☼DORAEMON:DECENTRALIZED ONTOLOGY-AWARE RELIABLE AGENT WITH ENHANCED MEMORY ORIENTED NAVIGATION

Anonymous authors

000

001

002

004

006

008 009 010

011

013

014

015

016

017

018

019

021

024

025

026

027

028

029

031

033

034

037

040

041

042

043

044

046

047

048

051

052

Paper under double-blind review

ABSTRACT

Adaptive navigation in unfamiliar environments is crucial for household service robots but remains challenging due to the need for both low-level path planning and high-level scene understanding. While recent vision-language model (VLM) based zero-shot approaches reduce dependence on prior maps and scenespecific training data, they face significant limitations: spatiotemporal discontinuity from discrete observations, unstructured memory representations, and insufficient task understanding leading to navigation failures. We propose DORAE-MON (Decentralized Ontology-aware Reliable Agent with Enhanced Memory Oriented Navigation), a novel cognitive-inspired, zero-shot, end-to-end framework consisting of Ventral and Dorsal Streams that mimics human navigation capabilities. The Dorsal Stream implements the Hierarchical Semantic-Spatial Fusion and Topology Map to handle spatiotemporal discontinuities, while the Ventral Stream combines CoDe-VLM and Exec-VLM to improve decision-making. Our approach also develops Nav-Ensurance to ensure navigation safety and efficiency. We evaluate DORAEMON on the HM3Dv1, HM3Dv2, MP3D, where it achieves state-of-the-art performance on both SR and SPL metrics, significantly outperforming existing methods. We also introduce a new evaluation metric (AORI) to assess navigation intelligence better. Comprehensive experiments demonstrate DORAEMON's effectiveness in zero-shot and end-to-end navigation without requiring prior map building or pre-training. Our code is available at https://anonymous.4open.science/r/DORAEMON-8D4D.

1 Introduction

Adaptive navigation in complex and unseen environments (Batra et al., 2020) is a key capability for household service robots. This task requires robots to move from a random starting point to the location of a target object without any prior knowledge of the environment. For humans, navigation appears almost trivial-however,navigation remains a highly challenging problem for robots: it demands not only low-level path planning to avoid obstacles and reach the destination, but also high-level scene understanding to interpret and make sense of the surrounding environment.

Most existing navigation methods rely on the construction of prior maps(Cadena et al., 2017)

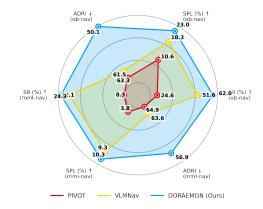


Figure 1: Performance comparison of end-to-end methods in object navigation(ob-nav) and multimodal lifelong navigation(mml).

or require extensive scene-specific data for task-oriented pre-training(Szot et al., 2021). Recently, some works(Yin et al., 2024; Zhong et al., 2024; Wu et al., 2024) have begun to explore zero-training and zero-shot navigation strategies, relying on textual descriptions of the current task, image inputs, and previously observed historical information, these approaches achieve navigation without dependence on environment or task-specific data, gradually shedding the reliance on scene priors.

Although zero-shot and zero-training navigation methods offer a novel perspective, they still face numerous challenges in practical applications. On the one hand, most current navigation methods are non-end-to-end, where the agent's spatial actions is mapped to a discrete set. These discrete actions result in paths that are neither smooth nor efficient. To align with a target, the agent may require multiple small-angle rotations. On the other hand, the primary bottleneck for current Vision-Language Models (VLMs) in long-range navigation is their inadequate memory mechanisms. Their reliance on discrete observational inputs prevents a cohesive understanding of spatiotemporal continuity. More critically, the prevalent approaches(Ramakrishnan et al., 2024; Nasiriany et al., 2024) of storing history as an unstructured log within a single-step decision paradigm fundamentally compromises their ability to perform effective long-term path planning.

Even though end-to-end methods like VLMnav(Nasiriany et al., 2024; Goetting et al., 2024) utilize historical information, they typically store this information in a flat, unstructured manner, which fundamentally limits their ability to perform long-range navigation. Additionally, VLMs sometimes insufficient understanding of task semantics often leads to poor decision-making, and the lack of reliable check mechanisms for navigation states frequently results in unreliable behaviors such as spinning in place during navigation tasks.

Inspired by cognitive science "Decentralized Ontology" principles (Bouquet et al., 2004), we propose the Decentralized Ontology-aware Reliable Agent with Enhanced Memory Oriented Navigation (DORAEMON), which consists mainly of a Ventral Stream and a Dorsal Stream. The core theoretical premise is that knowledge is inherently distributed and context-dependent, composed of multiple local perspectives, rather than the single, monolithic world model. The Ventral Stream processes object identity ("what") information, while the Dorsal Stream handles spatial ("where") processing in the human brain. The Dorsal Stream addresses spatio-temporal discontinuities through a Topology Map and a Hierarchical Semantic-Spatial Fusion, allowing our agent to reason accurately about target-environment relationships. Additionally, the Ventral Stream improves task understanding by utilizing a CoDe-VLM (Compositional Decomposition VLM) and Exec-VLM (Execution VLM) for navigation. Additionally, DORAEMON features a Nav-Ensurance system that enables agent to autonomously detect and respond to abnormal conditions, such as becoming stuck or blocked during navigation. To evaluate navigation performance more comprehensively, we propose a new metric called the Adaptive Online Route Index (AORI). Fig 2 conceptually illustrates limitations of traditional VLN methods and contrasts them with DORAEMON.

In summary, the main contributions of this work are:

- We propose DORAEMON, a novel adaptive navigation framework inspired by cognitive principles of decentralized knowledge, consisting of ventral and Dorsal Streams, enabling end-to-end and zero-shot navigation in completely unfamiliar environments without pretraining, offering plug-and-play compatibility with any VLMs.
- We propose the Dorsal Stream, which involves designing a Topology Map and a Hierarchical Semantic-Spatial Fusion Network to effectively manage spatio-temporal discontinuities.
 Additionally, we introduce the Ventral Stream, incorporating a synergistic reasoning component that combines CoDe-VLM for understanding ontological tasks and Exec-VLM for enhanced task comprehension and planning.
- We develop Nav-Ensurance, which includes multi-dimensional stuck detection and contextaware escape mechanisms. We propose a new evaluation metric called AORI to quantify the efficiency of the agents exploration. Our method demonstrates state-of-the-art performance across various navigation tasks.

2 Related Work

2.1 Zero-shot Navigation

Navigation methods are broadly supervised or zero-shot. Supervised approaches train visual encoders with reinforcement/imitation learning (Khandelwal et al., 2022; Maksymets et al., 2021; Ramrakhya et al., 2022; Chen et al., 2022) or build semantic maps from training data (Zhang et al., 2025a; Min et al., 2021; Zheng et al., 2022), struggling with novel scenarios due to data dependency. Zero-shot methods address this using open-vocabulary understanding, increasingly leveraging foundation

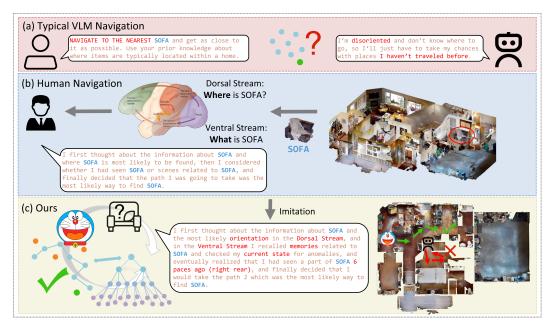


Figure 2: (a) Illustrates limitation of typical VLM navigation (red arrow). (b) DORAEMON's cognitive inspiration from human navigation. (c) Our DORAEMON method.

models like LLMs and VLMs. LLMs provide commonsense reasoning via object-room correlation (Yin et al., 2024; Zhou et al., 2023; Wu et al., 2024), semantic mapping (Yu et al., 2023), and chain-of-thought planning (Cai et al., 2025; Yin et al., 2024; Shah et al., 2023a), while VLMs align visual observations with textual goals. These foundation model-guided techniques include image-based methods mapping targets to visual embeddings (Wen et al., 2025; Gadre et al., 2023; Al-Halah et al., 2022) and map-based approaches using frontier (Zhong et al., 2024; Zhang et al., 2025a; Chen et al., 2023; Kuang et al., 2024; Yu et al., 2023; Shah et al., 2023a) or waypoint-based maps (Wu et al., 2024) with LLM/VLM reasoning. VLM-based strategies either use VLMs for recognition with traditional planning and extra perception models (Rahmanzadehgervi et al., 2024; Zhang et al., 2025b), or, like PIVOT (Nasiriany et al., 2024) and VLMnav (Goetting et al., 2024), directly produce actions end-to-end via visual prompting. Despite progress, many zero-shot methods, especially those processing observations independently, face challenges integrating temporal information and handling complex spatial reasoning in unfamiliar environments.

2.2 MEMORY MECHANISMS IN NAVIGATION

Memory representations in navigation systems have evolved through various architectures, including episodic buffers that maintain observation sequences (Goetting et al., 2024; Shah et al., 2023b; Hsu et al., 2022), spatial representations prioritizing geometric information (Zhong et al., 2024; Zhang et al., 2025b), graph-based semantic structures capturing object relationships (Yin et al., 2025; 2024), predictive world models attempting to forecast environmental states (Cao et al., 2024; Nie et al., 2025) and the memory capacity acquired through training(Zhu et al., 2025). These systems typically process semantic and spatial information separately, with limited integration between perception and reasoning modules. Most approaches focus on either building representations or enhancing reasoning mechanisms independently. Differently, DORAEMON integrates these aspects through a hierarchical semantic-spatial fusion network with bidirectional information flow between ventral and dorsal processing streams.

2.3 COGNITIVE NEUROSCIENCE INSPIRATION IN NAVIGATION

Navigation systems are influenced by cognitive neuroscience, recent models like CogNav(Cao et al., 2024) and BrainNav(Ling & Qianqian, 2025) incorporate cognitive elements, but they do not fully embody Decentralized Ontology. CogNav utilizes a finite state machine for cognitive states, but may

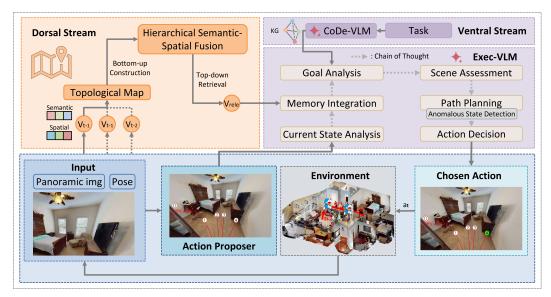


Figure 3: Architecture of the DORAEMON Navigation Framework.

have limitations in knowledge integration. BrainNav mimics biological functions but doesn't deeply engage in decentralized information processing. In contrast, DORAEMON is inspired by Decentralized Ontology(Bouquet et al., 2004), which suggests that human knowledge is organized through interconnected cognitive systems that enable context-dependent reasoning. It emphasizes the integration and bidirectional exchange of information between Dorsal Stream and Ventral Stream, allowing for the construction of semantic relationships that enhance spatial understanding and support flexible, context-aware navigation.

3 METHODS

Task Formulation We address the Navigation task (Batra et al., 2020), where an agent, starting from an initial pose, must locate and navigate to a target within a previously unseen indoor environment. At step t, the agent receives observation I_t , current pose P_t and a task specification T, which can be either a simple object category (e.g., "sofa") or an instruction (e.g., "find the red chair" or "the plant on the desk") for tasks like GOAT (Khanna et al., 2024). Based on these inputs, the agent must decide on an action a_t . While many prior works utilize a discrete action space, our end-to-end framework employs a continuous action representation in polar coordinates (r_t, θ_t) , where r_t specifies the forward distance to move, and θ_t denotes the change in orientation. Crucially, the action space also includes a stop action. The task is considered successful if the agent executes the stop action after meeting successive stop triggers in steps t and t+1. The trigger occurs when 1) the agent is within a predefined distance threshold $d_{success}$ of the target object; 2) the target object is visually confirmed within the agent's current observation I_t .

Methods Overview Our DORAEMON framework achieves end-to-end and zero-shot navigation through two decentralized cognitive-inspired streams, as depicted in Figure 3. Given an input with a panoramic image I_t and a pose P_t at step t, they are processed by Action Proposer (Appendix B) and Dorsal Stream(Section 3.1), respectively. In the Action Proposer, a candidate image $I_{\rm anno}^t$ is generated with a set of action candidates $A_{\rm final}^t$. Concurrently, the Dorsal Stream extracts semantic and spatial information from I_t using Hierarchical Semantic-Spatial Fusion and stores it within the Topology Map as node v_t . The relevant node $v_{\rm rele}$ can be accessed by up-down retrieval. After that, $v_{\rm rele}$ and $I_{\rm anno}^t$ are input to the Exec-VLM to select the best action based on the given information(Section 3.2.2). At the same time, the Exec-VLM receives a task-specific knowledge graph (KG) relevant to the task T, which is generated by the CoDe-VLM (Section 3.2.1) in the Ventral Stream (Section 3.2). The Exec-VLM integrates the information through chain of thought (Appendix I), identifies abnormal conditions (Section 3.3), and outputs the final action a_t . The agent performs this action a_t in the environment, navigates, and makes the next decision at step t+1.

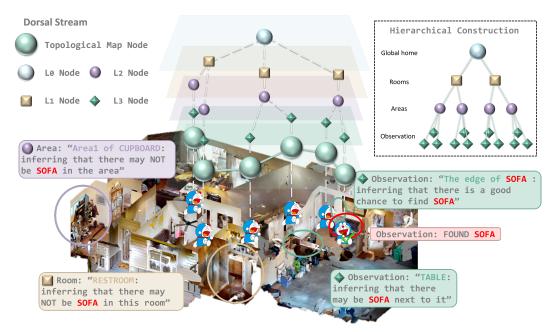


Figure 4: Architecture of Topological Map and Hierarchical Construction built in Dorsal Stream for spatio-temporal memory. The top view in the middle shows the content of different nodes during navigation, and the upper right part represents the Hierarchical Construction of a node.

3.1 Dorsal Stream

The Dorsal Stream, similar to the "where/how" pathway in cognition, is responsible for processing the spatial information to effectively navigate. As illustrated in Figure 4, at each step t, the agent constructs v_k on the Topology Map (Section 3.1.1). Subsequently, the Hierarchical Semantic-Spatial Fusion (Section 3.1.2) organizes the information into a hierarchical structure from the bottom up.

3.1.1 TOPOLOGICAL MAP

The topological map $\mathcal{G}=(\mathcal{V},\mathcal{E})$ is constructed incrementally. Each node $v_t\in\mathcal{V}$ encapsulates the agent's state at timestep t, defined as a tuple $(p_t,q_t,I_t,L_t,o_t,s_t)$. p_t and q_t denote the agent's position and orientation, which constitute the pose P_t . I_t is the visual observation, L_t is its corresponding language description, o_t is the target likelihood estimation, and s_t represents a semantic embedding of the observation (e.g., CLIP features). A new node v_{new} is added to \mathcal{V} based on spatio and temporal criteria: a new node is created if either the time elapsed since the last node addition $t_{\text{curr}}-t_{\text{prev}}$ exceeds a temporal threshold S_{update} , or if the agent's Euclidean distance from the previous node $\|p_{\text{curr}}-p_{\text{prev}}\|_2$ surpasses a spatial threshold δ_{sample} . Upon its creation, v_{new} is connected to its predecessor node v_{prev} via a new edge.

3.1.2 HIERARCHICAL SEMANTIC-SPATIAL FUSION

Hierarchical Construction. Building upon the information associated with the Topological Map nodes $v_t \in \mathcal{V}$, our module organizes information of v_t into a hierarchical structure. The nodes h_j on the hierarchical structure are defined as:

$$h_j = \left(id_j, \ l_j, \ \mathcal{P}_j, \ \mathcal{C}_j\right),\tag{1}$$

where id_j , $l_j \in \{L_0, L_1, L_2, L_3\}$, \mathcal{P}_j , \mathcal{C}_j correspond to unique string identifier, hierarchy level tag, parent node references, and child node references.

The memory hierarchy organizes nodes h_j into four semantic levels through structural and functional relationships (Appendix G): L_3 (Observation, directly linked to topological map nodes

 v_t), L_2 (Area), L_1 (Room), L_0 (Environment). The memory hierarchy is constructed bottom-up $(L_3 \to L_2 \to L_1 \to L_0)$ after an initial exploration phase or periodically. While the overall process involves sequential clustering or integration steps for each level transition, the specific logic and parameters differ between levels.

Hierarchical Memory Retrieval. To efficiently find relevant information within the constructed hierarchy (e.g., observations related to sofa), the system employs a top-down search. This search is guided by a scoring function $S(h_i)$ evaluated at nodes h_i during traversal the constructed hierarchy:

$$S(h_i) = \alpha_{\text{sem}} S_{\text{sem}}(h_i, T) + \alpha_{\text{spa}} S_{\text{spa}}(h_i) + \alpha_{\text{kev}} S_{\text{kev}}(h_i, T) + \alpha_{\text{time}} S_{\text{time}}(h_i), \tag{2}$$

where $S_{\rm sem}$ computes embedding similarity between node h_i and task T, $S_{\rm spa}$ measures proximity to the current position using an exponential decay function, $S_{\rm key}$ evaluates keyword overlap, and $S_{\rm time}$ prioritizes recent observations. For instance, semantic similarity is calculated via normalized cosine similarity, while spatial and temporal scores both rely on exponential decay models to reflect their diminishing influence over distance and time. (see more details in Appendix H).

3.2 VENTRAL STREAM

The Ventral Stream, analogous to the "what" pathway in human cognition, integrates two key components: CoDe-VLM (Compositional Decomposition VLM, Section 3.2.1) and Exec-VLM (Exection VLM, Section 3.2.2). Unlike prior models that encode task information into a single, entangled vector, our architecture explicitly disentangles task comprehension from execution. This decentralized design mirrors the Ventral Stream's approach to compositional understanding, first compiling knowledge and then acting upon it.

3.2.1 CODE-VLM: COMPILING TASKS INTO KNOWLEDGE GRAPHS

To build a deep and structured understanding of the task, CoDe-VLM acts as a semantic compiler. It leverages the vast world knowledge embedded within a VLM to on-the-fly compile an unstructured instruction T into a dynamic, task-specific knowledge graph (KG).

CoDe-VLM generates a graph structure encapsulating nodes and relational edges. This task KG, formed from extracted semantic attributes like general description, appearance, and location, constitutes our explicit and compositional representation of the task. This representation not only enables the agent to robustly verify objects encountered during navigation but also provides crucial priors for planning by interfacing with the spatial reasoning components of the Dorsal Stream.

3.2.2 EXEC-VLM: EXECUTING ACTIONS VIA GRAPH-BASED REASONING

The Exec-VLM serves as the agent's executive core, responsible for determining the optimal action by combining visual observations, spatial awareness from Dorsal Stream, and the structured task semantics provided by CoDe-VLM. Crucially, instead of making decisions in a high-dimensional, entangled feature space, Exec-VLM performs explicit reasoning on the task knowledge graph.

We steer this reasoning process using Chain-of-Thought (CoT). The CoT guides Exec-VLM to break down the complex navigation task into interpretable sub-steps: current state analysis, memory integration, goal analysis, scene assessment, path planning, and action decision. During the "goal analysis" step, for instance, the model directly queries the nodes and edges of the KG to confirm the target's identity and properties, rather than relying on a fragile memory of the initial instruction.

3.3 NAV-ENSURANCE

To enhance the evaluation of safety and efficiency in navigation, we present a new metric Area Overlap Redundancy Index (AORI) (Section 3.3.1). Additionally, we develop Nav-Ensurance, including Multimodal Stuck Detection (Section 3.3.2), context-aware escape strategies (Section 3.3.3), and adaptive precision navigation (Section 3.3.4) to ensure navigation systems reliably and effectively.

3.3.1 AREA OVERLAP REDUNDANCY INDEX (AORI)

We introduce the Area Overlap Redundancy Index (AORI) to quantify the efficiency of the agent's navigation strategy by measuring overlap in area coverage. A high AORI indicates excessive path overlap and inefficient exploration, specifically addressing the limitations of conventional coverage metrics that neglect temporal-spatial redundancy. AORI is formally defined as:

$$AORI = 1.0 - (w_c \cdot (1.0 - r_{\text{overlap}})^2 + w_d \cdot (1.0 - d_{\text{norm}})), \tag{3}$$

Where r_{overlap} represents the ratio of revisited areas, d_{norm} is the normalized density, and $w_c = 0.8, w_d = 0.2$ are weighting coefficients. Further details are provided in Appendix E.

3.3.2 Multimodal Stuck Detection

To detect if it is stuck, the agent analyzes its trajectory over a sliding window of T steps by computing two key metrics: the progress efficiency η and the rotational-to-translational ratio ρ .

$$\eta = \frac{\|p_T - p_0\|_2}{\sum_{t=1}^T \|p_t - p_{t-1}\|_2}, \quad \rho = \frac{\sum_{t=1}^T |\theta_t - \theta_{t-1}|}{\sum_{t=1}^T \|p_t - p_{t-1}\|_2}.$$
 (4)

A stuck state is confirmed if a weighted score $S = w_{\eta} \cdot \mathbb{I}[\eta < \tau_{\eta}] + w_{\rho} \cdot \mathbb{I}[\rho > \tau_{\rho}]$ remains above a threshold S_{th} for k consecutive windows. These metrics effectively detect situations where the agent makes little forward progress (low η) or is spinning in place (high ρ).

3.3.3 Context-aware Escape Strategies

When a stuck state is detected, the system selects an appropriate escape strategy based on the perceived information from Dorsal Stream(Section 3.1). For instance, in corner traps (perceived dead ends), a large turn is executed. In narrow passages, a small backward step followed by a randomized direction change is employed. If the environmental context is ambiguous, the agent will analyze recent successful movement directions and attempt to move perpendicularly, significantly improving escape capabilities from complex trap situations.

3.3.4 ADAPTIVE PRECISION NAVIGATION

As the agent nears the target object, it will activate a precision navigation mode. In this mode, the distance component d of all proposed actions (d,θ) is scaled down by a factor γ_{step} to enable fine-grained positioning adjustments:

$$a_{\text{precise}} = (d \cdot \gamma_{\text{step}}, \theta) \quad \text{for action } (d, \theta) \in A_{\text{actions}}.$$
 (5)

Additionally, when activating the precision navigation mode, the system can utilize visual analysis (using VLM) to create more detailed action options, thereby maximizing final positioning accuracy.

4 EXPERIMENTS

Datasets We evaluate our proposed DORAEMON within the Habitat simulator (Savva et al., 2019) on four large-scale datasets: HM3Dv1(Ramakrishnan et al., 2021)(Object Navigation), HM3Dv2(Yadav et al., 2023)(Object Navigation), and MP3D (Chang et al., 2017)(Object Navigation), GOAT(Khanna et al., 2024) (Multi-modal lifelong navigation, using HM3Dv2).

Implement Details and Evaluation Metrics The action space includes stop, move_forward where the distance parameter is sampled from the continuous range [0.5m, 1.7m], and rotate. We adopt standard metrics to evaluate navigation performance: Success Rate (SR), the percentage of episodes where the agent successfully stops near a target object; Success weighted by Path Length (SPL), defined as $\frac{1}{N} \sum_{i=1}^{N} S_i \frac{l_i}{\max(p_i, l_i)}$, rewarding both success and efficiency; and our proposed Area Overlap Redundancy Index (AORI) (Equation equation 3), which quantifies navigation by penalizing redundant exploration (lower is better). More information is set in the Appendix F.

Baselines We compare DORAEMON against several state-of-the-art navigation methods on the HM3Dv2, HM3Dv1, and MP3D. Our main comparison focuses on end-to-end approaches. Beyond these direct end-to-end counterparts, we also consider a broader set of recent methods for non-endto-end object navigation methods. More baseline details are set in the Appendix J.

METHODS COMPARISION

End-to-end Methods: We evaluate our approach on the HM3Dv2 (ObjectNav,val, Table 1 (a)) and HM3Dv1(GOAT, val, Table 1 (b)) with other end-to-end baselines. DORAEMON achieves state-of-the-art performance on both datasets, outperforming other methods by a significant margin.

Table 1: Comparison of end-to-end navigation methods on different benchmarks.

(a) ObjectNav benchmark

(b) GOAT I	benchmark
------------	-----------

AORI (%) ↓

64.9

56.9 10.5

Method	SR (%) ↑	SPL (%) ↑	AORI (%) ↓	Method	SR (%) ↑	SPL (%) ↑
Prompt-only	29.8	0.107	-	Prompt-only	11.3	3.7
PIVOT(Nasiriany et al., 2024)	24.6	10.6	63.3	PIVOT(Nasiriany et al., 202	(4) 8.3	3.8
VLMNav(Goetting et al., 2024)	51.6	18.3	61.5	VLMNav(Goetting et al., 20	024) 22.1	9.3
DORAEMON (Ours)	62.0	23.0	50.1	DORAEMON (Ours)	24.3	10.3
Improvement	20.2	10.0	18.5	Improvement	10.0	10.8

Non-end-to-end methods: Most methods are non-end-to-end, their reliance on fine-grained discrete actions is a significant departure from natural human behavior, underscoring the superiority of an end-to-end approach. To ensure a fair comparison with these methods that utilize a discrete action set A: move forward 0.25m, turn left/turn right 30°, look up/lookdown 30°, stop, and a common 500 steps episode limit, we conduct an additional set of experiments. In these, we normalize our agent's interactions by approximating an equivalent number of standard discrete steps for each of DORAEMON's actions. During our experiments, one DORAEMON step t was equivalent to about 9-10 non-end-to-end step t_n .

Table 2: Comprehensive comparison with state-of-the-art methods on ObjectNav benchmarks. TF refers to training-free, ZS refers to zero-shot, and E2E refers to end-to-end.

Method	ZS	TF	E2E	HM	3Dv1	HM	3Dv2	M	P3D
				SR(%)↑	SPL(%) ↑	SR(%)↑	SPL(%) ↑	SR(%)↑	SPL(%) ↑
ProcTHOR (Deitke et al., 2022)	×	×	×	54.4	31.8	-	-	-	-
SemEXP (Chaplot et al., 2020)	✓	×	×	-	-	-	-	36.0	14.4
Habitat-Web(Ramrakhya et al., 2022)	√	×	×	41.5	16.0	-	-	31.6	8.5
PONI (Ramakrishnan et al., 2022)	✓	×	×	-	-	-	-	31.8	12.1
ProcTHOR-ZS (Deitke et al., 2022)	√	×	×	13.2	7.7	-	-	-	-
ZSON (Majumdar et al., 2022)	✓	×	×	25.5	12.6	-	-	15.3	4.8
PSL (Sun et al., 2024)	√	×	×	42.4	19.2	-	-	-	-
Pixel-Nav (Cai et al., 2024)	✓	×	×	37.9	20.5	-	-	-	-
SGM (Zhang et al., 2024)	√	×	×	60.2	30.8	-	-	37.7	14.7
ImagineNav (Zhao et al., 2024)	✓	×	×	53.0	23.8	-	-	-	-
CoW (Gadre et al., 2023)	√	√	×	-	-	-	-	7.4	3.7
ESC (Zhou et al., 2023)	✓	✓	×	39.2	22.3	-	-	28.7	14.2
L3MVN (Yu et al., 2023)	√	√	X	50.4	23.1	36.3	15.7	34.9	14.5
VLFM (Yokoyama et al., 2024)	✓	✓	×	52.5	30.4	63.6	32.5	36.4	17.5
VoroNav (Wu et al., 2024)	√	√	×	42.0	26.0	-	-	-	-
TopV-Nav (Zhong et al., 2024)	✓	✓	×	52.0	28.6	-	-	35.2	16.4
InstructNav (Long et al., 2024)	√	√	X	-	-	58.0	20.9	-	-
SG-Nav (Yin et al., 2024)	✓	✓	×	54.0	24.9	49.6	25.5	40.2	16.0
DORAEMON (Ours)	✓	✓	✓	55.6	21.4	66.5	20.6	41.1	15.8

Compared to the non-end-to-end approach in the Table 2, DORAEMON achieves state-of-the-art performance on SR, despite normalizing our action to set A. Each action performed by ours corresponds to several actions in this set. In fact, we only run about 60 end-to-end steps, which further demonstrates the excellence of our DORAEMON.(details are provided in the Appendix C)

Ablation Studies. We perform comprehensive ablation studies to validate our design choices, with results summarized in Table 3. (a) Core Components: Table 3(a) shows that each component is crucial. Removing both the Dorsal and Ventral streams severely degrades performance, confirming their

 synergistic effect. Disabling the Nav-Ensurance mechanism also notably worsens the AORI, highlighting its effectiveness in error prevention. (b) Choice of VLM: The VLM ablation in Table 3(b) indicates that while Gemini-1.5-Pro is optimal, our framework remains highly effective with smaller models. This demonstrates our architecture's inherent strength and its plug-and-play nature, suggesting future compatibility with evolving VLMs. (c) Hyperparameter Sensitivity: The analysis in Table 3(c) reveals a trade-off between metrics. For example, setting TopK=12 yields the highest SR but at the cost of SPL. Our default hyperparameters are carefully chosen to achieve a robust and balanced performance across all metrics, rather than over-optimizing for a single one.

Table 3: A comprehensive ablation study of DORAEMON across different datasets, including variations in modules, VLMs, and hyperparameters. All experiments were evaluated over 100 episodes. Our DORAEMON uses the default hyperparameters: TopK=8, memory update interval=3, area grid size=2.

Method / Configuration	HM3Dv2			HM3Dv1			MP3D		
	SR (%) ↑	SPL (%) ↑	AORI (%) ↓	SR (%) ↑	SPL (%) ↑	AORI (%) ↓	SR (%) ↑	SPL (%) ↑	AORI (%) ↓
			(a) Ablation of	of different m	odules				
w/o Dorsal & Ventral Stream	51.6	18.3	61.5	48.4	18.9	53.7	38.8	13.9	64.3
w/o Dorsal & CoDe-VLM	54.0	19.8	59.1	51.2	19.4	52.5	40.2	14.2	63.8
w/o Dorsal Stream	59.0	22.7	56.3	53.8	20.5	51.1	40.9	14.6	65.1
w/o Nav-Ensurance	60.0	22.5	54.9	53.1	20.7	50.9	42.2	15.3	60.4
		(b) A	Ablation of diffe	rent VLMs (e	on HM3Dv2)				
Qwen-7B	49.5	20.6	68.7	-	-	-	-	-	-
Gemini-1.5-Flash	58.0	20.1	54.8	-	-	-	-	-	-
Gemini-2-Flash	59.0	21.5	57.9	-	-	-	-	-	-
		(c) Ai	blation of Hyper	rparameters	(on HM3Dv2)				
w/ TopK = 12	65.0	22.57	42.89	-	-	-	-	-	-
W/TopK = 4	59.0	22.78	42.03	-	-	-	-	-	-
w/ memory update interval = 1	53.0	19.94	45.54	-	-	-	-	-	-
w/ memory update interval = 5	62.0	23.61	44.22	-	-	-	-	-	-
w/ area grid size = 1	61.0	21.97	39.66	-	-	-	-	-	-
w/ area grid size = 3	61.0	22.95	43.57	-	-	-	-	-	-
DORAEMON (Ours, default)	61.0	23.7	48.8	55.6	21.4	49.1	41.1	15.8	59.3

4.2 NAVIGATION IN REAL WORLD

To validate the Sim-to-Real generalization of our model, we deployed our DORAEMON in a novel office environment. Despite the significant domain gap, the agent successfully completed navigation tasks. Figure 5 shows a representative trial. More demos are available on our project homepage.



Figure 5: DORAEMON's Performance in sim2real

5 CONCLUSION

In this paper, we present DORAEMON , a novel cognitive-inspired framework consisting of Ventral and Dorsal Streams that mimics human navigation capabilities. The Dorsal Stream implements the Hierarchical Semantic-Spatial Fusion and Topology Map to handle spatiotemporal discontinuities, while the Ventral Stream combines CoDe-VLM and Exec-VLM to improve decision-making. Our approach also develops Nav-Ensurance to ensure navigation safety and efficiency. Extensive experimental results demonstrate the superior performance of DORAEMON.

6 ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. Our research did not involve human subjects or animal experimentation. All datasets used, including HM3D and MP3D, were sourced in compliance with their respective usage guidelines, ensuring no violation of privacy. We have taken care to mitigate biases in our research process, and no personally identifiable information was used.

Beyond our current study, we recognize that the responsible deployment of our framework depends on addressing broader ethical and practical challenges. A central consideration is the choice of the foundational model, which presents a trade-off between high-performance proprietary models that sacrifice transparency (e.g., Gemini) and open-source models (e.g., Qwen) that enhance accessibility and privacy but may compromise performance. Furthermore, the transition from simulation to the real world introduces serious security and reliability risks that urgently require rigorous testing. Finally, we acknowledge that inherent biases in academic datasets may limit the generalizability of such models, while the use of cameras in private spaces like homes raises fundamental privacy concerns that must be carefully managed for trustworthy adoption.

7 REPRODICIBILITY STATEMENT

To foster reproducibility and facilitate future research, we have made our source code and experimental setup publicly available. The anonymous repository can be accessed at:

https://anonymous.4open.science/r/DORAEMON-8D4D

This repository contains the implementation of our proposed DORAEMON framework. Additionally, we provide detailed scripts, a README.md file with environment setup instructions, and all necessary configurations to reproduce the main experimental results presented in this paper.

To further ensure that our results are reproducible, the experimental setup, including training steps and hardware details, is described comprehensively within the paper. Furthermore, all evaluations were conducted on publicly available datasets, such as HM3D and MP3D, ensuring that evaluation results can be consistently and independently verified. We believe these measures will enable other researchers to readily reproduce our work and build upon it to further advance the field.

REFERENCES

- Ziad Al-Halah, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Zero experience required: Plug & play modular transfer learning for semantic visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17031–17041, 2022.
- Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *CoRR*, abs/2006.13171, 2020. URL https://arxiv.org/abs/2006.13171.
- Paolo Bouquet, Fausto Giunchiglia, Frank Van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. Contextualizing ontologies. *Journal of Web Semantics*, 1(4):325–343, 2004.
- Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2017.
- Wenzhe Cai, Siyuan Huang, Guangran Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pp. 5228–5234. IEEE, 2024.
- Yuxin Cai, Xiangkun He, Maonan Wang, Hongliang Guo, Wei-Yun Yau, and Chen Lv. Cl-cotnav: Closed-loop hierarchical chain-of-thought for zero-shot object-goal navigation with vision-language models. *arXiv* preprint arXiv:2504.09000, 2025.

- Yihan Cao, Jiazhao Zhang, Zhinan Yu, Shuzhen Liu, Zheng Qin, Qin Zou, Bo Du, and Kai Xu. Cognav: Cognitive process modeling for object goal navigation with llms. *arXiv preprint* arXiv:2412.10439, 2024.
 - Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.
 - Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020.
 - Junting Chen, Guohao Li, Suryansh Kumar, Bernard Ghanem, and Fisher Yu. How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers. *arXiv preprint arXiv:2305.16925*, 2023.
 - Peihao Chen, Dongyu Ji, Kunyang Lin, Weiwen Hu, Wenbing Huang, Thomas Li, Mingkui Tan, and Chuang Gan. Learning active camera for multi-object navigation. *Advances in Neural Information Processing Systems*, 35:28670–28682, 2022.
 - Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.
 - Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23171–23181, 2023.
 - Dylan Goetting, Himanshu Gaurav Singh, and Antonio Loquercio. End-to-end navigation with vision language models: Transforming spatial reasoning into question-answering. *arXiv* preprint *arXiv*:2411.05755, 2024.
 - Hao-Lun Hsu, Qiuhua Huang, and Sehoon Ha. Improving safety in deep reinforcement learning using unsupervised action planning. In 2022 International Conference on Robotics and Automation (ICRA), pp. 5567–5573. IEEE, 2022.
 - Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14829–14838, 2022.
 - Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. Goatbench: A benchmark for multi-modal lifelong navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16373–16383, 2024.
 - Yuxuan Kuang, Hai Lin, and Meng Jiang. Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models. *arXiv preprint arXiv:2402.10670*, 2024.
 - Luo Ling and Bai Qianqian. Endowing embodied agents with spatial reasoning capabilities for vision-and-language navigation. *arXiv preprint arXiv:2504.08806*, 2025.
 - Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zeroshot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.
 - Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zeroshot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35:32340–32352, 2022.

- Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. Thda: Treasure hunt data augmentation for semantic navigation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15374–15383, 2021.
 - So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*, 2021.
 - Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. *arXiv* preprint arXiv:2402.07872, 2024.
 - Dujun Nie, Xianda Guo, Yiqun Duan, Ruijun Zhang, and Long Chen. Wmnav: Integrating vision-language models into world models for object goal navigation. *arXiv preprint arXiv:2503.02247*, 2025.
 - Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind: Failing to translate detailed visual features into words. *arXiv* preprint *arXiv*:2407.06581, 2024.
 - Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Austin Clegg, John M Turner, Manolis Savva, Angel X Chang, and Dhruv Batra. Habitat-Matterport 3D Dataset (HM3D): 1000 large-scale 3D environments for embodied AI. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16203–16213, 2021.
 - Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18890–18900, 2022.
 - Santhosh Kumar Ramakrishnan, Erik Wijmans, Philipp Kraehenbuehl, and Vladlen Koltun. Does spatial cognition emerge in frontier models? *arXiv preprint arXiv:2410.06468*, 2024.
 - Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5173–5183, 2022.
 - Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied AI research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9339–9347, 2019.
 - Dhruv Shah, Michael Robert Equi, Błażej Osiński, Fei Xia, Brian Ichter, and Sergey Levine. Navigation with large language models: Semantic guesswork as a heuristic for planning. In *Conference on Robot Learning*, pp. 2683–2699. PMLR, 2023a.
 - Dhruv Shah, Michael Yang, Michael Laskin, Pieter Abbeel, and Sergey Levine. LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on Robot Learning (CoRL)*, pp. 1083–1093. PMLR, 2023b.
 - Xinyu Sun, Lizhao Liu, Hongyan Zhi, Ronghe Qiu, and Junwei Liang. Prioritized semantic learning for zero-shot instance navigation. In *European Conference on Computer Vision*, pp. 161–178. Springer, 2024.
- Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. Advances in neural information processing systems, 34:251–266, 2021.
 - Congcong Wen, Yisiyuan Huang, Hao Huang, Yanjia Huang, Shuaihang Yuan, Yu Hao, Hui Lin, Yu-Shen Liu, and Yi Fang. Zero-shot object navigation with vision-language models reasoning. In *International Conference on Pattern Recognition*, pp. 389–404. Springer, 2025.

- Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. Voronav: Voronoi-based zero-shot object navigation with large language model. *arXiv preprint arXiv:2401.02695*, 2024.
- Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4927–4936, 2023.
- Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. Advances in neural information processing systems, 37:5285–5307, 2024.
- Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Unigoal: Towards universal zero-shot goal-oriented navigation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19057–19066, 2025.
- Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pp. 42–48. IEEE, 2024.
- Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 35543560. IEEE, October 2023. doi: 10.1109/iros55552.2023.10342512. URL http://dx.doi.org/10.1109/IROS55552.2023.10342512.
- L Zhang, X Hao, Q Xu, Q Zhang, X Zhang, P Wang, J Zhang, Z Wang, S Zhang, and R MapNav Xu. A novel memory representation via annotated semantic maps for vlm-based vision-and-language navigation. *arXiv preprint arXiv:2502.13451*, 2025a.
- Mingjie Zhang, Yuheng Du, Chengkai Wu, Jinni Zhou, Zhenchao Qi, Jun Ma, and Boyu Zhou. Apexnav: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion. *arXiv preprint arXiv:2504.14478*, 2025b.
- Sixian Zhang, Xinyao Yu, Xinhang Song, Xiaohan Wang, and Shuqiang Jiang. Imagine before go: Self-supervised generative map for object goal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16414–16425, 2024.
- Xinxin Zhao, Wenzhe Cai, Likun Tang, and Teng Wang. Imaginenav: Prompting vision-language models as embodied navigator through scene imagination. *arXiv preprint arXiv:2410.09874*, 2024.
- Kaizhi Zheng, Kaiwen Zhou, Jing Gu, Yue Fan, Jialu Wang, Zonglin Di, Xuehai He, and Xin Eric Wang. Jarvis: A neuro-symbolic commonsense reasoning framework for conversational embodied agents. arXiv preprint arXiv:2208.13266, 2022.
- Linqing Zhong, Chen Gao, Zihan Ding, Yue Liao, Huimin Ma, Shifeng Zhang, Xu Zhou, and Si Liu. Topv-nav: Unlocking the top-view spatial reasoning potential of mllm for zero-shot object navigation. *arXiv preprint arXiv:2411.16425*, 2024.
- Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In International Conference on Machine Learning, pp. 42829–42842. PMLR, 2023.
- Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, et al. Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. *arXiv preprint arXiv:2507.04047*, 2025.

A USE OF LARGE LANGUAGE MODELS (LLMS)

Our framework centrally utilizes Large Language Models (LLMs) as the cognitive engine for its two key components: CoDe-VLM and Exec-VLM. The roles are clearly delineated: in the CoDe-VLM module, the LLM functions as a semantic compiler, translating high-level natural language instructions into a structured task knowledge graph. Subsequently, in the Exec-VLM module, the LLM acts as the executive core, performing step-by-step reasoning upon this graph to decide the optimal action.

Separate from this core function within our framework, an LLM was also utilized as a writing aid for the linguistic refinement of this manuscript. This assistance was strictly limited to improving clarity, grammar, and style. It is crucial to note that the ideation, methodology, experimental design, and data analysis were exclusively conducted by the human authors. The authors take full responsibility for all content, including the scientific claims and the final text.

B ACTION PROPOSER

DORAEMON employs an Action Proposer(Goetting et al., 2024) to generate a refined set of candidate actions, which the Exec-VLM then evaluates for the final action decision. As shown in Figure 6, first parameterized action candidates $A_{\rm init}^t$ are generated by the parameterized action space (Equation equation 6). Second, adaptive filtering (Equation equation 7) refines $A_{\rm cand}^t$ using exploration state \mathcal{V}_t and historical patterns \mathcal{H}_t . Safety-critical recovery (Equationequation 8) enforces a rotation cooldown γ through viability evaluation $\mathcal{F}(\cdot)$. Finally, the projection module visually encodes $A_{\rm final}^t$ into $I_{\rm anno}^t$ with numeric tagging (0 for rotation) to interface with VLM's semantic space.

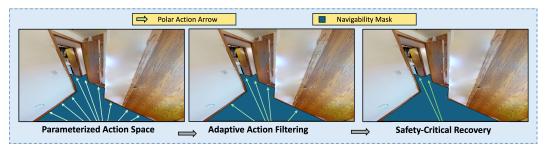


Figure 6: Action proposal: (a) Collision-free action generation within $\pm \theta_{\text{max}}$ FOV, (b) Exploration-aware filtering with $\Delta \theta$ angular resolution, (c) Safety-constrained and action projection.

Parameterized Action Space Define the action space through symbolic parameters:

$$A_{\text{init}}^{t} = \left\{ (\theta_{i}, \min (\eta r_{i}, r_{\text{max}})) \middle| \theta_{i} = k \Delta \theta, k \in \mathcal{K} \right\}.$$
 (6)

where $\mathcal{K} = [-\lfloor \theta_{\text{max}}/\Delta\theta \rfloor, \lfloor \theta_{\text{max}}/\Delta\theta \rfloor]$ ensures full FOV coverage. The safety margin η and collision check are derived from depth-based navigability analysis.

Adaptive Action Filtering Refinement combines exploration state V_t and historical search patterns \mathcal{H}_t :

$$A_{\text{cand}}^{t} = \left\{ (\theta_{i}, r_{i}) \in A_{\text{init}}^{t} \middle| \alpha(\mathcal{H}_{t}) \cdot s(\mathcal{V}_{t}) > \tau, \quad \min_{\theta_{j} \in A_{\text{cand}}} |\theta_{i} - \theta_{j}| \ge \theta_{\delta}. \right\}$$
(7)

where $\alpha(\cdot)$ models temporal search impact and $s(\cdot)$ quantifies spatial exploration potential.

Safety-Critical Recovery The next action set enforces, where $\mathcal{F}(\cdot)$ evaluates action viability and γ controls rotation cool down:

$$A_{\text{final}}^{t} = \begin{cases} \{(\pi, 0)\} & \text{if } , \mathcal{F}(A_{\text{cand}}^{t}) \land (t - t_{\text{rot}} > \gamma) \\ A_{\text{cand}}^{t} & \text{otherwise.} \end{cases}$$
(8)

Algorithm 1 Discrete Step Conversion

Require: Polar action (r, θ) , displacement unit $\Delta_r = 0.25$ m, angular unit $\Delta_\theta = 30^\circ$

```
1: if action is stop then
        return 1
                                                                                           2:
3: else
                                                                                          ⊳ Radial step calculation
4:
        s_r \leftarrow \lceil r/\Delta_r \rceil
        \theta_{\rm deg} \leftarrow 180 |\theta| / \pi
5:

    ▶ Radian-degree conversion

        s_{\theta} \leftarrow \lceil \theta_{\text{deg}} / \Delta_{\theta} \rceil
6:
                                                                                        7:
        N \leftarrow \max(s_r + s_\theta, 1)
                                                                                                 8:
        return N
9: end if
```

Action Projection The following phase focuses on visually anchoring these operational elements within the comprehensible semantic realm of the VLM. The projection component annotated visual depiction $I_{\rm anno}^t$ from $A_{\rm final}^t$ and I_t . We use numeric encoding, assigning a distinct code to each actionable option that is displayed on the visual interface. It is worth noting that rotation is assigned the code 0.

C STEPS CONVERSION

To establish temporal equivalence between DORAEMON's continuous actions and Habitat's discrete steps, we implement the conversion protocol formalized in Algorithm 1. Given a polar action $\mathbf{a}=(r,\theta)\in\mathbb{R}^+\times(-\pi,\pi]$ with radial displacement r meters and angular rotation θ radians:

This formulation enables direct comparison with baseline methods by normalizing both:

$$T_{\text{episode}} = \sum_{t=1}^{500} t_n \le 500 \tag{9}$$

where t_n denotes converted steps for action at time step t. During our experiments, one DORAE-MON step t was equivalent to about 9-10 t_n

We also presented examples of numerical conversions for steps in the experiment.

Table 4: Steps Conversion

(a) Steps Conversion for an end-to-end Step

End-to-end Action	Non-end-to-end Steps	Non-end-to-end Action
(1.27m, 53°)	9	$(1.5m, 75^{\circ})/0.25m \times 6 + 25^{\circ} \times 3$
(1.7m, 60°)	10	$(1.75m, 75^{\circ})/0.25m \times 7 + 25^{\circ} \times 3$
(1.1m, 93°)	9	$(1.25m, 100^{\circ})/0.25m \times 5 + 25^{\circ} \times 4$

(b) Steps Conversion for a Navigation

End-to-end Steps	Non-end-to-end Steps
3	23
16	95
42	300

D NAVIGATION CASE

Figure 7 depicts a full navigation episode in which our memoryaugmented agent searches for a chair in an unfamiliar indoor scene. The seven consecutive frames show the agents visual observations and planned motions from entry to target acquisition. Throughout the sequence, the agent (i) reasons about semantic priorschairs are more likely near tables or in living/dining areas; (ii) fuses transient visual evidence with its episodic memory to avoid revisiting explored regions; and (iii) selects actions that maximise expected information gain while respecting safety constraints. The case illustrates how the proposed memory module complements onthefly perception to yield efficient, goaldirected exploration in cluttered, realworld layouts.



 The agent is currently in a foyer where no chair is visible. Chairs are more likely deeper inside the house (e.g., living or dining rooms). Among the options, moving forward-right takes the agent directly into the open interior while avoiding an obstructing wall, whereas moving right is less direct and moving backward heads toward a closed door. Hence, Action 1 (forward-right) is chosen.



The agent is in a narrow hallway with no chair in sight. Chairs are more likely in a living space glimpsed through the partially open door to the right. Other directions head under stairs, down a narrow dead-end hall, or backward—all unlikely to reveal a chair. Therefore, Action 5 (enter through the open door on the right) is selected.



[left direction, 1.7m, 1 steps ago], [...], [...]

The recent memory of the hallway to my left suggests that's an already explored area, and going backward would be backtracking.

Û

The agent stands in a hallway beside a damaged door.
Through the opening it sees chairs in what looks like a dining room. To reach a chair quickly and avoid retracing explored paths, it chooses Action 1 (forward-left) to pass through the doorway into that room.



[backward direction, 1.7m, 1 steps ago], [...]

memory suggests the chair might be in this room. and the area in front of me seems to be an unexplored area.

Û

The agent is already in a combined living/dining room where chairs are visible, but to cover the still-unexplored central area it chooses Action 3, which moves deepest into the room.



[backward direction, 1.7m, 1 steps ago], [...]

"Memory suggests the chair might be nearby, potentially in the area I just came from, but I should prioritize exploring new spaces."

The agent is in a living/dining room and spots chairs around a small round table on the right. Since those visible chairs are a more reliable cue than vague earlier memories, it chooses Action 3 to move toward that table.



Figure 7: **Navigation case** Each row shows one decision step. *Left:* the green circle highlights the action selected for this step. *Upperright dashed green box* displays the most relevant episodic memory retrieved at this step. *Lowerright speech bubble* is the agents naturallanguage rationale that fuses (i) semantic priors, (ii) current visual evidence, and (iii) memory cues.

E DETAILED DESCRIPTION OF AORI

E.1 AREA OVERLAP REDUNDANCY INDEX (AORI)

The Area Overlap Redundancy Index (AORI) quantifies exploration efficiency through spatial overlap analysis. We formalize the computation with parameters from our implementation:

Parameter Basis:

- Map resolution: $5,000 \times 5,000$ grid (map_size=5000)
- Voxel ray casting resolution: 60×60 (voxel_ray_size=60)
- Exploration threshold: 3 observations per voxel (explore_threshold=3)
- Density scaling factor: $\eta = 0.8$ (e_i_scaling=0.8)

Step-wise Calculation: For each step $t \in [1, T]$:

1. Compute observed area $A_t = \bigcup_{i=1}^t \mathcal{V}(x_i, y_i)$ where $\mathcal{V}(x, y)$ is the visible region defined by:

$$\|\mathcal{V}(x,y)\| = \frac{\text{map_size}^2}{\text{voxel_ray_size}^2} \cdot \pi$$
 (10)

2. Calculate overlap ratio r_{overlap} :

$$r_{\text{overlap}} = \frac{\sum_{i=1}^{t-1} \mathbb{I}[\mathcal{V}(x_t, y_t) \cap \mathcal{V}(x_i, y_i) \ge \text{explore_threshold}]}{t - 1}$$
(11)

3. Compute normalized density $d_{\text{normalized}}$ using Poisson expectation:

$$d_{\text{normalized}} = \min\left(1, \frac{N_{\text{obs}}}{\lambda}\right), \quad \lambda = \eta \cdot \frac{\|A_t\|}{\text{map_size}^2} \cdot t$$
 (12)

where $N_{\rm obs}$ counts voxels with ≥ 3 visits, λ is expected active voxels

Boundary Cases:

- Optimal Case (AORI=0): When $r_{\text{overlap}} = 0 \& d_{\text{normalized}} = 0 \Rightarrow 1 (0.8 \cdot 1^2 + 0.2 \cdot 1) = 0$
- Worst Case (AORI=1): When $r_{\text{overlap}} = 1 \& d_{\text{normalized}} = 1 \Rightarrow 1 (0.8 \cdot 0 + 0.2 \cdot 0) = 1$

Calculation Examples:

• Case1: stay still (t=100 steps):

$$r_{\text{overlap}} = \frac{99}{99} = 1.0, \tag{13}$$

$$\lambda = 0.8 \cdot \frac{\pi (60/5000)^2}{1} \cdot 100 \approx 0.014,$$

$$d_{\text{norm}} = \min\left(1, \frac{100}{0.014}\right) = 1.0,$$

$$AORI = 1 - [0.8(1-1)^2 + 0.2(1-1)] = 1.0$$

• Case2: go around (t=500 steps):

$$r_{\text{overlap}} \approx \frac{38}{499} \approx 0.076,$$
 (14)

$$\lambda = 0.8 \cdot \frac{\pi (60/5000)^2}{1} \cdot 500 \approx 0.069,$$

$$d_{\text{norm}} = \min\left(1, \frac{62}{0.069}\right) = 1.0,$$

$$AORI = 1 - [0.8 \times (1 - 0.076)^2 + 0.2 \times (1 - 1)] \approx 0.285$$

F EXPERIMENTAL SETUP DETAILS

Implementation Details. The maximal navigation steps per episode are set to 40. The agent's body has a radius of 0.17m and a height of 1.5m. Its RGB-D sensors are positioned at 1.5m height with a -0.45 radian downward tilt and provide a 131° Field of View (FoV). For rotation, the agent selects an angular displacement corresponding to one of 60 discrete bins that uniformly discretize the 360° range. Success requires stopping within $d_{\rm success} = 0.3m$ of the target object and visually confirming it. Success requires stopping within $d_{\rm success} = 0.3m$ of the target object and visually confirming it. Our DORAEMON framework primarily utilizes ${\tt Gemini-1.5-pro}$ as the VLM and CLIP ViT-B/32 for semantic embeddings, with caching implemented for efficiency. Key hyperparameters include: topological map connection distance $\delta_{\rm connect} = 1.0m$, node update interval $S_{\rm update} = 3$ steps, L_1 hierarchical clustering weight w = 0.4, AORI grid resolution $\delta_{\rm grid} = 0.1m$, minimum obstacle clearance $d_{\rm min_obs} = 0.5m$, and various stuck detection thresholds (e.g., path inefficiency $\eta_{\rm path} < 0.25$, small area coverage $\delta_{\rm area_gain} < 0.35m^2$, high rotation/translation ratio $\rho_{\rm rot/trans} > 2.0$ for short paths when $\|{\rm path}\| < 0.5m$) and a precision movement factor $\gamma_{\rm step} = 0.1$.

G HIERARCHICAL CONSTRUCTION

- G.1 Level L_3 : Observation Anchoring
 - Input: Raw topological nodes $v_t \in \mathcal{V}$ from Eq 1
 - Process: Directly mapping to memory nodes

$$h_j^{(3)} = \left(id_j^{(3)}, L_3, \emptyset, \{v_t\} \right).$$
 (15)

• Output: $h_i^{(3)}$ nodes storing original p_t, \mathbf{s}_t from v_t

G.2 LEVEL L_2 : AREA FORMATION (L_3 L_2)

972

973

• Input: $\boldsymbol{h}_{j}^{(3)}$ nodes with spatial coordinates p_{t} 974 975 • Clustering: 976 1. Compute combined distance: 977 $d_{\text{comb}} = 0.4 ||p_i - p_j||_2 + 0.6 \left(1 - \frac{\mathbf{s}_i \cdot \mathbf{s}_j}{||\mathbf{s}_i|| ||\mathbf{s}_i||}\right).$ 978 (16)979 980 2. Apply adaptive threshold: 981 $\theta_1' = \begin{cases} 1.5\theta_1 & (|\mathcal{O}| > 20) \\ 0.8\theta_1 & (|\mathcal{O}| < 10) \\ \theta_1 & \text{otherwise.} \end{cases}$ 982 983 (17)984 985 3. Generate clusters using scipy.linkage + fcluster 986 • Functional Labeling: 987 988 $\texttt{area_type} = \arg\max_t \sum_{v \in \mathcal{C}_:^{(2)}} \sum_{k \in K_t} \mathbb{I}[k \in v.L_t].$ (18)989 990 991 • Output: $h_m^{(2)}$ nodes with: 992 - Parent: $h_n^{(1)}$ (L_1 room node). 993 994 - Children: $\{h_i^{(3)}\}$ (observations). 995 - Spatial boundary: Convex hull of p_t positions. 996 997 G.3 Level L_1 : Room Formation (L_2 L_1) 998 999 • Input: $h_m^{(2)}$ areas with spatial centroids P_A 1000 • Two-stage Clustering: 1. Spatial Pre-clustering: 1002 1003 $C_{\text{spatial}} = \text{fcluster}(\text{linkage}(d_{\text{spatial}}), \theta_2 = 3.0\text{m}).$ (19)2. Functional Refinement: $\mathcal{F}_s = \{\mathcal{A}_{s,f} | f = \text{MapToRoomFunction}(\text{area_type})\}.$ (20)1007 • Output: $h_n^{(1)}$ nodes containing: - Parent: $h_0^{(0)}$ (L_0 root) 1010 - Children: $\{h_m^{(2)}\}\ (L_2 \text{ areas})$ 1011 1012 G.4 Level L_0 : Environment Root 1013 1014 • Input: All $h_n^{(1)}$ room nodes 1015 • Consolidation: 1016 $h_0^{(0)} = \left(\texttt{GLOBAL_ROOT}, L_0, \emptyset, \{h_n^{(1)}\} \right).$ (21)1017 1018 • Function: Global access point for memory queries 1019 1020 MEMORY RETRIEVAL SCORING DETAILS 1021 1022 SCORING FUNCTION DECOMPOSITION 1023 1024 The retrieval score combines four evidence components through weighted summation: 1025 $S(h_i) = 0.45S_{\text{sem}} + 0.30S_{\text{spa}} + 0.20S_{\text{key}} + 0.05S_{\text{time}}.$

(22)

H.2 COMPONENT SPECIFICATIONS

H.2.1 SEMANTIC SIMILARITY

- **Input**: CLIP embeddings s_q (query) and s_i (node)
- Calculation:

$$S_{\text{sem}} = \frac{1}{2} \left(1 + \frac{\mathbf{s}_{q}^{\top} \mathbf{s}_{i}}{\|\mathbf{s}_{q}\| \|\mathbf{s}_{i}\|} \right) \in [0, 1].$$
 (23)

H.2.2 SPATIAL PROXIMITY

- Input: Agent position p_a , node position p_i
- Decay function:

$$S_{\text{spa}} = \exp\left(-\frac{\|p_a - p_i\|_2}{5.0}\right). \tag{24}$$

H.2.3 KEYWORD RELEVANCE

- Input: Query terms T, node keywords K_i (from L_t)
- Matching score:

$$S_{\text{key}} = \frac{|T \cap K_i|}{\max(|T|, 1)}.$$
(25)

H.2.4 TEMPORAL RECENCY

- Input: Current time t_c , observation time t_i
- Decay model:

$$S_{\text{time}} = \exp\left(-\frac{|t_c - t_i|}{600}\right). \tag{26}$$

H.3 PARAMETER CONFIGURATION

Table 5: Scoring Component Weights

Component	Symbol	Value
Semantic Similarity	$\alpha_{ m sem}$	0.45
Spatial Proximity	$\alpha_{ m spa}$	0.30
Keyword Relevance	$\alpha_{ m kev}$	0.20
Temporal Recency	α_{time}	0.05

H.4 SEARCH PROCESS

The beam search executes through these discrete phases:

Initialization Phase

- Start from root node(s): $\mathcal{F}_0 = \{h_{\text{root}}\}\$
- Set beam width: B=5

Iterative Expansion For each hierarchy level $l \in \{L_3, L_2, L_1, L_0\}$:

- Score all children: $S(h_{\text{child}}) \forall h_{\text{child}} \in \mathcal{C}(h_j), h_j \in \mathcal{F}_l$
- Select top-B nodes

1080 **Termination Conditions** 1081 1082 • Success: Reached L_0 nodes and selected top-K results 1083 • **Failure**: No nodes satisfy $S(h_i) > 0.4$ threshold 1084 1085 H.5 COMPUTATIONAL PROPERTIES 1086 • Time Complexity: $O(B \cdot D)$ for depth D = 41087 1088 • Memory Complexity: O(B) nodes per level 1089 • Score Normalization: 1090 $\sum_{k \in \{\text{sem,spa,key,time}\}} \alpha_k = 1.0.$ (27)1091 1092 1093 I CHAIN-OF-THOUGHT PROMPT 1094 1095 Our Exec-VLM leverages a structured Chain-of-Thought (CoT) prompt to guide the decision-1096 making process. The complete prompt is provided below: 1097 1098 TASK: NAVIGATE TO THE NEAREST [TARGET_OBJECT], and get as close to it as \hookrightarrow possible. Use your prior knowledge about where items are typically located within 1100 \hookrightarrow a home. 1101 There are [N] red arrows superimposed onto your observation, which 1102 → represent potential actions. 1103 These are labeled with a number in a white circle, which represent the \hookrightarrow location you would move to if you took that action. 1104 [TURN_INSTRUCTION] 1105 1106 Let's solve this navigation task step by step: 1107 1108 1. Current State Analysis: What do you observe in the environment? What → objects and pathways are visible? 1109 Look carefully for the target object, even if it's partially visible 1110 \hookrightarrow or at a distance. 1111 1112 2. Memory Integration: Review the memory context below for clues about → target location. 1113 - Pay special attention to memories containing or near the target 1114 → object 1115 - Use recent memories (fewer steps ago) over older ones 1116 - Consider action recommendations based on memory 1117 1118 3. Goal Analysis: Based on the target and home layout knowledge, where → is the [TARGET_OBJECT] likely to be? 1119 1120 4. Scene Assessment: Quickly evaluate if [TARGET_OBJECT] could 1121 → reasonably exist in this type of space: 1122 - If you're in an obviously incompatible room (e.g., looking for a \hookrightarrow [TARGET_OBJECT] but in a clearly different room type), choose 1123 → action 0 to TURN AROUND immediately 1124 1125 5. Path Planning: What's the most promising direction to reach the 1126 → target? Avoid revisiting 1127 previously explored areas unless necessary. Consider: 1128 - Available paths and typical room layouts - Areas you haven't explored yet 1129 1130 6. Action Decision: Which numbered arrow best serves your plan? Return 1131 → your choice as {"action": <action_key>}. Note: 1132 - You CANNOT GO THROUGH CLOSED DOORS, It doesn't make any sense to go 1133 → near a closed door. - You CANNOT GO THROUGH WINDOWS AND MIRRORS

```
- You DO NOT NEED TO GO UP OR DOWN STAIRS
- Please try to avoid actions that will lead you to a dead end to

→ avoid affecting subsequent actions, unless the dead end is very

→ close to the [TARGET_OBJECT]

- If you see the target object, even partially, choose the action
→ that gets you closest to it
```

J DETAILED DESCRIPTION OF BASELINE

To assess the performance of *DORAEMON*, we compare it with **16** recent baselines for (zeroshot) objectgoal navigation. Summaries are given below.

ProcTHOR (Deitke et al., 2022): A procedurally generated 10K scene suite for largescale Embodied AI.

ProcTHOR_ZS (Deitke et al., 2022): ProcTHOR_ZS trains in ProcTHOR and evaluates zeroshot on unseen iTHOR/RoboTHOR scenes to test crossdomain generalisation.

SemEXP (Chaplot et al., 2020): Builds an online semantic map and uses goaloriented exploration to locate the target object efficiently, achieving stateoftheart results in Habitat ObjectNav 2020.

HabitatWeb (Ramrakhya et al., 2022): Collects largescale human demonstrations via a browser interface and leverages behaviour cloning to learn objectsearch strategies.

PONI (Ramakrishnan et al., 2022): Learns a potential field predictor from static supervision, enabling interaction free training while preserving high navigation success.

ZSON (Majumdar et al., 2022): Encodes multimodal goal embeddings (text + images) to achieve zeroshot navigation towards previously unseen object categories.

PSL (Sun et al., 2024): Prioritised Semantic Learning selects informative targets during training and uses semantic expansion at inference for zeroshot instance navigation.

PixelNav (Cai et al., 2024): Introduces pixelguided navigation skills that bridge foundation models and ObjectNav, relying solely on RGB inputs.

SGM (Zhang et al., 2024): Imagine Before Go constructs a selfsupervised generative map to predict unseen areas and improve exploration efficiency.

ImagineNav (Zhao et al., 2024): Prompts visionlanguage models to imagine future observations, guiding the agent toward informationrich viewpoints.

CoW (Gadre et al., 2023): Establishes the Cows on Pasture benchmark for languagedriven zeroshot ObjectNav and releases baseline policies without indomain training.

ESC (Zhou et al., 2023): Employs soft commonsense constraints derived from language models to bias exploration, markedly improving zeroshot success over CoW.

L3MVN (Yu et al., 2023): Utilises large language models to reason about likely room sequences, while a visual policy executes the suggested path.

VLFM (Yokoyama et al., 2024): Combines VLM goallocalisation with frontierbased exploration, removing the need for reinforcement learning or taskspecific finetuning.

VoroNav (Wu et al., 2024): Simplifies the search space via Voronoi partitions and pairs this with LLMdriven semantic planning for improved zeroshot performance.

TopVNav (Zhong et al., 2024): Lets a multimodal LLM perform spatial reasoning directly on topview maps, with adaptive visual prompts for globallocal coordination.

SGNav (Yin et al., 2024): Online builds a 3D scene graph and uses hierarchical Chain-of-Thought prompting so an LLM can infer probable target locations.