2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

BINDER: Instantly Adaptive Mobile Manipulation with Open-Vocabulary Commands

Seongwon Cho*, Daechul Ahn*, Donghyun Shin, Hyeonbeom Choi, San Kim, Jonghyun Choi†

***Abstract*— Open-vocabulary mobile manipulation (OVMM) requires robots to follow language instructions, navigate, and manipulate while updating their world representation as the environment changes dynamically. However, most prior works update their world representation only at discrete milestones, such as waypoints or the end of an action step. Such sparse updates leave robots with limited awareness between updates, causing missed objects, delayed error detection, and slower replanning. To address this limitation, we propose BINDER (Bridging INstant and DEliberative Reasoning), a dual-process framework that separates strategic planning from continuous environmental monitoring. BINDER combines a Deliberative Response Module (DRM, a multimodal LLM for task planning) with an Instant Response Module (IRM, a Video-LLM for continuous monitoring). The DRM handles strategic planning through structured 3D scene updates and guides the IRM's focus, while the IRM processes video streams to update memory, proactively adjust actions, and trigger replanning when needed. This bidirectional coordination ensures continuous awareness without costly updates, enabling reliable and robust operation under dynamic conditions. We evaluate BINDER in three real-world environments where objects are moved during execution and show that it achieves substantially higher success rates and efficiency than state-of-the-art baselines, confirming its effectiveness for real-world deployment.**

I. Introduction

Open-Vocabulary Mobile Manipulation (OVMM) aims to enable robots to navigate unknown environments and manipulate objects based on open-vocabulary instructions [1], [2]. In real-world settings such as homes and offices, robots must cope with dynamic changes like object relocation and human movement, requiring both strategic planning and continuous monitoring. While prior approaches operated in fixed, pre-scanned environments without considering such changes [3], [4], [5], recent approaches incorporate environmental feedback through voxel maps [6], scene graphs [7], [8], and vision–language models for closed-loop reasoning [9].

However, these approaches rely on *intermittent scene perception*, leaving robots with limited awareness of environmental changes between updates. Because 3D semantic reconstruction is computationally expensive, environmental representations—whether voxel maps [6], [9] or scene graphs [7], [8], [4], or implicit or object-centric maps [5], [9], [10]—are refreshed only at *discrete intervals* [6], [7], [9], [8]. Even approaches employing powerful task planners (*e.g.*, GPT [7], [9]) remain limited by intermittent perception, as their reasoning may rely on outdated scene information.

All authors are with Seoul National University, Seoul 08826, Republic of Korea. * indicates equal contribution. †JC is with ECE, ASRI and IPAI in SNU and a corresponding author (jonghyunchoi@snu.ac.kr).

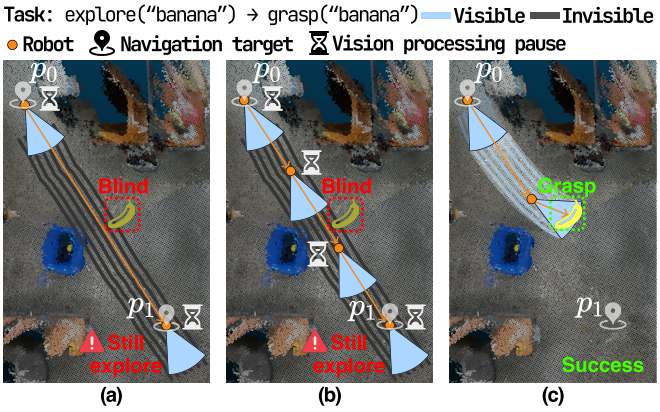

Fig. 1: **Limitations of existing OVMM approaches and the proposed BINDER.** Robots search for a banana while moving from navigation target p_0 to p_1. **(a)** Sparse-update approaches refresh perception only at navigation targets, leaving robots unaware during traversal and causing missed objects en-route. **(b)** Frequent updates at intermediate waypoints improve environmental awareness but require repeated vision processing pauses, causing inefficiency and still leaving perception gaps. **(c)** BINDER maintains continuous visual awareness en-route, enabling opportunistic detections and task execution without extra vision processing pauses.

Consider a robot searching for 'banana' as it travels from p_0 to p_1, as illustrated in Fig. 1. Even if the object lies directly on its path, approaches that update 3D semantic scenes only at navigation targets or after sub-actions (*e.g.*, grasping/placing) can miss this opportunity (Fig. 1-(a)). More frequent updates—at intermediate checkpoints [6] or during frontier expansions [11]—can still miss changes between reconstruction intervals (Fig. 1-(b)). This temporal unawareness, inherent to discrete-update approaches, triggers a chain of inefficiencies. Robots might ignore clearly visible objects while exploring, and during manipulation, they can fail to notice minor shifts that escalate into grasp failures, trajectory deviations, collisions, and task breakdowns.

Since 3D semantic scene reconstruction can take tens of seconds to minutes per update depending on scene complexity [7], [9], robots face an unsatisfactory trade-off: either pause frequently for accurate scene updates—delaying task completion—or continue moving with potentially outdated spatial information—risking critical oversights. While fast geometric reconstruction algorithms exist [12], [13], scaling them to the semantic level remains computationally prohibitive. To address this computational constraint, we argue

that instead of relying solely on monolithic 3D reconstruction, a heterogeneous perception strategy can offer a practical alternative by exploiting the complementary strengths of different sensing modalities: video streams provide continuous semantic awareness and detect salient environmental changes, while 3D reconstruction delivers the precise geometric information essential for OVMM task planning. By separating semantic monitoring from geometric perception, robots can maintain environmental awareness through video stream analysis while reserving computationally intensive 3D reconstruction for OVMM task planning.

To this end, we propose BINDER (**B**ridging **IN**stant and **DE**liberative **R**easoning), a dual-process framework inspired by cognitive theories [14], [15] that describe how humans navigate complex environments through fast, automatic monitoring (System 1) and slow, deliberative reasoning (System 2). Our framework operationalizes this cognitive division through two distinct modules (Fig. 2). The Instant Response Module (IRM) relies on a Video-LLM [16] to continuously process video streams, enabling opportunistic interventions during navigation and manipulation. Meanwhile, Deliberative Response Module (DRM) performs strategic planning using 3D semantic scene representations, which update upon navigation targets or when triggered by the IRM.

Furthermore, to enable mutual enhancement between these modules, we propose a bidirectional coordination method. Specifically, the DRM guides the IRM's monitoring attention based on current task context—whether navigating, searching, or manipulating—ensuring situation-appropriate monitoring, while the IRM provides environmental observations that enable context-aware planning and, when necessary, trigger immediate 3D reconstruction and replanning by the DRM (Sec. III-B). This heterogeneous perception strategy—combining scheduled reconstruction at navigation targets with on-demand reconstruction from video analysis—addresses the trade-off between temporal awareness and spatial precision that limits monolithic approaches.

We evaluate BINDER through extensive experiments across three real-world environments featuring diverse dynamic scenarios. When tested with dynamically appearing/disappearing objects and changing receptacles, BINDER demonstrates several key capabilities: immediate grasp correction during manipulation, early failure detection through temporal cues, opportunistic replanning when detecting targets mid-navigation, and dynamic task reordering based on environmental changes. Compared to state-of-the-art baselines [3], [6], [7], our approach shows significant improvements in handling dynamic situations—validating its potential for real-world OVMM deployment.

We summarize our contributions as follows:

- We identify intermittent scene perception as a limitation of current OVMM systems and propose BINDER, a dual-process framework that decouples continuous video monitoring from selective 3D reconstruction.
- We develop a bidirectional coordination mechanism enabling the IRM to trigger on-demand 3D updates while the DRM guides task-aware monitoring.
- We demonstrate through real-world experiments that BINDER effectively handles dynamic scenarios, significantly improving success rates and reducing task completion time compared to state-of-the-art baselines.

II. Related Work

A. Open Vocabulary Mobile Manipulation

Open-vocabulary mobile manipulation (OVMM) combines navigation, manipulation, and language understanding over extended horizons in dynamic environments. Two main paradigms exist: end-to-end vision–language–action (VLA) models [17], [18], [19], [20] capture rich multimodal correlations but suffer from high computational cost and limited long-horizon scalability, while modular pipelines [1], [3], [6], [7] decompose tasks into separate components using LLMs and VLMs [21], [22], [23] but accumulate errors over time. Unlike these approaches, we address intermittent scene perception by decoupling continuous video monitoring from selective 3D reconstruction.

B. Closed-Loop Recovery in Robotic Systems

Recent work incorporates closed-loop recovery to reduce cascading errors. COME-Robot [9] uses GPT-4V for iterative replanning, while RACER [24] employs supervisor-actor loops for feedback. However, these systems assess outcomes only at the keyframe or action-completion level, constraining responsiveness. Our IRM issues continue/adjust/replan signals during execution for timely corrections.

C. Scene Representations for OVMM

Robust scene representations enable OVMM by preserving object-level semantics and relations for long-horizon reasoning. Graph-based methods fuse multi-view evidence and support scalable queries: ConceptGraphs [25] builds an open-vocabulary scene graph; HOV-SG [4] adds a floor–room–object hierarchy for large-scale, multi-floor navigation; and DovSG [7] performs local, in-place 3D updates during interaction without full reconstruction. Voxel/field methods encode language-conditioned 3D maps: CLIP-Fields [5] enables continuous queries via implicit fields; VLMaps [22] grounds features in explicit voxel grids for language-driven navigation; and DynaMem [6] introduces efficient updates for long horizons. Yet all rely on discrete refreshes, so mid-execution changes can be missed and maps drift. In contrast, our dual-process design maintains continuous awareness and triggers 3D updates when needed.

III. Approach

OVMM in dynamic settings demands continuous perception and adaptive planning to handle appearing/relocating objects and to monitor/correct manipulation errors. Yet prior systems use *intermittent scene perception* (limited by compute constraints), leaving robots with limited awareness between discrete updates. BINDER is a dual-process framework that decouples strategic planning from continuous monitoring, delivering strong reasoning with real-time environmental awareness under dynamic conditions.

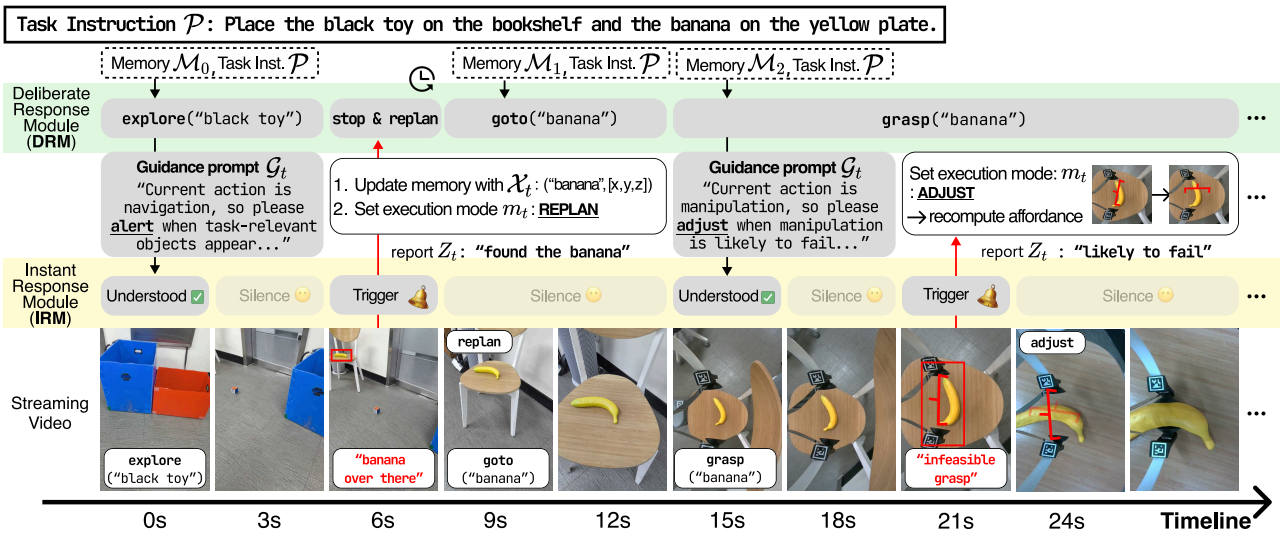

Fig. 2: **Illustration of dual-process reasoning in BINDER.** Our proposed framework, BINDER, consists of two modules operating in parallel: *Deliberative Response Module (DRM)* and *Instant Response Module (IRM)*. Based on the task instruction (inst.) and memory, DRM issues high-level actions (*e.g.*, `explore(''black toy'')`) and guides IRM's attention. In parallel, IRM continuously monitors the video stream in the background. When a task-relevant event occurs—such as opportunistically detecting the task-relevant object (6s) or diagnosing a grasp failure (21s)—IRM immediately generates a report, prompting DRM to replan for navigation or adjust the grasp for manipulation. This bidirectional coordination enables both continuous responsiveness and adaptive planning, addressing the intermittent scene perception of prior OVMMs.

A. BINDER: *Dual-Process Framework for OVMM*

Existing approaches for OVMM reveal a fundamental limitation: they apply the same computationally expensive 3D semantic scene reconstruction for all perception tasks, creating an unnecessary trade-off between awareness and efficiency. Frequent updates ensure awareness but degrade efficiency, while sparse updates [6], [7], [9] maintain speed but miss critical changes.

Dual-process architecture. We posit that this trade-off stems from treating all perception tasks as equally demanding: previous approaches apply computationally expensive 3D reconstruction uniformly, without distinguishing between tasks that require geometric precision and those that do not. While planning tasks necessarily require precise 3D geometry for manipulation and navigation decisions, we argue that monitoring tasks—detecting new objects or environmental changes—can be effectively handled through continuous video analysis, avoiding costly reconstruction overhead during navigation without sacrificing environmental awareness. This natural division between computationally-intensive planning and relatively lightweight monitoring parallels how humans navigate complex environments—through both fast, automatic monitoring (System 1) and slow, deliberative response (System 2), as described in dual-process theories [14], [15]. Inspired by this well-established cognitive architecture, we decouple continuous environmental monitoring from costly periodic 3D reconstruction.

To operationalize this separation, we introduce two specialized modules as illustrated in Fig. 2: the **Instant Response Module (IRM)** powered by a Video-LLM maintains continuous environmental monitoring through video streams during execution, analogous to System 1's automatic processing; while the **Deliberative Response Module (DRM)** performs strategic planning using 3D semantic scene representations at navigation targets, mirroring System 2's deliberative reasoning. This architectural division allows each module to optimize for its primary objective—the IRM for temporal responsiveness, the DRM for spatial precision—enabling both continuous awareness and sophisticated reasoning without the compromises of current monolithic approaches.

B. *Dual-Process Modules*

DRM-IRM coordination. While the DRM and IRM serve distinct roles, effective OVMM requires coordination between continuous monitoring and strategic planning (Fig. 2). We achieve this through bidirectional information flow between the modules, as illustrated in Fig. 3. The DRM provides task-specific guidance prompt $\mathcal{G}_t$ that dynamically reconfigures the IRM's attention—shifting from "identify task-relevant objects and receptacles" during exploration to "monitor gripper-object alignment and placement stability" during manipulation. Conversely, the IRM supplies continuous environmental feedback: during exploration, newly detected or relocated objects asynchronously update the object registry $\mathcal{R}_t$ without full map reconstruction; during manipulation, it enables reactive control through immediate local corrections or escalation to the DRM when local adjustments fail. This bidirectional coordination ensures the system remains both deliberate and responsive.

DRM. To implement the planning component of this coordination, we employ a multimodal LLM as the DRM, which operates at navigation targets or when triggered by the

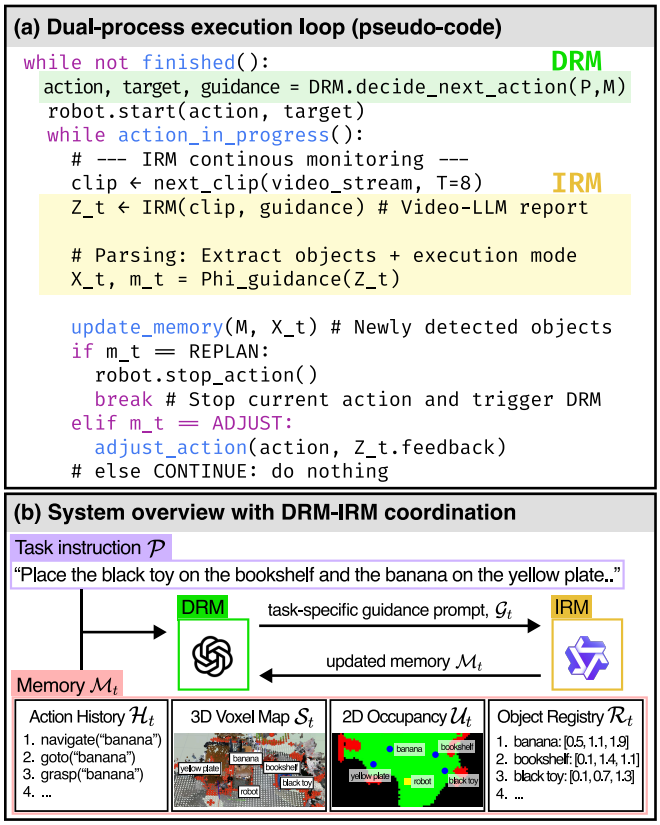

Fig. 3: **Flowchart of dual-process execution in BINDER.** (a) Pseudocode of the execution loop: the DRM issues high-level actions *and task-specific guidance*, while the IRM continuously monitors video and outputs execution modes (CONTINUE/ADJUST/REPLAN) and object updates that drive local corrections or trigger replanning. (b) System overview: the DRM uses task instructions and memory to generate plans and guidance, while the IRM monitors environmental changes to update memory status and trigger timely replanning under dynamic conditions.

IRM. Upon activation, the robot executes a `look_around` primitive to capture surrounding views and performs 3D semantic scene reconstruction following prior work [6]. This reconstruction updates the memory $\mathcal{M}_t$ that maintains: (1) a 3D semantic scene representation $\mathcal{S}_t$, *i.e.*, 3D voxel map, (2) a 2D occupancy projection $\mathcal{U}_t$ derived from $\mathcal{S}_t$ for effective spatial reasoning, encoding navigable areas, obstacles, and semantic labels, (3) action history $\mathcal{H}_t = \{a_1, ..., a_t\}$, and (4) an object registry $\mathcal{R}_t = \{(c_i, p_i)\}_{i=1}^{N_t}$ accumulating N_t discovered objects with category c_i and position $p_i = (x_i, y_i, z_i)$. Using the task instruction $\mathcal{P}$ and memory $\mathcal{M}_t$, the DRM generates planning decisions:

$$a_{t+1}, o_{t+1}, \mathcal{G}_{t+1} = \text{DRM}(\mathcal{P}, \mathcal{M}_t) \tag{1}$$

This yields three outputs: (1) next action $a_{t+1} \in \{\texttt{go\_to}, \texttt{explore}, \texttt{grasp}, \texttt{place}\}$, (2) target specification o_{t+1} (coordinates for `go_to`, locations for `explore`, or object/receptacle IDs for manipulation), and (3) task-specific guidance $\mathcal{G}_{t+1}$ that refocuses the IRM's attention for the upcoming phase. The robot's controller then executes the action-target pair (a_{t+1}, o_{t+1}).

IRM. For continuous perception during task execution, we employ a Video-LLM [16] as the IRM, enabling continuous environmental monitoring throughout navigation and manipulation. The Video-LLM processes video clips v_t (recent frames from the continuous stream) with task-specific guidance prompt $\mathcal{G}_t$ provided by the DRM to generate a structured language report Z_t that describes detected objects, task progress, and potential issues:

$$Z_t = \text{Video-LLM}(v_t, \mathcal{G}_t). \tag{2}$$

Since the Video-LLM generates free-form language outputs whose structure varies with task context, we employ a guidance-conditioned parsing module $\Phi_{\mathcal{G}_t}$ (detailed procedures are in Sec. III-C) to extract actionable information:

$$\Phi_{\mathcal{G}_t} : Z_t \mapsto (\mathcal{X}_t, m_t), \tag{3}$$

where detected object information $\mathcal{X}_t$ contains object category and position pairs (c_i, p_i) used to update the object registry $\mathcal{R}_t$, and execution mode $m_t \in \{\text{CONTINUE}, \text{ADJUST}, \text{REPLAN}\}$ specifies the appropriate robot behavior. Here, $\mathcal{G}_t$ is a text prompt from the DRM that reconfigures $\Phi_{\mathcal{G}_t}$'s extraction target: during navigation it instructs $\Phi_{\mathcal{G}_t}$ to match object mentions in Z_t against task targets via embedding similarity to populate $\mathcal{X}_t$; during manipulation it instead shifts to assessing grasp quality and object stability cues to determine m_t.

The execution mode m_t enables three levels of adaptation: (i) CONTINUE maintains current execution when no issues are detected, (ii) ADJUST applies immediate corrections for minor deviations (*e.g.*, grasp refinement), and (iii) REPLAN triggers DRM invocation for 3D semantic scene reconstruction and strategy revision when crucial environmental changes occur (*e.g.*, target object appearing unexpectedly). This design ensures the IRM functions as an effective continuous monitor—detecting opportunities and threats between discrete 3D updates—while maintaining computational efficiency through selective DRM activation.

C. Task Execution Strategies

Exploration and navigation. Our exploration strategy builds upon the value-guided frontier selection method from DynaMem [6], which combines temporal and semantic value maps to compute exploration values $V_i = V_i^T + V_i^S$, where V_i^T prioritizes least-recently-seen areas and V_i^S measures semantic similarity to target objects. We enhance this approach through DRM-based intelligent selection, as pure value-based ranking may overlook contextual cues visible from the current position. Specifically, the DRM evaluates the top-k frontier candidates (empirically set to $k = 3$, balancing computational efficiency with coverage):

$$f^* = \underset{f \in \text{top-}k(V)}{\arg\max}\ \text{DRM}(I_f, \mathcal{P}, \mathcal{M}_t) \tag{4}$$

where I_f is the image captured by orienting the camera toward frontier f from the current position. This enables the

DRM to leverage visual context alongside task instruction $\mathcal{P}$ and memory $\mathcal{M}_t$ for context-aware destination selection. Once f^* is determined, the robot generates a trajectory using A* path planning [26] and begins navigation.

During transit, our IRM continuously monitors the environment—a key departure from DynaMem's fixed waypoint pausing—enabling opportunistic replanning when needed. The IRM processes the video stream to detect task-relevant objects and environmental changes, generating reports Z_t that describe the current scene. The guidance-conditioned parsing module $\Phi_{\mathcal{G}_t}$ extracts actionable information from these free-form language outputs, simultaneously determining both detected objects $\mathcal{X}_t$ and execution mode m_t. When $\Phi_{\mathcal{G}_t}$ identifies object mentions in Z_t matching task targets from $\mathcal{P}$ (via embedding similarity [27]), it triggers an asynchronous localization step to compute precise 3D positions for $\mathcal{X}_t$. OWL-ViT [28] detects the 2D bounding box B_i of each matched object i, which is then lifted to 3D position p_i using RGB-D projection:

$$p_i = \text{median}\{T_t^{\text{cam}\to\text{world}}(u, v, d(u, v)) : (u, v) \in B_i\}, \quad (5)$$

where $T_t^{\text{cam}\to\text{world}}$ denotes the standard camera-to-world transformation using RGB-D measurements and camera parameters at time t, following [29].

The median operation within B_i improves robustness to depth noise and background pixels. The resulting positions form $\mathcal{X}_t$ and are merged into $\mathcal{R}_{t+1}$ without interrupting motion. Meanwhile, $\Phi_{\mathcal{G}_t}$ determines m_t based on the overall scene context—typically returning CONTINUE but opportunistically selecting REPLAN when detecting nearer targets or obstacles that invalidate the current plan.

Manipulation. Our manipulation approach builds on OK-Robot [3], which combines AnyGrasp [30] with LangSAM [31] filtering for grasping and uses point cloud-based height computation for placing. We extend this framework with event-triggered visual feedback through the IRM, enabling reactive adjustments in dynamic environments.

During manipulation, $\mathcal{X}_t$ typically remains empty as objects are already localized, while $\Phi_{\mathcal{G}_t}$ focuses on extracting the execution mode m_t from the IRM's reports Z_t. Similar to the navigation phase, $\Phi_{\mathcal{G}_t}$ analyzes Z_t using embedding similarity [27] to identify manipulation-specific cues such as grasp quality indicators, object stability assessments, and environmental changes. When the IRM detects misalignments during grasping (m_t = ADJUST), we perform local grasp recomputation: AnyGrasp generates new candidates within a constrained region around the current target, selecting the highest-scoring pose with minimal reorientation. This enables immediate corrections without costly full replanning overhead. For placing, the IRM monitors object stability and receptacle availability continuously throughout execution. When issues arise, $\Phi_{\mathcal{G}_t}$ returns m_t = ADJUST, triggering height recomputation or alternative receptacle selection based on the problem detected. Critical failures—such as repeated grasp failures or unavailable receptacles—result in m_t = REPLAN, engaging the DRM for strategic revision. The DRM then performs a targeted 3D reconstruction update and generates alternative strategies, such as selecting different objects or modifying task sequences.

This hierarchical error recovery, progressing from local adjustments to strategic replanning, ensures robust manipulation in scenarios where OK-Robot's open-loop approach would require manual intervention.

(a) Mobile manipulation tasks (b) Manipulation task

Fig. 4: **Experimental environments.** (a) **Mobile manipulation:** We evaluate BINDER in a controlled office and two real-world sites with various complexity; objects and receptacles form diverse scenes under identical configurations. (b) **Tabletop manipulation:** Base motion is limited to forward–backward; three objects and receptacles are on a table; 30 trials per condition to isolate the IRM's contribution.

IV. EXPERIMENTS

We evaluate BINDER in real-world environments to assess its robustness against environmental changes introduced during task execution and its effectiveness on long-horizon multi-object tasks compared to baselines.

A. Experimental Settings

Robot setups. We use a Hello Robot Stretch SE3 as our mobile robot platform [32] equipped with RGB-D (RealSense D435i) camera for perception.

Implementation details. Our system builds upon DynaMem's 3D voxel representation [6]. We employ GPT-4o as our DRM and Qwen2.5VL (3B) [16] as our Video-LLM. The Video-LLM processes 1-second video clips at 8 fps with an inference time of about 0.5 seconds per clip, requiring approximately 12 GB of VRAM on a single NVIDIA A6000 (48 GB) without interfering with the navigation, perception, or control pipelines sharing the same GPU.

B. Task Setup

Multi-step tasks in dynamic environments. Following previous work [9], we systematically evaluate multi-step task execution by defining three task categories with increasing complexity: **Task 1:** Single object → single receptacle. **Task 2:** Two objects → two receptacles. **Task 3:** Three objects → three receptacles. Experiments are conducted in three environments: a controlled office, a studio apartment, and a three-room apartment (Fig. 4-(a)). We evaluate all three task categories in the office (40 trials each) and focus on Task 3 in the homes (10 trials each), using identical code across all settings. We vary three key factors: (1) **Scenes**: Each

TABLE I: **Real-world office environment evaluation across Task 1–3.** The three task categories contain 1, 2, and 3 object→receptacle subtasks respectively, testing increasing difficulty from single-step to long-horizon execution. We report four metrics: SR for full completion, PSR for subgoal progress, SPL for efficiency, and PSPL for partially completed tasks. For Task 1, SPL and PSPL are equivalent.

Method	Task 1 (1 subtask)		Task 2 (2 subtasks)				Task 3 (3 subtasks)			
	SR ↑	SPL ↑	SR ↑	PSR ↑	SPL ↑	PSPL ↑	SR ↑	PSR ↑	SPL ↑	PSPL ↑
OK-Robot	0.23	0.20	0.05	0.19	0.05	0.19	0.03	0.27	0.03	0.13
DovSG	0.28	0.25	0.13	0.23	0.13	0.16	0.08	0.36	0.05	0.23
DynaMem	0.60	0.42	0.43	0.71	0.29	0.40	0.15	0.62	0.09	0.47
BINDER (**Ours**)	**0.93**	**0.69**	**0.78**	**0.88**	**0.68**	**0.71**	**0.63**	**0.85**	**0.48**	**0.72**

TABLE II: **Real-world two home environments evaluation on Task 3.** Results are shown for a studio apartment and a three-room apartment. We report Success Rate (SR) and Success weighted by Path Length (SPL).

Method	Studio		3-Room	
	SR	SPL	SR	SPL
OK-Robot	0.20	0.10	0.20	0.18
DovSG	0.20	0.20	0.40	0.33
DynaMem	0.30	0.15	0.50	0.36
BINDER (**Ours**)	**0.70**	**0.57**	**0.80**	**0.62**

TABLE III: **Ablation study of proposed dual-process components.** We evaluate four variants of BINDER. Experiments are conducted on Task 3 (three objects → three receptacles) in the office environment with 10 trials per variant. Metrics include SR, PSR, and SPL.

Configuration	Components		SR	PSR	SPL	PSPL
	DRM	IRM				
Neither	✘	✘	0.30	0.43	0.22	0.28
DRM only	✔	✘	0.40	0.57	0.38	0.50
IRM only	✘	✔	0.60	0.83	0.47	0.59
DRM+IRM (BINDER)	✔	✔	**0.80**	**0.93**	**0.63**	**0.82**

unique object-receptacle arrangement defines a distinct initial state, maintained consistently across methods. (2) **Queries**: Task instructions specify randomly sampled object-receptacle pairs (1–3 pairs based on task category). (3) **Dynamics**: We introduce two position perturbations per query—typically moving objects during approach and receptacles during transport—simulating real-world dynamics.

Metrics. Following prior work [33], we report *Success Rate* (SR) for full completion, *Partial Success Rate* (PSR) for completed subgoals, and *Success weighted by Path Length* (SPL) for path efficiency relative to expert demonstrations from voxel-derived occupancy grids. For multi-subgoal tasks, we introduce *Partial Success weighted by Path Length* (PSPL), extending SPL by averaging efficiency over completed subgoals rather than requiring full task completion.

Baselines. We compare BINDER with three strong baselines for the OVMM task: OK-Robot [3], DynaMem [6], and DovSG [7]. Since OK-Robot and DynaMem are designed for single object-receptacle tasks, we extend them to multi-object settings by sequentially executing each object-receptacle pair in the instruction query. OK-Robot and DovSG require global pre-scanning to build environment maps, with performance highly sensitive to scan quality. To ensure fair comparison, we perform five scans per scene and report results using the best-quality map. For DovSG, we replace the Stretch SE3's default D435i camera with a RealSense D455 RGB-D camera following their original implementation, as ACE-based pose estimation was unreliable with the default hardware.

C. Quantitative Results

Quantitative results in office environment. Table I demonstrates BINDER's consistent superiority across all task complexities. In single-object tasks (Task 1), BINDER achieves an SR of 0.93 compared to 0.60 for the best baseline (DynaMem). This advantage amplifies with task complexity: in Task 2 (two subtasks), BINDER reaches 0.78 versus DynaMem's 0.43, and in Task 3 (three subtasks), maintains 0.63—over 4× higher than any baseline. Moreover, BINDER excels in partial task completion, achieving a PSR of 0.85 in Task 3 compared to DynaMem's 0.62, indicating robust recovery from individual failures.

These improvements stem from our dual-process design: the IRM enables real-time corrections and dynamic adjustments through continuous monitoring, while the DRM ensures efficient exploration via top-k frontier evaluation (Sec. III-C). This is reflected in improved SPL/PSPL metrics and shorter trajectories (Table V). Our heterogeneous compute strategy effectively resolves the fundamental trade-off in existing approaches—OK-Robot and DovSG suffer from stale perception, while DynaMem incurs costly reconstruction pauses. By decoupling strategic planning (DRM) from lightweight monitoring (IRM), BINDER achieves temporal continuity and spatial precision, translating to higher success rates and efficiency under dynamic conditions.

Quantitative results in home environments. Table II shows BINDER's clear advantages in both home settings, with success rates improving by roughly 0.4 in the one-room studio and 0.3 in the three-room layout compared to DynaMem. These gains stem from the IRM's opportunistic

TABLE IV: **Effect of IRM on tabletop manipulation tasks.** We compare a baseline without IRM against our system with IRM enabled, averaging results over 30 manipulation trials with restricted base motion. Metrics are overall success rate and average execution time, highlighting how IRM improves reliability with minimal time overhead.

Configuration	SR ↑	Avg. Time (sec.) ↓
DRM only	0.53	**61**
DRM + IRM (BINDER)	**0.77**	66

TABLE V: Comparison of completion time and trajectory length for Task 3 in the office environment. We report average completion time (minutes) and path length (meters), computed over successful trials from 40 runs. Results compare our method against DynaMem [6].

Method	Avg. Time (min.) ↓	Avg. Length (m) ↓
DynaMem [6]	33.83	39.60
BINDER	**21.90**	**28.35**

target detection during navigation and the DRM's timely replanning for changed receptacle states. Efficiency metrics mirror these improvements: SPL improves by $\sim 3.8\times$ in the studio and $\sim 1.7\times$ in the 3-room apartment, confirming that our dual-process design enhances both reliability and path efficiency under dynamic conditions.

D. In-depth analysis

Ablation study. Table III evaluates four variants on Task 3 (10 trials each): *DRM + IRM* (full system), *DRM only* (no continuous monitoring), *IRM only* (no DRM guidance), and *Neither* (discrete updates only). *DRM only* shows modest reliability gains (SR: $0.30 \rightarrow 0.40$, PSR: $0.43 \rightarrow 0.57$) but substantial path efficiency improvements (SPL: $0.22 \rightarrow 0.38$, PSPL: $0.28 \rightarrow 0.50$) via contextual frontier evaluation avoiding unnecessary detours. *IRM only* improves reliability through continuous perception (SR: $0.30 \rightarrow 0.60$, PSR: $0.43 \rightarrow 0.83$), but lacks task-specific focus, relying on generic descriptions without DRM guidance.

The combined *DRM + IRM* achieves the highest performance (SR: 0.80, PSR: 0.93, SPL: 0.63, PSPL: 0.82). This confirms our dual-process synergy: the IRM provides temporal continuity through opportunistic detections and micro-corrections, while the DRM supplies task-specific guidance and geometry-based replanning, outperforming either alone.

Effect of IRM on manipulation. To further evaluate the IRM's contribution, we conduct a tabletop study restricting base motion to forward–backward (Fig. 4b). Using diverse objects and receptacles, we run 30 manipulation trials with randomly sampled object–receptacle pairs, comparing with and without the IRM. Table IV shows the IRM increases success rate from 0.53 to 0.77 with minimal overhead (61s $\rightarrow$ 66s), primarily from gripper re-alignment during local adjustments. This demonstrates how continuous monitoring detects minor shifts to enable immediate corrections.

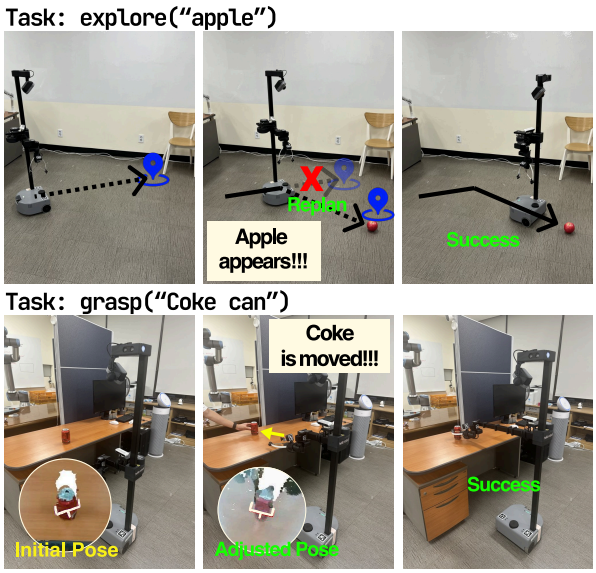

Fig. 5: **Qualitative examples of BINDER in dynamic environments.** Top: *Exploration.* An apple appears mid-navigation; the IRM detects it and triggers DRM replanning, leading to efficient target acquisition. Bottom: *Manipulation.* A Coke can is displaced during grasp; the IRM detects the shift, adjusts the pose, and completes the action without full replanning. Together, DRM and IRM maintain temporal awareness and spatial precision under dynamic changes.

Completion time and path efficiency. Table V shows that despite additional modules (IRM and DRM), BINDER achieves faster execution and shorter trajectories than DynaMem [6], which shows the highest SR among baselines in the office environment. This aligns with our motivation (Sec. I): while prior systems repeatedly pause for map updates, BINDER's IRM provides continuous monitoring, triggering 3D reconstruction updates only when exploring new areas. Object localization occurs directly through the IRM during navigation, yielding smoother execution. DynaMem [6] trajectories are approximately $1.4\times$ longer, as robots travel additional distance before recognizing objects.

E. Qualitative Results

We illustrate in Fig. 5 how BINDER adapts to dynamic changes. In the top example, an apple appears mid-navigation; the IRM detects it and triggers a REPLAN, allowing the DRM to update the plan and grasp the object efficiently. In the bottom example, a Coke can is physically displaced during the grasp attempt; the IRM issues an ADJUST event, enabling rapid re-alignment and successful completion without restarting the manipulation pipeline. These examples show how continuous monitoring and deliberative replanning work together to maintain temporal awareness and spatial precision in real-world execution.

V. Conclusion

We presented BINDER, a dual-process framework that addresses OVMM's core limitation—*intermittent scene perception*—by decoupling continuous video monitoring (IRM) from selective 3D reconstruction and planning (DRM) via bidirectional coordination. Across an office and two real-world homes, BINDER consistently improved metrics and reduced time and path length over baselines. Ablations confirmed the roles of DRM and IRM, and tabletop studies showed higher manipulation reliability. By maintaining continuous awareness between updates while preserving geometry-accurate planning at key decision points, BINDER advances OVMM toward robust real-world deployment.

Acknowledgements

This work was partly supported by the IITP grants (RS-2022-II220077, RS-2022-II220113, RS-2022-II220959, RS-2022-II220871, RS-2021-II211343 (SNU AI), RS-2025-25442338 (AI Star Fellowship-SNU)) funded by the Korea government (MSIT), grants (RS-2025-25462891 (US-KOR BARI), RS-2025-25453780) funded by MOTIR, a grant of Korean ARPA-H Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (RS-2025-25424639), and the BK21 FOUR program, SNU in 2025, and the Artificial Intelligence Industrial Convergence Cluster Development Project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City.

References

[1] S. Yenamandra, A. Ramachandran, K. Yadav, A. S. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. Clegg, J. M. Turner, Z. Kira, M. Savva, A. X. Chang, D. S. Chaplot, D. Batra, R. Mottaghi, Y. Bisk, and C. Paxton, "Homerobot: Open-vocabulary mobile manipulation," in *CoRL*, 2023.

[2] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, 2023.

[3] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto, "Ok-robot: What really matters in integrating open-knowledge models for robotics," in *RSS*, 2024.

[4] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation," in *RSS*, 2024.

[5] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, "Clip-fields: Weakly supervised semantic fields for robotic memory," in *RSS*, 2023.

[6] P. Liu, Z. Guo, M. Warke, S. Chintala, C. Paxton, N. M. M. Shafiullah, and L. Pinto, "Dynamem: Online dynamic spatio-semantic memory for open world mobile manipulation," in *ICRA*, 2025.

[7] Z. Yan, S. Li, Z. Wang, L. Wu, H. Wang, J. Zhu, L. Chen, and J. Liu, "Dynamic open-vocabulary 3d scene graphs for long-term language-guided mobile manipulation," *RA-L*, vol. 10, no. 5, pp. 4252–4259, 2025.

[8] M. Mohammadi, D. Honerkamp, M. Büchner, M. Cassinelli, T. Welschehold, F. Despinoy, I. Gilitschenski, and A. Valada, "More: Mobile manipulation rearrangement through grounded language reasoning," in *IROS*, 2025.

[9] P. Zhi, Z. Zhang, Y. Zhao, M. Han, Z. Zhang, Z. Li, Z. Jiao, B. Jia, and S. Huang, "Closed-loop open-vocabulary mobile manipulation with gpt-4v," in *ICRA*, 2025.

[10] D. Qiu, W. Ma, Z. Pan, H. Xiong, and J. Liang, "Open-vocabulary mobile manipulation in unseen dynamic environments with 3d semantic maps," *arXiv*, 2024.

[11] D. Honerkamp, M. Büchner, F. Despinoy, T. Welschehold, and A. Valada, "Language-grounded dynamic scene graphs for interactive object search with mobile manipulation," *RA-L*, 2024.

[12] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," *ISMAR*, pp. 127–136, 2011.

[13] F. Tosi, Y. Zhang, Z. Gong, E. Sandström, S. Mattoccia, M. R. Oswald, and M. Poggi, "How nerfs and 3d gaussian splatting are reshaping slam: a survey," *arXiv*, 2024.

[14] P. C. Wason and J. S. B. T. Evans, "Dual processes in reasoning?" *Cognition*, vol. 3, no. 2, pp. 141–154, 1974.

[15] D. Kahneman, *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux, 2011.

[16] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, "Qwen2.5-vl technical report," *arXiv*, 2025.

[17] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *CoRL*, 2023.

[18] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source vision-language-action model," in *CoRL*, 2024.

[19] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "π_0: A vision-language-action flow model for general robot control," *arXiv*, 2024.

[20] L. X. Shi, B. Ichter, M. Equi, L. Ke, K. Pertsch, Q. Vuong, J. Tanner, A. Walling, H. Wang, N. Fusai, A. Li-Bell, D. Driess, L. Groom, S. Levine, and C. Finn, "Hi robot: Open-ended instruction following with hierarchical vision-language-action models," in *ICML*, 2025.

[21] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *ICRA*, 2023.

[22] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *ICRA*, 2023.

[23] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler, "Open-vocabulary queryable scene representations for real world planning," in *ICRA*, 2023.

[24] Y. Dai, J. Lee, N. Fazeli, and J. Chai, "Racer: Rich language-guided failure recovery policies for imitation learning," in *ICRA*, 2025.

[25] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *ICRA*, 2024.

[26] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[27] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mpnet: Masked and permuted pre-training for language understanding," in *NeurIPS*, 2020.

[28] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, *et al.*, "Simple open-vocabulary object detection," in *ECCV*, 2022.

[29] B. Cheng, L. Sheng, S. Shi, M. Yang, and D. Xu, "Back-tracing representative points for voting-based 3d object detection in point clouds," in *CVPR*, 2021.

[30] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *T-RO*, 2023.

[31] L. Medeiros, "Lang-segment-anything: Sam with text prompt," https://github.com/luca-medeiros/lang-segment-anything, 2023.

[32] Hello Robot Inc., "Stretch se3 mobile manipulator robot," https://hello-robot.com/product, 2023, accessed: 2024-01-01.

[33] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks," in *CVPR*, 2020.