Empowering Precise Embodied Agents with Executable Analytic Concepts as Semantic-Physical Blueprints

Mingyang Sun[1,2,4], **Jiude Wei**[2], **Qichen He**[2,3], **Donglin Wang**[4], **Cewu Lu**[2,3] and **Jianhua Sun**[3*]
[1]Zhejiang University, Hangzhou, China
[2]Shanghai Innovation Institute, Shanghai, China
[3]Shanghai Jiao Tong University, Shanghai, China
[4]Westlake University, Hangzhou, China

Abstract

A core challenge for embodied agents is the "semantic-to-physical gap"—the difficulty of mapping symbolic reasoning to precise execution. While Vision-Language Models (VLMs) enhance agent task planning, they often fail in problem classes requiring accurate alignment between functional geometry and physical constraints, such as articulated object manipulation or precision assembly. To address these challenges, we propose GRACE, an agent framework that adopts Executable Analytic Concepts (EAC) as a core knowledge annotation paradigm for object understanding. Under this paradigm, the agent does not merely perceive objects as unstructured visual data; instead, it interprets and annotates them as "Semantic-Physical Blueprints", which provide a structured cognitive representation that encodes geometric primitives, mechanical affordances, and manipulation constraints. Central to our framework is a policy scaffolding mechanism, which allows the agent to dynamically ground VLM-based insights into these instantiated blueprints. This enables the agent to model and resolve complex decision-making tasks involving parameterized object descriptions and constrained motion planning. Experimental results demonstrate that agents utilizing this paradigm achieve high robustness in complex manipulation tasks. Furthermore, we demonstrate that GRACE supports automated concept discovery, offering a scalable and modular solution for high-precision embodied intelligence.

1 Introduction

The realization of embodied agents capable of autonomous reasoning and high-fidelity physical interaction remains a fundamental objective of artificial intelligence research. Recent advancements in Large Language Models (LLMs) [Achiam *et al.*, 2023] and Multimodal Vision-Language Models (VLMs) [Zhang *et al.*, 2024; Hurst *et al.*, 2024] have endowed these agents with a sophisticated substrate of world knowledge, enabling them to interpret open-vocabulary instructions and infer complex goal states without the need for exhaustive task-specific demonstrations. This shift has catalyzed several core cognitive faculties in contemporary agentic research, most notably hierarchical task planning [Ahn *et al.*, 2022; Driess *et al.*, 2023], reflexive error detection [Duan *et al.*, 2024a], and spatial-semantic grounding [Huang *et al.*, 2025; Huang *et al.*, 2023]. Consequently, agent architectures have converged toward a dominant hierarchical paradigm, where a high-level "reasoning layer" leveraging VLMs provides semantic abstractions and task-level guidance to a low-level "execution layer" responsible for physical interaction [Ma *et al.*, 2024; Shi *et al.*, 2025].

However, a fundamental theoretical gap persists in the agent's cognitive model: the "semantic-to-physical gap". While VLMs excel at 2D image understanding and linguistic dialogue, they lack an inherent formal model of physical intelligence. Existing agent engineering methods often attempt to bridge this by either fine-tuning models into end-to-end controllers—which suffers from high data costs and poor generalization—or by feeding semantic features directly into control policies. The latter is particularly inefficient, as it forces the agent to relearn fundamental physical principles from scratch without a structured representation language to link symbolic concepts to metric-space actions. Consequently, embodied agents still struggle with the precise numerical reasoning and kinematic awareness required for tasks demanding high accuracy.

To bridge the semantic knowledge inferred by VLMs and the physical realm in which robots operate, we leverage a novel knowledge representation paradigm for embodied agents based on the notion of Analytic Concepts [Sun *et al.*, 2024]. An analytic concept serves as a procedural, mathematically defined schema that captures the generalized physical commonality of an object or task. We introduce **GRACE** (From VLM-based Grounding to Robotic manipulation through Analytic Concept Execution), an agent framework that operationalizes these concepts as semantic-physical blueprints. In this paradigm, the agent does not merely recognize an object; it annotates the environment by selecting and instantiating a mathematical blueprint from a concept library. This transforms the agent's internal reasoning from abstract chain-of-thought (COT) into an Executable Analytic Concept

*Corresponding Author

(EAC)—a physics-grounded plan whose parameters are directly compatible with motion planners. Unlike traditional scene graphs or point clouds, an EAC provides a structured, programmable representation encoding constituent parts, spatial topology, and kinematic primitives. Crucially, EACs are not just static annotations but dynamic computational templates that that undergo parameter reification—automatically estimating physical attributes from sensory inputs to translate high-level semantics into actionable physical constraints.

By mediating between symbolic reasoning and physical execution through this structured paradigm, our approach provides a systematic architectural framework for developing embodied agents with inherent interpretability and high-precision capabilities. Our main contributions are:

- **A Unified Conceptual Framework for Embodied Cognition:** We introduce GRACE, a plug-and-play agent architecture that elicits the physical intelligence of VLMs through structured, physics-aware representations. This framework provides a formalized interface between high-level cognitive reasoning and low-level physical execution, offering a new perspective on how agents internalize world knowledge.
- **The EAC Paradigm for Embodied Cognition:** We develop a systematic methodology that adopts EAC as a knowledge annotation paradigm. The policy scaffolding pipeline enables the agent to translate semantic knowledge into "Semantic-Physical Blueprints," effectively grounding symbolic reasoning into a mathematically rigorous and executable domain.
- **Extensive Empirical Validation:** We demonstrate our approach's outstanding performance in a wide range of manipulation tasks, showcasing the remarkable success rate in both simulated and real-world environments. We also highlight the compatibility of our EAC-based approach with VLA architecture.

2 Problem Formulation

The fundamental objective of an embodied agent is the reification of semantic intent into metric-space execution. We formalize this as a mapping problem where an agent must translate high-level, potentially under-specified human language into precise physical interactions.

2.1 Input and Output Spaces

At any decision epoch t, the agent is provided with:

- **Language Instruction** (l): A high-level, often abstract or under-specified natural language command.
- **Visual Observation** (V_t): A snapshot of the environment (e.g., an RGB-D images or point cloud) providing the spatial context of the relevant objects.
- **Robot Proprioception** (S_t): The robot state, including current joint configurations, end-effector pose, and gripper status. This ensures the generated actions are kinematically feasible and context-aware.

We define the aggregate observation at time t as $\mathcal{O}_t = \{V_t, S_t\}$. The agent's goal is to generate an Executable Trajectory: $\tau = \{a_0, a_1, \ldots, a_H\}$, where each action $a_k \in \mathbb{R}^7$ represents the robot's Cartesian displacement and gripper configuration. The first six dimensions encode translation and rotation vectors, while the last component $g_k \in \{0, 1\}$ indicates the binary gripper state.

2.2 The Semantic-to-Physical Mapping

The core challenge of high-precision embodied interaction lies in operationalizing the mapping function $\mathcal{M}$:

$$\mathcal{M} : (l, \mathcal{O}_t, \mathcal{C}) \to \tau \quad \text{s.t.} \quad \tau \in \Omega_\Phi$$

where $\mathcal{C}$ denotes a library of structured prior knowledge, and Ω_Φ represents the motion manifold constrained by the object's physical properties Φ. Unlike generic end-to-end approaches, $\mathcal{M}$ transitions the task from a high-dimensional regression problem to a knowledge-driven instantiation process. Central to this work is that by incorporating $\mathcal{C}$ as a repository of Analytic Concepts, the agent leverages pre-defined object-centric physical schemas to disambiguate under-specified instructions l and anchor them into the metric-geometric context of $\mathcal{O}_t$. A task is successful only if the agent navigates Ω_Φ with high precision, ensuring semantic fulfillment while strictly adhering to the mechanical invariants of the environment.

3 Executable Analytic Concept

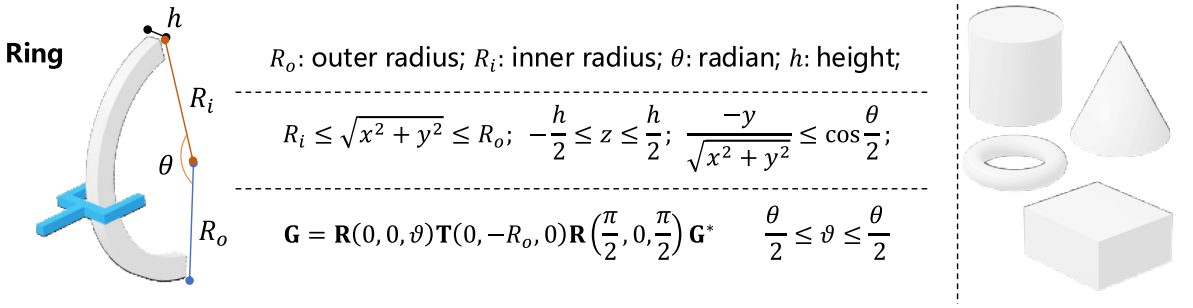
To bridge the representational gap between symbolic reasoning and metric execution, we leverage the Analytic Concept (AC) [Sun *et al.*, 2024] as a structured knowledge annotation paradigm that allows the agent to interpret the world through "Semantic-Physical Blueprints."

3.1 Formal Definition

An AC, denoted as $\mathcal{E}$, is an explicit, programmatic and hierarchical representation that formalizes the agent's internal model of an object or part. Unlike flat representations, an AC is defined recursively to capture the compositional nature of articulated systems:

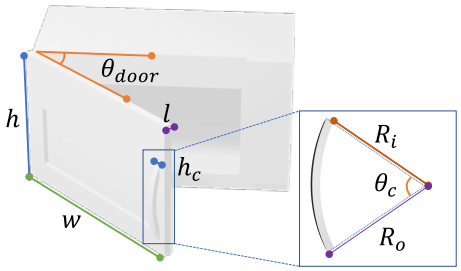
$$\mathcal{E} = \langle \mathcal{P}, \mathcal{T}, \Phi \rangle$$

- Constituent Parts ($\mathcal{P}$): A set of lower-level Part ACs $\{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n\}$. At the lowest level of this hierarchy, the part is grounded as a Geometric Concept in Fig. 1a, defined by parametric surfaces $S(p; \theta_g) = 0$ (e.g., a cylinder for a handle or a plane for a cabinet face).
- Spatial Topology ($\mathcal{T}$): A set of transformations $\{T_{ij}\}$ that define the spatial configuration and relative poses between constituent parts in $\mathcal{P}$. This ensures the structural integrity of the composite object (e.g., the handle's fixed offset relative to the door).
- Kinematic Primitive (Φ): A library of Intention-Conditioned Guidance Functions that instantiate specific control laws $\mathbf{u}(t)$ based on the interaction semantics. For complex $\mathcal{E}$, the root-level Φ is generated by inheriting and composing part-level functions and their associated parameters.



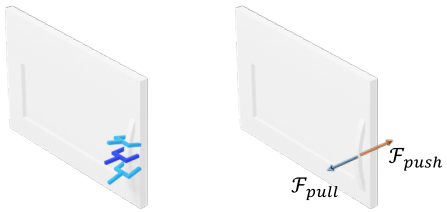
(a) Geometric Concept Assets

```python
# Structural Blueprint Code
class Curve_Handle:
    def __init__(self, curveL, radian, innerRadius, outerRadius
        , rotation, position, ...):
        self.shape = Ring(curveL, radian, innerRadius,
            outerRadius)
        ...
class Sunken_Door():
    def __init__(self, size, sunken_size, handle_size,
        handle_pos, rotation, position, ...):
        self.bottom_shape = Cuboid(size)
        self.top_shape = Rectangular_Ring(sunken_size)
        self.handle = Curve_Handle(handle_size, handle_pos)
```



(b) Structural Blueprint

```python
# Manipulation Blueprint Code
def get_grasp_pose(obj, initial_pose):
    if (isinstance(obj, Curve_Handle)):
        local_pose = translate_world2local(pose, init_pose)
        grasp_pose = obj.apply_pose(local_pose)
    if (isinstance(obj, Sunken_Door)):
        ...
    return translate_local2world(grasp_pose)
def push(obj):
    force_direction = get_direction(obj.axis...)
def pull(obj):
    force_direction = get_direction(obj.axis...)
```



(c) Manipulation Blueprint

Figure 1: Example of analytic concepts. (a) Geometric Concept Assets. Each asset exposes its free parameters (top), canonical structure (mid), and partial affordance cues (bottom). (b) Structural Blueprint: higher-level objects or parts are procedurally composed by wiring multiple geometric assets together, forming a parametric graph with their spatial layout and structural relationships. (c) Manipulation Blueprint: parameterised routines compute grasp poses and force directions that exploit the affordances encoded in the underlying structure.

3.2 Blueprint Instantiation

The AC serves as a programmatic template. To bridge the gap between abstract knowledge and physical interaction, the agent performs Knowledge Annotation to instantiate a task-specific Semantic-Physical Blueprint $\mathcal{B}$ as an Executable Analytic Concept (EAC). Unlike static templates, the instantiation of $\mathcal{B}$ is a dynamic mapping conditioned on the current task l and visual observation V_t:

$$\mathcal{B} = \text{Instantiate}(\mathcal{E}, V_t, l).$$

To achieve this, the agent need to identifies the task-relevant constituent parts $\mathcal{P}_\mathcal{B} \subseteq \mathcal{P}$ (see Fig. 1b) and specific procedural guidance function $\mathbf{u}_\mathcal{B}$ (see Fig. 1c). By extracting metric features from V_t via neural regressor tools, the agent instantiates the Geometric Parameters and Transition Parameters. This blueprint acts as a policy scaffold, providing the necessary geometric and physical "rails" that guide the agent's low-level motion planner to generate the final trajectory τ. Specifically, by generalizing the concepts towards certain objects, various knowledge associated with the concepts can be automatically propagated to all these objects.

4 The Agent Framework

4.1 Agentic Workflow Overview

The GRACE agent orchestrates a hierarchical decision-making pipeline that bridges abstract intent with deterministic physical control. As illustrated in Fig. 2, the workflow is divided into three cognitive stages:

- Task Parsing: The agent resolves linguistic ambiguity by grounding instructions in the visual scene.
- Policy Scaffolding: The agent performs a sequence of specialized tool calls to construct an Executable Ana-

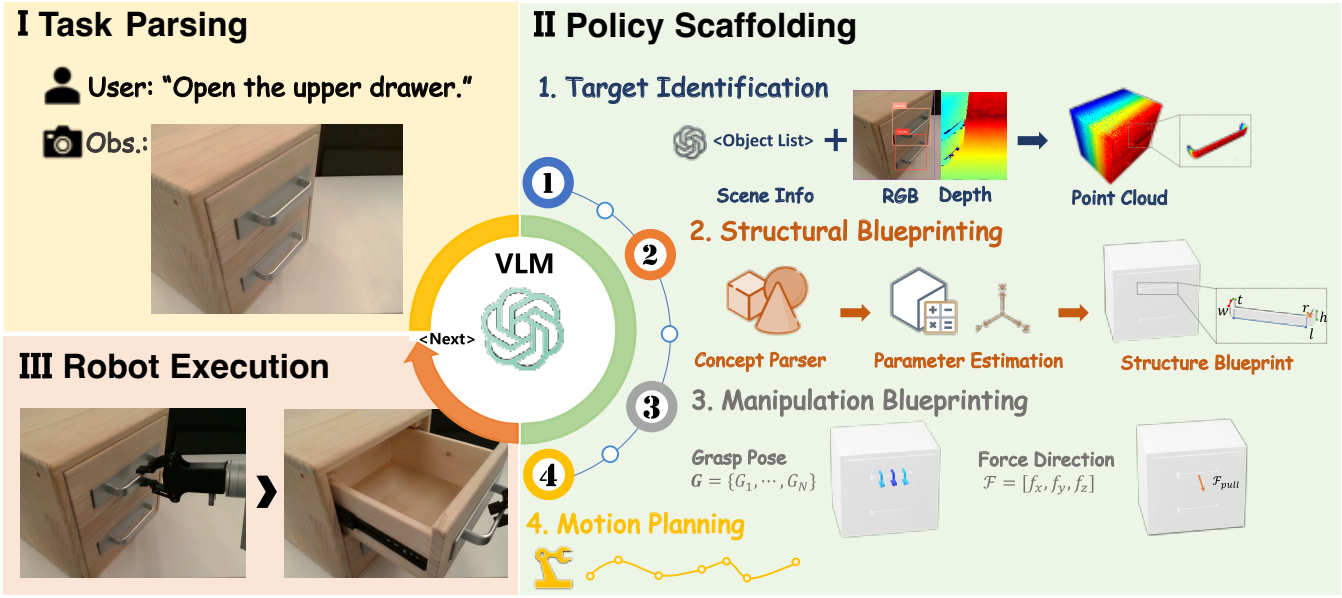


Figure 2: An overview of the proposed method GRACE. (I) **Task Parsing**: A Vision–Language Model (VLM) parses the natural-language instruction based on the current RGB image. (II) **Policy Scaffolding**: The process includes: 1. segmenting the target object or part from images and back-projecting it to a partial point cloud; 2. parsing the analytic concept and estimating geometric parameters to instantiate the structural blueprint; 3. constructing the manipulation blueprint to produce feasible grasp poses and force directions; 4. generating a joint-space trajectory via a motion-planning module. (III) **Robot Execution**: The trajectory is executed to complete the task.

lytic Concept (EAC). This transforms "insight" into a structured, parameter-driven blueprint.

- Policy Execution: The agent exports the EAC's computational parameters to a downstream policy model for low-level motor control.

We give the pseudocode of the overall process in the appendix.

4.2 Spatial-Aware Task Parsing

Object Centric Symbolic Graphing. As the foundational stage for perception and language grounding, the agent first interprets the l by constructing a structured graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of object nodes—each represented as a structured dictionary containing id, name, and state—and $\mathcal{E}$ constitutes a set of directed spatial relationships between objects, each expressed as a triple $e_{ij} = (v_i, r, v_j)$. This object-centric symbolic graph provides a semantically rich and structurally explicit representation for subsequent reasoning stages. This process distills the "what" and "where" from the command, delivering a clean symbolic input for downstream reasoning.

Hierarchical Task Decomposition. For long-horizon tasks, the agent functions as a high-level planner, decomposing the primary task into discrete sub-tasks $\{(l_1, d_1), \ldots, (l_n, d_n)\}$. Each sub-task is associated with a computable verification condition d_i, ensuring the agent can autonomously validate successful completion before proceeding. Each sub-task then undergoes an execution loop, as depicted in Fig. 2.

4.3 Policy Scaffolding

The Policy Scaffolding stage represents the core cognitive layer of the GRACE agent. In this phase, the agent transitions from semantic reasoning to physical grounding by executing a series of inter-tool protocols that construct the Executable Analytic Concept (EAC). This process is governed by a structured workflow that transforms raw observation into instantiated physical blueprints.

Target Identification. Upon receiving the structured object graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the agent initiates the Perceptual Grounding Protocol. The agent ground its geometric structure in a formalized representation by interfacing with a Library of Analytic Concepts. This library acts as an Analytic Knowledge Base containing parametric models of structural archetypes. This involves calling a sequence of Visual Foundation Tools:

- **Semantic Concept Retrieval:** The agent interfaces with a Library of Analytic Concepts, which serves as an organized repository of common structural archetypes (e.g., primitive geometries, specific handle designs). Each concept is paired with a short natural-language synopsis. Based on the sub-task instructions identified in the task parsing stage, the agent prunes the library and performs a VLM-driven query to match the visual part to its corresponding symbolic archetype.
- **Open-Vocabulary Localization:** The agent invokes GroundingDINO [Liu *et al.*, 2024] to identify and localize target part or object nodes within the scene.
- **Instance Segmentation Tool:** The Segment Anything Model (SAM) [Kirillov *et al.*, 2023] is triggered to gen-

erate high-fidelity 2D masks M for task-relevant entities.

- **3D Back-Projection:** Using the depth map, the agent executes a geometric tool call to back-project M into the 3D space, producing object-centric point clouds P.

Structural Parameter Estimation. Once the symbolic concept is retrieved and the point cloud is segmented, the agent transitions to Parameter Reification. This process transforms the abstract concept into an executable program by estimating concrete metric variables from the point cloud P. These parameters are of two types:

- **Structural parameters** encode the concept's intrinsic geometry of the analytic concept (e.g., the size l, w, h of a sunken door). To estimate them, we encode the point cloud P into a deep feature vector using a Point Transformer [Zhao *et al.*, 2021]. This feature vector is then fed into multiple specialized MLP heads, each regressing a specific structural parameter.
- **6-DoF pose parameters** locate the concept's global position and orientation. We leverage the FoundationPose [Wen *et al.*, 2024] to estimate the 6-DoF pose of the target part in the global coordinate system. These are recovered analytically by combining the object's known simulation pose with the newly estimated structural variables.

Manipulation Blueprinting. While the structural blueprint defines the geometry, the Manipulation Blueprinting protocol specifies the functional "how-to". The agent treats manipulation knowledge as an Executable API. Firstly, the agent retrieves candidate Manipulation Functions (e.g., "pull-type grasp") from the knowledge base. The VLM selects the module that best satisfies the high-level intent, mapping semantic perception directly onto a mathematical interaction law.

With the parameters, a physically grounded grasp pose $\mathbf{G}$ can be calculated according to the initial grasp pose $\mathbf{G}^*$ by solving the analytic equations of the selected module. For instance, a curved handle's interaction is governed by the transformation:

$$\mathbf{G} = \mathbf{R}(0,0,\vartheta)\mathbf{T}(0,-R_o,0)\mathbf{R}(\frac{\pi}{2},0,\frac{\pi}{2})\mathbf{G}^*, \quad -\frac{\theta_c}{2} \le \vartheta \le \frac{\theta_c}{2}.$$

Once $\mathbf{G}$ is fixed, the force-direction formula—conditioned by the verb or manipulation type chosen by the VLM (e.g., *pull* vs. *push*)—is invoked to produce the vector $\mathcal{F}$, ensuring that the applied force is semantically aligned with the selected action and correctly oriented on the target part. The final output of the scaffolding process is an instantiated EAC, exported as a lightweight Python Object. This serves as the Standard Execution Protocol for the downstream motion planner, closing the loop from abstract language to deterministic control.

4.4 Low-Level Motion Execution

Blueprint Execution. The structural and manipulation blueprints jointly output two quantities in the *local* frame of the target part: a grasp pose $\mathbf{G}_{\text{local}} = (\boldsymbol{t}_{\text{local}}, \boldsymbol{r}_{\text{local}})$, and a force direction $\mathcal{F}_{\text{local}}$. Running the blueprint therefore reduces to transforming the local descriptors into the world frame and then feeding them to a standard motion planning stack.

Transformation to World Coordinates. Let $\mathbf{M} \in \mathbb{R}^{4\times4}$ denote the homogeneous transform of the target part with respect to the world frame, obtained from perception or simulation. For every point–set or inequality description F in the blueprint we apply $F\big((x,y,z,1)^\top\big) \le 0 \implies F\big(\mathbf{M}^{-1}(x,y,z,1)^\top\big) \le 0$, thereby re-expressing all structural constraints globally. The grasp pose is mapped by $\mathbf{G}_{\text{world}} = \mathbf{M}\mathbf{G}_{\text{local}}$. For rotationally symmetric geometries we additionally enforce a minimal-rotation constraint on $\boldsymbol{r}_{\text{local}}$ to obtain a unique orientation. The force vector is transformed analogously: $\mathcal{F}_{\text{world}} = \mathbf{R}\,\mathcal{F}_{\text{local}}$, where $\mathbf{R}$ is the rotational part of $\mathbf{M}$.

Motion Planning Interface. The instantiated EAC parameters are forwarded to a policy model (e.g., a standard motion-planning stack or a learned neural controller). This model realizes the final joint-space command sequence by synthesizing a collision-free path that strictly adheres to the force vector $\mathcal{F}_{\text{world}}$ and pose $\mathbf{G}_{\text{world}}$. By decoupling the analytic "what" from the motor "how", GRACE enables high-precision execution across diverse hardware platforms.

5 Experiments

In this section, we conduct comprehensive experiments and analyses to answer the following key questions: (1) How does the GRACE framework perform against state-of-the-art baselines across diverse manipulation tasks? (2) To what extent can the agent accurately discover the analytic concept and ground their functional affordances in complex articulated objects? (3) Can the integrated agentic pipeline successfully execute long-horizon, multi-stage tasks in real-world settings?

5.1 Manipulation Evaluation in Simulation

To rigorously evaluate the execution efficacy of the GRACE framework, we leverage SimplerEnv [Li *et al.*, 2024c] as our primary simulation benchmark. SimplerEnv is chosen for its high-fidelity physical modeling and its standardized suite of real-world robotic manipulation tasks, which provide a reliable proxy for physical hardware constraints. We conduct extensive quantitative comparisons on Google Robot and Widow-X task suites, benchmarking GRACE against state-of-the-art Vision-Language-Action (VLA) models including Octo [Ghosh *et al.*, 2024], OpenVLA [Kim *et al.*, 2024], and several concurrent spatial-aware baselines [Qi *et al.*, 2025; Qu *et al.*, 2025; Li *et al.*, 2024b]. This evaluation specifically targets the agent's ability to transcend the precision limitations of end-to-end black-box models by utilizing the EAC-based decision framework.

On the four Widow-X tasks (Table 1), GRACE powered by GPT-4o achieves an average success rate of 86.1%, clearly outperforming the strongest published baseline, SoFar (58.3%). Although it is not the best on every single task, GRACE never performs poorly, maintaining consistently high scores across the entire suite. The pattern repeats on the Google-robot tasks (Table 2): GRACE(GPT-4o) attains 90.1% mean success, exceeding the best prior result by almost 30 pp. Notably, on the articulated Open/Close Drawer task the jump is the largest, rising from 29.7% (SoFar) and

Model	**Put Spoon on Towel**		**Put Carrot on Plate**		**Stack Green Block on Yellow**		**Put Eggplant in Basket**		**Avg**
	Grasp Spoon	Success	Grasp Carrot	Success	Grasp Block	Success	Grasp Eggplant	Success	
RT-1-X	16.7%	0.0%	20.8%	4.2%	8.3%	0.0%	0.0%	0.0%	1.1%
Octo-small	77.8%	47.2%	27.8%	9.7%	40.3%	4.2%	87.5%	56.9%	30.0%
OpenVLA	4.1%	0.0%	33.3%	0.0%	12.5%	0.0%	8.3%	4.1%	1.0%
RoboVLM	37.5%	20.8%	33.3%	25.0%	8.3%	8.3%	0.0%	0.0%	13.5%
RoboVLM (FT)	54.2%	29.2%	25.0%	25.0%	45.8%	12.5%	58.3%	58.3%	31.1%
SpatialVLA	25.0%	20.8%	41.7%	20.8%	58.3%	25.0%	79.2%	70.8%	34.4%
SpatialVLA (FT)	20.8%	16.7%	29.2%	25.0%	62.5%	29.2%	**100.0%**	**100.0%**	42.7%
SoFar	62.5%	58.3%	75.0%	66.7%	**91.7%**	70.8%	66.7%	37.5%	58.3%
SpatialVLA-EAC	**91.7%**	**87.5%**	79.2%	62.5%	75.0%	50.0%	79.2%	79.2%	69.8%
GRACE(Qwen2.5-VL)	83.3%	83.3%	**79.2%**	**79.2%**	87.5%	83.3%	91.7%	91.7%	84.4%
GRACE(GPT-4o)	83.3%	83.3%	**79.2%**	**79.2%**	87.5%	**87.5%**	95.8%	95.8%	**86.1%**

Table 1: SimplerEnv simulation evaluation results for the WindowX Robot task. We report both the final success rate ("Success") along with partial success (e.g., "Grasp Spoon"). "FT" denotes performance of the fine-tuned models.

Model	Variant Aggregation			Visual Matching			**Avg**
	Pick Coke Can	**Move Near**	**Open/Close Drawer**	**Pick Coke Can**	**Move Near**	**Open/Close Drawer**	
RT-1-X	49.0%	32.3%	29.4%	56.7%	31.7%	59.7%	43.1%
Octo-Base	0.6%	3.1%	1.1%	17.0%	4.2%	22.7%	8.11%
OpenVLA	54.5%	47.7%	17.7%	16.3%	46.2%	35.6%	36.3%
RoboVLM	68.3%	56.0%	8.5%	72.7%	66.3%	26.8%	49.8%
RoboVLM(FT)	75.6%	60.0%	10.6%	77.3%	61.7%	43.5%	54.8%
SpatialVLA	89.5%	71.7%	36.2%	81.0%	69.6%	59.3%	67.9%
SpatialVLA(FT)	88.0%	72.7%	41.8%	86.0%	77.9%	57.4%	70.6%
SoFar	90.7%	74.0%	29.7%	**92.3%**	**91.7%**	40.3%	69.6%
SpatialVLA-EAC	88.9%	77.9%	83.3%	86.1%	79.2%	85.4%	83.4%
GRACE(Qwen2.5-VL)	90.3%	87.5%	88.9%	91.7%	88.9%	84.7%	88.7%
GRACE(GPT-4o)	**91.7%**	**87.5%**	**90.3%**	90.3%	**91.7%**	**88.9%**	**90.1%**

Table 2: SimplerEnv simulation evaluation results for the Google Robot setup. We present success rates for the "Variant Aggregation" and "Visual Matching" approaches. "FT" denotes performance of the fine-tuned models.

36.2% (SpatialVLA) to 90.3% with GRACE for "Variant Aggregation", highlighting the advantage of EACs when precise kinematic reasoning is required.

To isolate the contribution of analytic concepts, we retrofit SpatialVLA by replacing its native, end-to-end action output with EAC-guided motion planning when the gripper approaches the target; this variant is denoted *SpatialVLA-EAC*. The simple swap boosts SpatialVLA's average success to 69.8% on Widow-X and to 83.4% on the Google robot, demonstrating that EACs can be used as a plug-and-play module to substantially enhance existing VLA architectures. Finally, GRACE's performance is insensitive to the underlying VLM. The fully open-source Qwen2.5-VL backend trails GPT-4o by only 1–2 pp on both robot families, yet still outperforms every external baseline, confirming that the bulk of the gain comes from the analytic-concept layer rather than the choice of language model.

5.2 Evaluation of Concept Grounding

This experiment evaluates the agent's capacity to perform analytic concept discovery and functional affordance grounding within complex articulated objects. Unlike general manipulation, success in this context requires the agent to correctly identify the structural blueprints and adhere to the resulting kinematic manifolds. We conduct these evaluations within the SAPIEN environment, utilizing the standardized settings established by Where2Act [Mo *et al.*, 2021]. The task requires the agent to transition an articulated object (e.g., cabinets, faucets, drawers) from an initial state to a target configuration. We compare our method against three baselines, i.e., Where2Act, Where2Explore [Ning *et al.*,] and ManipLLM [Li *et al.*, 2024a], each representative of a distinct modelling paradigm for articulated–object manipulation. To isolate the contribution of VLM reasoning, we also report an ablated variant, GRACE-w/o-VLM, in which the concept-selection step is replaced by ground-truth ontology labels.

Quantitative Results. Table 3 demonstrates that GRACE(GPT-4o) achieves the highest scores across all categories. For instance, it attains 0.65 for "faucet" objects and 0.91 on "cabinet" doors, significantly outperforming ManipLLM, which scores 0.26 and 0.71, respectively.

Objects						
Where2Act	0.14	0.68	0.27	0.23	0.15	0.15
UMPNet	0.44	0.54	0.28	0.54	0.28	0.25
ManipLLM	0.65	0.71	0.77	0.43	0.65	0.26
w/o-VLM	**0.85**	**0.91**	**0.90**	**0.70**	**0.78**	**0.65**
(GPT-4o)	0.84	0.85	0.88	**0.70**	0.72	0.60

Table 3: Comparison of performance on different objects (icons represent object categories).

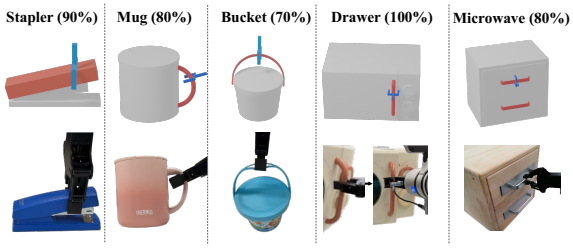


Figure 3: Visualize the grasping outcomes alongside their corresponding EACs with target parts highlighted in red.

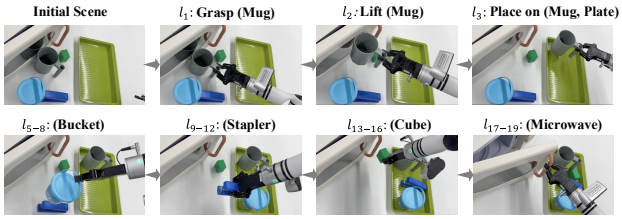


Figure 4: Visualization of a long-horizon task execution. Task: *Place the 4 tabletop objects onto the plate and open the microwave.*

These results decisively surpass both pixel-level affordance methods and the LLM-based ManipLLM. The substantial numerical margins underscore the advantage of integrating VLM-based reasoning with analytically grounded control.

Replacing the oracle concept labels with GPT-4o's automatic selection reduces performance only marginally—from an average of 0.80 to 0.77. This negligible 3% drop indicates that the bottleneck for the remaining failure cases is primarily localized within occasional semantic misclassifications by the VLM, rather than inherent limitations of the analytic concepts themselves. Once the correct concept is retrieved, the resulting execution remains highly deterministic and reliable.

Qualitative Insights in Real World. To validate our approach qualitatively, we illustrate the execution success and generated trajectories for five distinct articulated objects in Fig. 3. These real-world outcomes demonstrate the effectiveness of EAC for physics-grounded planning. The visualizations affirm that the agent's cognitive framework functions as intended: the Visual Language Model successfully completes Part Identification and EAC Construction, while the underlying manipulation blueprints manage the complexities of physical compliance. This modular design enables the robot to perform high-precision interactions consistently.

5.3 Long-Horizon Tasks in Real-world

To validate the deployment-readiness and generalization of the GRACE framework, we conducted real-world experiments in a tabletop environment using a Realman RM75 robotic arm equipped with a parallel gripper. To rigorously test the system's integration, we designed a long-horizon manipulation tasks involving diverse objects. This task requires the agent to maintain coherent semantic reasoning while executing a series of precise, multi-stage physical interactions. As illustrated in Fig. 4 , the agent autonomously decomposed the high-level goal into a sequence of 19 distinct instructions. The successful execution of this long-horizon chain underscores the efficacy of our decoupled architecture: the VLM ensures inter-stage semantic continuity, while the EACs guarantee intra-stage kinematic adherence.

6 Related Work

Foundation Model Paradigms in Robotics. Current research leveraging foundation models for manipulation can be categorized into two primary paradigms. The first paradigm employs monolithic end-to-end Vision-Language-Action (VLA) models [Kim *et al.*, 2024; Ghosh *et al.*, 2024; Shao *et al.*, 2025] that directly map multimodal observations to low-level actions, though they are often bottlenecked by massive data requirements and limited zero-shot precision. The second paradigm adopts a hierarchical structure where a high-level VLM generates intermediate representations—such as symbolic code, sparse keypoints, or discrete skill selections—to guide a separate low-level policy [Ahn *et al.*, 2022; Huang *et al.*, 2023; Duan *et al.*, 2024b; Huang *et al.*, 2024; Pan *et al.*, 2025]. GRACE focuses on how an agent can establish a reliable intermediate structured representation through Executable Analytic Concepts. By formalizing these concepts, our framework provides the deterministic, numerically computable geometric and physical "rails" necessary to bridge fluid semantic reasoning with precise, kinematically compliant execution. This structured alignment ensures that the agent's high-level intent is grounded in the object's inherent mechanical manifold, overcoming the sparsity and lack of physical depth in existing intermediate representations.

7 Conclusion

In this paper, we introduced GRACE, a plug-and-play agentic framework that bridges the gap between high-level semantic reasoning and high-precision physical interaction. By grounding visual observations into Executable Analytic Concepts, GRACE effectively translates open-vocabulary instructions into kinematically compliant robot actions. Extensive experiments on simulation and real world demonstrate marked gains in manipulation success rates, particularly on kinematically challenging tasks. In future work, we plan to extend GRACE to multi-fingered hands and to explore on-the-fly concept refinement from real-world interaction data.

Acknowledgments

The authors declare that this research received no external funding.

References

[Achiam *et al.*, 2023] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[Ahn *et al.*, 2022] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[Driess *et al.*, 2023] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. 2023.

[Duan *et al.*, 2024a] Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation. *arXiv preprint arXiv:2410.00371*, 2024.

[Duan *et al.*, 2024b] Jiafei Duan, Wentao Yuan, Wilbert Pumacay, Yi Ru Wang, Kiana Ehsani, Dieter Fox, and Ranjay Krishna. Manipulate-anything: Automating real-world robots using vision-language models. In *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 5326–5350. PMLR, 2024.

[Ghosh *et al.*, 2024] Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.

[Huang *et al.*, 2023] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

[Huang *et al.*, 2024] Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024*, pages 9488–9495. IEEE, 2024.

[Huang *et al.*, 2025] Haifeng Huang, Xinyi Chen, Yilun Chen, Hao Li, Xiaoshen Han, Zehan Wang, Tai Wang, Jiangmiao Pang, and Zhou Zhao. Roboground: Robotic manipulation with grounded vision-language priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22540–22550, 2025.

[Hurst *et al.*, 2024] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[Kim *et al.*, 2024] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024.

[Kirillov *et al.*, 2023] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[Li *et al.*, 2024a] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18061–18070, 2024.

[Li *et al.*, 2024b] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *CoRR*, abs/2412.14058, 2024.

[Li *et al.*, 2024c] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 3705–3728. PMLR, 2024.

[Liu *et al.*, 2024] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[Ma *et al.*, 2024] Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*, 2024.

[Mo *et al.*, 2021] Kaichun Mo, Leonidas J. Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 6793–6803. IEEE, 2021.

[Ning *et al.*,] Chuanruo Ning, Ruihai Wu, Haoran Lu, Kaichun Mo, and Hao Dong. Where2explore: Few-shot affordance learning for unseen novel categories of articulated objects. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

[Pan *et al.*, 2025] Mingjie Pan, Jiyao Zhang, Tianshu Wu, Yinghao Zhao, Wenlong Gao, and Hao Dong. Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 17359–17369. Computer Vision Foundation / IEEE, 2025.

[Qi *et al.*, 2025] Zekun Qi, Wenyao Zhang, Yufei Ding, Runpei Dong, Xinqiang Yu, Jingwen Li, Lingyun Xu, Baoyu Li, Xialin He, Guofan Fan, Jiazhao Zhang, Jiawei He, Jiayuan Gu, Xin Jin, Kaisheng Ma, Zhizheng Zhang, He Wang, and Li Yi. Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. *CoRR*, abs/2502.13143, 2025.

[Qu *et al.*, 2025] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[Shao *et al.*, 2025] Rui Shao, Wei Li, Lingsen Zhang, Renshan Zhang, Zhiyang Liu, Ran Chen, and Liqiang Nie. Large vlm-based vision-language-action models for robotic manipulation: A survey. *arXiv preprint arXiv:2508.13073*, 2025.

[Shi *et al.*, 2025] Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025.

[Sun *et al.*, 2024] Jianhua Sun, Yuxuan Li, Longfei Xu, Nange Wang, Jiude Wei, Yining Zhang, and Cewu Lu. Conceptfactory: Facilitate 3d object knowledge annotation with object conceptualization. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[Wen *et al.*, 2024] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 17868–17879. IEEE, 2024.

[Zhang *et al.*, 2024] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024.

[Zhao *et al.*, 2021] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.