Dynamic Topology Awareness: Breaking the Granularity Rigidity in Vision-Language Navigation

Jiankun Peng, Jianyuan Guo, Ying Xu, Yue Liu, Jiashuang Yan, Xuanwei Ye, Houhua Li, Xiaoming Wang

***Abstract*—Vision-Language Navigation in Continuous Environments (VLN-CE) presents a core challenge: grounding high-level linguistic instructions into precise, safe, and long-horizon spatial actions. Explicit topological maps have proven to be a vital solution for providing robust spatial memory in such tasks. However, existing topological planning methods suffer from a "Granularity Rigidity" problem. Specifically, these methods typically rely on fixed geometric thresholds to sample nodes, which fails to adapt to varying environmental complexities. This rigidity leads to a critical mismatch: the model tends to over-sample in simple areas, causing computational redundancy, while under-sampling in high-uncertainty regions, increasing collision risks and compromising precision. To address this, we propose DGNav, a framework for Dynamic Topological Navigation, introducing a context-aware mechanism to modulate map density and connectivity on-the-fly. Our approach comprises two core innovations: (1) A Scene-Aware Adaptive Strategy that dynamically modulates graph construction thresholds based on the dispersion of predicted waypoints, enabling "densification on demand" in challenging environments; (2) A Dynamic Graph Transformer that reconstructs graph connectivity by fusing visual, linguistic, and geometric cues into dynamic edge weights, enabling the agent to filter out topological noise and enhancing instruction adherence. Extensive experiments on the R2R-CE and RxR-CE benchmarks demonstrate DGNav exhibits superior navigation performance and strong generalization capabilities. Furthermore, ablation studies confirm that our framework achieves an optimal trade-off between navigation efficiency and safe exploration. The code is available at https://github.com/shannanshouyin/DGNav.**

***Index Terms*—Vision-Language Navigation, Topological Map, Graph Neural Networks**

I. Introduction

VISION-LANGUAGE Navigation (VLN) stands as a pivotal task in embodied AI, requiring agents to interpret natural language instructions and navigate through unseen environments to reach target locations [1]. As the research focus shifts from discrete graphs to continuous environments (VLN-CE) [2], agents face escalated challenges: they must not only perform high-level semantic reasoning but also execute low-level precise control [3], [4]. In such complex tasks, endowing agents with explicit spatial memory and robust planning capabilities is imperative for success.

Current approaches typically fall into two paradigms. End-to-End learning methods directly map raw visual observations and instructions to actions [5], [6], [7]. While effective in short-horizon tasks, these "black-box" models often incur prohibitive computational costs and struggle to guarantee safety due to the lack of explicit geometric constraints [2], [8], [9]. In contrast, Explicit Map-based approaches have emerged as a robust alternative [10], [11], [12], [13]. By constructing a structured topological representation of the environment, these methods decouple high-level planning from low-level control. This modular hierarchy not only provides agents with interpretable spatial memory for long-horizon navigation but also imposes physical constraints necessary for precise obstacle avoidance.

Despite their effectiveness, existing topological planning methods are constrained by a fundamental bottleneck we term "Granularity Rigidity". Most state-of-the-art methods rely on environment-agnostic thresholding rules to structure the agent's spatial memory and environmental perception. Specifically, these approaches predominantly utilize static geometric distances to determine the bias matrix of the Graph Transformer [12], [13], [14], thereby dominating the edge connectivity weights regardless of the actual scene complexity. This reliance on fixed heuristics creates a structural decoupling between the map representation and the actual navigational complexity, manifesting in two critical limitations: First, the physical granularity is non-adaptive to environmental entropy. In structurally simple, low-uncertainty regions (e.g., straight corridors), fixed thresholds often result in node redundancy, introducing unnecessary computational noise. Conversely, in cluttered or open areas with high navigational uncertainty, the resulting graph becomes catastrophically sparse, failing to provide sufficient candidate waypoints for precise local maneuvering. Second, and more critically, the reliance on static geometry leads to "Navigational Myopia" [13], [15], [16], [17]. Because the bias terms in conventional graph transformers encode only static geometric distance information, the model's ability to attend to the semantic attributes of graph nodes is weakened, causing the planner to exhibit locally greedy behaviors. For instance, when commanded to "walk down the hall and take the second left", the agent often assigns excessive attention to the first encountered door purely due to its physical proximity, resulting in premature turns and task failure. In essence, geometric proximity does not inherently imply navigational correctness. Yet, existing

Corresponding author: Ying Xu.

Jiankun Peng, Yue Liu, Jiashuang Yan, Xuanwei Ye, Houhua Li is with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; and is also with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: pengjiankun24@mails.ucas.ac.cn; liuyue212@mails.ucas.ac.cn; yanjiashuang22@mails.ucas.ac.cn; yexuanwei24@mails.ucas.ac.cn; lihouhua24@mails.ucas.ac.cn).

Jianyuan Guo is with the Department of Computer Science, City University of Hong Kong, Hong Kong, SAR, China (e-mail: jianyguo@cityu.edu.hk)

Ying Xu and Xiaoming Wang are with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (e-mail: xuying@aircas.ac.cn; wxm@aircas.ac.cn).

rigid topologies lack the "semantic elasticity" required to dynamically reconstruct graph connectivity in alignment with complex, multi-step linguistic instructions.

To break this rigidity, we propose DGNav (Dynamic Graph Navigation), a unified framework that introduces Dynamic Topology Awareness to enable flexible and context-sensitive navigation. DGNav innovates on two levels. At the physical structure level, we devise a Scene-Aware Adaptive Strategy. By analyzing the dispersion of predicted waypoints, this strategy dynamically adjusts the graph construction threshold (γ), achieving an adaptive granularity that "maintains sparsity in low-uncertainty simple regions to enhance efficiency, while dynamically densifying in high-uncertainty open or complex areas to ensure precise coverage". At the semantic logic level, we propose a Dynamic Graph Transformer. Unlike traditional planners that rely on static edges, our module fuses visual similarity, instruction relevance, and geometric priors to dynamically generate context-aware edge weights. This mechanism allows the agent to functionally "disconnect" nearby geometric noise (e.g., the incorrect first turn) and establish "semantic shortcuts" to distant, task-relevant landmarks, effectively mitigating the myopia problem.

Unlike prior graph-based planners that operate on a fixed topology or static edge priors, DGNav explicitly adapts both graph granularity and connectivity online based on scene uncertainty and instruction semantics. The main contributions of this work are summarized as follows: (1) We identify the "Granularity Rigidity" problem in VLN-CE and propose the DGNav framework. Central to this framework, we introduce a Scene-Aware Adaptive Strategy, which, for the first time, dynamically modulates graph construction thresholds based on waypoint uncertainty, effectively resolving the conflict between node sparsity in complex regions and computational redundancy in open areas. (2) We propose a Multi-modal Dynamic Graph Transformer. Unlike traditional static geometric connectivity, this module reconstructs edge weights by fusing visual semantics, instruction relevance, and geometric constraints, thereby suppressing topological noise and significantly enhancing instruction fidelity in path planning. (3) We conduct extensive experiments on both R2R-CE and RxR-CE benchmarks. Results demonstrate that DGNav establishes a new performance benchmark, exhibiting significant superiority in terms of navigation accuracy and efficiency compared to both end-to-end learning baselines and other explicit map-based approaches. Furthermore, we investigate multiple advanced training strategies to analyze the impact of implicit geometric decoupling on performance upper bounds, providing empirical insights for enhancing navigation robustness in complex scenarios.

II. RELATED WORK

A. VLN in Continuous Environments

Early Vision-Language Navigation research primarily operated in discrete environments where agents navigate between pre-defined nodes [1]. To bridge the gap to real-world robotics, Krantz et al. [2], [18] introduced VLN in Continuous Environments (VLN-CE), requiring agents to execute low-level controls in continuous spaces. Existing approaches generally fall into two categories: end-to-end and modular methods. End-to-end approaches directly map pixel inputs to actions, relying on implicit neural representations for reasoning [10], [19]—a paradigm recently bolstered by Vision-Language Models (VLMs) [5], [6], [7]. However, regardless of the backbone capacity, these methods inherently function as "black boxes" lacking explicit spatial structures. Consequently, they struggle to guarantee navigational safety and long-horizon consistency [20], as the absence of geometric constraints often leads to unstable trajectories and frequent collisions. Conversely, modular approaches [8], [9], [12] decompose the task into waypoint prediction and local planning, offering better interpretability and stability. Our work follows the modular paradigm, aiming to enhance the robustness of the spatial planning module.

B. Topological Mapping for Navigation

Explicit spatial memory is pivotal for long-horizon navigation [21]. Topological maps, which represent environments as graphs of nodes and edges, have become a dominant solution due to their compactness compared to metric maps. Chen et al. [13], [22] explored the application of topological graphs in VLN-CE based on offline pre-constructed prior maps. Building on these foundations, ETPNav [12] advanced the field by proposing an "evolving" strategy that generates candidate nodes online and merges them incrementally. Despite these advancements, ETPNav and similar followers [14], [23] inherit a critical limitation: granularity rigidity. First, regarding graph construction, they typically rely on fixed distance thresholds to merge nodes, ignoring the varying complexity of local scenes. Second, regarding edge connectivity, they predominately utilize static geometric distances (Euclidean distance) to determine edge weights. As argued in Section I, this static geometric prior fails to capture semantic connectivity, often leading the agent to prioritize geometrically close but semantically irrelevant nodes [17], [24], [25].

C. Dynamic Graph Learning in Navigation

Graph Neural Networks (GNN) [26] and Transformers [27] have revolutionized the processing of non-Euclidean data. In the context of deep learning, the Transformer architecture can be viewed as a Graph Attention Network (GAT) [28] operating on a fully connected graph, where attention weights represent dynamic edge strengths. While GNN have been applied to VLN to model environment topology [12], [13], [14], [22], most existing methods treat the graph structure as static or purely geometric, only updating node features. In contrast, our DGNav extends the concept of dynamic graph learning to the topological structure itself. We introduce a mechanism to dynamically generate context-aware edge weights by fusing visual, linguistic, and geometric cues. By doing so, our Dynamic Graph Transformer functions as a learnable reasoning module that filters out topological noise and reconstructs the graph connectivity based on navigation context, rather than relying solely on physical proximity.

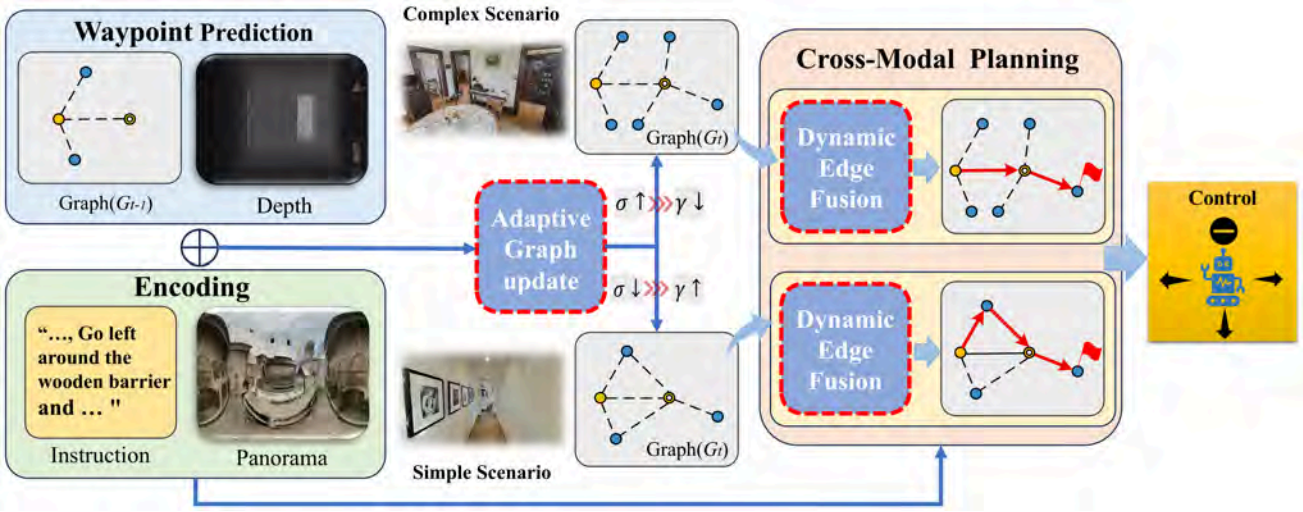

Fig. 1. The overall framework of DGNav. DGNav dynamically adjusts its navigation strategy according to the estimated scene complexity σ. Specifically, higher scene complexity leads to denser topological graph construction, while simpler environments favor sparser representations. The graph merging threshold γ controls the resulting graph granularity and is inversely correlated with σ, enabling adaptive trade-offs between navigation safety and efficiency.

III. METHOD

A. Problem Formulation and Overview

The task of navigation in continuous environment is usually formulated as a sequential decision-making process. We address the VLN task in continuous environments. At step t, the agent receives a visual observation O_t (RGB-D panorama) and a natural language instruction L. The goal is to predict a continuous action $a_t \in A$ (e.g., turning 15°, moving forward 0.25 meters, and stopping) to reach the target. To handle long-horizon navigation, we maintain an online topological map $G_t = \langle v_t, \varepsilon_t \rangle$, where nodes v_t represent navigable waypoints and edges ε_t denote connectivity.

Following prior works in vision-and-language navigation [9], [12], [13], the agent incrementally constructs a topological graph to represent its explored environment. At each step, the agent leverages depth maps to infer navigable candidate locations, which are instantiated as graph nodes representing spatial positions, while panoramic RGB-D observations are used to extract semantic features associated with these nodes. Edges are then established between nodes based on geometric reachability, encoding feasible transitions for continuous navigation. As the agent moves, newly observed locations are added to the graph, while previously visited nodes are retained to provide a persistent spatial memory. This online topological representation enables the planner to reason over both local observations and accumulated global structure during long-horizon navigation.

Building upon this formulation, as illustrated in Fig.1, the proposed DGNav framework operates in a modular pipeline. First, the Waypoint Prediction module takes the depth map at step t and the topological graph from the previous step G_{t-1} as inputs to predict candidate ghost nodes, which serve as potential navigable extensions of the current graph. These candidates are then processed by the Adaptive Graph Update module (our first innovation, detailed in Section III-B). This module analyzes the local scene complexity and adaptively regulates the graph granularity—generating a dense topology for complex scenarios to ensure safety, while maintaining a sparse topology for simple scenarios to enhance efficiency. Simultaneously, an Encoding module extracts feature representations from the RGB-D panorama and the natural language instruction. These encoded features, along with the updated graph G_t, are injected into the Cross-Modal Planning module. Inside this planner, the Dynamic Edge Fusion module (our second innovation, detailed in Section III-C) fuses multimodal cues—spanning static geometry, visual semantics, and natural language instructions—to construct a dynamic adjacency matrix $\mathbf{E}$, which is then consumed by the Cross-Modal Graph Transformer to reason about the optimal path. Finally, the high-level plan is executed by the Control module to output low-level actions.

B. Scene-Aware Adaptive Topology

Waypoint Prediction and Granularity Definition: As illustrated in Fig.2, our pipeline initiates with a Waypoint Prediction module, which perceives the local geometry by processing the depth map at step t and the topological graph from the previous step G_{t-1}. This module generates a set of candidates ghost nodes, denoted as $C_t = \{c_{t,i}\}_{i=1}^{N_c}$. Subsequently, the Adaptive Graph Update module determines the status of these candidates: whether to instantiate them as new landmarks or merge them into the existing graph. This decision is strictly governed by a distance threshold γ. Specifically, a candidate is merged if its Euclidean distance to the nearest neighbor is less than γ; otherwise, the graph is expanded. Thus, γ serves as a critical controller for the granularity of environmental abstraction. A small γ yields a dense topology (high-fidelity but computationally redundant), while a large γ results in a sparse topology (efficient but potentially losing detail). Existing rigid methods typically fix γ, failing to accommodate varying scene requirements.

Metric of Scene Complexity: To break this rigidity, we leverage the spatial distribution of candidates C_t as a proxy for

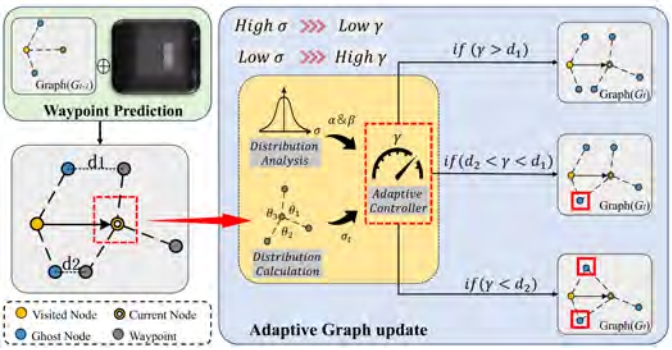

Fig. 2. Illustration of the Scene-Aware Adaptive Strategy. The process begins with Waypoint Prediction, generating raw ghost nodes from the depth map. These nodes are then passed to the Adaptive Graph Update module. Based on the angular dispersion (σ) of the candidates, the controller dynamically adjusts the merging threshold γ. The figure demonstrates three scenarios: in simple environments (low σ), a larger γ yields a sparse graph for efficiency; in complex environments (high σ), a smaller γ results in a dense graph for safety.

local scene complexity. Specifically, given N_c valid candidate nodes predicted at the current step, let θ_i denote the relative angle of the i-th candidate with respect to the agent's heading. We quantify the spatial dispersion σ_t by calculating the standard deviation of these angles:

$$\sigma_t = \sqrt{\frac{1}{N_c} \bullet \sum_{i=1}^{N_c} \left(\theta_i - \bar{\theta}\right)^2}. \tag{1}$$

where $\bar{\theta}$ represents the mean angle of all candidates. As depicted in the three branches of Fig. 2, a high σ_t (e.g., at an intersection where candidates diverge) implies a complex decision boundary, while a low σ_t (e.g., in a corridor where candidates cluster forward) suggests a simple geometric structure. Accordingly, we establish a linear inverse control law to dynamically modulate γ_t:

$$\gamma_t = CLIP\left(\alpha - \beta \bullet \sigma_t,\ \gamma_{min}, \gamma_{max}\right). \tag{2}$$

This formulation ensures that in complex regions (high σ_t), γ_t is decreased to construct a dense graph for safety, whereas in open areas (low σ_t), γ_t is increased to form a sparse graph for efficiency.

Statistical Calibration of Parameters: The coefficients α and β in our control law are not heuristically tuned hyperparameters but are derived through rigorous statistical calibration. Specifically, we utilized the fully converged ETPNav baseline model [12] to perform inference across the Val-Seen split (i.e., a subset of instructions from the R2R-CE training set, used to strictly avoid data leakage from unseen environments), collecting the angular dispersion σ_t of candidate ghost nodes at every navigation step. As illustrated in Fig. 3, we plot the statistical distribution histogram of σ_t. It can be observed that the distribution exhibits a distinct Normal Distribution (Gaussian-like) profile. This indicates that the complexity of most navigation scenarios clusters around the central mean, while extremely open or cluttered scenarios reside in the tails.

Theoretical Justification for Linear Mapping: Based on this Gaussian observation, we theoretically justify the choice

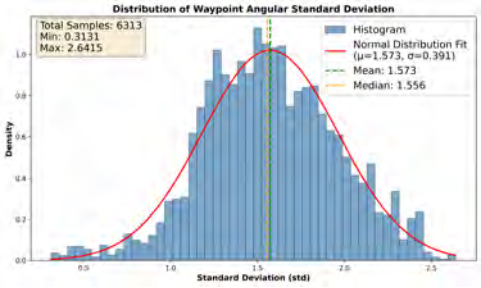

Fig. 3. Statistical distribution of angular dispersion (σ_t) for parameter calibration. Data is collected from the ETPNav baseline on the R2R-CE Val-Seen split. The distribution exhibits a Gaussian-like profile, which serves as the empirical basis for calibrating α and β.

of a linear control law over non-linear alternatives (e.g., Sigmoid or Exponential). From an information-theoretic perspective, a linear transformation f(x)=$\alpha - \beta x$ preserves the maximum entropy property of the Gaussian source distribution P(σ_t) [29], [30]. In contrast, non-linear mappings like Sigmoid introduce saturation regions at the distribution tails where gradients vanish ($\nabla f \rightarrow 0$). This would compress distinct high-uncertainty states (the "cluttered" tail) into indistinguishable threshold values, causing information loss in the most critical scenarios. Therefore, the linear mapping acts as an optimal first-order approximation, maintaining constant sensitivity across the full dynamic range of scene complexity. To empirically validate this analysis, we provide a comparative study of different mapping functions in Section IV-C, demonstrating the superior stability of our linear approach compared to non-linear variants.

Conditional Linear Mapping Strategy: Applying dynamic adjustment across the entire uncertainty range may introduce unnecessary topological fluctuations in stable environments. To address this issue, we adopt a conditional control strategy that activates adaptive granularity only when scene complexity exceeds a critical level. Specifically, the median dispersion σ_{med} is used to delineate a stable regime, within which the merging threshold is fixed at a conservative baseline γ_{fix}. When $\sigma_t > \sigma_{med}$, the threshold is progressively relaxed to allow finer topological refinement, with the adjustment bounded by the maximum dispersion σ_{max}. This design ensures that structural adaptation is triggered only in decision-critical regions while preserving stability in common cases. The specific mapping relationship is as follows:

$$\gamma_t = \begin{cases} \gamma_{fix} & if\ \sigma_t \leq \sigma_{med} \\ \gamma_{fix} - \left(\frac{\sigma_t - \sigma_{med}}{\sigma_{max} - \sigma_{med}}\right)(\gamma_{fix} - \gamma_{min}) & if\ \sigma_t > \sigma_{med} \end{cases} \tag{3}$$

This formulation guarantees continuity at σ_{med} while activating adaptive thresholding only in high-uncertainty regimes. Empirical validation in Section IV-C, comparing global versus conditional linear mappings, confirms that the conditional approach more effectively captures scene complexity to attain higher navigation success rates.

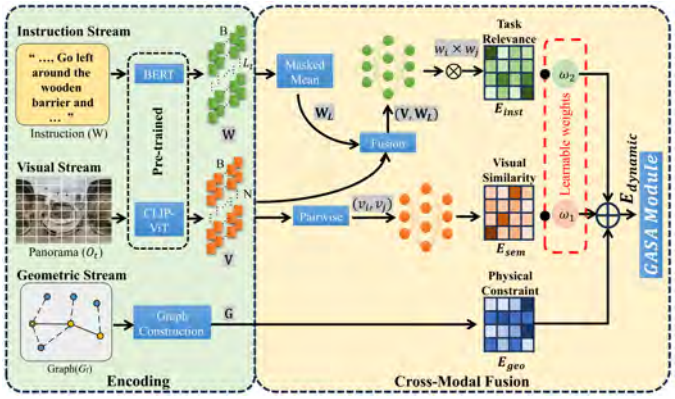

Fig. 4. The architecture of Multimodal Encoding and Dynamic Edge Fusion. The visual encoder and instruction encoder extract node features ($\mathbf{V}$) and word features ($\mathbf{W}$), respectively. The Dynamic Edge Fusion module constructs the graph connectivity by fusing the geometric map ($\mathbf{E_{geo}}$), pairwise visual similarity ($\mathbf{E_{sem}}$), and instruction relevance ($\mathbf{E_{inst}}$ derived from $\mathbf{W_L}$). The resulting matrix $\mathbf{E_{dynamic}}$ guides the Graph Transformer to perform context-aware planning

C. Dynamic Graph Transformer

Feature Encoding: As illustrated in Fig. 4, the system processes multimodal inputs into a unified feature space. For the linguistic stream, we employ a pre-trained BERT [31] to encode instructions on R2R-CE, and RoBERTa [32] for multilingual instruction encoding on RxR-CE, yielding the word feature sequence $\mathbf{W} \in \mathbb{R}^{L\times D}$, where L is the instruction length. We further aggregate $\mathbf{W}$ to obtain the global instruction token $\mathbf{W_L} \in \mathbb{R}^D$ via pooling. For the visual stream, the topological map consists of N navigable nodes. We extract the panoramic features of each node using a pre-trained CLIP-ViT [33], [34], fused with orientation embeddings, to generate the visual node features $\mathbf{V} = \{v_1, ..., v_N\} \in \mathbb{R}^{N\times D}$.

Dynamic Edge Fusion: To enable semantic reasoning, we reconstruct the graph connectivity by fusing three distinct streams into a dynamic adjacency matrix $\mathbf{E_{dynamic}}$.

- *Geometric Stream*: Provides the hard constraint of the physical world. We calculate the normalized Euclidean distance $\mathbf{E_{geo}^{(i,j)}}$, representing basic spatial reachability.
- *Visual Stream*: Introduces semantic consistency to soften physical constraints. To capture semantic continuity, we compute the pairwise similarity between visual nodes v_i and v_j. By passing them through a non-linear mapping, we derive the semantic adjacency matrix:

$$\mathbf{E}_{sem}^{(i,j)} = MLP\left([v_i; v_j]\right). \tag{4}$$

  This term enables the model to connect regions that are visually coherent even if they are not immediate physical neighbors.
- *Instruction Stream*: To align the topology with the task, we compute the relevance between each node v_i and the global instruction $\mathbf{W_L}$. The instruction-guided edge weight is derived via the outer product of relevance scores:

$$w_i = MLP\left([v_i; \mathbf{W_L}]\right). \tag{5}$$

$$\mathbf{E}_{inst}^{(i,j)} = w_i \bullet w_j. \tag{6}$$

  This term effectively filters task-irrelevant topological noise by suppressing paths that do not align with the instruction description.

The final dynamic adjacency matrix $\mathbf{E_{dynamic}}$ is formulated by superimposing learnable semantic residuals onto the fixed physical baseline:

$$\mathbf{E_{dynamic}} = \mathbf{E_{geo}} + \omega_1 \bullet \mathbf{E_{sem}} + \omega_2 \bullet \mathbf{E_{inst}}. \tag{7}$$

Here, ω_1 and ω_2 are learnable scalar coefficients. This architecture allows the model to smoothly transition from pure geometric navigation to semantic-aware planning, preventing instability during the early stages of training.

Graph Transformer with Dynamic Bias: The constructed dynamic adjacency matrix $\mathbf{E_{dynamic}}$ serves as a learnable structural bias for the graph reasoning module. We employ a multi-layer Graph Transformer to update the node features. Let $\hat{\mathbf{H}}^{\mathbf{l}} = \left\{h_1^l, ..., h_N^l\right\}$ denote the node features at layer l. The update process for the $(l+1)$-th layer is formalized as:

$$\hat{\mathbf{H}}^{\mathbf{l+1}} = GASA\left(\mathbf{H^l}, \mathbf{E_{dynamic}}\right) + \mathbf{H^l}. \tag{8}$$

$$\mathbf{H}^{\mathbf{l+1}} = FFN\left(LN\left(\hat{\mathbf{H}}^{\mathbf{l+1}}\right)\right) + \hat{\mathbf{H}}^{\mathbf{l+1}}. \tag{9}$$

where FFN denotes the Feed-Forward Network and LN denotes Layer Normalization. Specifically, the Graph-Aware Self-Attention (GASA) mechanism [12], [13], [26], [28] incorporates the dynamic topology directly into the attention calculation. For a single attention head, the output is computed as:

$$\mathrm{GASA}\left(\mathbf{H}^l, \mathbf{E}_{\mathrm{dynamic}}\right) = \mathrm{Softmax}\left(\frac{\mathbf{H}^l\mathbf{W}_Q\left(\mathbf{H}^l\mathbf{W}_K\right)^{\mathsf{T}}}{\sqrt{d_k}} + \mathbf{E}_{\mathrm{dynamic}}\right)\left(\mathbf{H}^l\mathbf{W}_V\right). \tag{10}$$

Here, $\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_v}$ are learnable projection matrices. By replacing the static geometric mask with our $\mathbf{E_{dynamic}}$, the attention mechanism is forced to attend to nodes that are semantically relevant ($\omega_1 \bullet \mathbf{E_{sem}}$) and task-aligned ($\omega_2 \bullet \mathbf{E_{inst}}$), effectively creating "soft" semantic edges that bypass the rigid constraints of the physical graph

IV. EXPERIMENTS

A. Experimental Setup

Datasets: We evaluate our method on two standard Vision-Language Navigation datasets in continuous environments (VLN-CE), i.e., R2R-CE and RxR-CE,, both built upon the Matterport3D (MP3D) [1] simulation platform. MP3D serves as the visual backbone, providing 90 diverse indoor building environments with high-fidelity RGB-D scans. Unlike discrete navigation settings, the VLN-CE [2] benchmark allows agents to move freely in 3D space without being constrained to pre-defined graph nodes, imposing stricter requirements on spatial awareness and obstacle avoidance. (1) R2R-CE: This dataset is a continuous reconstruction of the Room-to-Room (R2R) benchmark. It contains 5611 trajectories across 90 scenes, paired with human-annotated English instructions. The dataset

TABLE I
END-TO-END VS. EXPLICIT MAP-BASED APPROACHES

Methods		Val-Seen					Val-Unseen					Test-Unseen				
		TL↓	NE↓	OSR↑	SR↑	SPL↑	TL↓	NE↓	OSR↑	SR↑	SPL↑	TL↓	NE↓	OSR↑	SR↑	SPL↑
End-to-End	Seq2Seq [35]	9.26	7.12	46	37	35	8.64	7.37	40	32	30	8.85	7.91	36	28	25
	Sim2Sim [11]	11.18	4.67	61	52	44	10.69	6.07	52	43	36	11.43	6.17	52	44	37
	CWP-CMA [10]	11.47	5.20	61	51	45	10.90	6.20	52	41	36	11.85	6.30	49	38	33
	CWP-RecBert [10]	12.50	5.02	59	50	44	12.23	5.74	53	44	39	13.51	5.89	51	42	36
	Navid [5]	-	-	-	-	-	7.63	5.47	49	37	36	-	-	-	-	-
	Uni-Navid [6]	-	-	-	-	-	9.71	5.58	53	47	43	-	-	-	-	-
	StreamVLN [7]	-	-	-	-	-	-	5.43	63	53	47	-	-	-	-	-
Explicit Map-based	CM2 [16]	12.05	6.10	50.7	42.9	34.8	11.54	7.02	41.5	34.3	27.6	13.9	7.7	39	31	24
	WG-MGMap [36]	10.12	5.65	52	47	43	10.00	6.28	48	39	34	12.30	7.11	45	35	28
	DUET-CE [13]	-	-	-	-	-	13.08	5.16	62	54	46	-	-	-	-	-
	GridMM [37]	12.69	4.21	69	59	51	13.36	5.11	61	49	41	13.31	5.54	56	46	39
	Safe-VLN [14]	13.71	3.35	79	71	60	15.00	4.48	68	60	47	15.44	5.01	54	56	45
	OVL-MAP [17]	11.82	3.90	73	68	59	11.45	4.69	65	58	50	11.64	4.98	62	57	48
	ETPNav [12]	11.78	3.95	72	66	59	11.99	4.71	65	57	49	12.87	5.12	63	55	48
	Ours (DGNav)	10.79	3.78	72	66	**60**	11.64	4.66	65	59	**50**	14.20	**4.92**	**64**	56	47

is split into Training, Val-Seen, Val-Unseen, and Test sets. The Val-Unseen and Test splits, which involve environments not encountered during training, serve as the primary benchmarks for evaluating the generalization capability of our method. (2) RxR-CE: To further verify the robustness of DGNav in complex, long-horizon tasks, we also conduct experiments on the Room-across-Room (RxR) continuous benchmark [38]. Compared to R2R-CE, RxR-CE features significantly longer path trajectories, more fine-grained instruction descriptions, and a larger scale of instructions (over $10\times$ that of R2R), covering multiple languages (English, Hindi, and Telugu). This dataset presents a more challenging testbed, demanding superior topological memory capabilities over extended time horizons.

Evaluation Metrics: To comprehensively assess navigation performance, we adopt the standard evaluation protocols for VLN-CE [39], [40]. We report the following metrics: (1) Basic Navigation Metrics: Trajectory Length (TL), Navigation Error (NE), Success Rate (SR), and Oracle Success Rate (OSR). (2) Efficiency and Fidelity Metrics: Success weighted by Path Length (SPL), which balances success with path efficiency, along with Normalized Dynamic Time Warping (nDTW) and Spatial Dynamic Time Warping (SDTW), which evaluate the spatiotemporal alignment between the predicted and ground-truth trajectories.

Following community conventions [41], [42], for the R2R-CE benchmark, we prioritize SR and SPL as the primary indicators of goal-reaching capability. For the RxR-CE benchmark, given its strict instruction-following requirements, nDTW and SDTW are considered the most critical metrics for measuring the agent's fidelity to the described path.

Implementation Details:

1)*Training and Optimization*: We implemented our model using PyTorch and conducted all experiments in the Habitat simulator [43] on two NVIDIA RTX 4090 GPUs. Following the established training protocol of ETPNav [12], we adopt a two-stage training strategy. First, we initialize the ViT [34] and BERT [31], [32] using weights pre-trained on large-scale datasets. Second, we fine-tune the entire model on the R2R-CE and RxR-CE datasets. For R2R-CE, we employ the AdamW optimizer with a learning rate of 1e-5, a batch size of 8 (4 per GPU), and train for 20,000 iterations. For RxR-CE, we train for 25,000 iterations with the same learning rate of 1e-5.

2)*Dynamic Graph Parameters*: For the dynamic graph module, the weight of the geometric stream is fixed at $\omega_1 = 1.0$. The learnable semantic weights, ω_2 and ω_3, are initialized to 0.1, and their specific learning rate is set to 1e-4 to facilitate rapid convergence from scratch.

3)*Adaptive Strategy*: To ensure the stability of topological learning, we keep γ fixed at the baseline value (0.5m) [12] during the training phase. The Scene-Aware Adaptive Strategy is activated exclusively during the inference phase. At this stage, the merging threshold γ is dynamically modulated between 0.25m and 0.5m based on the real-time σ_t. This decoupled strategy ensures stable feature representation learning while endowing the agent with the flexibility to handle extreme scenarios in unseen test environments. This adaptive mechanism does not rely on any additional supervision or privileged information during inference.

B. Comparison with Other Methods

Table I presents the performance comparison between DGNav and a series of classic navigation baselines. To provide a clear contextual analysis, we categorize baselines into two distinct paradigms: End-to-End Learning Methods (ranging from classic sequential models to recent VLM-based agents) and Explicit Map-based Methods (including topological, grid-based, Semantic planners).

Comparison with End-to-End Learning Methods: We first compare DGNav with some classic end-to-end baselines. As shown in Table I, these methods, which rely on implicit memory states, generally suffer from poor long-horizon navigability. DGNav achieves substantial superiority over these baselines, demonstrating the decisive advantage of our hierarchical planning framework that decouples perception from control. Furthermore, compared to recent Large Vision-Language Models (VLMs) based approaches such as NaVid

TABLE II
COMPARISON WITH CLASSIC BASELINES ON RXR-CE

Methods	Val-Seen					Val-Unseen				
	NE↓	SR↑	SPL↑	nDTW↑	SDTW↑	NE↓	SR↑	SPL↑	nDTW↑	SDTW↑
Seq2Seq [35]	-	-	-	-	-	12.10	13.9	11.9	30.8	-
CWP-CMA [10]	-	-	-	-	-	8.76	26.59	22.16	47.05	-
CWP-RecBert [10]	-	-	-	-	-	8.98	27.08	22.65	46.71	-
StreamVLN [7]	-	-	-	-	-	6.72	48.6	42.5	60.2	-
ETPNav [12]	5.55	57.26	47.67	64.15	47.57	5.80	53.07	44.16	61.49	43.92
Ours (DGNav)	**5.43**	**58.48**	**48.54**	**65.49**	**48.60**	6.00	**53.78**	**44.37**	**62.04**	**44.49**

[5] and Uni-NaVid [6], DGNav maintains a strong competitive edge.

While these VLM-based agents achieve breakthroughs in zero-shot reasoning, their lack of explicit geometric constraints often leads to navigational instability in continuous environments. DGNav significantly outperforms Uni-NaVid by a large margin, confirming that despite the rise of large models, explicit spatial memory remains a crucial component for ensuring precise and robust navigation.

Comparison with Explicit Map-based Methods: Within the map-based category, DGNav establishes a new benchmark for topological planning. Versus Classic Map Baselines: Compared to early metric map approaches, DGNav exhibits substantial performance gains across all metrics, validating the efficacy of our dynamic graph construction over static mapping strategies.

When benchmarked against recent advanced baselines, DGNav demonstrates distinct advantages arising from its adaptive mechanism. Specifically, in comparison with GridMM [37], which employs fine-grained grid maps, DGNav achieves significantly lower NE, suggesting that our adaptive topology offers greater flexibility for precise localization than rigid grid representations. Furthermore, while DGNav trails marginally behind Safe-VLN [14] in terms of SR and NE, it achieves a significantly shorter TL and substantially outperforms Safe-VLN in SPL on the Val-unseen split. This trade-off indicates that DGNav exhibits more intelligent behavior: rather than adopting overly conservative strategies that prolong paths, our method ensures high success rates while maintaining superior efficiency, reflecting a deeper understanding of both environmental scenes and linguistic instructions. Finally, even against OVL-Map [17], a formidable competitor incorporating object-level semantic mapping, DGNav maintains a crucial edge in SR (+1%) and NE (-0.03m). This finding highlights that static accumulation of semantic labels is insufficient; instead, DGNav's "Dynamic Edge Fusion" mechanism intelligently filters and leverages semantic cues based on instruction requirements, thereby enabling optimal navigation decisions in complex, unseen environments.

Finally, compared to our direct baseline ETPNav, DGNav achieves comprehensive improvements across NE, OSR, SR, and SPL on the R2R-CE Val-unseen split. On the more challenging Test-unseen split, while SPL shows a slight decrease (-1%), DGNav demonstrates enhanced robustness with improvements in NE (-0.2m), OSR (+1%), and SR (+1%). This specific trade-off reflects DGNav's superior robustness: when facing highly unfamiliar and complex test environments, the adaptive graph mechanism encourages the agent to adopt a more cautious exploration strategy (e.g., moving slightly more to verify landmarks in ambiguous regions) to ensure precise localization and successful target convergence, rather than making hasty turns for shorter paths that risk task failure. The official results on the Test-unseen split are publicly available on the R2R-CE leaderboard.

Results on RxR-CE Benchmark (Table II): To verify the generalization capability of DGNav in handling complex, long-horizon instructions, we further conducted evaluations on the RxR-CE dataset. Compared to R2R-CE, RxR-CE features significantly longer trajectories (avg. length increased by $\approx$ 50%) and more fine-grained instruction descriptions, imposing rigorous demands on the agent's spatiotemporal memory. Table II presents the comparison results against classic methods (Seq2Seq [35], CWP [10]) and recent baselines (ETPNav [12], StreamVLN [7]).

Results indicate that DGNav achieves a comprehensive superiority over the ETPNav baseline on the Val-Seen split. Specifically, SR is improved by +1.22%, and SPL by +0.87%. More notably, on metrics prioritizing path fidelity—crucial for the RxR task—nDTW and SDTW are boosted by +1.34% and +1.03%, respectively. This demonstrates that in familiar environments, DGNav's dynamic semantic graph can precisely align with the subtle spatial cues described in the instructions.

On the critical Val-Unseen split, DGNav maintains this advantage. Despite the extreme difficulty of this dataset, we achieve consistent improvements in SR and SPL. Particularly on the core metrics measuring "instruction following fidelity", DGNav again surpasses ETPNav, with nDTW reaching 62.04% (+0.55%) and SDTW hitting 44.49% (+0.57%). While there is a marginal increase in NE (5.80m to 6.00m), this is not contradictory to the improvement in SR (53.07% to 53.78%). This phenomenon typically stems from DGNav performing more local adjustments near the goal area (e.g., fine-tuning based on detailed descriptions), which, while resulting in a slight deviation from the absolute geometric center in Euclidean terms, leads to a more accurate satisfaction of the task completion criteria. Overall, the RxR-CE results compellingly prove that DGNav excels not only at "reaching the destination" but also at "navigating in the prescribed manner", a direct manifestation of the deep semantic understanding enabled by our dynamic graph approach.

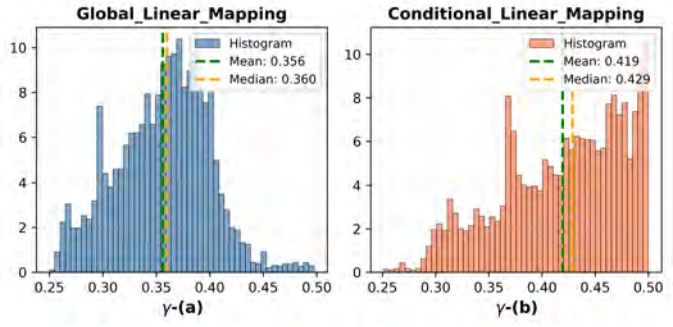

Fig. 5. Distribution of Dynamic Thresholds (γ_t) under Different Mapping Strategies on R2R-CE Val-Unseen. (a) Global Linear Mapping exhibits a dispersed distribution, indicating frequent but potentially noisy adjustments. (b) Conditional Linear Mapping shows a long-tailed distribution, reflecting a selective intervention strategy. Note: To clearly visualize the distribution of dynamic adjustments, the dominant peak at the baseline ($\gamma = 0.5$) is omitted; data at the lower bound ($\gamma = 0.25$) is also excluded to maintain visual symmetry.

C. Ablation Study

1) *Analysis of Adaptive Mapping Strategies*

To investigate how to optimally map the predicted scene uncertainty (σ_t) to the graph construction threshold γ_t, we compare two distinct linear mapping strategies:

- Global Linear Mapping: Directly maps the full range of σ_t to the γ interval.
- Conditional Linear Mapping: Introduces a median value σ_{med} as a trigger gate. When $\sigma_t < \sigma_{med}$ (simple scenes), γ_t is held at the baseline of 0.5 to maintain stability; the linear mapping to reduce the threshold is triggered only when $\sigma_t > \sigma_{med}$ (complex scenes).

Impact of Triggering Strategy (Global vs. Conditional): Fig. 5 visualizes the histogram of γ_t values on the Val-Unseen split for both strategies (excluding the boundary values of 0.25 and 0.5 for visualization clarity). It can be observed that the Global Strategy yields a dispersed distribution of γ, implying that the model frequently micro-adjusts the threshold even in relatively simple environments, which may introduce unnecessary topological noise. In contrast, the Conditional Strategy exhibits a distinct long-tailed distribution. This pattern indicates that the strategy successfully achieves "Selective Intervention": the vast majority of simple steps are maintained at the optimal baseline $\gamma_{base} = 0.5$ (the dominant peak not shown in the plot), while dynamic adjustments are precisely confined to the tail-end complex scenarios where uncertainty is high.

The quantitative results in Table III further validate the superiority of the Conditional Strategy. Compared to the Global Strategy, the Conditional Strategy achieves a higher SR (58.56%) and Path SPL (50.08%). This suggests that establishing a "stability gate" to prevent over-sensitivity in simple scenes, while concentrating resources on addressing sparsity in complex regions, is key to achieving robust navigation. Thus, we adopt the Conditional strategy as our baseline.

Choice of Mapping Function: We validated the superiority of linear mapping against non-linear variants (Sigmoid and Exponential) within the Conditional framework. As illustrated in Fig. 6 and Fig. 7, the Sigmoid function performs the worst; its S-shaped curve suffers from gradient saturation, leading to an overly aggressive distribution ($\gamma \approx 0.25$) that generates excessive topological noise. Conversely, the Exponential function exhibits a "Conservative Bias" due to its convex nature, causing "Late Activation" that fails to provide sufficient granularity for moderately complex scenes. In contrast, Linear Mapping achieves the best performance (Table III) by maintaining Constant Sensitivity across the dynamic range. It effectively preserves the entropy of the uncertainty signal, striking an optimal equilibrium that avoids both the "over-reaction" of Sigmoid and the "under-reaction" of Exponential, ensuring precise alignment between topological granularity and environmental complexity.

TABLE III
ABLATION STUDY ON ADAPTIVE MAPPING STRATEGIES

Model	Mapping Strategy	Mapping Function	Val-unseen				
			N_{node}	NE↓	OSR↑	SR↑	SPL↑
DGNav	Global	Linear	27.50	4.72	63.30	57.10	49.43
	Condi.	**Linear**	**23.88**	**4.66**	**64.82**	**58.56**	**50.08**
	Condi.	Sig.	28.29	4.75	63.24	57.10	49.19
	Condi.	Exp.	24.73	4.73	64.55	58.02	50.03

2) *Effectiveness of Dynamic Graph Construction*

To verify the necessity of dynamically adjusting the threshold γ, we compared our proposed dynamic scheme against three static threshold settings ($\gamma \in \{0.25,\ 0.40,\ 0.50\}$) and a random threshold strategy. Table IV presents the detailed comparisons on the R2R-CE and RxR-CE validation splits, where N_{node} denotes the average number of generated nodes, reflecting graph density and computational overhead.

Blind Density vs. Intelligent Adaptation: Results indicate that simply lowering the threshold significantly increases graph density but leads to a degradation in navigation performance. This suggests that blindly adding redundant nodes in simple areas not only wastes computational resources but also introduces topological noise that distracts the agent, thereby increasing the difficulty of model learning. Similarly, the Random strategy fails to outperform the baseline, further confirming that the performance gain stems not from random perturbations, but from the precise capture of scene complexity by σ_t.

In contrast, our Dynamic strategy achieves the best performance across both datasets. Specifically, on R2R-CE, the Dynamic strategy leads in all metrics (NE, OSR, SR, SPL). Notably, this improvement is achieved with minimal additional structural complexity. Since the computational load of the Graph Transformer scales with the number of nodes, the marginal increase in average node count (N_{node} increases by only ≈ 0.4) implies that DGNav incurs negligible runtime overhead compared to the baseline, while yielding significant performance gains. On RxR-CE, this advantage is even more pronounced. The Dynamic strategy not only tops the SR and SPL metrics but also significantly outperforms static and random schemes in nDTW and SDTW, which measure path fidelity. This confirms that the dynamic topology better accommodates the reliance on fine-grained spatial landmarks

TABLE IV
EFFECTIVENESS OF DYNAMIC γ STRATEGY

Model	γ	R2R-CE(Val-unseen)						RxR-CE(Val-unseen)					
		N_{node}	TL↓	NE↓	OSR↑	SR↑	SPL↑	N_{node}	NE↓	SR↑	SPL↑	nDTW↑	SDTW↑
DGNav	0.25	31.61	11.56	4.76	62.43	56.39	49.23	46.67	6.07	53.07	44.08	61.55	43.98
	0.40	25.90	11.46	4.75	63.19	56.66	49.03	37.40	6.02	53.35	44.07	61.53	44.21
	0.50	23.14	11.58	4.69	64.82	58.02	49.71	32.62	6.01	53.43	44.11	61.87	44.25
	Random	26.97	11.42	4.80	62.43	56.23	48.88	39.16	6.04	53.04	43.84	60.83	44.00
	Dynamic	23.88	11.64	**4.66**	**64.82**	**58.56**	**50.08**	33.98	**6.00**	**53.78**	**44.37**	**62.04**	**44.49**

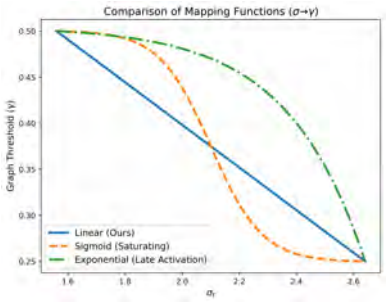

Fig. 6. Statistical distribution of dynamic thresholds (γ_t) under non-linear mapping functions on the Val-Unseen split. (a) The Sigmoid mapping exhibits a polarized distribution heavily concentrated near the lower bound ($\gamma \approx 0.25$), leading to excessive node generation. (b) The Exponential mapping shows a skewed distribution towards the conservative baseline ($\gamma \to 0.5$), indicating a "late activation" response to scene complexity.

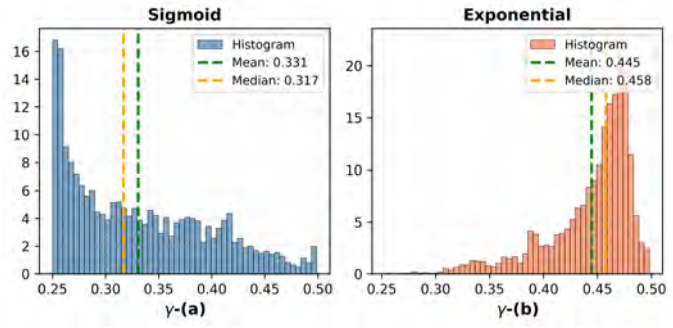

Fig. 7. Comparison of mapping function shapes ($\sigma_t \to \gamma_t$). The Linear function provides constant sensitivity, whereas Sigmoid suffers from saturation and Exponential exhibits late activation.

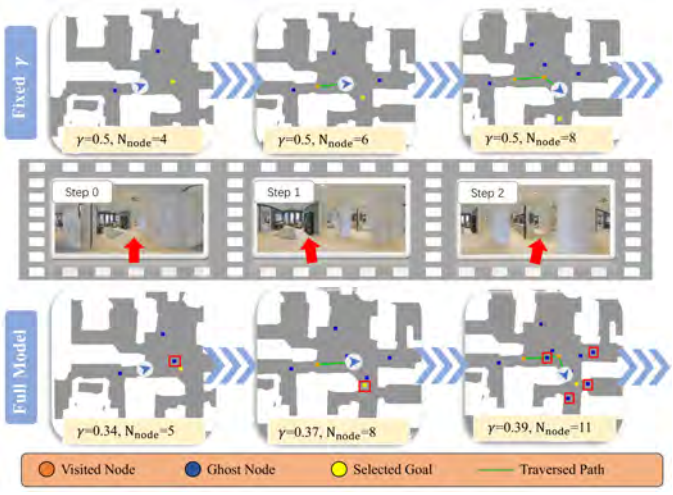

Fig. 8. Visualization of Scene-Aware Adaptive Graph Construction. Comparison of graph generation in a complex environment within the same episode. (Top) Fixed γ: Constrained by a static threshold, the baseline generates a sparse set of ghost nodes. (Bottom) DGNav: The Scene-Aware Adaptive Strategy detects high local uncertainty and dynamically lowers the threshold, triggering a dense generation of Ghost Nodes (N_{node} ↑).

TABLE V
ABLATION STUDY ON DYNAMIC EDGE FUSION MODULES

Model	E_{sem}	E_{inst}	Val-unseen					
			Steps	TL↓	NE↓	OSR↑	SR↑	SPL↑
DGNav	-	-	80.04	11.44	4.76	63.62	56.82	48.74
	√	-	71.00	10.38	4.80	61.94	55.85	49.52
	-	√	77.38	11.05	4.88	61.77	56.01	48.71
	√	√	80.36	11.64	**4.66**	**64.82**	**58.56**	**50.08**

in long-horizon navigation.

To provide a more intuitive validation of this mechanism, we visualize the graph generation process across the first three navigation steps of the same episode in Fig. 8. From the top-down view, it is clearly observable that at the initial pose (Step 0), facing multiple navigable directions, the baseline model is constrained by a fixed threshold and generates only 3 ghost nodes. In contrast, our dynamic strategy perceives the environmental complexity and actively lowers the γ value, generating 4 ghost nodes to cover more potential paths. This advantage persists through the second and third steps; notably, in the complex multi-junction scenario at Step 2, the dynamic strategy generates 4 additional ghost nodes compared to the fixed baseline. This significant node increment strongly confirms that DGNav effectively aligns topological granularity with the immediate environmental complexity through "densification on demand".

3) *Impact of Multi-modal Dynamic Edge Fusion*

Next, we investigate the contribution of different modal edge weights in the dynamic graph. To evaluate the specific impacts of $\mathbf{E_{sem}}$ and $\mathbf{E_{inst}}$ on graph connectivity, we conducted component ablation studies on the R2R-CE Val-Unseen dataset. Considering the computational cost of re-training, this part of the ablation is primarily performed on R2R-CE. All experiments were conducted with the Adaptive γ strategy enabled.

As shown in Table V, relying solely on geometric distance as a baseline limits the model's semantic understanding of the environment. Interestingly, when introducing $\mathbf{E_{sem}}$ alone, we observe a significant drop in average Steps and TL and an

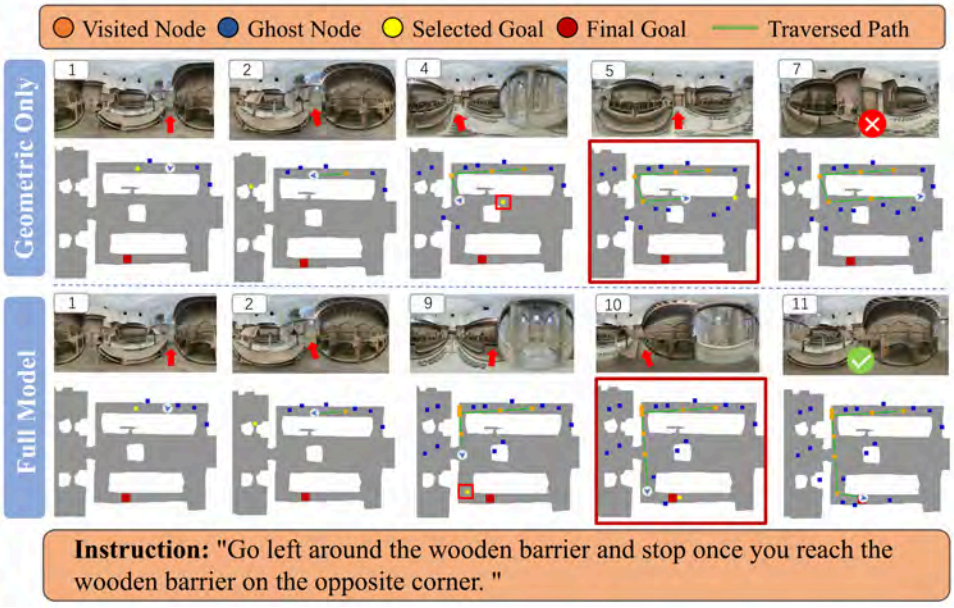

Fig. 9. Qualitative Analysis: Static Geometric Priors vs. Dynamic Semantic Weights. (Top) Geometric Only: Lacking semantic filtering, the agent is misled by static geometric priors and selects the incorrect turn solely due to its proximity. (Bottom) DGNav: Empowered by Multi-modal Dynamic Edge Fusion, the agent effectively integrates visual and linguistic cues to suppress the weight of the incorrect path. It correctly identifies the instructional intent, ignores the geometric distractor, and boldly explores the open space to successfully reach the target.

increase in SPL, but a slight decline in SR. This suggests that pure visual similarity connections make the agent "aggressive and efficient", prone to taking visual shortcuts; however, without instruction constraints, these shortcuts sometimes lead to erroneous navigation decisions. In contrast, the Full Model (DGNav), which fuses both $\mathbf{E_{sem}}$ and $\mathbf{E_{inst}}$, achieves the best performance, with SR jumping to 58.56% and SPL reaching 50.08%. This demonstrates the synergy between the two modalities: $\mathbf{E_{sem}}$ identifies potential semantic shortcuts to enhance efficiency, while $\mathbf{E_{inst}}$ acts as a "task-alignment filter" to verify that these shortcuts comply with the instruction descriptions. The combination allows the agent to optimize path planning while maintaining a high success rate.

To further qualitatively validate the effectiveness of the Multi-modal Dynamic Edge Fusion module, we present a complete navigation episode in Fig. 9. In this scenario, the agent faces two consecutive left-turn intersections, while the instruction explicitly requires it to "reach the wooden barrier" before turning. Lacking semantic perception, the Geometric-Only baseline is misled by static geometric priors: it executes the turn prematurely upon merely "seeing" the barrier rather than truly "reaching" it, causing a complete deviation from the target path. This reveals a fundamental limitation of geometry-dominated planners. In contrast, DGNav, leveraging dynamic edge weights, effectively fuses visual and linguistic cues to suppress the connectivity of the incorrect turn. In this large-scale environment, it overcomes the geometric bias and makes a decision that aligns better with the instructional semantics, boldly choosing a farther but correct exploration path to successfully reach the destination. This case study intuitively demonstrates the critical role of dynamic edge fusion in correcting geometric biases and enhancing instruction adherence.

4) *Exploration of Advanced Training Strategies*

To further enhance semantic dependency, we investigated three training variants: (1) *Node Gating*: A learnable MLP head to filter nodes based on predicted topological importance; (2) *Geometric Dropout*: Randomly masking the geometric stream to prevent overfitting to explicit distances; (3) *Geometric Annealing*: A curriculum strategy where dropout linearly increases to stabilize the transition from geometric to semantic reliance.

Analysis of Results (as shown in Table VI):

- *Redundancy of Node Gating*: Introducing gating causes a slight drop in SPL. This suggests that our core Adaptive Threshold (γ) is already sufficient for density control, and additional filtering likely removes necessary stepping-stone nodes, causing detours.
- *Enhanced Exploration via Geometric Dropout*: Enabling dropout significantly increases TL and OSR. This indicates that reducing geometric priors "liberates" exploration capabilities, encouraging the agent to venture into unknown areas, albeit at the cost of path efficiency.
- *Performance Upper Bound via Geometric Annealing*: The annealing strategy achieves the highest SR and lowest NE. This confirms that balancing "geometry-guided convergence" and "semantics-driven localization" is crucial. Although its SPL is lower than standard DGNav, this strategy effectively curbs risky geometric shortcuts, steering the agent towards conservative but accurate paths—a promising direction for safety-critical scenarios.

Considering the balance between computational efficiency and navigation performance, we retained the standard DGNav for our main experiments. However, these explorations compellingly demonstrate that further decoupling geometric dependency through implicit training strategies is an effective pathway to enhancing the robustness of VLN agents.

TABLE VI
ANALYSIS OF ADVANCED TRAINING STRATEGIES

Method	Gate	Drop	Ann	Val-unseen				
				TL	NE↓	OSR↑	SR↑	SPL↑
DGNav	-	-	-	11.64	4.70	64.82	58.56	**50.08**
	√	-	-	12.35	4.77	64.82	58.40	49.28
	√	√	-	14.08	4.71	66.83	58.56	47.21
	√	√	√	13.55	**4.60**	**67.43**	**59.11**	48.16

V. CONCLUSION

In this paper, we address the "Granularity Rigidity" dilemma inherent in existing topological planning methods for Vision-Language Navigation in Continuous Environments (VLN-CE). We propose DGNav, a framework that breaks the constraints of fixed construction thresholds and sole geometric metrics by introducing a dynamic topological perception mechanism. Furthermore, the Dynamic Graph Transformer reconstructs graph connectivity by fusing visual semantics, instruction relevance, and geometric constraints, effectively suppressing topological noise and enhancing the fidelity of path planning. Extensive experiments on R2R-CE and RxR-CE benchmarks demonstrate that DGNav achieves strong performance against strong baselines, exhibiting particular strengths in path efficiency and robust spatiotemporal alignment.

REFERENCES

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3674–3683, 2018.

[2] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," in *European Conference on Computer Vision*, pp. 104–120, Springer, 2020.

[3] Y. Xu, Y. Liu, M. Lei, M. Gao, Z. Fang, and C. Jiang, "Joint pseudo-range and doppler positioning method with leo satellites 'signals of opportunity," *Satellite Navigation*, vol. 6, no. 1, p. 10, 2025.

[4] Y. Liu, Y. Xu, M. Lei, M. Gao, Z. Fang, and Y. Mao, "A joint high-precision frequency and code phase estimation algorithm for iridium next satellites' signals of opportunity," *IEEE Transactions on Instrumentation and Measurement*, 2025.

[5] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, "Navid: Video-based vlm plans the next step for vision-and-language navigation," *arXiv preprint arXiv:2402.15852*, 2024.

[6] J. Zhang, K. Wang, S. Wang, M. Li, H. Liu, S. Wei, Z. Wang, Z. Zhang, and H. Wang, "Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks," *arXiv preprint arXiv:2412.06224*, 2024.

[7] M. Wei, C. Wan, X. Yu, T. Wang, Y. Yang, X. Mao, C. Zhu, W. Cai, H. Wang, Y. Chen, *et al.*, "Streamvln: Streaming vision-and-language navigation via slowfast context modeling," *arXiv preprint arXiv:2507.05240*, 2025.

[8] 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 *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 4018–4028, 2021.

[9] J. Krantz, A. Gokaslan, D. Batra, S. Lee, and O. Maksymets, "Waypoint models for instruction-guided navigation in continuous environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15162–15171, 2021.

[10] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15439–15449, 2022.

[11] J. Krantz and S. Lee, "Sim-2-sim transfer for vision-and-language navigation in continuous environments," in *European conference on computer vision*, pp. 588–603, Springer, 2022.

[12] D. An, H. Wang, W. Wang, Z. Wang, Y. Huang, K. He, and L. Wang, "Etpnav: Evolving topological planning for vision-language navigation in continuous environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[13] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16537–16547, 2022.

[14] L. Yue, D. Zhou, L. Xie, F. Zhang, Y. Yan, and E. Yin, "Safe-vln: Collision avoidance for vision-and-language navigation of autonomous robots operating in continuous environments," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 4918–4925, 2024.

[15] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 538–547, 2019.

[16] G. Georgakis, K. Schmeckpeper, K. Wanchoo, S. Dan, E. Miltsakaki, D. Roth, and K. Daniilidis, "Cross-modal map learning for vision and language navigation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15460–15470, 2022.

[17] 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 Robotics and Automation Letters*, 2025.

[18] J. Gu, E. Stefani, Q. Wu, J. Thomason, and X. Wang, "Vision-and-language navigation: A survey of tasks, methods, and future directions," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7606–7623, 2022.

[19] S. Chen, P.-L. Guhur, C. Schmid, and I. Laptev, "History aware multimodal transformer for vision-and-language navigation," *Advances in neural information processing systems*, vol. 34, pp. 5834–5847, 2021.

[20] J. Yan, Z. Fang, R. Sun, M. Gao, Y. Mao, C. Jiang, and Y. Xu, "Araim protection level optimization based on feedback-structure subset grouping," *IEEE Sensors Journal*, 2025.

[21] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2017.

[22] K. Chen, J. K. Chen, J. Chuang, M. Vázquez, and S. Savarese, "Topological planning with transformers for vision-and-language navigation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11271–11281, 2021.

[23] R. Liu, W. Wang, and Y. Yang, "Vision-language navigation with energy-based policy," *Advances in Neural Information Processing Systems*, vol. 37, pp. 108208–108230, 2024.

[24] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological slam for visual navigation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12875–12884, 2020.

[25] O. Kwon, N. Kim, Y. Choi, H. Yoo, J. Park, and S. Oh, "Visual graph memory with unsupervised representation for visual navigation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15890–15899, 2021.

[26] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[29] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[30] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[33] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable

visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PmLR, 2021.
[34] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
[35] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.
[36] P. Chen, D. Ji, K. Lin, R. Zeng, T. Li, M. Tan, and C. Gan, "Weakly-supervised multi-granularity map learning for vision-and-language navigation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38149–38161, 2022.
[37] Z. Wang, X. Li, J. Yang, Y. Liu, and S. Jiang, "Gridmm: Grid memory map for vision-and-language navigation," in *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 15625–15636, 2023.
[38] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," *arXiv preprint arXiv:2010.07954*, 2020.
[39] F. Zhu, Y. Zhu, V. Lee, X. Liang, and X. Chang, "Deep learning for embodied vision navigation: A survey," *arXiv preprint arXiv:2108.04097*, 2021.
[40] C. Xiong, Y. Huang, F. Yu, C. Chen, Y. Wang, S. Xia, and L. Pei, "Sensing, social, and motion intelligence in embodied navigation: A comprehensive survey," *arXiv preprint arXiv:2508.15354*, 2025.
[41] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.
[42] G. Ilharco, V. Jain, A. Ku, E. Ie, and J. Baldridge, "General evaluation for instruction conditioned navigation using dynamic time warping," *arXiv preprint arXiv:1907.05446*, 2019.
[43] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A platform for embodied ai research," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346, 2019.