Neuro-Symbolic Task Routing in Semantic Voxel Environments for Generalist Robotic Manipulation

Anindya Jana
Physical AI, TCS Research
Tata Consultancy Services
Kolkata, India
anindya.jana@tcs.com

Snehasis Banerjee
Physical AI, TCS Research
Tata Consultancy Services
Kolkata, India
snehasis.banerjee@tcs.com

Arup Kumar Sadhu
Physical AI, TCS Research
Tata Consultancy Services
Kolkata, India
arupk.sadhu@tcs.com

Ranjan Dasgupta
Physical AI, TCS Research
Tata Consultancy Services
Kolkata, India
ranjan.dasgupta@tcs.com

***Abstract*—Vision-Language-Action (VLA) models have demonstrated significant potential in robotic manipulation but suffer from a performance gap when comparing generalist models to task-specific specialists. Furthermore, deploying these models in unstructured indoor environments requires robust mapping and navigation capabilities. In order to close this gap, this paper presents a unified framework that combines a novel neuro-symbolic task configuration engine with a modular semantic mapping system. In order to make navigation and grounding easier, our method first builds a time-constrained semantic voxel map of the surroundings. By using relaxed waypoint algorithms, we optimize exploration efficiency by 50%. When a task configuration framework arrives at a workspace, it analyzes the spatial arrangement using zero-shot object detection and dynamic scene graph generation. 78–100% of the performance difference between generalist and specialist approaches is recovered by this analysis, which directs the manipulation request to the best specialized model. Empirical findings show that our system maintains real-time computational feasibility appropriate for practical implementation while increasing success rates from 74% to 98% in challenging long-horizon tasks.** [1]

***Index Terms*—Vision-Language-Action, Neuro-symbolic AI, Semantic Mapping, Robotic Manipulation, Physical AI.**

I. Introduction

The integration of computer vision, natural language processing, and robotics has led to the emergence of Vision-Language-Action (VLA) models, such as OpenVLA [1] and RT-2 [2]. These foundation models promise a paradigm shift in embodied AI by directly mapping multimodal observations of visual scenes and linguistic instructions to continuous control actions. However, despite their remarkable zero-shot performance on controlled benchmarks, the application of these models in unstructured real-world settings exposes critical limitations. We point out two essential barriers to adoption: the "teleportation assumption" in manipulation research and the "competence-generalization trade-off" in policy learning.

The first barrier is the artificial split between navigation and manipulation. Modern robotic architectures tend to compartmentalize these into separate silos: navigation systems are solely concerned with reaching coordinate waypoints [3] [4], whereas manipulation policies usually assume the robot is already pre-positioned directly in front of the target workspace. This compartmentalized strategy is inadequate for real-world applications, where a service robot needs to first semantically interpret a large, unseen environment (e.g., "find the kitchen") before attempting any manipulation task. Developing a strong "world model" that closes the gap between room-scale exploration and table-scale interaction is an open problem.

The second challenge is the degradation of fine-grained manipulation performance when scaling to different task distributions. Although generalist VLA models [5] can try a large number of different tasks, our analysis shows that they are always behind task-specific specialists. The performance difference is especially large when tasks demand different cognitive modalities, such as high geometric reasoning versus long-term planning. A single "one-size-fits-all" policy may have difficulty switching between these modalities, resulting in success probabilities that are 4-21% lower than their specialized counterparts.

To tackle these interrelated challenges, we introduce a novel neuro-symbolic framework that combines a modular semantic mapping framework with an intelligent task configuration engine. To address these challenges holistically, our proposed framework seamlessly integrates mobility and manipulation into a unified operational sequence:

Firstly, we leverage a time-constrained exploration module inspired by VLA-3D [6]. By fusing real-time RGB-D streams with open-vocabulary OwlViT embeddings [7], we construct a queryable semantic voxel map. This allows the agent to autonomously locate workspaces under strict operational time limits, effectively solving the "teleportation" problem.

Secondly, we introduce a *Task Configuration Framework* to solve the manipulation gap. Instead of relying on a single generalist policy, we employ a neuro-symbolic router that analyzes the scene's spatial structure using dynamic scene graphs. This router intelligently directs user instructions to the optimal specialized model—whether it be a spatial reasoning expert for precise placement or a long-horizon expert for multi-step sequencing [8]. This "mixture-of-specialists" approach recovers nearly all the performance lost by generalist models while maintaining the flexibility of a unified system.

Thirdly, contemporary approaches consider navigation and manipulation as separate silos. The navigation system typically centers around the concept of reaching a waypoint [3], and the manipulation policy typically assumes that the robot is already

[1] Video: https://www.youtube.com/watch?v=BL4NTgXr_mc

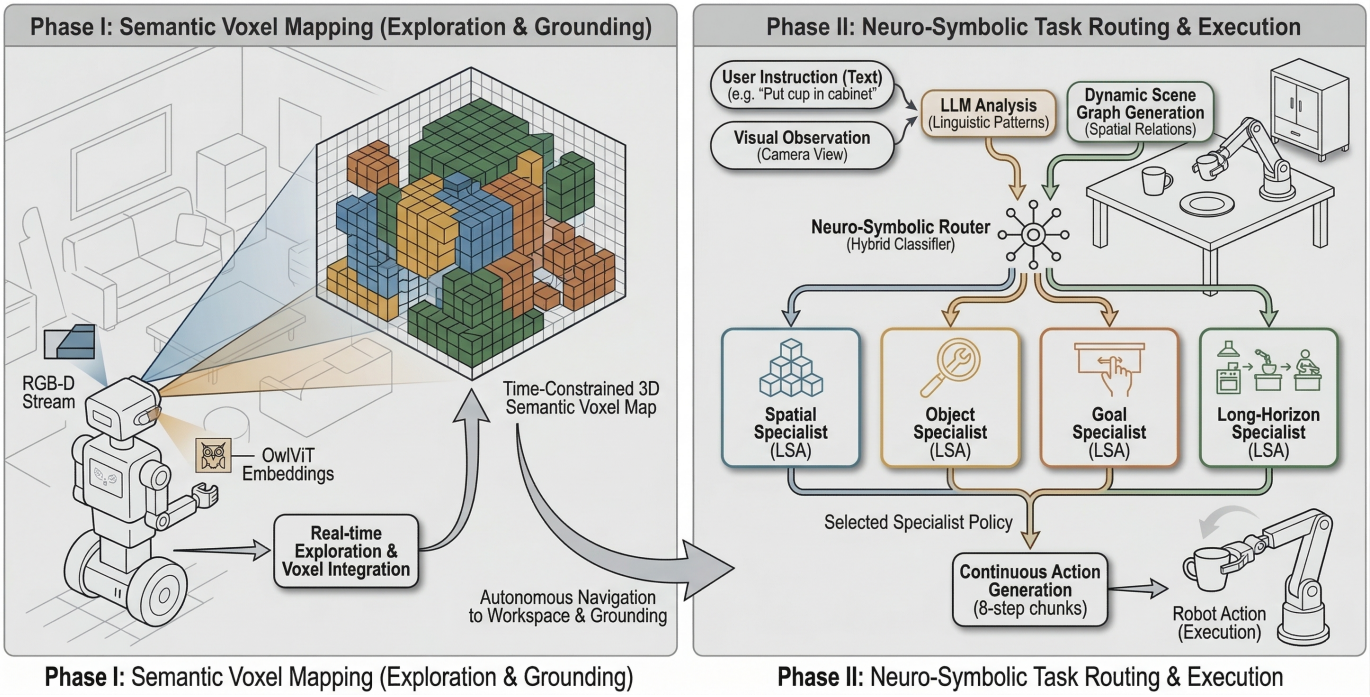

Fig. 1. Neuro-Symbolic Task Routing in Semantic Voxel Environments – an illustration of the two phases from perception to task execution.

positioned in front of the target object. To address this, we propose a unified system.

Finally, and most importantly to this work, we address the manipulation performance gap. Our analysis shows that generalist VLA models are 4-21% behind specialists depending on the task type. To address this, we propose a *Task Configuration Framework*. This neuro-symbolic approach analyzes the instruction and the visual scene graph to successfully direct the request to the most appropriate specialized model (e.g., a model that is fine-tuned for spatial reasoning or long-horizon sequences). Our key contributions are enlisted below:

1) **Algorithmic Routing Innovation:** A novel neuro-symbolic task classification algorithm that logically fuses visual-semantic grounding (via dynamic scene graphs) and spatial reasoning, advancing beyond standard LLM-based routing to achieve 93–98% accuracy.
2) **Systematic Formulation of the VLA Performance Gap:** We introduce a quantifiable framework to measure and recover the "competence-generalization trade-off" between generalist and specialist policies without requiring system-wide retraining.
3) Empirical validation showing that dynamic model routing improves manipulation success rates from roughly 74% to near-specialist levels of 95–98% with negligible latency overhead.

II. Related Work

A. Vision-Language-Action Models

The end-to-end control paradigm in robotics has undergone a dramatic transformation with the emergence of VLA models. Traditional approaches such as RT-1 [9] proved the effectiveness of the Transformer architecture in processing image and language-based instructions to produce discretized actions. Later models such as RT-2 and PaLM-E [10] incorporated reasoning modules from large language models (LLMs) into the control pipeline. More recently, OpenVLA [1] has employed an encoder-decoder structure to produce continuous action spaces, leading to increased accuracy.

However, generalist models trained on diverse datasets, such as Octo [5] or the newly proposed $\pi 0$ [11], have difficulty competing with specialists fine-tuned for specific domains. The "jack-of-all-trades" problem is most pronounced in fine-grained domains such as spatial reasoning or long-horizon sequencing, where generalist policies are prone to catastrophic forgetting or negative transfer due to conflicting task modalities. Our proposed solution does not involve training a more generalist model but instead provides a sophisticated selection strategy for a set of pre-existing specialized experts.

B. Task Adaptation and Mixture of Experts

To overcome the shortcomings of generalist policies, various task-conditioned learning approaches have been investigated. Marza et al. [12] introduced visual adapters that adjust pre-trained features according to downstream tasks. Along similar lines, Mixture of Experts (MoE) models, such as AdaMoE [13] and HiMoE-VLA [14], selectively engage domain-specific sub-networks according to input tokens.

These approaches generally depend on implicit "soft routing" in a single huge neural network, where gating networks learn to route computations. This can cause routing collapse

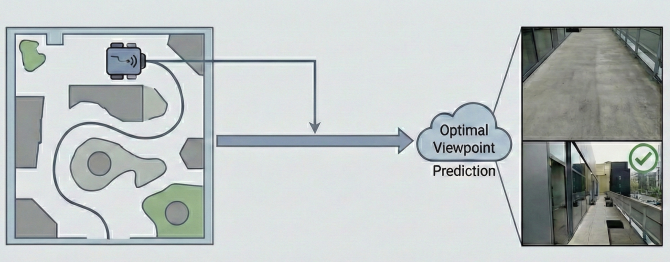

Fig. 2. Path-Level Active Localization with ActLoc-based Optimal Viewpoint Prediction

or a lack of specialization. By contrast, our framework uses an explicit and interpretable neuro-symbolic classification module to selectively route tasks to entirely different, lightweight specialist adapters (LSA). Our "hard routing" strategy maximizes domain specialization, avoids task interference, and enables the modular incorporation of new specialists without system-wide retraining.

C. Semantic Mapping and Navigation

However, for a robot to manipulate objects in the wild, it first needs to navigate towards them. While conventional SLAM methods are concerned with geometric reconstruction via occupancy grids [15], more contemporary methods have emphasized semantic representation, in which geometric representations are supplemented with linguistic descriptions. Voxel maps [16] are a dense 3D representation of the environment that is particularly well-suited to navigation, while open-vocabulary detectors such as OwlViT [7] and CLIP [17] enable these maps to be queried by natural language.

Our work draws on ideas from VLA-3D [6] and SORT3D [18], effectively building a "searchable" physical world. In contrast to approaches that require pre-computed maps or time-consuming offline processing, our system builds semantic voxel maps in real-time during a time-constrained exploration process, which directly enables the transition from room-scale navigation to table-scale manipulation.

D. Neuro-Symbolic Scene Understanding

Closing the gap between raw perception and actionable reasoning is a fundamental task. Scene graphs [19], [20] provide a structured representation of the world, where objects are represented as nodes and relations as edges. Although they have been conventionally used for static scene analysis, recent approaches such as RoboEXP [21] and SG-Nav [22] have extended them to active robotics. There has been work based on ontologies for scene understanding [23] [24]. Our approach further extends this line of work by producing *dynamic* manipulation-centric scene graphs. We leverage these graphs not only for representation but also as a direct input to our task classifier, allowing the system to reason about spatial constraints (e.g., "inside," "supported by") that linguistic models have difficulty capturing [8].

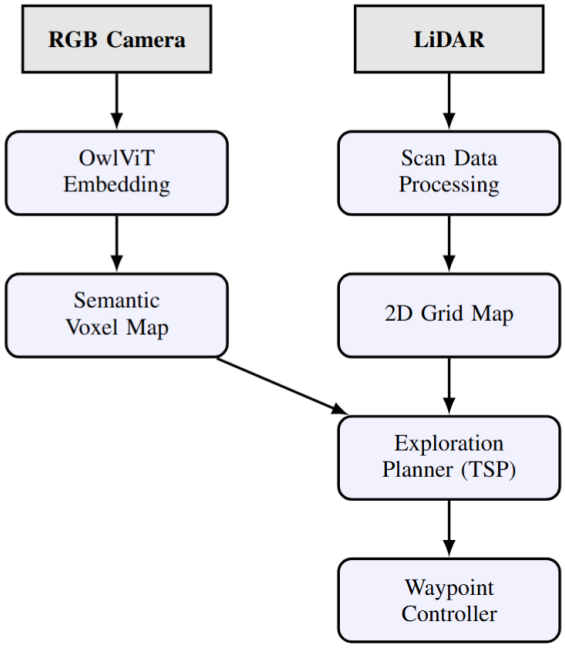

Fig. 3. **Phase I Architecture:** The system processes parallel visual and geometric streams to build a semantic voxel map and execute autonomous exploration.

III. System Architecture

Our framework operates in two distinct phases: (A) Environmental Exploration and Semantic Mapping, and (B) Neuro-Symbolic Task Configuration and Execution.

Hardware Allocation: To manage computational load and ensure real-time responsiveness, processing is distributed across two tiers:

- **On-Board (Robot CPU/GPU):** High-frequency control loops, real-time, continuous voxel map updates, autonomous path planning, and the execution of the lightweight VLA manipulation policies (LSA adapters).
- **Cloud/Edge Compute:** Computationally heavy, one-time perceptual tasks, including open-vocabulary embedding extraction (OwlViT), LLM-based instruction parsing, and dynamic scene graph generation.

While traditional manipulation research often assumes a fixed starting state to *mobile manipulation* approach, a generalist agent must first autonomously locate relevant objects within unstructured environments (Phase I) before fine-grained interaction can occur (Phase II). This integration removes the "teleportation assumption" common in baselines, ensuring the manipulation policies are grounded in a realistic, navigable world model.

A. Phase I: Semantic Voxel Mapping

To enable the robot to operate in unknown indoor environments, we employ a modular architecture built upon the Robot Operating System (ROS) [25]. This phase constructs the world model required for navigation (Fig. 3).

1) Vision: The visual information from the RGB camera of the robot is processed to detect objects and learn high-dimensional embeddings with OwlViT (`google/owlvit-base-patch32`) [7]. The embeddings are used to create a voxel map, which encodes both geometric information (x, y, z) and semantic features.

2) Autonomous Exploration Logic: Our system utilizes a greedy frontier-based exploration strategy augmented with a Traveling Salesperson Problem (TSP) solver to optimize path efficiency. The planner identifies "frontier" points (boundaries between explored and unexplored space) and samples candidate viewpoints to maximize coverage. A distance transform method is applied to the 2D grid to identify safe, traversable points while maintaining a buffer from obstacles. The TSP solver then computes the optimal sequence for visiting these points, minimizing backtracking.

3) Semantic Voxel Construction: Simultaneously, we construct a dense 3D semantic map. We employ OwlViT [7] to extract open-vocabulary embeddings from real-time RGB frames. These embeddings are projected into 3D space using depth data and aggregated into a voxel grid with a resolution of 0.15m. Each voxel v_i stores not just occupancy information, but also a centroid c_i and an aggregated semantic feature vector f_i. This allows for rich spatial querying; for example, locating "the potted plant on the file cabinet" involves computing the cosine similarity between the query text embedding and the stored voxel features. [16]

B. Phase II: Neuro-Symbolic Task Configuration

Once the robot has navigated to the target workspace, control shifts to the VLA manipulation framework (Fig. 4). To maintain direct alignment with the LIBERO benchmark suites evaluated in Section IV, we categorize manipulation tasks into four distinct taxonomies: **Spatial** (geometric reasoning), **Object** (visual retrieval), **Goal** (state changes), and **Long-Horizon** (multi-step sequencing).

1) Parallel Classification Pathways: The system employs two parallel pathways to classify the task:

- **Language Branch:** Uses an LLM to analyze instruction text for linguistic patterns (e.g., "open" implies goal-oriented).
- **Visual Branch:** Generates a dynamic scene graph from the RGB stream to identify spatial relationships (e.g., "next to", "inside") that language may under-specify.

2) Specialist Selection & Execution: A **Hybrid Classifier** fuses the outputs of these branches. Based on the classification, the **Specialist Router** dynamically loads the appropriate OpenVLA model fine-tuned via Lightweight Specialist Adapter (LSA), in contrast to LoRA (Low-Rank Adaptation) based model finetuning. Finally, the selected model generates continuous control actions (8-step chunks) to execute the task.

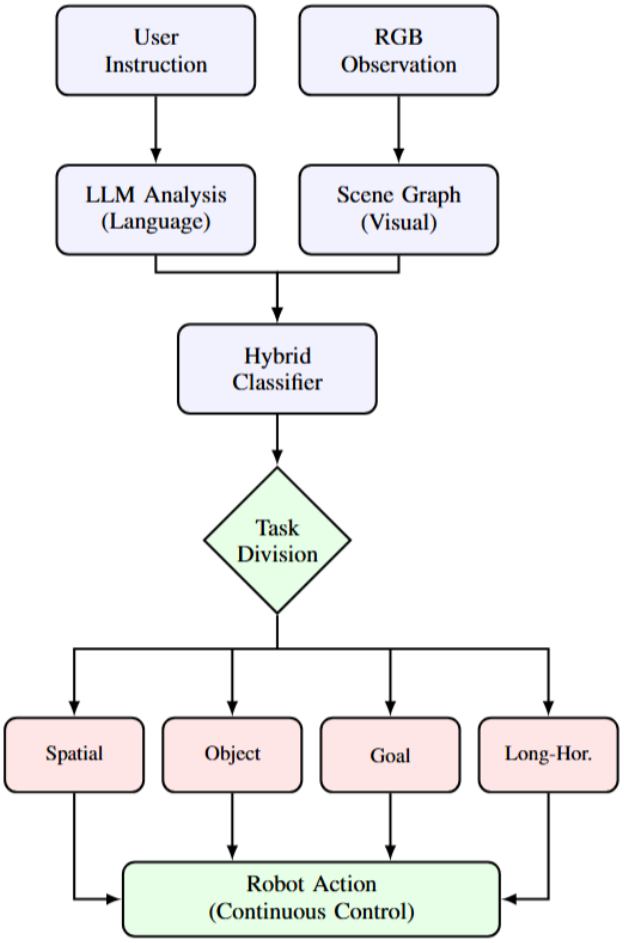

Fig. 4. **Phase II Architecture:** The Neuro-Symbolic Configurator analyzes instructions and scene graphs to route tasks to one of four specialized OpenVLA models.

IV. Experiments and Results

A. Experimental Setup

To validate the unified architecture, we conducted evaluations across two complementary benchmarks corresponding to our system's distinct phases:

- **Navigation & Grounding (Phase I):** We utilized high-fidelity simulated indoor environments (e.g., apartments, offices) within the AI Habitat and Unity frameworks, as seen in Fig. 6.
- **Manipulation (Phase II):** We evaluated the system using the LIBERO benchmark [26] (see Fig. 5). This benchmark is designed for lifelong robot learning research with controlled distribution shifts. It includes 130 unique manipulation tasks grouped into four distinct task suites: [26]in 5. This benchmark is designed for lifelong robot learning research with controlled distribution shifts. It includes 130 unique manipulation tasks grouped into four distinct task suites:
  - **LIBERO-Spatial:** 10 tasks focused on geometric reasoning and relative positioning (e.g., "place the bowl next to the plate").
  - **LIBERO-Object:** 10 tasks emphasizing visual recognition and retrieval of specific items from cluttered scenes.

Fig. 5. LIBERO simulation environment showing Franka Panda robot performing manipulation task in a kitchen scene

Fig. 6. Real-time visualization of the exploration process. The system builds a 2D grid map while simultaneously populating a 3D semantic voxel map for object grounding.

- **LIBERO-Goal:** 10 tasks involving functional state changes and procedural knowledge (e.g., "open the drawer").
- **LIBERO-10 (Long-Horizon):** 10 complex tasks requiring multi-step planning and temporal abstraction (e.g., "pick up the book and place it in the back compartment").

This taxonomy fits well with our neuro-symbolic classification categories. It serves as an ideal testbed for assessing how effective our routing framework is. The environment setup reflects typical indoor exploration conditions, using a Franka Panda robot configuration. Each task underwent testing across 50 trials, resulting in a total of 2,000 episodes.

1) OpenVLA Model Architecture: We utilize OpenVLA as our base vision-language-action architecture. OpenVLA is a 7-billion parameter model that integrates a vision encoder for processing multi-view images, a pre-trained language model (Llama-based) for instruction understanding, and an action decoder for continuous control [1]. The model processes dual-camera observations (third-person and wrist views) through convolutional encoders, concatenates visual features with instruction embeddings, and generates seven-dimensional continuous action predictions: end-effector position (3D), orientation (3D axis-angle representation), and binary gripper state [1], [2].

B. Navigation and Grounding

1) Efficiency of Exploration and Mapping: One of the most critical requirements in the mapping phase is to meet the time constraints. In our initial implementation with the representative office scene, the total time taken for the exploration process was around 8 minutes and 42 seconds. However, with the optimization of the voxel size to 0.15 m and the planner's waypoint interval to 1.25 m, we were able to cut down the exploration time by 50%, taking only 4 minutes and 17 seconds. This ensures that even with an operational time of 10 minutes, the robot has more than 5 minutes for query processing and manipulation.

2) Semantic Query Resolution: Beyond exploration, we evaluated the system's ability to ground natural language queries into the generated semantic map. The system demonstrated robust performance across three distinct query types defined in the challenge:

- **Numerical Queries:** The system successfully aggregated semantic voxel counts to answer numerical queries. For instance, for the query "How many chairs are there?", the system correctly identified and counted the instances to respond with "12".
- **Object Reference:** Using OwlViT embeddings, the system correctly located specific objects. For the query "Find the potted plant on the file cabinet," the system responded with correct 3D coordinates (e.g., $[1.15, 1.26, 1.33]$) for visualization markers.
- **Complex Grounding:** The system disambiguated references using spatial reasoning. For "Find the paper cup on the table closest to the projector screen," the system correctly identified the target coordinates $[-1.63, -1.98, 0.81]$ based on relative distances in the scene graph.

C. Validation and Performance

To validate the generalizability of these parameters, we evaluated this optimized configuration across 10 distinct AI

Habitat indoor environments (including apartments, offices, and retail spaces). Across these unseen environments, the system achieved a 92% success rate for our primary navigation metric: *successfully identifying and navigating to the target workspace within a strict 10-minute operational time budget*. The robot has over 5 minutes available for query processing and manipulation.

We first established performance bounds by evaluating task-specific specialists against a single unified generalist model. As shown in Table I, there is a distinct trade-off between specialization and generalization.

TABLE I
BASELINE MANIPULATION SUCCESS RATES (%) ACROSS FOUR TASK SUITES. SPECIALIST MODELS (DIAGONAL, BOLD) ACHIEVE NEAR-OPTIMAL WITHIN-DISTRIBUTION PERFORMANCE, WHILE GENERALIST MODEL EXHIBITS SUBSTANTIAL DEGRADATION.

Eval Suite	**Specialist Models**				**Gen**
	Spatial	**Object**	**Goal**	**Long**	**Model**
Spatial	**99.4**	85.7	82.0	76.0	91.0
Object	83.3	**98.7**	85.7	79.0	85.0
Goal	85.7	80.0	**96.7**	70.3	80.0
Long	72.0	68.0	70.3	**95.0**	74.0
Mean	85.1	83.1	83.7	80.1	**82.5**

Table I presents a cross-evaluation matrix. The diagonal entries (bold) represent the performance of specialist models on their native task suite (e.g., OpenVLA-Spatial on Spatial tasks). These models achieve exceptional success rates, ranging from 95.0% to 99.4%.

However, as indicated by the rightmost column, the Generalist model is always behind these peak performances. For instance, in Long-Horizon tasks, the generalist performs only 74.0% as opposed to the specialist's 95.0% performance, which is a difference of 21 percentage points. The off-diagonal elements make it clear that although specialists are strong performers, they are poor generalizers to out-of-distribution tasks (for example, the Long-Horizon specialist performs only 76.0% in Spatial tasks). This information highlights the need for our routing mechanism, which must choose the appropriate specialist based on the current task context in order to reap the benefits of both worlds.

D. Task Classification Accuracy and Ablation Studies

The efficacy of our routing depends on accurate classification. Our Scene Graph approach achieved 98% accuracy on Spatial tasks and 95% on Long-Horizon tasks. This represents a 13 percentage point improvement over using a compact Llama-3 (1B) model alone.

To validate the neuro-symbolic components, we conducted several ablation studies:

- **Scene Graph Components:** Turning off spatial relationship detection decreased accuracy from 95.5% to 87.2%. This is expected, as geometric reasoning (such as 'between' or 'inside') is essential to differentiate spatial reasoning tasks from generic object manipulation. Turning off object count features further decreased accuracy to 89.8%.
- **Detection Thresholds:** The confidence threshold of 0.15 was optimal for the OWLv2 detector. Reducing it to 0.05 introduced noise (false positives) and decreased accuracy to 91.3%, while increasing it to 0.30 resulted in missed detections, lowering accuracy to 92.7%.
- **Robustness:** Randomly removing 20% of objects from the knowledge base did not decrease accuracy below 93.1%, indicating the system's robustness to partial perception failure.

E. End-to-End Manipulation Performance

Integrating our task configurator led to near-specialist performance across all domains. As shown in Table II, the generalist baseline suffered significant performance degradation, particularly in *Long-Horizon* tasks where the success rate dropped to 74.0%. This failure mode often stemmed from "catastrophic forgetting" within the sequence—for example, the model might successfully pick up an object but fail to execute the subsequent placement correctly due to modal interference from other task types.

In contrast, our Scene Graph Configurator closed 78% to 100% of this performance gap (calculated as the ratio of performance gained by our method over the generalist, relative to the maximum possible gain defined by the specialist in Table I). For Long-Horizon tasks specifically, success rates surged from 74.0% (Generalist) to 95.0% (Ours), effectively matching the oracle specialist. This massive improvement is attributed to the router's ability to identify the "KITCHEN_SCENE" context and route the task to the *OpenVLA-Long* specialist, which is fine-tuned to maintain temporal coherence over extended sequences. Similarly, for Goal-Oriented tasks, performance improved from 80.0% to 93.0%. Qualitative analysis of rollout videos revealed that our specialized agents exhibited significantly less "jitter" and hesitation during approach phases compared to the generalist, executing more confident and stable grasps.

TABLE II
END-TO-END MANIPULATION SUCCESS RATES (%)

Task Suite	**Generalist**	**LLM-Only**	**Scene Graph (Ours)**
Spatial	91.0	94.0	**98.0**
Object	85.0	91.0	**96.0**
Goal	80.0	88.0	**93.0**
Long-Horizon	74.0	89.0	**95.0**
Average	82.5	90.5	**95.5**

F. Discussion on Specialist vs. Generalist Trade-offs

The substantial performance improvements from task configuration raise important questions about optimal architecture design for VLA models. The current approach maintains

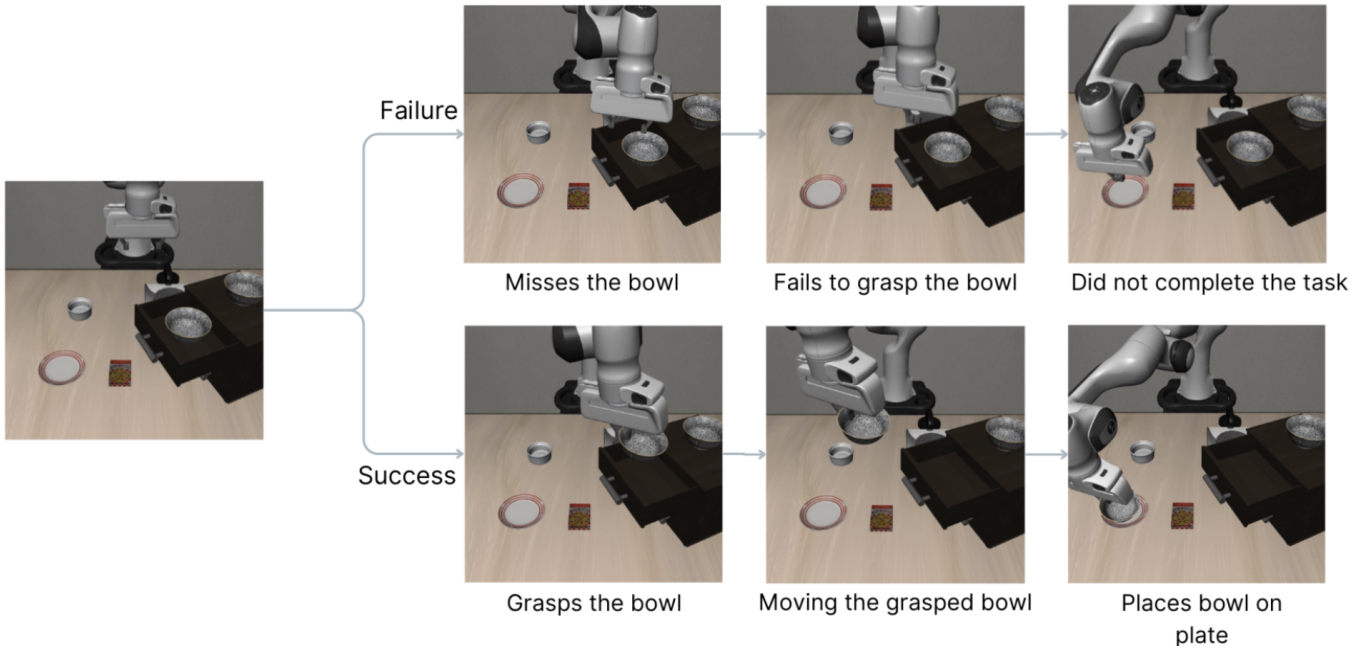

Fig. 7. Comparison of VLA execution. Top: Baseline method fails to secure the bowl. Bottom: Our method successfully grasps and places the bowl.

completely separate models for each domain, maximizing specialization at the cost of parameter efficiency and deployment complexity (Black et al. 2024; Team et al. 2024).

Alternative architectures could achieve similar specialization with reduced overhead through parameter sharing. Task-conditioned models could incorporate explicit task category inputs that modulate internal representations through adaptive normalization or attention mechanisms (Marza et al. 2024). Mixture-of-experts architectures could learn to route computation through specialized subnetworks based on input characteristics (Li et al. 2024b; Chen et al. 2025).

Regarding deployment scalability, our use of 'LSA' adapters avoids the prohibitive cost of maintaining multiple full models. This approach requires only 25GB of GPU memory for training and incurs negligible latency (0.7–6.5ms per step), making it strictly more efficient than maintaining separate full-scale policies. While we currently employ 'hard routing' to maximize reliability, future iterations could explore 'soft routing' or hierarchical mixtures-of-experts to allow for continuous task blending without sacrificing the reliability gains demonstrated here. However, our empirical results suggest that complete model specialization provides substantial benefits. The 15-20% performance gaps between specialists and generalists in complex domains indicate fundamental representational trade-offs that task-conditioned modulation may not fully address (Din et al. 2025; Wang et al. 2025).

G. Computational Overhead

The added latency for scene graph generation and classification is approximately 350–650ms. Since this is a one-time cost per episode, it amortizes to roughly 0.7–1.3ms per control step for a typical 500-step episode, maintaining the system's suitability for real-time 20Hz control loops.

V. Limitations

While our framework demonstrates robust performance, we acknowledge several limitations that define the scope for future research.

A. Static Environment Assumption

The current architecture assumes that the global map stays mostly the same. The semantic voxel map is built during the initial exploration phase and is used as a static guide for navigation. If the environment changes significantly, such as moving furniture, after mapping but before starting the manipulation task, the robot's navigation plan might not work. Updating the map in real-time is still a challenge.

B. Dependency on Cloud APIs

The initial high-level query processing and language-based classification depend on external VLM APIs. This creates a reliance on network connectivity and varying response times. Although the low-level manipulation tasks run locally, the system's ability to start tasks is at risk in areas with no communication.

C. Hard Routing Constraints

Our task configuration engine employs "hard routing," activating a single specialist model per task. While computationally efficient, this precludes the possibility of blending behaviors from multiple domains simultaneously. For hybrid tasks that might require both the fine-grained dexterity of the *Object* specialist and the sequential planning of the *Long-Horizon*

specialist, a soft-routing or mixture-of-experts approach might yield better results.

VI. Conclusion and Future Work

This paper has presented a comprehensive, end-to-end solution that bridges the gap between large-scale environmental navigation and fine-grained robotic manipulation. By integrating a modular semantic mapping framework with a novel neuro-symbolic task configuration engine, we have shown that a practical approach exists for generalist robots to maintain the high performance level of specialized experts.

Our study has identified a fundamental flaw in the current VLA deployment approach: the "one-model-fits-all" approach causes a significant performance loss, especially in complex, long-horizon tasks where the success rate can be degraded by more than 20%. Our proposed approach—a visual scene graph analysis-driven dynamic routing approach—has successfully recovered 78-100% of the performance loss. By systematically grouping tasks into spatial, object, goal-oriented, and long-horizon categories, and routing them to specialized 'LSA' adapters, we have approached oracle performance levels (up to 98% success rate) while retaining the flexibility of a generalist architecture. Most importantly, this is accomplished with a highly manageable one-time initialization cost (350–650ms), resulting in an overhead of roughly 1ms per step, making it amenable for real-time control.

In our Phase I semantic voxel mapping, we have ensured that the robot is capable of autonomously identifying and moving to workspaces in unstructured environments with tight time constraints. This makes our manipulation skills more realistic and grounded in the physical world, going beyond tabletop static benchmarks.

Moving ahead, there are a number of research directions that have been identified. Firstly, we would like to remove the assumption of a static environment by developing real-time continuous mapping techniques that update the semantic voxel grid in real-time, enabling the robot to handle dynamic changes such as the movement of furniture. Secondly, to further increase autonomy in communication-denied environments, we would like to distill the high-level reasoning power of cloud-based VLMs into smaller language models that can be executed on-board. Finally, although our "hard routing" approach has been successful, we would like to explore hierarchical mixture-of-experts (MoE) models to investigate "soft routing" approaches that might combine the expertise of different models for novel tasks. Additionally, inclusion of LLM synced safe planning with contextual reasoning is expected to complement the proposed architecture in uncertain scenarios [27] [27] [28] [29] [30].

Acknowledgment

For coding assistance Github Copilot[2] is used, while Gemini[3] is used for polishing author-written text; but ideas, structure, base content, flowcharts and thought process is original.

[2]https://github.com/features/copilot

[3]https://gemini.google.com

References

[1] M. J. Kim *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[2] A. Brohan *et al.*, "Rt-1-x: Cross-embodiment robotic manipulation with vla models," in *CoRL*, 2023.

[3] J. Zhang, "Autonomous navigation and collision avoidance for ground robots," GitHub repository, 2020.

[4] S. Banerjee, S. Paul, R. Roychoudhury, A. Bhattacharya, C. Sarkar, A. Sau, P. Pramanick, and B. Bhowmick, "Teledrive: an embodied ai based telepresence system," *Journal of Intelligent & Robotic Systems*, vol. 110, no. 3, p. 96, 2024.

[5] O. M. Team *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2305.16334*, 2023.

[6] H. Zhang *et al.*, "Vla-3d: A dataset for 3d semantic scene understanding and navigation," *arXiv preprint arXiv:2411.03540*, 2024.

[7] M. Minderer *et al.*, "Simple open-vocabulary object detection with vision transformers," in *ECCV*, 2022.

[8] S. Agrawal, M. Shridhar, and D. Fox, "Active scene understanding for robotic manipulation with foundation models," in *RSS*, 2024.

[9] A. Brohan *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[10] D. Driess *et al.*, "Palm-e: An embodied multimodal language model," in *ICML*, 2023.

[11] K. Black *et al.*, "Pi0: A generalist robot policy with vla architectures," *arXiv preprint arXiv:2410.00000*, 2024.

[12] P. Marza *et al.*, "Task-conditioned adaptation of visual features in multi-task policy learning," in *CVPR*, 2024.

[13] A. Rokah, D. Veress, C. Caulk, and S. Sharan, "Mixture-of-experts models in vision: Routing, optimization, and generalization," 2026. [Online]. Available: https://arxiv.org/abs/2601.15021

[14] X. Chen *et al.*, "Himoe-vla: Hierarchical mixture-of-experts for generalist vision-language-action models," *OpenReview*, 2025.

[15] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[16] M. Muglikar, Z. Zhang, and D. Scaramuzza, "Voxel map for visual slam," in *ICRA*, 2020, pp. 4181–4187.

[17] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[18] N. Zantout *et al.*, "Sort3d: Spatial object-centric reasoning toolbox for zero-shot 3d grounding using large language models," *arXiv preprint arXiv:2504.18684*, 2025.

[19] I. Armeni *et al.*, "3d scene graphs: A structure for unified semantics, 3d space, and camera," in *ICCV*, 2019.

[20] Q. Gu *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," *arXiv preprint arXiv:2309.16650*, 2023.

[21] X. Li *et al.*, "Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation," *arXiv preprint arXiv:2402.15487*, 2024.

[22] H. Yin *et al.*, "Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation," in *NeurIPS*, 2024.

[23] S. Banerjee, P. Pramanick, C. Sarkar, and B. Purushothaman, "Ontoscene: Ontology guided indoor scene understanding for cognitive robotic tasks." in *ISWC (Posters/Demos/Industry)*, 2021.

[24] S. Banerjee and B. Purushothaman, "Semnav: How rich semantic knowledge can guide robot navigation in indoor spaces." in *ISWC (Demos/Industry)*, 2020, pp. 398–400.

[25] M. Quigley *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, 2009.

[26] B. Liu *et al.*, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *arXiv preprint arXiv:2306.03310*, 2023.

[27] S. Singh, K. Swaminathan, N. Dash, R. Singh, S. Banerjee, M. Sridharan, and M. Krishna, "Adaptbot: Combining llm with knowledge graphs and human input for generic-to-specific task decomposition and knowledge refinement," in *ICRA*. IEEE, 2025, pp. 4345–4351.

[28] S. Banerjee, "Towards safe and reliable robot task planning." in *AISafety@ IJCAI*, 2020.

[29] D. Mukherjee, S. Banerjee, S. Bhattacharya, and P. Misra, "A context-aware recommendation system considering both user preferences and learned behavior," in *CITA*. IEEE, 2011, pp. 1–7.

[30] D. Mukherjee, S. Banerjee, and P. Misra, "Towards efficient stream reasoning," in *OTM ODBASE*. Springer Berlin Heidelberg Berlin, 2013, pp. 735–738.