# Critical Analysis of Language-Grounded Dynamic Scene Graphs for Interactive Object Search with Mobile Manipulation (MoMa-LLM)

## Abstract

The integration of Large Language Models (LLMs) into robotic systems has catalyzed a paradigm shift in Embodied AI, promising agents capable of interpreting abstract human instructions and executing long-horizon tasks in unstructured environments. However, a significant "grounding gap" remains between the semantic reasoning capabilities of LLMs and the physical, geometric realities of robot perception. This seminar report provides an exhaustive critical analysis of "Language-Grounded Dynamic Scene Graphs for Interactive Object Search with Mobile Manipulation" (MoMa-LLM), presented by Honerkamp et al. The paper introduces a novel framework that bridges this gap by constructing open-vocabulary Dynamic 3D Scene Graphs (DSGs) in real-time as a robot explores an unknown environment. By interleaving a hierarchical scene representation—comprising rooms, objects, and a navigational Voronoi graph—with an object-centric action space, the authors propose a method for zero-shot, interactive object search (IOS) that outperforms conventional heuristics and learning-based baselines.

This report dissects the MoMa-LLM architecture, evaluating its three primary contributions: the scalable construction of dynamic scene graphs in unexplored settings, the structured knowledge extraction mechanism designed to interface with LLM planners, and the introduction of the Area Under the Efficiency Curve (AUC-E) metric for evaluating search efficiency independent of time budgets. The analysis situates MoMa-LLM within the broader trajectory of semantic navigation, contrasting it with precursor methods like SayPlan and Hydra, and evaluating its performance against recent critiques regarding low-level action costs and rearrangement capabilities found in subsequent literature (e.g., Inter-LLM, MORE). The review concludes that while MoMa-LLM establishes a robust standard for semantic grounding in exploration, its reliance on high-level semantic reasoning without tightly coupled low-level cost feedback presents limitations in complex, cluttered environments.

## 1. Introduction

### 1.1 The Evolution of Mobile Manipulation and Semantic Reasoning

The field of mobile manipulation stands at the confluence of navigation, perception, and physical interaction, representing one of the most challenging frontiers in autonomous robotics. Unlike passive navigation tasks, where a robot must merely traverse from a starting point to a goal coordinates, mobile manipulation requires the agent to physically alter its environment to achieve its objectives. This distinction is paramount in the context of domestic service robots, where the environment is not a static museum but a dynamic, clutter-filled living space. To function effectively in such domains, a robot must possess not only geometric awareness—knowing where obstacles are—but also semantic understanding—knowing what objects are and how they relate to one another.1

Historically, approaches to this problem have been bifurcated. Classical methods relied on geometric mapping and explicit task planning, often utilizing PDDL (Planning Domain Definition Language) domains that required rigorous, hand-coded definitions of state and transition rules. These systems offered guarantees of correctness but suffered from brittleness; they could not handle the "open-world" nature of human environments where new objects and unexpected configurations are the norm. Conversely, the deep learning revolution introduced end-to-end learning methods, specifically Reinforcement Learning (RL) and Imitation Learning (IL). While these methods demonstrated remarkable success in specific skills, they often struggled with long-horizon reasoning and generalization to completely unseen environments, as noted in evaluations of baselines like HIMOS.2

The recent advent of Large Language Models (LLMs) has offered a potential solution to the semantic reasoning bottleneck. LLMs, trained on vast corpora of internet text, possess an inherent "world model" encoded in their parameters. They understand that "breakfast" typically involves "cereal," "milk," and a "kitchen," and that "milk" is usually found inside a "fridge".1 This common-sense reasoning is precisely what classical planners lacked and what RL agents struggled to learn from sparse rewards. However, applying LLMs to robotics introduces the "grounding problem." An LLM operating on text does not inherently understand the robot's coordinate frame, the geometry of obstacles, or the temporal dynamics of a changing scene. It can hallucinate plausible-sounding but physically impossible plans, such as instructing a robot to "pick up the table" or "walk through the wall".4

### 1.2 The Interactive Object Search (IOS) Challenge

To benchmark progress in this domain, the robotics community has formulated the task of Interactive Object Search (IOS). This task serves as a canonical "grand challenge" for embodied intelligence. Unlike passive Object Navigation (ObjectNav), where an agent merely locates a visible target, IOS requires the agent to physically modify the environment—opening doors, cabinets, and drawers—to reveal occluded targets.1

This introduces a Partially Observable Markov Decision Process (POMDP) of extreme complexity. The state space includes not only the robot's pose and the map of the environment but also the articulation states of every manipulable object (e.g., is the drawer open or closed?) and the contents of every container. In realistic household environments, the search space is vast and unstructured. An agent tasked with "finding something for breakfast" must deduce that milk is likely in a fridge (requiring navigation to the kitchen and manipulation of the fridge door) or that cereal might be in a cabinet.5

The complexity is exacerbated by the "exploration-exploitation" dilemma in a semantic context. Traditional frontier-based exploration, which drives robots to the boundary of known space, is semantically blind. It is as likely to explore a bathroom as a kitchen when searching for food. To be efficient, an agent must use semantic priors to bias its exploration. MoMa-LLM addresses this by leveraging the LLM not just for task sequencing, but for *semantic exploration*—deciding which unknown area is most promising based on the task description.1

### 1.3 The MoMa-LLM Proposition

The paper under review, "Language-Grounded Dynamic Scene Graphs for Interactive Object Search with Mobile Manipulation" (MoMa-LLM), addresses these deficiencies by proposing the Dynamic 3D Scene Graph (DSG) as the mediating data structure between the robot's perception and the LLM's reasoning.1

The core hypothesis of MoMa-LLM is that raw perceptual data (point clouds, voxel grids) is too granular for LLMs, while simple object lists are too sparse, lacking topological context. The DSG serves as a "lingua franca," abstracting the environment into nodes (rooms, objects) and edges (spatial relationships) that can be textually encoded for the LLM. Crucially, this graph is built *online*. As the robot explores, it detects frontiers (unexplored areas) and updates the graph, allowing the LLM to reason about *where to look next* based on partial information.

The authors validate this approach through a rigorous set of experiments in the iGibson simulator and on a real-world Toyota HSR robot, comparing against state-of-the-art baselines like HIMOS (Hierarchical RL) and ESC (Commonsense Exploration).1 This report will critically analyze the methodology, experimental rigor, and theoretical contributions of this work, positioning it within the rapidly evolving landscape of language-guided robotics.

## 2. Related Work

To critically analyze MoMa-LLM, one must situate it within three intersecting fields: 3D Scene Representations, LLM-based Planning, and Semantic Search. The landscape of related work highlights both the novelty of MoMa-LLM and the specific gaps it attempts to fill.

### 2.1 3D Scene Graphs (3DSG) in Robotics

Scene graphs have evolved from static computer vision tools to dynamic robotic maps. A 3DSG hierarchically organizes a scene: a building contains rooms, rooms contain objects, and objects have attributes.7

Static vs. Dynamic Representations:

Early works, such as the seminal "3D Scene Graph" by Armeni et al., focused on offline generation from complete scans. These representations were comprehensive but static, unsuitable for robots operating in changing environments. The transition to dynamic scene graphs was marked by systems like Hydra (Hughes et al.) 4, which represents the state-of-the-art in real-time 3DSG construction. Hydra utilizes a mesh-based approach to abstract geometry into topology, constructing a "places" graph that represents navigable space.

The MoMa-LLM Divergence:

While MoMa-LLM draws inspiration from Hydra, it diverges significantly in its room segmentation strategy. Hydra relies heavily on geometric constrictions—detecting narrow passages or bottlenecks—to define room boundaries. This geometric approach fails in modern "open concept" architectural layouts where a kitchen and living room might flow into each other without a narrow hallway. MoMa-LLM introduces a probabilistic door-detection mechanism rooted in the Voronoi graph to separate regions.1 By integrating semantic cues (door detections) into the topological generation, MoMa-LLM argues for a more semantically robust segmentation than purely geometric methods.

### 2.2 Large Language Models for Task Planning

The utilization of LLMs for planning is characterized by the method of grounding—how the abstract text of the LLM is connected to the physical reality of the robot.

SayCan and Inner Monologue:

Early systems like SayCan attempted to ground LLMs via value functions derived from reinforcement learning policies. The LLM would generate a list of potential steps (e.g., "pick up apple"), and the RL policy would provide a "feasibility score" for each. While innovative, these systems are computationally expensive and typically struggle with long-horizon exploration in unknown maps because they require pre-trained policies for every conceivable action.9

SayPlan:

SayPlan represents the most direct comparator to MoMa-LLM. Like MoMa-LLM, SayPlan uses 3DSGs for planning. However, a critical distinction lies in the assumption of knowledge. SayPlan assumes a pre-mapped, static environment. It focuses on searching a large, known graph for a specific object, utilizing a mechanism to expand and collapse graph nodes to manage token limits.9 MoMa-LLM advances this paradigm by handling the exploration phase, where the graph is incomplete and dynamically growing. This capability to handle the "unknown" is a significant leap over static planners.

Unstructured Prompts:

Simple baselines in the literature often feed raw object detections or unstructured lists to an LLM. MoMa-LLM argues—and empirically demonstrates—that this leads to context overflow and poor spatial reasoning. The "Structured Knowledge Extraction" in MoMa-LLM is a direct response to the inefficiency of unstructured prompting, providing a filtered, hierarchical view of the world.1

### 2.3 Semantic and Interactive Object Search

Object search has matured from geometric frontier exploration to semantic guidance.

Frontier Exploration:

Classical methods, such as those proposed by Yamauchi, rely on frontier-based exploration. The agent moves to the boundary between known and unknown space. While efficient for geometric mapping (coverage), this approach is semantically blind and inefficient for finding specific objects.1

Semantic Priors (ESC, ObjectNav):

Methods like ESC (Exploration with Soft Commonsense constraints) utilize pre-trained models to predict where objects should be. ESC employs Probabilistic Soft Logic to score frontiers based on commonsense rules (e.g., "beds are in bedrooms").6 While effective, ESC relies on fixed logic rules or pairwise correlations. MoMa-LLM replaces these with the general, generative reasoning of an LLM, potentially allowing for more nuanced deductions (e.g., inferring that a "dirty plate" implies a need to find a dishwasher or sink, a relationship that might not be captured in simple co-occurrence statistics).

Interactive Search (HIMOS):

Schmalstieg et al. introduced the interactive search task involving the physical opening of doors and drawers. HIMOS (Hierarchical Interactive Multi-Object Search) utilizes Hierarchical Reinforcement Learning to solve this. While effective, RL policies are often opaque, data-hungry, and difficult to transfer to new robot embodiments without retraining.2 MoMa-LLM aims to achieve similar or better performance in a zero-shot manner, utilizing the "world knowledge" frozen in the LLM rather than learning from scratch.

## 3. Paper Summary: Problem Statement and Methodology

The MoMa-LLM framework is a modular pipeline transforming raw sensor data into high-level actions. The architecture is tripartite: Perception & Mapping, Dynamic Scene Graph Generation, and LLM-Grounded Planning.

### 3.1 Problem Statement: The Partially Observable Challenge

The paper formalizes the problem as a **Semantic Interactive Object Search (SIOS)** task in unexplored environments. This is mathematically formulated as a Partially Observable Markov Decision Process (POMDP) defined by the tuple $\mathcal{M} = (S, A, O, T, P, r, \gamma)$ 1:

* **State ($S$):** The complete state of the world, including the robot pose $(x, y, \theta)$, the geometric map of the environment, the semantic class $c\_i$ of every object $o\_i$, and crucially, the state $s\_i$ of every articulated object (e.g., $drawer\_{kitchen\\_1} = closed$).
* **Observation ($O$):** The robot receives an RGB-D image $I\_t$, its pose, and semantic segmentations.
* **Action ($A$):** A hybrid action space comprising high-level primitives (e.g., navigate, open, explore) that are executed by low-level continuous controllers.
* **Goal ($g$):** A natural language instruction (e.g., "Find me something for breakfast").

The core challenge lies in the mapping $O \to A$. In a POMDP, the agent must maintain a belief state. In classical robotics, this belief is often a probabilistic grid. In MoMa-LLM, the belief state is explicitly structured as the **Dynamic Scene Graph (DSG)**. The agent must infer that a high-level semantic goal ("breakfast") implies a sequence of geometric and interactive actions (navigating to the kitchen, opening the fridge) based on this structured belief state.

### 3.2 Methodology: The MoMa-LLM Architecture

#### 3.2.1 Perception and Dynamic RGB-D Mapping

The foundation of the system is a 3D volumetric map. The robot ingests posed RGB-D frames equipped with semantic labels.

* **Semantic Mapping:** Point clouds are projected into a global frame. A voxel grid $\mathcal{M}\_t$ accumulates occupancy and semantic labels.
* **BEV Abstraction:** To facilitate navigation, the 3D voxel grid is collapsed into a 2D Bird's-Eye-View (BEV) occupancy map $\mathcal{B}\_t$. This distinguishes between occupied space, free space $\mathcal{F}\_t$, and unexplored space.
* **Critique of Perception:** It is crucial to note that the paper assumes ground-truth semantics in simulation to isolate the reasoning component from perception noise.1 While standard in reasoning-focused literature, this is a significant assumption. In the real-world deployment, they utilize specific detectors (e.g., markers for handles) and YOLO-based detection, acknowledging that perception noise is a separate, non-trivial challenge.

#### 3.2.2 Topological Abstraction: The Voronoi Graph

A key innovation in MoMa-LLM is the use of a **Generalized Voronoi Diagram (GVD)** for topology and safety.

* GVD Construction: The GVD represents the set of points equidistant to the nearest obstacles. Formally,  
    
  $$\mathcal{F}^{2} = \{ x \in \mathbb{R}^2 : d\_i(x) = d\_j(x) \le d\_k(x), \forall k \}$$  
    
  where $d\_i(x)$ is the distance to obstacle $i$.1
* **Strategic Advantage:** By navigating on the Voronoi graph, the robot inherently maximizes safety (distance to walls). This is particularly critical for mobile manipulation platforms like the Fetch or HSR, which have wide bases and extendable arms that create collision risks. The nodes of the GVD also serve as "anchors" for object locations, providing a topological rather than purely metric reference system.
* **Graph Pruning:** The raw GVD is dense and continuous. MoMa-LLM sparsifies this into a topological graph $\mathcal{G\_V} = (\mathcal{V}, \mathcal{E})$, removing dead ends and redundant nodes to create a streamlined navigational backbone.1

#### 3.2.3 Dynamic Scene Graph (DSG) Construction

The DSG $\mathcal{G}\_S$ serves as the bridge between geometry and semantics. It is constructed hierarchically through a novel room segmentation strategy.

Room Segmentation via Door Probabilities:

Unlike Hydra, which uses geometric constrictions, MoMa-LLM uses a semantic-geometric heuristic centered on doors.

1. **Door Detection:** The system detects doors (open or closed) in the sensory stream.
2. Kernel Density Estimation: The distribution of door detections is modeled as a mixture of Gaussians (Eq. 1 in 1):  
     
   $$\rho\_{\mathcal{N}}(x, H) = \frac{1}{N\_D} \sum\_{i=1}^{N\_D} K\_H(x - x\_i)$$  
     
   where $x\_i$ represents the center coordinates of detected doors and $K\_H$ is a Gaussian kernel.
3. **Graph Cutting:** Edges in the Voronoi graph that traverse regions of high door probability $\rho\_{\mathcal{N}}$ are severed. This partitions the Voronoi graph into disjoint subgraphs $\mathcal{G}\_{\mathcal{V}}^R$, each corresponding to a distinct region or room. This approach is robust to layouts where semantic boundaries (doors) do not strictly correlate with geometric bottlenecks.

Object-Room Assignment:

To avoid "through-wall" assignment errors—where an object in one room is erroneously assigned to an adjacent room due to Euclidean proximity—MoMa-LLM assigns object $o$ to a room node $n\_o$ by minimizing a path cost that penalizes travel distance on the Voronoi graph (Eq. 2 in 1). This ensures that an object is assigned to the room from which it is accessible, preserving the functional topology of the environment.

Open-Vocabulary Room Classification:

Once regions are segmented, they must be semantically labeled. MoMa-LLM employs an LLM (GPT-3.5) for this task.

* **Prompting Strategy:** The LLM is provided with a list of objects contained within the region (e.g., "bed, lamp, wardrobe").
* **Inference:** The LLM infers the room type (e.g., "Bedroom"). This allows for open-vocabulary classification (e.g., "Office," "Pantry," "Workshop") without a fixed label set, adapting to diverse environments.1

#### 3.2.4 Structured Knowledge Extraction and Planning

The crux of the paper is how this graph is presented to the planner (GPT-4). The authors argue that raw data representations fail to leverage the reasoning capabilities of LLMs effectively.

The Failure of Raw JSON:

The authors demonstrate that feeding a raw JSON dump of the scene graph to the LLM (the "Unstructured LLM" baseline) leads to poor performance.1 JSON is token-heavy and lacks inherent spatial prioritization, causing the LLM to get "lost in the details" or hallucinate connections.

Structured Prompting:

MoMa-LLM creates a "summarized" textual representation that acts as an information bottleneck:

1. **Scene Structure:** A hierarchical list: Room -> Objects (State). Distances are binned into natural language adjectives ("near," "far") to facilitate linguistic reasoning over numerical comparison.1
2. **Affordances:** Object states are encoded directly (e.g., "closed fridge").
3. **Frontiers as Semantic Entities:** This is a crucial innovation. Unexplored areas are not just treated as geometric voids but are attached to rooms as semantic entities. A frontier is described as "unexplored area in the kitchen" or "unexplored area leading out of the living room".1 This allows the LLM to reason: "If the milk is not in the current view of the kitchen, I should explore the *unexplored area in the kitchen*."

Dynamic History Alignment:

To maintain the Markov property without exploding the token count, MoMa-LLM summarizes history. However, simply listing past actions (visited room 1) is problematic because room 1 might be reclassified as kitchen later as more objects are discovered.

* **Re-alignment:** The system tracks interaction positions. When the map updates, past actions are re-mapped to the *current* semantic labels. The history presented to the LLM is always consistent with the current map state.1

### 3.3 Action Space

The high-level action space available to the LLM is object-centric and abstract:

* navigate(room, object)
* go\_to\_and\_open(room, object)
* close(room, object)
* explore(room)
* done()

Low-level execution is handled by standard motion planners ($A^\*$) and manipulation policies ($N^2M^2$), creating a hierarchical control loop.1

## 4. Experiments

The evaluation of MoMa-LLM is comprehensive, spanning simulation (iGibson) and real-world deployment. The authors introduce a new metric, AUC-E, to address flaws in standard evaluation protocols for search tasks.

### 4.1 Metrics: The Introduction of AUC-E

Standard metrics in ObjectNav are Success Rate (SR) and Success Weighted by Path Length (SPL).

* **The Critique:** SR and SPL depend heavily on the maximum time budget allowed. A method that finds an object in 500 steps is a "failure" if the budget is 499, but a "success" if the budget is 501. This binary cutoff masks the efficiency profile of the agent.1
* **AUC-E (Area Under the Efficiency Curve):** The authors propose plotting the Success Rate as a function of the time budget (steps). The Area Under this Curve (AUC-E) provides a scalar value representing the agent's performance across *all* possible time budgets. A high AUC-E indicates the agent finds objects quickly and reliably. This provides a more nuanced view of "search efficiency," rewarding agents that succeed early while still acknowledging those that succeed eventually.1

### 4.2 Simulation Results (iGibson)

The experiments utilized the iGibson simulator with a Fetch robot. The task involved finding specific objects (e.g., "find the milk") in large, procedurally populated apartments.

Quantitative Analysis (Table I in 1):

| **Method** | **Success Rate (SR)** | **SPL** | **AUC-E** | **Object Interactions** | **Distance Traveled** | **Infeasible Actions** |
| --- | --- | --- | --- | --- | --- | --- |
| **MoMa-LLM (Ours)** | **97.7%** | **63.6** | **87.2** | 3.9 | 18.2 | 0.19 |
| **ESC-Interactive** | 95.4% | 62.7 | 84.5 | 4.1 | 19.6 | - |
| **HIMOS (RL)** | 93.7% | 48.5 | 77.4 | 4.8 | 35.9 | - |
| **Unstructured LLM** | 86.3% | 59.4 | 77.6 | 3.6 | 18.5 | 0.41 |
| **Random** | 93.1% | 50.2 | 77.0 | 5.7 | 32.9 | - |

**Key Findings:**

* **MoMa-LLM Dominance:** MoMa-LLM outperforms all baselines in SR, SPL, and AUC-E. The high SPL (63.6) compared to HIMOS (48.5) confirms that the LLM-guided agent takes significantly more direct paths to the target, avoiding the jittery or redundant exploration often seen in RL agents.
* **Efficiency vs. Learning:** Comparing MoMa-LLM to HIMOS reveals the trade-off between learning and reasoning. HIMOS achieves high success but at the cost of efficiency. The RL agent, learning from scratch, struggles to internalize the semantic priors that the LLM possesses natively. It explores exhaustively rather than semantically.
* **Heuristics vs. Reasoning:** ESC-Interactive, which uses a commonsense model to score frontiers, performs strongly (95.4% SR), validating the utility of semantic priors. However, MoMa-LLM still edges it out, suggesting that the "long-horizon" planning capability of the LLM (sequencing multiple rooms) adds value over the greedy, stepwise heuristic of ESC.1
* **The Importance of Structure:** The failure of the "Unstructured LLM" baseline (86.3% SR) is a critical finding. It demonstrates that the power of the LLM is contingent on the representation. Without the structured graph summary, the LLM hallucinates more frequently (0.41 infeasible actions per episode vs. 0.19) and fails to maintain context.1

**Ablation Studies:**

* **No Frontiers:** Removing frontier information dropped performance drastically to 79.4% SR. This proves that explicitly representing the unknown is vital for exploration. An agent that only knows what it has seen cannot plan where to go next.
* **No History:** Removing history led to a drop in efficiency (SPL 63.0 vs 63.6), as the agent would occasionally loop or revisit areas, losing track of its search progress.1

### 4.3 Real-World Transfer

The system was deployed on a Toyota HSR in a physical apartment comprising 4 rooms and 54 objects.

* **Setup:** Perception utilized 2D LiDAR for SLAM and AR markers for handle detection to simplify the manipulation perception challenge.
* **Results:** MoMa-LLM achieved an 80% success rate (8/10 trials), comparable to ESC in success but with significantly higher efficiency (Distance traveled: 17.9m vs 33.9m for ESC).
* **Qualitative Insight:** The real-world experiments demonstrated robustness to "combined rooms" (e.g., kitchen-dining) which plague geometric segmentation methods. The open-vocabulary classifier correctly identified these hybrid spaces, allowing the agent to reason about them appropriately (e.g., looking for both food and plates in the same zone).1 The system's ability to transfer zero-shot from simulation to the real world validates the "Language-Grounded" hypothesis—the reasoning capability holds up regardless of the domain shift in low-level perception.

## 5. Discussion

The MoMa-LLM framework represents a significant step forward in grounded robotic reasoning, yet a critical analysis reveals both its transformative potential and its architectural limitations.

### 5.1 The Power of Structure in Grounding

The most significant contribution of MoMa-LLM is the empirical validation of **Structured Knowledge Extraction**. The grounding gap is often framed as a lack of training data. MoMa-LLM suggests it is fundamentally a problem of *data organization*. By converting the continuous world into a discrete, semantic graph and then summarizing that graph into structured text, the system aligns the robot's state representation with the LLM's native modality (text). The explicit encoding of "Frontiers" as semantic entities (e.g., "Unexplored area in the Kitchen") is a subtle but profound shift. It transforms "exploration" from a geometric task (go to coordinates $x,y$) to a semantic task (search the kitchen), which allows the LLM to apply commonsense priors (e.g., "Food is in the kitchen, so explore the kitchen frontier"). This effectively bridges the gap between geometric SLAM and semantic planning.

### 5.2 Efficiency vs. Learning: A Shift in Paradigm

The comparison with HIMOS 3 underscores a shift in the robotics paradigm from *learning* to *reasoning*. HIMOS learns exploration policies from massive interaction data. MoMa-LLM infers exploration strategies from the semantic priors of the LLM. The results indicate that for household search, the semantic priors are strong enough that "zero-shot reasoning" beats "trained policy." The RL agent struggles with the sheer size of the state space (hundreds of objects and articulation states), whereas the LLM handles the combinatorics naturally via language.1 This suggests that for high-level semantic tasks, the "pre-training" of the LLM on human knowledge is a more efficient starting point than RL from scratch.

### 5.3 Limitations and Emerging Critiques

While MoMa-LLM represents the state-of-the-art for 2024, recent literature (2025) highlights intrinsic limitations in its design.

#### 5.3.1 The "Cost-Blindness" of High-Level Reasoning

A major critique, articulated in the **"Inter-LLM"** paper (Yang et al., 2025) 15, is that MoMa-LLM decouples high-level reasoning from low-level execution costs. The hierarchical architecture treats actions as abstract function calls. The LLM might decide to explore(kitchen) because it makes semantic sense, ignoring that the path to the kitchen is blocked by clutter that makes navigation costly or risky.

* **The Consequence:** The generated plans are semantically logical but may be physically inefficient or infeasible. Inter-LLM proposes an "interleaved" approach where the motion planner provides cost feedback *before* the LLM finalizes the plan. MoMa-LLM only receives binary success/failure feedback *after* an attempt, leading to potentially wasted attempts in difficult environments.1

#### 5.3.2 Granularity and Rearrangement

The **"MORE"** paper (Mohammadi et al., 2025) critiques the granularity of MoMa-LLM's graph for **rearrangement tasks**.4 MoMa-LLM filters the scene graph heavily to fit the LLM context window, often summarizing multiple objects into lists. However, for rearrangement (moving multiple objects to specific configurations), the robot needs to track specific instances and their precise spatial relations. MoMa-LLM's aggressive summarization may discard the geometric nuances required for complex manipulation sequences, limiting its utility to search tasks where "finding" is the terminal state.

#### 5.3.3 The "Re-computation" Bottleneck

MoMa-LLM recomputes the scene graph at every time step.1 While feasible for the relatively slow decision cycle of object search (navigate -> stop -> think), this prevents high-frequency replanning. As environments scale to "large scene graphs" with thousands of objects 15, this $O(N)$ re-computation will become prohibitive. Incremental graph updates (as seen in Hydra) are necessary for scalability to larger, continuous operations.

### 5.4 The "Hallucination" Defense

A common criticism of LLM planners is hallucination. MoMa-LLM employs a "filtering" defense. By only exposing the LLM to objects explicitly detected and present in the graph (and strictly enforcing valid function arguments via the prompt structure), the system constrains the LLM's output space. The experiments show a low rate of "Infeasible Actions" (0.19), suggesting this constraining strategy is effective. However, it also limits "hypothetical" reasoning. The LLM cannot say "I expect to find keys on a table" if the table hasn't been seen yet; it can only say "Explore the room." This limits the specificity of exploration compared to methods that might hallucinate potential object locations to guide search more aggressively.

## 6. Future Directions and Implications

The analysis of MoMa-LLM points toward several high-impact research directions:

1. **Cost-Aware Grounding:** Integrating cost maps (e.g., traversability costs) directly into the LLM prompt or using an API that returns cost estimates before execution could address the Inter-LLM critique.15
2. **Multimodal LLMs (VLMs):** MoMa-LLM uses text-only LLMs (GPT-4). Replacing the room classification module with a Vision-Language Model (VLM) that looks at the panoramic image of the room could improve robustness against segmentation errors.17
3. **From Search to Rearrangement:** Extending the framework to tasks where the robot must *change* the scene graph (e.g., "clean the room") requires tracking the *history of object states* more granularly than the current system allows.4
4. **Open-Set Perception:** The current reliance on ground-truth semantics (in sim) or specific detectors (in real) is a bottleneck. Integrating "Open-Vocabulary Object Detection" (e.g., Owl-ViT, YOLO-World) directly into the node generation pipeline would create a truly fully open-vocabulary system.19

In conclusion, "Language-Grounded Dynamic Scene Graphs for Interactive Object Search with Mobile Manipulation" establishes a robust standard for semantic grounding in exploration. By treating the Scene Graph as a dynamic, structured prompt, MoMa-LLM effectively operationalizes the "world knowledge" of LLMs for robotics. While the separation of "Mind" (LLM) and "Body" (Planner) remains a friction point, the architectural blueprint of dynamic, language-grounded graphs is a foundational step toward truly intelligent mobile manipulators.

#### Works cited

1. menon\_1\_honerkamp24ral-1.pdf
2. Learning Long-Horizon Robot Exploration Strategies for Multi-Object Search in Continuous Action Spaces - Semantic Scholar, accessed on January 18, 2026, <https://www.semanticscholar.org/paper/Learning-Long-Horizon-Robot-Exploration-Strategies-Schmalstieg-Honerkamp/915234d691be2d9efdbdcc178fbff28a381c0863>
3. Learning Hierarchical Interactive Multi-Object Search for Mobile Manipulation, accessed on January 18, 2026, <https://www.researchgate.net/publication/372313553_Learning_Hierarchical_Interactive_Multi-Object_Search_for_Mobile_Manipulation>
4. MORE: Mobile Manipulation Rearrangement through Grounded Language Reasoning - Toronto Intelligent Systems Lab, accessed on January 18, 2026, <https://tisl.cs.toronto.edu/publication/MORE__Mobile_Manipulation_Rearrangement_Through_Grounded_Language_Reasoning/MORE__Mobile_Manipulation_Rearrangement_Through_Grounded_Language_Reasoning.pdf>
5. MoMa-LLM, accessed on January 18, 2026, <https://moma-llm.cs.uni-freiburg.de/>
6. ESC: Exploration with Soft Commonsense Constraints for Zero-shot Object Navigation - OpenReview, accessed on January 18, 2026, <https://openreview.net/pdf?id=GydFM0ZEXY>
7. [2403.08605] Language-Grounded Dynamic Scene Graphs for Interactive Object Search with Mobile Manipulation - arXiv, accessed on January 18, 2026, <https://arxiv.org/abs/2403.08605>
8. Language-Grounded Dynamic Scene Graphs for Interactive Object Search with Mobile Manipulation - OpenReview, accessed on January 18, 2026, <https://openreview.net/pdf/f0ccb8e5511b7ca0d776b046872e4ed72e6e3233.pdf>
9. Language-Grounded Dynamic Scene Graphs for Interactive Object Search with Mobile Manipulation, accessed on January 18, 2026, <https://semrob.github.io/docs/rss_semrob2024_cr_paper26.pdf>
10. LOOKPLANGRAPH: EMBODIED INSTRUCTION FOL- LOWING METHOD WITH VLM GRAPH AUGMENTATION - OpenReview, accessed on January 18, 2026, <https://openreview.net/pdf?id=B47cCZfJFa>
11. [2301.13166] ESC: Exploration with Soft Commonsense Constraints for Zero-shot Object Navigation - arXiv, accessed on January 18, 2026, <https://arxiv.org/abs/2301.13166>
12. Language-Grounded Dynamic Scene Graphs For Interactive Object Search With Mobile Manipulation | PDF - Scribd, accessed on January 18, 2026, <https://www.scribd.com/document/787749169/2403-08605v4>
13. N$^{2}$M$^{2}$: Learning Navigation for Arbitrary Mobile Manipulation Motions in Unseen and Dynamic Environments - Semantic Scholar, accessed on January 18, 2026, <https://www.semanticscholar.org/paper/5051947cbac4223a68edc8e7e7319b5cdb2dc712>
14. MoMa-LLM: Scene Graphs for Mobile Object Search, accessed on January 18, 2026, <https://www.emergentmind.com/papers/2403.08605>
15. Interleaved LLM and Motion Planning for Generalized Multi-Object Collection in Large Scene Graphs - arXiv, accessed on January 18, 2026, <https://arxiv.org/html/2507.15782v1>
16. [2507.15782] Interleaved LLM and Motion Planning for Generalized Multi-Object Collection in Large Scene Graphs - arXiv, accessed on January 18, 2026, <https://arxiv.org/abs/2507.15782>
17. Explainable Saliency: Articulating Reasoning with Contextual Prioritization - CVF Open Access, accessed on January 18, 2026, <https://openaccess.thecvf.com/content/CVPR2025/papers/Chen_Explainable_Saliency_Articulating_Reasoning_with_Contextual_Prioritization_CVPR_2025_paper.pdf>
18. zubair-irshad/Awesome-Robotics-3D - GitHub, accessed on January 18, 2026, <https://github.com/zubair-irshad/Awesome-Robotics-3D>
19. Open-Vocabulary and Semantic-Aware Reasoning for Search and Retrieval of Objects in Dynamic and Concealed Spaces, accessed on January 18, 2026, <https://autonomousrobots.nl/assets/images/workshops/2025_iros/accepted_papers/paper_8_Open.pdf>