**59.65**	**51.36**

Test-Unseen split. This demonstrates that our method is applicable to the state-of-the-art map-based approach, showcasing its generalization ability across different map-based policies.

The results on the RxR-CE dataset (in Table II) indicate that CoMaR also significantly boosts the performance of the ETPNav baseline on the Val-Unseen split, demonstrating CoMaR's generalization capability on long-distance VLN-CE datasets. Specifically, our CoMaR outperforms ETPNav by approximately 3% on SR and SPL, and by around 2% on the NDTW and SDTW metrics. Our method surpasses the current state-of-the-art method, HNR, on most metrics, showcasing the competitiveness of CoMaR.

C. Ablation Study

In this section, we conduct ablation experiments to verify the effectiveness of the designs in CoMaR.

Ablation of Route Representation: We conduct an ablation study to compare two route representations: a sequence of node features versus a sequence of directed edge features. As shown in Table III, we evaluate their effects on both the route-based policy and CoMaR, highlighting the necessity of directed edges and the multiplex graph. For the single route-based policy, using node features significantly degrades SR and SPL by 2% and 6% compared to directed edges, since node features lack the directional specificity required for procedural knowledge. In contrast, each directed edge incorporates egocentric directional visual cues to form a directed association between nodes, better capturing route procedures. This design choice also impacts CoMaR: when using node features, CoMaR slightly underperforms the single map-based policy, whereas switching to directed edges yields over a 2% gain on key metrics, underscoring the importance of directed edge features for route representation.

Ablation of Dual-Stream Encoding: As shown in the first, second, and last rows of Table IV, we assess Dual-Stream Encoding (**DSE**) by comparing CoMaR with single-policy variants (map-based or route-based). Both single-policy variants underperform CoMaR by at least 2% across all metrics, indicating that CoMaR effectively integrates the complementary strengths of the two policies. Furthermore, we observe that the route-based policy consistently performs worse than the map-based policy. This is because the map-based policy is a more robust planning policy compared to the route-based policy, supporting more effective exploration and backtracking. This phenomenon is also consistent with the existing theory of human spatial navigation [33].

Ablation of Hierarchical Policy Integration: We conducted ablation studies on Hierarchical Policy Integration (**HPI**), and the results are presented in the third, fourth, and the last rows of Table IV. Specifically, removing logit-level fusion reduces SPL

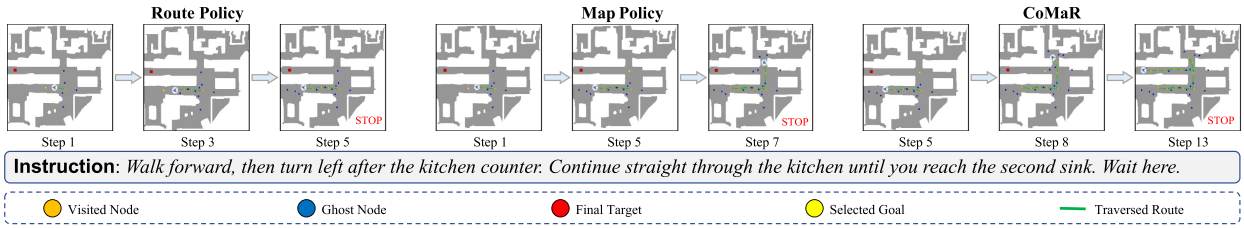

Fig. 3. Qualitative comparison between CoMaR and single-policy variants

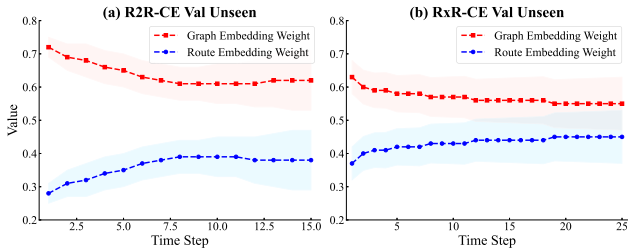

Fig. 4. Qualitative analysis of Feature-Level Collaboration

by 1%. This is because the logit-level fusion acts as a voting mechanism, enhancing the robustness of each decision-making step by combining the logits of the route-based and map-based policies. Removing feature-level collaboration decreases SR by 1% and OSR by 2%. This is because the collaborative policy supports in-depth reasoning based on graph embeddings and route embeddings, improving the agent's navigation capability in complex environments.

D. Qualitative Analysis

To further investigate how CoMaR outperforms the individual route-based and map-based policies, we visualized a typical case on R2R-CE. As shown in Fig. 3, the route-based policy has difficulty grounding region-level landmarks (e.g., kitchen) and struggles to recover from navigation errors. The map-based policy managed to backtrack in the correct direction through spatial reasoning, but it failed to navigate under procedural guidance, leading to navigation failure. CoMaR integrates the advantages of both the map-based and route-based policies. After sufficiently exploring the environment, it reached the target location by following the sequential requirements of the instruction.

To study the relative importance of the map-based and route-based policies at each navigation step, we analyzed the gating weights of the graph and route embeddings on the unseen splits of the R2R-CE and RxR-CE datasets. Specifically, for each timestep, we calculated the mean and standard deviation of these weights across all trajectories in the unseen set. The experimental results are shown in Fig. 4. We find that the gating weight on graph embeddings is consistently higher than that on route embeddings, suggesting that the map-based policy plays a dominant role in the VLN task. Meanwhile, the route weight gradually increases over time, indicating that procedural guidance from the route-based policy becomes more prominent as the trajectory lengthens.

TABLE V
COMPARISON UNDER THE MONOCULAR SETTING ON R2R-CE VAL UNSEEN IN SIMULATION

Method	NE↓	OSR↑	SR↑	SPL↑	FPS↑
Route-Based Policy	5.81	49.4	40.2	33.2	**2.13**
Map-Based Policy (g3D-LF [22])	5.70	59.5	47.2	34.6	2.10
$\text{CoMaR}_{\text{Ours}}$	**5.29**	**60.2**	**49.1**	**38.0**	2.09

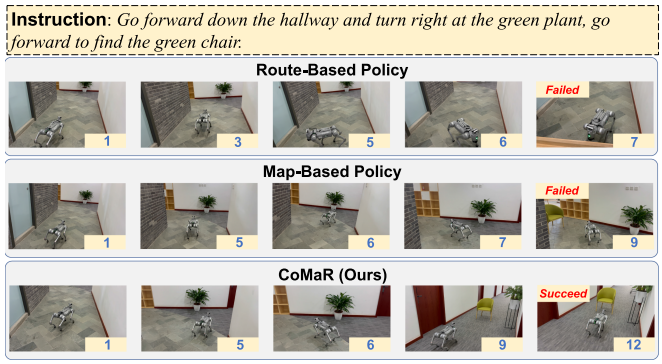

Fig. 5. Real-world qualitative comparison between CoMaR and single-policy variants. The numbers indicate the navigation steps.

E. Real-World Experiment

For real-world deployment, we adopt a monocular setup due to the practical limitations of panoramic sensing and apply CoMaR to the g3D-LF backbone [22]. We first validate this setting on R2R-CE Val Unseen in simulation, comparing CoMaR with single-policy variants. As shown in Table V, CoMaR improves SR by 1.9% and SPL by 3.4% over g3D-LF, while maintaining comparable inference speed (2.09 vs. 2.10 FPS). This suggests that CoMaR remains effective under monocular perception and the additional CoMaR modules introduce minimal extra computation, incurring only negligible system-level overhead.

With the same monocular models validated above, we further compare CoMaR with the map-based and route-based policies in a real-world office-corridor scenario. All methods are executed from the same initial position and follow the same instruction: "*Go forward down the hallway and turn right at the green plant, go forward to find the green chair.*" To increase ambiguity at the junction, we add a visually identical distractor green chair on the left branch. As shown in Fig. 5, CoMaR successfully completes the task, briefly exploring both branches while selecting the correct route. In contrast, the map-based baseline g3D-LF commits to the left branch due to insufficient procedural alignment, whereas the route-based policy fails to reach the junction due

to limited spatial reasoning. Overall, these results highlight the value of jointly leveraging map-based spatial reasoning and route-based procedural alignment in real-world applications.

V. Conclusion

This letter proposes CoMaR, a collaborative policy learning framework for VLN-CE that combines a map-based policy for spatial reasoning with a route-based policy for procedural alignment. CoMaR enables effective collaboration via: (i) spatio-procedural topological mapping, which builds a multiplex graph encoding both environment layout and procedural knowledge; (ii) dual-stream encoding for instruction grounding and route–instruction consistency assessment; and (iii) hierarchical policy integration through feature-level collaboration and logit-level fusion. Extensive comparisons and ablations validate CoMaR's effectiveness and generalization across baselines and datasets, and we further demonstrate its practical benefits through real-world experiments.

References

[1] P. Anderson et al., "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3674–3683.

[2] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the Nav-Graph: Vision-and-language navigation in continuous environments," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 104–120.

[3] D. An et al., "BEVBert: Multimodal map pre-training for language-guided navigation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 2737–2748.

[4] S. Wen, Z. Zhang, Y. Sun, and Z. Wang, "OVL-map: An online visual language map approach for vision-and-language navigation in continuous environments," *IEEE Robot. Autom. Lett.*, vol. 10, no. 4, pp. 3294–3301, Apr. 2025.

[5] D. An et al., "ETPNav: Evolving topological planning for vision-language navigation in continuous environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 7, pp. 5130–5145, Jul. 2025.

[6] Z. Wang et al., "Lookahead exploration with neural radiance representation for continuous vision-language navigation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 13753–13762.

[7] J. Zhang et al., "NaVid: Video-based VLM plans the next step for vision-and-language navigation," in *Proc. Robot.: Sci. Syst.*, 2024, doi: 10.15607/RSS.2024.XX.079.

[8] H. Liu et al., "NaVid-4D: Unleashing spatial intelligence in egocentric RGB-D videos for vision-and-language navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2025, pp. 10607–10615.

[9] F. Chersi and N. Burgess, "The cognitive architecture of spatial navigation: Hippocampal and striatal contributions," *Neuron*, vol. 88, pp. 64–77, 2015.

[10] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 4392–4412.

[11] J. Zhang et al., "Uni-NaVid: A video-based vision-language-action model for unifying embodied navigation tasks," in *Proc. Robot.: Sci. Syst.*, 2025, doi: 10.15607/RSS.2025.XXI.013.

[12] A.-C. Cheng et al., "NaVILA: Legged robot vision-language-action model for navigation," in *Proc. Robot.: Sci. Syst.*, 2025, doi: 10.15607/RSS.2025.XXI.018.

[13] G. Georgakis et al., "Cross-modal map learning for vision and language navigation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15439–15449.

[14] Z. Wang, X. Li, J. Yang, Y. Liu, and S. Jiang, "GridMM: Grid memory map for vision-and-language navigation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 15625–15636.

[15] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "Think global, act local: Dual-scale graph transformer for vision-and-language navigation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16537–16547.

[16] H. Tan and M. Bansal, "LXMERT: Learning cross-modality encoder representations from transformers," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 5100–5111.

[17] J. Li, R. R. Selvaraju, A. D. Gotmare, S. Joty, C. Xiong, and S. C. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 9694–9705.

[18] R. Lu, J. Meng, and W.-S. Zheng, "PRET: Planning with directed fidelity trajectory for vision and language navigation," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 72–88.

[19] M. Z. Irshad, N. C. Mithun, Z. Seymour, H.-P. Chiu, S. Samarasekera, and R. Kumar, "Semantically-aware spatio-temporal reasoning agent for vision-and-language navigation in continuous environments," in *Proc. Int. Conf. Pattern Recognit.*, 2022, pp. 4065–4071.

[20] P. Chen et al., "Weakly-supervised multi-granularity map learning for vision-and-language navigation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 38149–38161.

[21] R. Liu, W. Wang, and Y. Yang, "Vision-language navigation with energy-based policy," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2024, pp. 108208–108230.

[22] Z. Wang and G. H. Lee, "g3D-LF: Generalizable 3D-language feature fields for embodied tasks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2025, pp. 14191–14202.

[23] A. Chang et al., "Matterport3D: Learning from RGB-D data in indoor environments," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 667–676.

[24] M. Savva et al., "Habitat: A platform for embodied AI research," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9338–9346.

[25] P. Anderson et al., "On evaluation of embodied navigation agents," 2018, *arXiv:1807.06757*.

[26] P. Anderson et al., "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3674–3683.

[27] G. I. Magalhaes, V. Jain, A. Ku, E. Ie, and J. Baldridge, "General evaluation for instruction conditioned navigation using dynamic time warping," in *Proc. NeurIPS Visually Grounded Interaction Lang. Workshop*, 2019.

[28] Y. Hong, Z. Wang, Q. Wu, and S. Gould, "Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15418–15428.

[29] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[30] S. Raychaudhuri, S. Wani, S. Patel, U. Jain, and A. Chang, "Language-aligned waypoint (LAW) supervision for vision-and-language navigation in continuous environments," in *Proc. Conf. Assoc. Comput. Linguistics*, 2021, pp. 4018–4028.

[31] J. Chen, B. Lin, X. Liu, L. Ma, X. Liang, and K.-Y. K. Wong, "Affordances-oriented planning using foundation models for continuous vision-language navigation," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 23568–23576.

[32] D. An et al., "1st place solutions for RxR-habitat vision-and-language navigation competition," 2022, *arXiv:2206.11610*.

[33] D. Anggraini, S. Glasauer, and K. Wunderlich, "Neural signatures of reinforcement learning correlate with strategy adoption during spatial navigation," *Sci. Rep.*, vol. 8, 2018, Art. no. 10110.