



 Latest updates: <https://dl.acm.org/doi/10.1145/3746027.3755832>

RESEARCH-ARTICLE

CogDDN: A Cognitive Demand-Driven Navigation with Decision Optimization and Dual-Process Thinking

YUEHAO HUANG, Zhejiang University, Hangzhou, Zhejiang, China

LIANG LIU, AIVIVO LTD, Milton, Cambridge, U.K.

SHUANGMING LEI, Zhejiang University, Hangzhou, Zhejiang, China

YUKAI MA, Zhejiang University, Hangzhou, Zhejiang, China

HAO SU, Zhejiang University, Hangzhou, Zhejiang, China

JIANBIAO MEI, Zhejiang University, Hangzhou, Zhejiang, China

[View all](#)

Open Access Support provided by:

Zhejiang University

AIVIVO LTD



PDF Download
3746027.3755832.pdf
22 December 2025
Total Citations: 0
Total Downloads: 44

Published: 27 October 2025

[Citation in BibTeX format](#)

MM '25: The 33rd ACM International Conference on Multimedia
October 27 - 31, 2025
Dublin, Ireland

Conference Sponsors:
SIGMM

CogDDN: A Cognitive Demand-Driven Navigation with Decision Optimization and Dual-Process Thinking

Yuehao Huang [*] yuehaohuang@zju.edu.cn Zhejiang University Hangzhou, China	Liang Liu [*] liangliu.vivoai@vivo.com vivo AI Lab Hangzhou, China	Shuangming Lei shuangminglei@zju.edu.cn Zhejiang University Hangzhou, China	Yukai Ma yukaima@zju.edu.cn Zhejiang University Hangzhou, China
Hao Su haosu@@zju.edu.cn Zhejiang University Hangzhou, China	Jianbiao Mei jianbiaomei@zju.edu.cn Zhejiang University Hangzhou, China	Pengxiang Zhao zhaopengxiang@zju.edu.cn Zhejiang University Hangzhou, China	Yaqing Gu yaqinggu@zju.edu.cn Zhejiang University Hangzhou, China
	Yong Liu [†] yongliu@iipc.zju.edu.cn Zhejiang University Hangzhou, China	Jiajun Lv [†] lvjiajun314@zju.edu.cn Zhejiang University Hangzhou, China	

Abstract

Mobile robots are increasingly required to navigate and interact within unknown and unstructured environments to meet human demands. Demand-driven navigation (DDN) enables robots to identify and locate objects based on implicit human intent, even when object locations are unknown. However, traditional data-driven DDN methods rely on pre-collected data for model training and decision-making, limiting their generalization capability in unseen scenarios. In this paper, we propose CogDDN, a VLM-based framework that emulates the human cognitive and learning mechanisms by integrating fast and slow thinking systems and selectively identifying key objects essential to fulfilling user demands. CogDDN identifies appropriate target objects by semantically aligning detected objects with the given instructions. Furthermore, it incorporates a dual-process decision-making module, comprising a Heuristic Process for rapid, efficient decisions and an Analytic Process that analyzes past errors, accumulates them in a knowledge base, and continuously improves performance. Chain of Thought (CoT) reasoning strengthens the decision-making process. Extensive closed-loop evaluations on the AI2Thor simulator with the ProcThor dataset show that CogDDN outperforms single-view camera-only methods by 15%, demonstrating significant improvements in navigation accuracy and adaptability. The project page is available at <https://yuehaohuang.github.io/CogDDN/>.

^{*}Equal contribution

[†]Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '25, Dublin, Ireland

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-2035-2/2025/10
<https://doi.org/10.1145/3746027.3755832>

CCS Concepts

- Computing methodologies → Cognitive robotics.

Keywords

Demand-driven navigation; Dual-Process; Chain of Thought; Closed-loop

ACM Reference Format:

Yuehao Huang, Liang Liu, Shuangming Lei, Yukai Ma, Hao Su, Jianbiao Mei, Pengxiang Zhao, Yaqing Gu, Yong Liu, and Jiajun Lv. 2025. CogDDN: A Cognitive Demand-Driven Navigation with Decision Optimization and Dual-Process Thinking. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3746027.3755832>

1 Introduction

Mobile robots are becoming indispensable in dynamic and unstructured environments such as households, hospitals, and warehouses. To operate effectively and naturally within these human-centric spaces, they must move beyond simple command execution to interpret and respond to implicit human demands. This capability is particularly critical when target objects are uncertain or not explicitly specified. For instance, a hungry person instinctively seeks food based on internal cues, rather than following a direct instruction like "find the apple on the kitchen counter". Similarly, demand-driven navigation (DDN) [32, 33] enables robots to locate objects that fulfill such implicit needs without relying on predefined object lists or known locations. However, as illustrated in Figure 1 (a), conventional data-driven DDN methods [32, 33] are fundamentally constrained by their heavy reliance on extensive, pre-collected datasets for training and decision-making. This dependence curtails their adaptability and generalization when encountering novel scenarios or vaguely phrased instructions.

Recent breakthroughs in Large Language Models (LLMs) and Vision-Language Models (VLMs) have catalyzed a new era in robotic

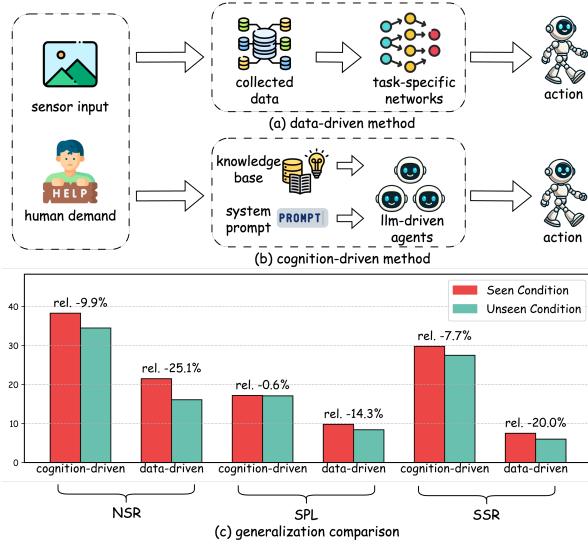


Figure 1: Comparison between cognition-driven and data-driven methods. (a) illustrates the process of the data-driven method, (b) shows the process of the cognition-driven method, and (c) compares the generalization capabilities of both methods using NSR, SPL and SSR metrics, with relative decrease rates marked between seen and unseen conditions.

intelligence, as these models possess remarkable reasoning capabilities and extensive world knowledge that are exceptionally well-suited for complex navigation tasks. Leveraging these advancements, cognition-driven approaches [27, 30, 35, 43, 47] employ these models to help agents interpret high-level instructions and strategically locate targets. The DDN task, which inherently fuses natural language with real-time visual perception, demands that an agent infer and act upon subtle user intent. As depicted in Figure 1 (b), our proposed framework synergistically integrates LLMs and VLMs for DDN, enabling sophisticated and nuanced reasoning over ambiguous instructions. The embedded VLMs enhance navigation by jointly processing visual and linguistic inputs, effectively bridging the semantic gap between high-level intent and spatial context. This cognition-driven, multi-agent approach significantly improves the agent’s spatial and contextual understanding, yielding more accurate, robust, and adaptable navigation. Figure 1 (c) shows that the cognition-driven method generalizes better than traditional data-driven approaches, exhibiting a substantially smaller performance drop when transitioning from seen to unseen scenarios.

This paper introduces CogDDN, a novel VLM-based framework for DDN that is inspired by the human dual-process cognitive framework and designed for robust, closed-loop decision-making. A key differentiator of our approach is its emulation of human cognitive and learning mechanisms to solve complex search tasks. Unlike many contemporary VLM-based navigation systems [3, 20] that rely on supplementary data such as multiple camera views, depth information, or pre-built maps, our system uses only egocentric, front-facing visual input. This design choice intentionally aligns our agent’s perception with natural human search behavior and adheres to the practical constraints of real-world robotic hardware.

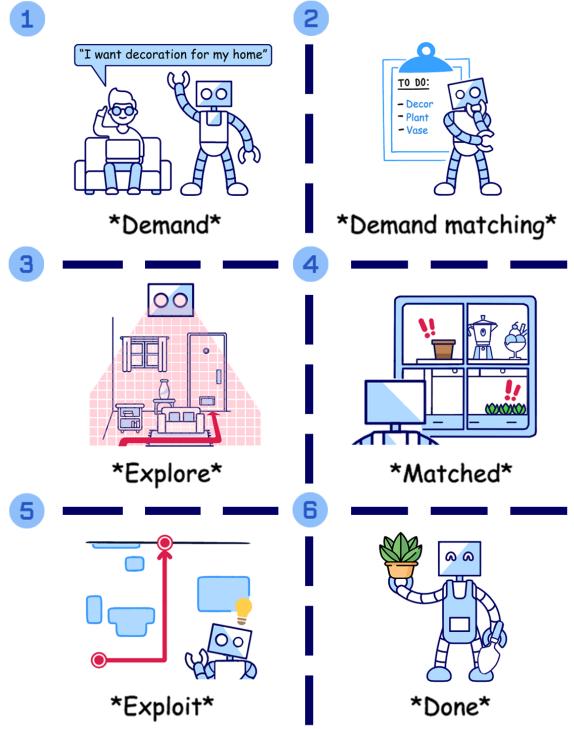


Figure 2: CogDDN Framework for Demand-Driven Navigation Tasks. Upon receiving the instruction "I want decoration for my home", the agent continuously processes real-time observation perspectives and object information inputs. The agent continues this process until it identifies a match to the instruction. At this point, it transitions to the exploitation phase and moves towards the target object.

Consequently, our system addresses a more realistic and challenging navigation task by mirroring the perceptual limitations and sequential decision-making inherent in human object search.

Analogous to human cognitive processes, CogDDN incorporates a demand-matching module that leverages the semantic reasoning power of LLMs. This module translates ambiguous natural language instructions into a set of relevant, potential target objects, enabling the agent to identify and prioritize items that satisfy the user’s underlying needs. Furthermore, we introduce a core decision-making system founded on dual-process theory [8, 13, 38], which simulates human cognition via two distinct yet complementary components: a Heuristic Process (System-I) and an Analytic Process (System-II) [23, 24]. The Heuristic Process, akin to human intuition, facilitates rapid and efficient decisions based on prior knowledge. In contrast, the Analytic Process employs deliberate, step-by-step reasoning to refine these decisions, particularly in novel or challenging situations, and updates the knowledge base accordingly. Prior to navigation, we construct a knowledge base of high-quality, generalizable strategies. The Heuristic Process, continuously refined through supervised fine-tuning (SFT), draws upon this accumulated knowledge to guide initial actions. Meanwhile, the Analytic Process

is invoked to update this knowledge base upon encountering obstacles or failures, creating a virtuous cycle of continuous learning and performance improvement. In contrast to direct end-to-end methods, our system integrates Chain of Thought (CoT) [40] reasoning, enhancing the agent's ability to logically reason through complex scenarios and make transparent, informed decisions, rather than merely outputting final actions.

The main contributions of this paper are summarized as follows:

- We propose CogDDN, a novel VLM-based framework for DDN tasks that simulates human-like cognitive mechanisms and learning processes to navigate based on implicit demands.
- We introduce a dual-process decision-making module wherein the empirical Heuristic Process learns from the rational Analytic Process via a self-supervised mechanism, eliminating the need for human intervention.
- CogDDN uses the Analytic Process and a knowledge accumulation mechanism to progressively enrich a transferable knowledge base, enabling continual learning and effective generalization to novel navigation environments.
- Extensive experiments on AI2Thor simulator [16] with the ProcThor dataset [7] demonstrate that CogDDN outperforms existing single-view camera-only methods by 15%.

2 Related Works

2.1 Demand-driven Navigation

Demand-driven Navigation (DDN) [32, 33] represents an advanced navigation paradigm where an agent must interpret and act upon high-level human demands expressed in natural language. This task diverges significantly from traditional visual navigation frameworks such as Visual Object Navigation (VON) [5, 9, 15, 28, 29, 50], which typically requires finding a specific object category (e.g., "find a chair"), and Visual Language Navigation (VLN) [3, 10, 12, 41, 44, 46, 51, 52], which relies on explicit, step-by-step instructions (e.g., "walk past the table and turn left"). In contrast, DDN centers on fulfilling an underlying user need (e.g., "I'm thirsty"), compelling the agent to infer which objects in the environment are relevant to satisfying the stated demand. This formulation makes DDN a more flexible and dynamic, shifting the focus from locating specific objects to addressing abstract human demands. The agent is thus required to perform a higher level of semantic interpretation and contextual inference based on the current visual scene, fostering a more generalized and human-aligned approach to navigation. Therefore, the primary challenge in DDN lies in this sophisticated process of interpreting abstract human needs and grounding them within a rich, dynamic visual context to guide action.

2.2 Transitioning from Data-Driven to Cognition-Driven Navigation

Early approaches to robotic navigation were predominantly data-driven, relying heavily on imitation learning [14, 26] and reinforcement learning [6, 37] to train reactive policies. These methods learn to map sensory inputs directly to actions by training on large-scale, often manually labeled, datasets. While such approaches have demonstrated promising results within their training distributions,

they frequently exhibit brittleness and fail to generalize to out-of-distribution scenarios [4, 45]. This limitation arises primarily from a lack of deep semantic reasoning and contextual awareness, as the models often learn spurious correlations in the data rather than robust, causal relationships. Furthermore, these data-intensive methods typically incur high computational costs during extensive training and deployment [39, 42, 49].

Motivated by the success of pre-trained vision-language models in a wide array of multi-modal reasoning tasks [18, 19, 21, 22, 36], recent navigation research has shifted towards cognition-driven paradigms. This new wave of research adopts large-scale models as powerful cognitive priors, aiming to instill more robust and human-like navigation strategies [27, 43, 47]. Inspired by human reasoning, these systems unify perception, action, and language to support flexible, language-driven goal specification [25] and a deeper semantic understanding of the world [2]. Unlike traditional methods that depend on structured instructions or explicit spatial maps, VLM-based approaches empower agents to interpret open-ended natural language commands and navigate effectively in complex, unfamiliar environments [31, 48]. These modern systems emphasize multi-modal grounding, context-aware planning, and continual learning. To enhance transparency and robustness, they often incorporate explicit reasoning techniques, such as Chain-of-Thought (CoT) prompting, which encourages the model to generate a sequence of logical steps, thereby supporting more interpretable and generalizable navigation behavior.

3 Method

In this section, we describe the design of CogDDN, a closed-loop, anthropomorphic indoor navigation system. As shown in Figure 3, the system consists of five key components: 3D Robot Perception for identifying objects, an LLM for Demand Matching (Section 3.2), a Knowledge Base for storing experiences (Section 3.3), and a dual-process decision-making module with the Heuristic Process (Section 3.4) and the Analytic Process (Section 3.5). We begin by formally defining the DDN task in Section 3.1, followed by detailed descriptions of each component.

3.1 Task Definition

CogDDN tasks require agents to locate the relevant objects in a given environment based on human demands and perceptual inputs. These tasks can be formally described as a **Partially Observable Markov Decision Process (POMDP)**, defined as

$$\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{E}) \quad (1)$$

where \mathcal{S} represents the state space (the agent's current state), \mathcal{O} is the observation space (including human demands and images), \mathcal{A} is the action space (e.g., MoveAhead, RotateLeft, LookDown, etc.), and \mathcal{E} is the action execution function. For instance, a user may state, "I am thirsty". In response, the agent must identify an object capable of quenching thirst based on the current visual observation and then execute the actions to locate the object. After each action, the agent receives updated observations and generates new actions based on the updated information to continue searching for the object. The task is considered complete when the distance between the agent and the target object is reduced to within 1.5 meters.

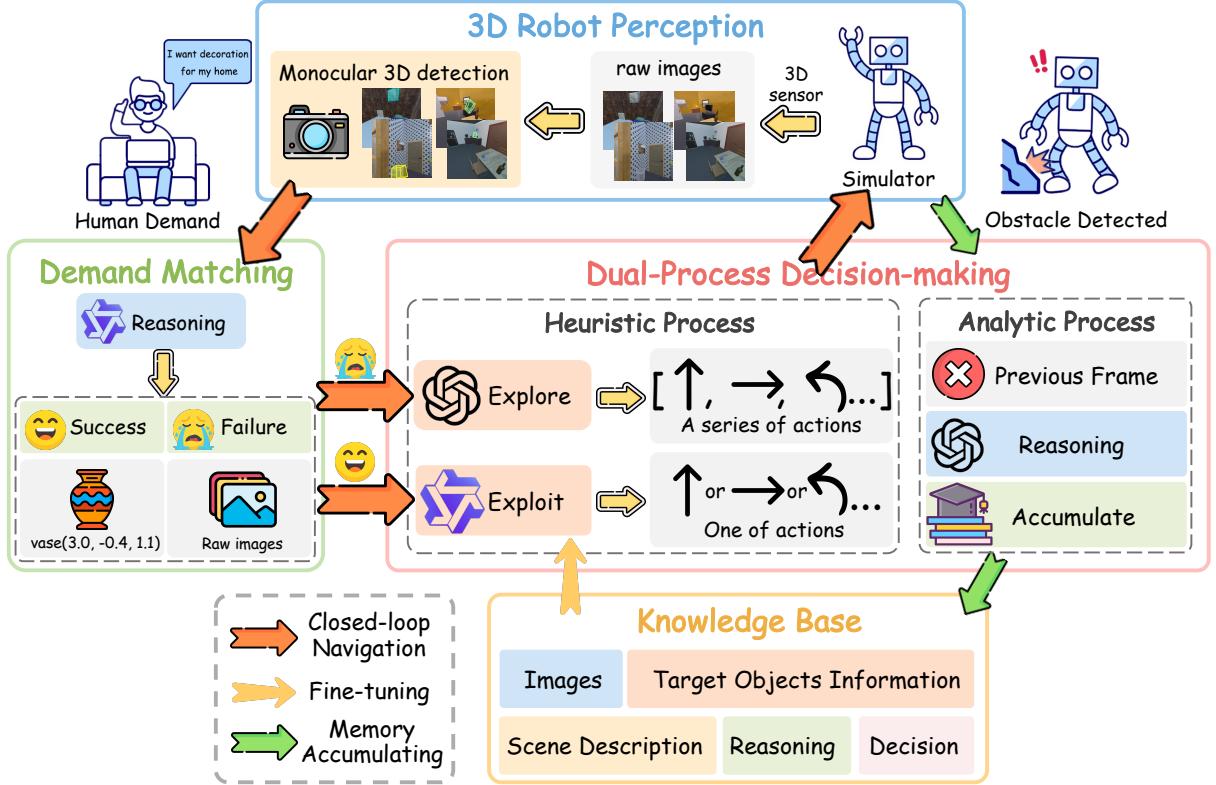


Figure 3: The detailed architecture of our proposed CogDDN. The 3D Robot Perception module identifies objects based on the robot’s perception of the environment. The robot then uses the detected objects and the human demand as input prompts for the demand matching module, which identifies the matched objects. This information enters the dual-process decision module, which drives scene description, reasoning, and decision-making. If no object matches the instruction, the Heuristic Process triggers the explore module to output a series of decisions for exploring unknown areas. Conversely, suppose the system finds a matching object. In that case, it activates the exploit module, refined through the knowledge base, to approach the target object. The Analytic Process analyzes the situation using a VLM whenever the system encounters obstacles, and it stores the corrected information as experience in the knowledge base.

3.2 Demand Matching

A foundational principle of demand-driven navigation is that objects capable of fulfilling the same high-level human demand often share a set of key, underlying properties. For instance, paintings, houseplants, and statues are all suitable candidates for decorating a space. Although physically distinct, they all enhance visual appeal and complement the ambient environment. This intrinsic relationship between abstract human demands and object properties is grounded in commonsense or universal knowledge.

While LLMs are proficient at general reasoning based on instructions and properties, they can encounter difficulties when a perfect match for a given request is unavailable in the immediate environment. In such cases, standard LLMs may suggest suboptimal or functionally adjacent objects. For example, if a user requests, "I need something to hold my flowers", and no vase is present, the model might suggest a mug. Although a mug can technically hold water and flowers, it is a less-than-ideal choice domestically. This tendency to over-generalize based on basic affordances can significantly reduce the model’s accuracy in selecting the most suitable

object that aligns with user expectations. To mitigate this issue, we employ supervised fine-tuning (SFT) to further train the LLM, allowing it to better align object affordances with nuanced user requirements and contextual appropriateness. The fine-tuning enhances the model’s capacity to identify and recommend the most fitting objects, ensuring it avoids suggesting less optimal options, even when an exact match is not readily apparent.

Formally, the Demand Matching module implements a target objects generation function $\mathcal{F} : \mathcal{O} \rightarrow \mathcal{P}, \mathcal{T}$, where \mathcal{O} represents the space of observation space including the human demands and visual information from images, \mathcal{P} is the latent space of inferred demand properties, and \mathcal{T} is the resulting space of candidate target objects. By leveraging a fine-tuned LLM for demand matching, the system benefits from an enhanced ability to reason over complex instructions and subtle object properties. This specialized capability prevents it from forcing matches between human demands and available objects when a direct correspondence is absent, ensuring more appropriate suggestions even in ambiguous scenarios.

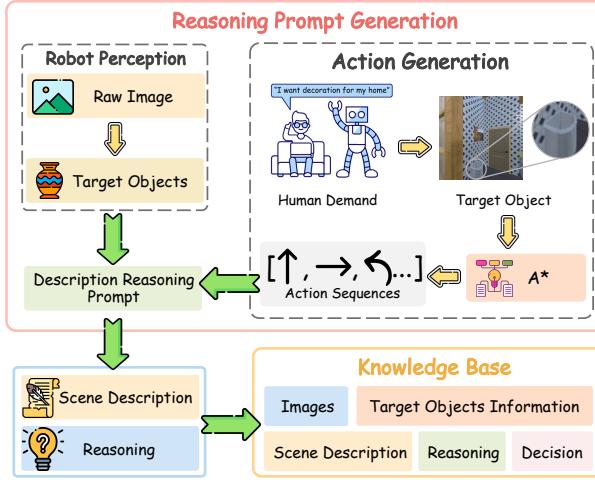


Figure 4: The Construction of the Knowledge Base. A random instruction is selected, and the target object is directly obtained from the simulator. The A^* algorithm generates a trajectory consisting of actions based on the existing grid map and the target object’s position. During the execution of these actions, the target object’s position is determined using the current viewpoint image. When a matching object is detected, the VLMs generate a scene description and reasoning, which is added to the knowledge base.

3.3 Knowledge Base

In indoor navigation scenarios, humans typically prioritize the approximate location of the target object and the scene description, focusing on navigable areas. This approach helps mitigate information overload, enhances reaction times, and reduces cognitive load. Consequently, we use scene descriptions to delineate passable areas, providing a sufficient contextual understanding of the environment. Additionally, Chain of Thought (CoT) reasoning enhances the model’s inference process by leveraging the visual model’s language-based reasoning capabilities. This approach contrasts with direct end-to-end methods by explicitly incorporating reasoning steps before generating final action decisions.

Inspired by LeapVAD [23], we generate additional data with outputs divided into three components: 1) **Scene Description**: This component outlines the environment based on the current viewpoint image, highlighting passable areas and task-related objects that may affect mobility. 2) **Reasoning**: This component leverages the visual-language model’s spatial awareness and commonsense reasoning capabilities based on the target’s location and the scene description to guide the agent’s navigation. 3) **Decision**: Drawing from the output of the reasoning component, the model generates the optimal decision for the current navigation task, ensuring the most effective action. This process incorporates knowledge transfer mechanisms to enhance navigation performance.

Building on this foundational approach of leveraging scene descriptions and reasoning for navigation optimization, we construct a knowledge base that integrates environmental data with dynamic decision-making processes. As illustrated in Figure 4, during the

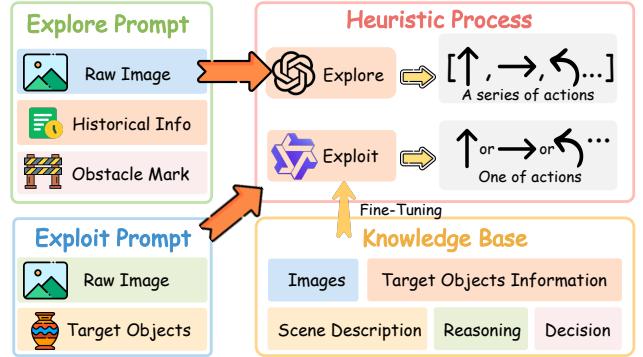


Figure 5: Detailed procedure of the Heuristic Process. When no target object is present, the **Explore** module directly utilizes the VLMs and the current viewpoint image, historical information, and obstacle mark to generate a series of actions for exploring unknown areas. In contrast, the **Exploit** module leverages a VLM fine-tuned with knowledge base information to output a single action that progressively moves towards the target object.

execution of the trajectory generated by the A^* algorithm [11], once the target object is detected, the current viewpoint image I , target object O_m , and final action S are input into the VLMs to generate scene descriptions D and reasoning R , which are then recorded as an experience in the knowledge base. This accumulated experience can be progressively transferred to the Heuristic Process. As detailed in Section 3.4, we utilize the collected data $\{O_m, I, D, R, S\}$ to train the Heuristic Process’s ability to follow instructions and adhere to formatting standards.

3.4 Heuristic Process

While the VLMs can effectively generate reasoning that aligns with decisions from path planning algorithms like A^* algorithm [11], they often struggle to directly infer the causal relationship between reasoning and decisions. Moreover, their relatively slow processing speed can lead to redundancy and repetition, limiting their practical application in real-time indoor navigation. To address these challenges, we introduce the Heuristic Process in CogDDN. This process is inspired by human navigation, where efficiency is gained through repeated practice and environmental familiarity, eventually allowing individuals to navigate with minimal cognitive effort. Formally, the Heuristic Process module implements an action execution function $\mathcal{E} : \mathcal{T} \times \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$, where \mathcal{T} represents the space of target objects, \mathcal{S} depicts the state space, and \mathcal{A} is the action space, as presented in Table 1. As demonstrated in Figure 5, it consists of two modules: **Explore** and **Exploit**, both leveraging Chain-of-Thought (CoT) reasoning to adapt to new environments and optimize learned navigation execution. The Heuristic Process is expressed as:

$$D, R, S = \begin{cases} \text{Explore}(I, S', X) & \text{if } \text{len}(O_m) = 0 \\ \text{Exploit}(I, O_m) & \text{else} \end{cases} \quad (2)$$

Table 1: CogDDN Action Space

Action	Definition
MoveAhead	Move forward by 0.25 meters.
RotateLeft	Rotate the agent 30° to the left.
RotateRight	Rotate the agent 30° to the right.
LookUp	Tilt the agent's camera upward by 30°.
LookDown	Tilt the agent's camera downward by 30°.
Done	Indicate that the goal has been found.

Explore. The Explore module is activated when the Demand Matching module fails to identify a suitable target object within the agent's current field of view. In this mode, the system defaults to a broad, information-seeking search of the environment. It generates exploratory actions by leveraging the VLM to interpret its visual perspective and reason about promising, unvisited areas. The objective is to efficiently survey the surroundings to uncover previously overlooked objects or navigational paths. As illustrated in Figure 2, this process begins with the system generating a concise Scene Description D from the current viewpoint image I . Subsequently, it engages in explicit Reasoning R to formulate a strategy for exploration. Based on this reasoning, the system makes a Decision S , which comprises a sequence of actions designed to navigate toward unvisited regions. To prevent inefficient behavior, the module incorporates a short-term memory of prior actions and recent rotations S' to minimize redundant movements. Furthermore, if an obstacle impedes progress, a flag X is set to true, prompting the system to dynamically replan and generate new exploratory actions, which ensures continuous progress without getting stuck in loops or depleting resources.

Exploit. Once the Demand Matching module successfully identifies a target object, the system transitions from exploration to exploitation. The Exploit module shifts the agent's strategy from a broad, open-ended search to precise, goal-directed action. This phase leverages a version of the VLM that has been fine-tuned on prior successful experiences stored in the knowledge base, optimizing it for goal achievement. This fine-tuning enhances the model's ability to make accurate predictions and decisions based on subtle environmental cues, enabling the system to execute complex navigation tasks with high precision. Guided by a Chain-of-Thought(CoT) process, the system first generates a Scene Description D based on the current viewpoint image I and the information about the target objects O_m . It then performs focused Reasoning R to infer the most direct course of action toward the target, given D and O_m . Finally, it produces the most optimal Decision S , a single, high-confidence action focused on progressing toward the target. By concentrating on a single, decisive action rather than a sequence of potential actions, the system improves its decisiveness and computational efficiency, ensuring rapid and effective goal attainment.

3.5 Analytic Process

The Analytic Process serves as the deliberative reasoning component of our framework, designed to reflect upon and learn from

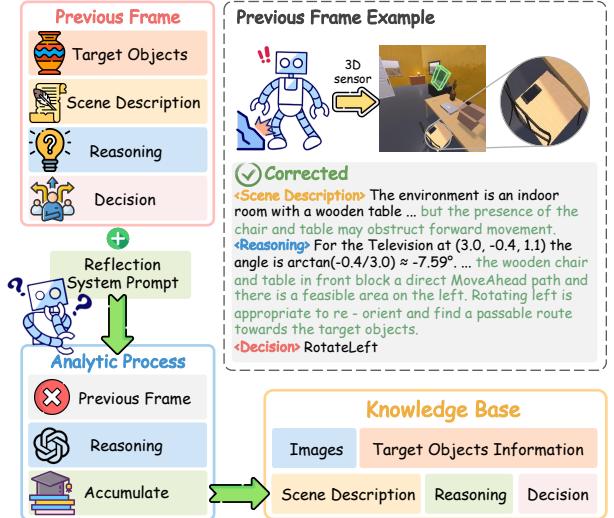


Figure 6: Detailed procedure of the Analytic Process. When the Heuristic Process encounters an obstruction, the Analytic Process intervenes by analyzing the previous frame to identify errors and generate corrected samples. These corrected samples are subsequently integrated into the knowledge base, supporting continuous learning.

navigation failures, as illustrated in Figure 6. Through comprehensive pre-training on diverse, large-scale datasets, VLMs naturally accumulate a vast repository of world knowledge. This enables them to address complex challenges with sophisticated reasoning and insight, which is a capability that aligns perfectly with the objectives of the Analytic Process. This process moves beyond the rapid, intuitive responses of the Heuristic Process and instead relies on detailed causal analysis and deep contextual understanding to draw sound, generalizable inferences from navigational errors encountered in indoor environments.

Specifically, within the closed-loop operation of our system, any obstruction or failure encountered during the execution of the Heuristic Process triggers this reflective mechanism. When triggered, the Analytic Process utilizes the extensive world knowledge embedded within the VLMs to diagnose the situation. To do this, it ingests a snapshot of the state preceding the failure, including the target object's location, the scene description, and the specific reasoning and decision that led to the error. The system then carefully analyzes the root cause of the event, identifies the flaw in the previous logic, and generates a corrected chain of reasoning along with a revised decision. The valuable insights gained from this reflective process, a successful pairing of a problem state with a corrected solution, are subsequently integrated into the knowledge base. This integration establishes an iterative learning loop, enabling the system to learn from its failures continuously and progressively enhance its decision-making capabilities. This virtuous cycle ultimately leads to more informed and accurate navigation strategies for future tasks and directly improves the performance of the Heuristic Process over time.

Table 2: Performance comparison with single-view camera-only methods on DDN tasks. Seen Scene: Train scene from ProcThor, Unseen Scene: Test scene from ProcThor, Seen Ins: Instructions used for training, Unseen Ins: Instructions used for testing, NSR: Navigation success rate, SPL: Navigation success rate weighted by the path length, SSR: Selection success rate.

Method	Seen Scene						Unseen Scene					
	Seen Ins			Unseen Ins			Seen Ins			Unseen Ins		
	NSR	SPL	SSR	NSR	SPL	SSR	NSR	SPL	SSR	NSR	SPL	SSR
Random [32]	5.2	2.6	3.0	3.7	2.6	2.3	4.8	3.3	2.8	3.5	1.9	1.4
VTN-demand [32]	6.3	4.2	3.2	5.2	3.1	2.8	5.0	3.2	2.8	6.6	4.0	3.3
VTN-CLIP-demand [32]	12.0	5.1	5.7	10.7	3.5	5.0	10.0	3.6	4.0	9.3	3.9	4.0
VTN-GPT* [32]	1.6	0.5	0	1.4	0.4	0.5	1.3	0.2	0.3	0.9	0.4	0.5
ZSON-demand [32]	4.2	2.7	1.9	4.6	3.1	2.0	4.1	2.9	1.2	3.5	2.4	1.1
ZSON-GPT [32]	4.0	1.1	0.3	3.6	1.9	0.3	2.5	0.7	0.2	3.2	0.9	0.2
CLIP-Nav-MiniGPT-4 [32]	4.0	4.0	2.0	3.0	3.0	2.0	4.0	3.7	2.0	5.0	5.0	3.0
CLIP-Nav-GPT* [32]	5.0	5.0	4.0	6.0	5.5	5.0	5.5	5.3	4.0	4.0	3.0	2.0
FBE-MiniGPT-4 [32]	3.5	3.0	2.2	3.5	3.2	2.0	3.5	3.5	2.0	4.0	4.0	3.5
FBE-GPT* [32]	5.0	4.3	4.3	5.5	5.0	5.5	4.5	4.3	4.5	5.5	5.0	5.5
GPT-3-Prompt* [32]	0.3	0.01	0	0.3	0.01	0	0.3	0.01	0	0.3	0.01	0
MiniGPT-4 [32]	2.9	2.0	2.5	2.9	2.0	2.5	2.9	2.0	2.5	2.9	2.0	2.5
DDN [32]	21.5	9.8	7.5	19.3	9.4	4.5	14.2	6.4	5.7	16.1	8.4	6.0
CogDDN(ours)	38.3	17.2	29.8	37.5	18.0	28.6	33.3	16.4	27.1	34.5	17.1	27.5

4 Experiments

4.1 Data preparation

Our experiments are conducted in the AI2Thor [16] simulator using the ProcThor dataset [7]. Following the setup of DDN [32], we evaluate our model on 600 scenes (200 from each of ProcThor’s train, validation, and test splits), containing 109 object categories. We use the instruction sets provided by DDN [32], comprising 1692 training, 241 validation, and 485 test instructions.

Data for 3D Robot Perception. For the 3D Robot Perception Module, we collected 55K frames from AI2THOR, split into 70% training, 10% validation, and 20% testing sets. Each frame includes annotations for object category, 2D bounding box, and 3D box attributes. The 3D box information was obtained by extracting the minimal bounding box from the object’s point cloud, which was generated using the simulator’s depth and segmentation maps.

Data for Demand Matching. For the Demand Matching Module, we created a dataset of 10.7K instruction-attribute pairs. The construction process involved using LLMs to generate common attributes for objects associated with a given instruction. This was structured into a QA format, where the question (Q) was an instruction and a set of candidate objects, and the answer (A) was the shared attributes and a list of correct objects. The dataset was split into 7.2K training, 1.4K validation, and 2.1K test samples.

Data for Heuristic Process. For the Heuristic Process, we generated a dataset of 72K expert trajectories in AI2THOR. An optimal path was computed for each instruction using the A* algorithm [11]. Once the target object was detected in the agent’s field of view, we used a VLM to generate a rationale for the path taken. This data was formatted into a VQA structure, where the question (Q) was the target’s location and the answer (A) was the scene description, reasoning, and decision sequence. This dataset, used to populate

our Knowledge Base, was split into 80% for training and 20% for validation.

4.2 Implementation Details

For the 3D Robot Perception module, we utilize UniMODE [17]. The model is trained for 30,000 iterations with a batch size of 64 on four RTX6000 GPUs, a process that takes approximately 10 hours. Input images are resized from 448×448 to 640×512 pixels. We use the AdamW optimizer with a learning rate of 1.2×10^{-3} , and a weight decay of 1.0×10^{-2} . A CosineAnnealing scheduler is used, with a warm-up factor of 0.3333 for the first 2000 iterations. To ensure stability, gradient clipping is applied at a threshold of 35.0.

For the Demand Matching, we conduct SFT on Qwen2-VL-7B [34] with LoRA, training with a batch size of 1 for 5 epochs on 4 RTX6000 GPUs in 1 hour. Key hyperparameters include a learning rate of 1×10^{-5} , gradient accumulation of 8, and a cosine scheduler. LoRA is applied with a rank of 64 and alpha of 128, while freezing the ViT layers.

For the Heuristic Process, we conduct SFT on Qwen2-VL-7B for 5 epochs using knowledge base samples, which takes about 14 hours, with consistent hyperparameters from the Demand Matching.

4.3 Evaluation in Closed-Loop Navigation

We conduct closed-loop experiments in AI2Thor, a widely-used open-source simulator, to evaluate the performance of CogDDN on 400 scenes using three metrics: Navigation Success Rate (NSR), Navigation Success Rate weighted by the path length (SPL) [1], and Selection Success Rate (SSR). Table 2 compares our method with competitive methods offered by DDN [32] on the ProcThor benchmark.

As shown in Table 2, CogDDN outperforms all single-view camera-only methods. With the LLM’s reasoning capabilities, the Demand

Table 3: Performance comparison with SOTA methods on DDN tasks. I: RGB image, D: Depth map.

Method	Modality	Reflection	NSR	SPL
InstructNav [20]	D+I	✗	30.0	14.2
DDN [32]	I	✗	16.1	8.4
CogDDN(ours)	I	✓	34.5	17.1

Table 4: Ablation study of Exploit and CoT.

Method	Single Action	NSR	SPL	SSR
CogDDN(w/o Exploit)	✓	24.2	13.9	19.2
CogDDN(w/o Exploit)	✗	29.1	12.4	20.2
CogDDN(w/o CoT)	✓	27.3	11.5	21.1
CogDDN	✓	34.5	17.1	27.5

Matching module exhibits high generalization, with minimal performance gaps between seen and unseen instructions. The main gap between seen and unseen scenes stems from the 3D Robot Perception. Additionally, as shown in Figure 1 (c), CogDDN demonstrates a smaller relative drop between seen and unseen conditions, proving its strong generalization ability. Furthermore, as shown in Table 3, we compare CogDDN with SOTA methods InstructNav [20] on unseen scenes and instructions. Remarkably, CogDDN achieves comparable performance, even though InstructNav leverages depth maps as additional input.

4.4 Ablation Study

We conduct ablation studies on Exploit CoT, and Reflection in a closed-loop navigation system. The experiments were carried out on both unseen scenes and unseen instructions, demonstrating the generalization and continuous learning capabilities of our CogDDN.

Ablation Study of Exploit. The Exploit module uses a fine-tuned VLM to generate precise, deliberate single-step actions critical for approaching the target. To assess the impact of fine-tuning, we replace the Exploit module with a vanilla GPT-4 model not adapted for navigation and evaluate performance in two settings: (i) generating one action per step and (ii) generating a full action sequence. As shown in Table 4, this substitution causes a significant performance drop, underscoring the importance of task-specific fine-tuning for accurate, step-wise decisions in the Exploit phase.

Ablation study of CoT. In both the Heuristic and Analytic Processes, we use CoT to guide the VLM in generating high-quality decisions. To evaluate its effect, we remove CoT and generate decisions directly. As shown in Table 4, this results in a significant performance drop, underscoring its importance within the dual-process framework. In the Heuristic Process, CoT helps leverage environmental information for decision-making. In the Analytic Process, CoT enables the VLM to analyze obstacles and uncover root causes, resulting in more accurate and informed decisions.

Ablation study of Reflection. Reflection is crucial for enhancing the continuous learning capabilities of our proposed CogDDN. This mechanism promotes self-improvement by analyzing navigational failures and integrating corrected experiences into the knowledge base, allowing the agent to proactively accumulate knowledge in unseen environments. We conducted a closed-loop experiment

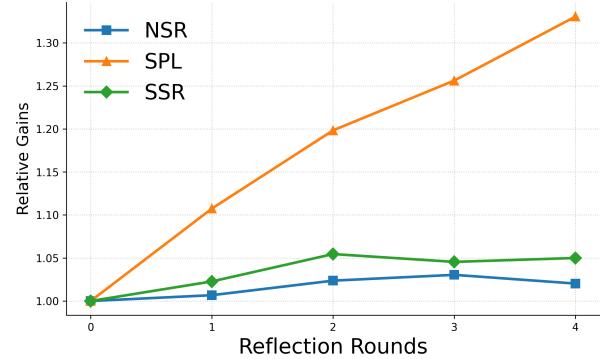


Figure 7: Ablation of Reflection. Four rounds of experiments were conducted, each consisting of 500 epochs. In each round, the experiences generated by the reflection mechanism were integrated into the knowledge base, and the VLM was fine-tuned accordingly.

over four iterative rounds of 500 epochs each to evaluate this capability. In each round, experiences generated by the reflection mechanism were used to augment the knowledge base and fine-tune the VLM. As shown in Figure 7, after four rounds of iterative learning, both SSR and NSR showed only marginal gains, which is an expected outcome as the reflection mechanism primarily enables the agent to recover from obstacles after they are encountered, leading to only modest improvements in these goal-completion metrics. In contrast, the SPL exhibited a significant increase. This substantial improvement in SPL demonstrates that as the agent accumulates experience, it learns to anticipate potential obstacles and take proactive detours, resulting in more efficient paths. This outcome underscores the effective continuous learning capability of CogDDN, as its performance demonstrably improves with experience.

5 Conclusion

In this paper, we introduced CogDDN, a novel dual-process closed-loop navigation system designed to emulate human attentional mechanisms. Central to our approach is a dual-process decision-making module that simulates human cognition, where a deliberative process is enhanced by Chain of Thought (CoT) reasoning to navigate complex scenarios. Furthermore, we integrated a cumulative knowledge base that enables the agent to continuously learn and self-improve from its experiences. Our comprehensive evaluations demonstrate that CogDDN achieves state-of-the-art performance in closed-loop navigation tasks while requiring minimal training data, underscoring its efficiency and practical potential.

Limitations and Broader Impacts. The current CogDDN system incorporates only short-term memory in the Explore module, lacking long-term memory to identify previously explored areas. Additionally, employing GPT-4 as a decision-making module in the Explore phase is computationally expensive and impractical for real-world deployment. Moreover, updating the Heuristic Process requires SFT whenever new experiences are added to the Knowledge Base, leading to inefficiency. In the future, we will design a new system focusing on long-term memory and end-to-end navigation.

Acknowledgments

This work is supported by the State Key Laboratory of Industrial Control Technology, China (Grant No. ICT2024A09).

References

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. 2018. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757* (2018).

[2] Lei Han Hamidreza Kasaei Tingguang Li Bangguo Yu, Yuzhen Liu and Ming Cao. 2024. VLN-Game: Vision-Language Equilibrium Search for Zero-Shot Semantic Navigation. *arXiv:2411.11609* (2024).

[3] Wenzhe Cai, Siyuan Huang, Guangan Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. 2024. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5228–5234.

[4] Yuxin Cai, Xiangkun He, Maonan Wang, Hongliang Guo, Wei-Yun Yau, and Chen Lv. 2025. CL-CoTNav: Closed-Loop Hierarchical Chain-of-Thought for Zero-Shot Object-Goal Navigation with Vision-Language Models. *arXiv preprint arXiv:2504.09000* (2025).

[5] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. 2020. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems* 33 (2020), 4247–4258.

[6] Jinyu Chen, Chen Gao, Erli Meng, Qiong Zhang, and Si Liu. 2022. Reinforced structured state-evolution for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15450–15459.

[7] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weils, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. 2022. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. *Advances in Neural Information Processing Systems* 35 (2022), 5982–5994.

[8] Jonathan St BT Evans and Keith E Stanovich. 2013. Dual-process theories of higher cognition: Advancing the debate. *Perspectives on psychological science* 8, 3 (2013), 223–241.

[9] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. 2022. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421* 3, 4 (2022), 7.

[10] Dylan Goetting, Himanshu Gaurav Singh, and Antonio Loquercio. 2024. End-to-End Navigation with Vision Language Models: Transforming Spatial Reasoning into Question-Answering. *arXiv preprint arXiv:2411.05755* (2024).

[11] Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* 4, 2 (1968), 100–107.

[12] Haodong Hong, Yanyuan Qiao, Sen Wang, Jiajun Liu, and Qi Wu. 2025. General Scene Adaptation for Vision-and-Language Navigation. *arXiv preprint arXiv:2501.17403* (2025).

[13] Daniel Kahneman. 2011. Fast and slow thinking. *Allen Lane and Penguin Books, New York* (2011).

[14] Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. 2023. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10813–10823.

[15] DA Sasi Kiran, Kritika Anand, Chaitanya Kharyal, Gulshan Kumar, Nandiraju Gireesh, Snehasis Banerjee, Ruddra Dev Roychoudhury, Mohan Sridharan, Brojeshwar Bhowmick, and Madhava Krishna. 2022. Spatial relation graph and graph convolutional network for object goal navigation. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 1392–1398.

[16] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weils, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. 2017. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474* (2017).

[17] Zhuoliang Li, Xiaogang Xu, SerNam Lim, and Hengshuang Zhao. 2024. UniMODE: Unified monocular 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16561–16570.

[18] Guangyi Liu, Pengxiang Zhao, Liang Liu, Zhiming Chen, Yuxiang Chai, Shuai Ren, Hao Wang, Shibo He, and Wenchoao Meng. 2025. Learnact: Few-shot mobile gui agent with a unified demonstration benchmark. *arXiv preprint arXiv:2504.13805* (2025).

[19] Guangyi Liu, Pengxiang Zhao, Liang Liu, Yaxuan Guo, Han Xiao, Weifeng Lin, Yuxiang Chai, Yue Han, Shuai Ren, Hao Wang, et al. 2025. Llm-powered gui agents in phone automation: Surveying progress and prospects. *arXiv preprint arXiv:2504.19838* (2025).

[20] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. 2024. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882* (2024).

[21] Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. 2025. UI-R1: Enhancing Efficient Action Prediction of GUI Agents by Reinforcement Learning. *arXiv preprint arXiv:2503.21620* (2025).

[22] Yukai Ma, Jianbiao Mei, Xuemeng Yang, Licheng Wen, Weihua Xu, Jiangning Zhang, Xingxing Zuo, Botian Shi, and Yong Liu. 2024. Licrocc: Teach radar for accurate semantic occupancy prediction using lidar and camera. *IEEE Robotics and Automation Letters* (2024).

[23] Yukai Ma, Tiantian Wei, Naifeng Zhong, Jianbiao Mei, Tao Hu, Licheng Wen, Xuemeng Yang, Botian Shi, and Yong Liu. 2025. LeapVAD: A Leap in Autonomous Driving via Cognitive Perception and Dual-Process Thinking. *arXiv preprint arXiv:2501.08168* (2025).

[24] Jianbiao Mei, Yukai Ma, Xuemeng Yang, Licheng Wen, Xinyu Cai, Xin Li, Daocheng Fu, Bo Zhang, Pinlong Cai, Min Dou, et al. 2024. Continuously learning, adapting, and improving: A dual-process approach to autonomous driving. *arXiv preprint arXiv:2405.15324* (2024).

[25] Reihaneh Mirjalili, Michael Krawez, Florian Walter, and Wolfram Burgard. 2024. VLM-Vac: Enhancing Smart Vacuums through VLM Knowledge Distillation and Language-Guided Experience Replay. *ArXiv abs/2409.14096* (2024). <https://api.semanticscholar.org/CorpusID:272826708>

[26] Khanh Nguyen, Debadatta Dey, Chris Brockett, and Bill Dolan. 2019. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12527–12537.

[27] Dujun Nie, Xianda Guo, Yiqun Duan, Ruijun Zhang, and Long Chen. 2025. WM-Nav: Integrating Vision-Language Models into World Models for Object Goal Navigation. <https://api.semanticscholar.org/CorpusID:276776282>

[28] Anwesan Pal, Yiding Qiu, and Henrik Christensen. 2021. Learning hierarchical relationships for object-goal navigation. In *Conference on Robot Learning*. PMLR, 517–528.

[29] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. 2022. Poni: Potential functions for object-goal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18890–18900.

[30] Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. 2022. LM-Nav: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action. In *6th Annual Conference on Robot Learning*. <https://openreview.net/forum?id=UW5A3SweAH>

[31] Daenu Song, Jing Liang, Amirreza Payandeh, Amir Hossain Raj, Xuesu Xiao, and Dinesh Manocha. 2025. VLM-Social-Nav: Socially Aware Robot Navigation Through Scoring Using Vision-Language Models. *IEEE Robotics and Automation Letters* 10, 1 (2025), 508–515. doi:10.1109/LRA.2024.3511409

[32] Hongcheng Wang, Andy Guan Hong Chen, Xiaoqi Li, Mingdong Wu, and Hao Dong. 2023. Find what you want: Learning demand-conditioned object attribute space for demand-driven navigation. *Advances in Neural Information Processing Systems* 36 (2023), 16353–16366.

[33] Hongcheng Wang, Peiqi Liu, Wenze Cai, Mingdong Wu, Zhengyu Qian, and Hao Dong. 2024. MO-DDN: A Coarse-to-Fine Attribute-based Exploration Agent for Multi-object Demand-driven Navigation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[34] Peng Wang, Shuai Bai, Sinan Tan, Shiji Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191* (2024).

[35] Tian Wang, Junning Fan, and Pai Zheng. 2024. An LLM-based vision and language cobot navigation approach for Human-centric Smart Manufacturing. *Journal of Manufacturing Systems* 75 (Aug. 2024), 299–305. doi:10.1016/j.jmsy.2024.04.020 Publisher Copyright: © 2024 The Society of Manufacturing Engineers.

[36] Xudong Wang, Weihong Ren, Xi'ai Chen, Huijie Fan, Yandong Tang, and Zhi Han. 2024. Uni-YOLO: Vision-Language Model-Guided YOLO for Robust and Fast Universal Detection in the Open World. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 1991–2000.

[37] Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. 2018. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 37–53.

[38] Peter C Wason and J St BT Evans. 1974. Dual processes in reasoning? *Cognition* 3, 2 (1974), 141–154.

[39] Jiateng Wei, Quan Lu, Ning Jiang, Siqi Li, Jingyang Xiang, Jun Chen, and Yong Liu. 2024. Structured Optimal Brain Pruning for Large Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 13991–14007.

[40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[41] Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. 2024. Voronav: Voronoi-based zero-shot object navigation with large language model. *arXiv preprint arXiv:2401.02695* (2024).

[42] Jingyang Xiang and Sai Qian Zhang. 2024. DFRot: Achieving Outlier-Free and Massive Activation-Free for Rotated LLMs with Refined Rotation. *arXiv preprint arXiv:2412.00648* (2024).

[43] Zecheng Yin, † Chonghao Cheng, Yinghong Liao, Zhihao Yuan, Shuguang Cui, and Zhen Li. 2024. Navigation with VLM framework: Go to Any Language. *ArXiv abs/2410.02787* (2024). <https://api.semanticscholar.org/CorpusID:273163041>

[44] Lu Yue, Dongliang Zhou, Liang Xie, Feitian Zhang, Ye Yan, and Erwei Yin. 2024. Safe-vln: Collision avoidance for vision-and-language navigation of autonomous robots operating in continuous environments. *IEEE Robotics and Automation Letters* (2024).

[45] Zhaohuan Zhan, Lisha Yu, Sijie Yu, and Guang Tan. 2024. Mc-gpt: Empowering vision-and-language navigation with memory map and reasoning chains. *arXiv preprint arXiv:2405.10620* (2024).

[46] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. 2024. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852* (2024).

[47] Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Peng-wei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. 2025. MapNav: A Novel Memory Representation via Annotated Semantic Maps for VLM-based Vision-and-Language Navigation. *ArXiv abs/2502.13451* (2025). <https://api.semanticscholar.org/CorpusID:276449634>

[48] Yue Zhang, Ziqiao Ma, Jialu Li, Yanyuan Qiao, Zun Wang, Joyce Chai, Qi Wu, Mohit Bansal, and Parisa Kordjamshidi. 2024. Vision-and-Language Navigation Today and Tomorrow: A Survey in the Era of Foundation Models. *ArXiv abs/2407.07035* (2024). <https://api.semanticscholar.org/CorpusID:271064503>

[49] Ying Zhang, Peng Zhang, Mincong Huang, Jingyang Xiang, Yujie Wang, Chao Wang, Yineng Zhang, Lei Yu, Chuan Liu, and Wei Lin. 2024. Qqq: Quality quattuor-bit quantization for large language models. *arXiv preprint arXiv:2406.09904* (2024).

[50] Qianfan Zhao, Lu Zhang, Bin He, Hong Qiao, and Zhiyong Liu. 2023. Zero-shot object goal visual navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025–2031.

[51] Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. 2024. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*. Springer, 260–278.

[52] Gengze Zhou, Yicong Hong, and Qi Wu. 2024. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 7641–7649.