# CoRAL: Contact-Rich Adaptive LLM-based Control for Robotic Manipulation

Berk Cicek*, Mert K. Er*, Ozgur S. Oguz
LiRA Lab, Department of Computer Engineering
Bilkent University, Ankara, Türkiye
Emails: {berk.cicek, kaan.er, ozgur.oguz}@bilkent.edu.tr
*These authors contributed equally to this work.

***Abstract*—While Large Language Models (LLMs) and Vision-Language Models (VLMs) demonstrate remarkable capabilities in high-level reasoning and semantic understanding, applying them directly to contact-rich manipulation remains a challenge due to their lack of explicit physical grounding and inability to perform adaptive control. To bridge this gap, we propose CoRAL (Contact-Rich Adaptive LLM-based control), a modular framework that enables zero-shot planning by decoupling high-level reasoning from low-level control. Unlike black-box policies, CoRAL utilizes Large Language Models (LLMs) not as direct controllers, but as cost designers that synthesize context-aware objective functions for a sampling-based motion planner (MPPI). To address the ambiguity of physical parameters in visual data, we introduce a neuro-symbolic adaptation loop: a Vision-Language Model provides semantic priors for environmental dynamics (e.g., mass, friction estimates), which are then explicitly refined in real-time via online system identification, while the LLM iteratively modulates the cost function structure to correct strategic errors based on interaction feedback. Furthermore, a retrieval-based memory unit allows the system to reuse successful strategies across recurrent tasks. This hierarchical architecture ensures real-time control stability by decoupling high-level semantic reasoning from reactive execution, effectively bridging the gap between slow LLM inference and dynamic contact requirements. We validate CoRAL on both simulation and real-world hardware across challenging and novel tasks, such as "flipping objects against walls" leveraging extrinsic contacts. Experiments demonstrate that CoRAL outperforms state-of-the-art VLA and foundation-model-based planner baselines by boosting success rates over 50% on average in unseen contact-rich scenarios, effectively handling sim-to-real gaps through its adaptive physical understanding. Website: *https://sites.google.com/view/lira-coral***

## I. Introduction

Foundational models have demonstrated significant success in various fields, leading to increased efforts to apply these models within robotics [4, 25]. Typically, these models are integrated into robotic control pipelines through **imitation learning**, most notably in the form of Vision-Language-Action (VLA) systems [22, 34, 24]. However, existing VLA frameworks struggle to effectively handle contact-rich manipulation tasks, which constitute a substantial portion of daily interactions [7, 30, 29]. In this work, we specifically refer to a challenging subset of these tasks as contact- and force-critical, where success depends not merely on geometric collision avoidance, but on the precise, active regulation of extrinsic contact forces to manipulate objects (e.g., pivoting against a wall - Fig. 1). These tasks pose significant challenges, as they require not only precise trajectory planning but also sophisticated interaction force management and adaptive control strategies. Achieving success in such complex scenarios typically necessitates extensive training through teleoperation or detailed dynamic modeling, methods that are labor-intensive and reduce generalizability.

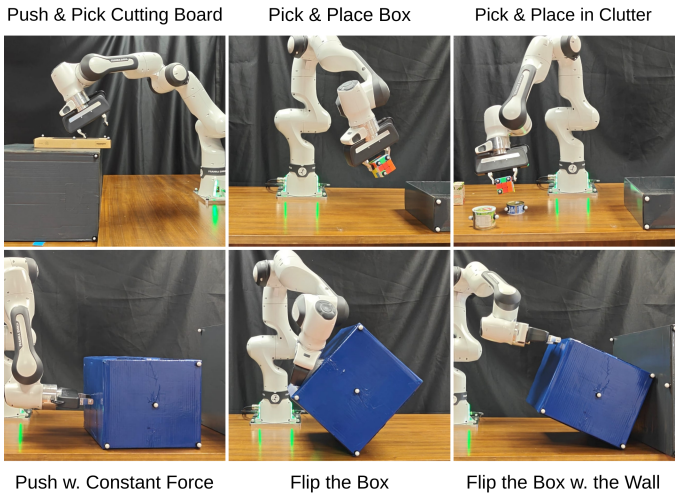


Fig. 1: Real-world execution of CoRAL across six different manipulation tasks.

Humans, by contrast, rely on initial estimations, subsequently refine their strategies based on sensory feedback, and adjust interactions accordingly [5, 11, 14]. Similar to this cognitive framework, we propose a novel modular system, **Contact-Rich Adaptive LLM-based Control (CoRAL)**, that integrates reasoning, planning, and control modules into a cohesive architecture. Our model begins by estimating 6-DoF object poses from RGB-D data using FoundationPose [27], and then a VLM generates *semantic beliefs* regarding physical parameters such as mass and friction from the estimated object poses, the environment image, and the textual task description (Fig. 2). The planning stage generates initial contact strategies and actions, which are executed in the evaluation environment through reactive control modules. The outcomes from these actions are continuously monitored, with the interaction history and control feedback being used for iterative refinement of plans.

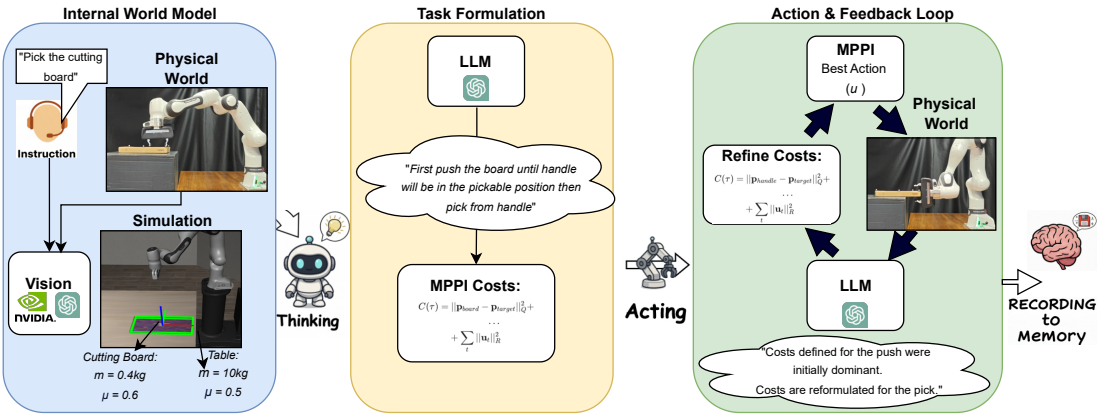


Fig. 2: The conceptual workflow of CoRAL, illustrated with the "pick the cutting board" task.

A key innovation of our approach is the strategic integration of vision and Large Language Models (LLMs) with motion planners and controllers, substantially enhancing action explainability and enabling more effective reasoning. Our modular structure clearly delineates roles: vision module manages parameter estimation for environment modeling, while LLM provides symbolic reasoning, initial contact strategies, and cost estimations. The reactive controller then applies these symbolic outputs in the evaluation world, establishing a tight feedback loop between high-level strategy and low-level sensory information. Our main contributions are:

- We propose a novel framework that employs Large Language Models (LLMs) not as direct controllers, but as context-aware *cost designers* for a high-frequency motion planner, enabling zero-shot planning for dynamic, contact- and force-critical manipulation.
  We introduce a modular architecture that decouples perception (providing semantic physics priors) from reasoning (formulating control objectives). This separation offers a transparent and adaptive alternative to monolithic VLA architectures, allowing the system to maintain robustness even when initial visual estimates are imperfect.
- We develop a hierarchical closed-loop mechanism where reactive control ensures contact stability, while a slower LLM-driven loop performs online system identification to adapt physical parameter beliefs and strategy mid-execution.
- Our framework's long-term adaptability is enhanced by a retrieval-based memory unit that stores successful physical parameters and contact strategies to bootstrap effective solutions for novel, yet semantically similar tasks.

We evaluate CoRAL on a challenging suite of manipulation tasks, including novel contact- and force-critical problems, such as picking up a thin object from a table and standardized benchmarks from the LIBERO suite [18]. Furthermore, we validate the system's robustness against real-world physical uncertainties by deploying it on a physical robot. Our experiments and detailed ablation studies confirm that this modular structure offers a robust, data-efficient alternative for contact-rich manipulation, providing explainability and adaptability in zero-shot regimes where end-to-end models struggle.

## II. Related Work

**From End-to-End Policies to Decoupled Reasoning.** Foundation models have shifted robotic manipulation towards VLA models learning general-purpose policies [4, 22]. Leading examples like OpenVLA [13], $\pi_0$ [2], and RT-X [23] directly map multimodal inputs to low-level actions. While powerful, their reliance on imitation learning makes them data-dependent and often brittle in novel physical scenarios involving complex contact dynamics. To overcome this, emerging frameworks decouple high-level reasoning from low-level control. ThinkAct [8], Inner Monologue [9], and ECoT [32] leverage LLMs to generate reasoning steps guiding separate, learned policies. Similarly, MolmoAct [15] produces mid-level spatial plans, while OneTwoVLA [16] formalizes this as System 1 (acting) and System 2 (reasoning). CoRAL aligns with this decoupling but takes a distinct neuro-symbolic path: we ground LLM reasoning directly in a controller rather than a learned model.

**Integrating Foundation Models with Motion Planners and Controllers.** Alternatively, foundation models can guide traditional motion planners using semantic understanding. VoxPoser [10] and IMPACT [17] use VLMs to generate static 3D cost maps for planners like MPPI or RRT*, while VLMPC [33] embeds a VLM within an MPC [6] loop. In code generation, Eureka [20] and DrEureka [21] synthesize RL reward functions but operate *offline*. Closest to our approach, Language-to-Rewards (L2R) [31] generates rewards for *real-time* MPC. However, L2R relies on static physical assumptions and lacks mechanisms to correct model mismatches during execution. CoRAL advances this by elevating the LLM to a high-level strategist capable of online adaptation. Rather than merely identifying goals, our LLM formulates the structure of the MPPI [28] cost function and symbolic contact strategies, grounding commonsense reasoning directly into the optimal control problem.

**Tackling Contact-Rich Manipulation.** Contact-rich manipulation requires nuanced force control beyond simple trajectory generation. Recent works like ForceVLA [30], TLA [7], VLA-Touch [1], RDP [29], and FACTR [19] explicitly integrate force or tactile data into learned policies. While effective, this hardware-centric approach creates a data bot-

tleneck, requiring difficult-to-collect specialized multimodal datasets [4]. CoRAL leverages real-time force feedback within the MPPI controller but eliminates the need for prior demonstration datasets. We use the LLM to formulate high-level strategies and cost functions explicitly reasoning about forces, which the controller executes adaptively. This neuro-symbolic approach combines physical sensing with zero-shot reasoning, avoiding the imitation learning bottleneck while achieving precise, force-aware control.

## III. Methodology

CoRAL is a neuro-symbolic framework designed for zero-shot, contact-rich manipulation. It strategically decouples high-level reasoning from low-level control by integrating a vision pipeline that continuously tracks object poses and enriches the world model with semantic physics priors inferred by the VLM, an LLM acting in two distinct roles (Task Formulation and Online Adaptation), a Memory Unit for experience retrieval, and a Model Predictive Path Integral controller (MPPI) for reactive execution. The overall architecture, which features nested feedback loops for rapid and robust adaptation, is illustrated in Figure 3. Below, we detail each component of this architecture.

### A. Environment Perception and World Model Initialization

The first step is to translate raw visual, textual, and geometric inputs into a structured, physics-aware world model. Our perception pipeline achieves this through a two-stage process that first establishes the geometric state of the scene and then initializes it with semantic physical beliefs. The process involves two core steps:

1) **Pose Estimation and Tracking:** We employ **FoundationPose** [27], a state-of-the-art pose estimation model, to determine and continuously track the 6-DoF poses of all interactable objects. This model takes the RGB-D camera images $I$, and the known 3D geometric models of the objects, $M$, as input. The output is a real-time stream of estimated pose data for each object in the scene.
2) **Semantic Physics Priors:** Direct estimation of physical dynamics from static images is ill-posed. Therefore, we utilize the VLM not as a ground-truth estimator, but to generate *semantic priors*. Given the visual input and task context $T$, the VLM infers likely physical properties (e.g., distinguishing a "heavy metal tool" from a "light foam block"). These priors initialize the mass and friction coefficients in $\theta$, which are explicitly treated as beliefs subject to refinement by the online adaptation module.

The combined output of this perception pipeline is a structured set of world parameters, $\theta$. For each object, $\theta$ contains its semantic label (derived from the input 3D model), its continuously tracked pose from FoundationPose, and its semantic physics priors from the VLM. These parameters are crucial as they are used to initialize and continuously update the internal `Planning World` that the MPPI planner operates on.

While our current implementation focuses on mass and friction—the dominant parameters for rigid-body contact dynamics in our evaluated tasks—the framework is intentionally designed to be parameter-agnostic. The VLM prompt and JSON schema (detailed in supplementary material) can be extended to query additional properties such as stiffness for soft objects, damping coefficients for viscous interactions, or center-of-mass offsets for asymmetric objects. Such properties can be incorporated into MPPI's rollout dynamics without architectural changes, enabling broader deployment beyond rigid-body manipulation.

### B. LLM-driven Task Formulation and Memory Retrieval

With the perceived world, the system formulates a concrete plan. This is handled by the 'LLM (Task Formulation)' module, which can generate a plan from scratch or leverage past experiences from the 'Memory Unit'.

**Memory Retrieval:** Before invoking the LLM, the system queries the 'Memory Unit' with the current world parameters $\theta$ and natural language task description $T$. Our memory module is based on Retrieval-Augmented Generation (RAG), storing successful "experience episodes" indexed by task definitions and environmental parameters. Instead of relying on predefined similarity metrics, the LLM embeds the current task into a latent semantic space to retrieve the most relevant past experience:

$$(J_{\text{mem}}, C_{\text{mem}}) = \text{RAG}_{\text{Retrieve}}(T, \theta) \tag{1}$$

where $J_{\text{mem}}$ denotes the final cost function that led to a successful episode, and $C_{\text{mem}}$ denotes the corresponding contact strategy. If a sufficiently similar and successful past experience is found, its stored plan $(J_{\text{mem}}, C_{\text{mem}})$ is retrieved and used as the initial plan, bypassing the initial LLM call and accelerating performance.

**Plan Generation from Scratch:** If no suitable memory is found, the 'LLM (Task Formulation)' module is invoked. It acts as a high-level strategist, translating the task $T$ and world parameters $\theta$ into a formal optimization problem. Its output is an initial plan tuple $(J_0, C_0)$, where:

- **Initial MPPI Cost Function ($J_0$):** The LLM generates the mathematical structure and relative weights of a cost function. Specifically, for a given task, the LLM provides a structured cost functional, for instance:

$$\begin{aligned} J_0(\mathbf{x}_{0:H}, \mathbf{u}_{0:H-1}) = \sum_{t=0}^{H-1} \Big[ & w_d \left\| \mathbf{p}_{\text{target}} - \mathbf{p}_{\text{obj}}(t) \right\|^2 \\ & + w_c \, \mathbb{I}\{\text{no contact at } t\} \\ & + w_u \left\| \mathbf{u}_t \right\|^2 \Big] \end{aligned} \tag{2}$$

  Here, the weights $w_d, w_c, w_u$ and the cost terms are determined by the LLM based on the task description (e.g., for a pushing task, $w_c$ would be high). In the example cost function, $\mathbf{p}_{\text{obj}}(t)$ is the object's tracked position at time $t$, and $\mathbb{I}\{\cdot\}$ is an indicator function penalizing the absence of contact. This expression is only an illustrative

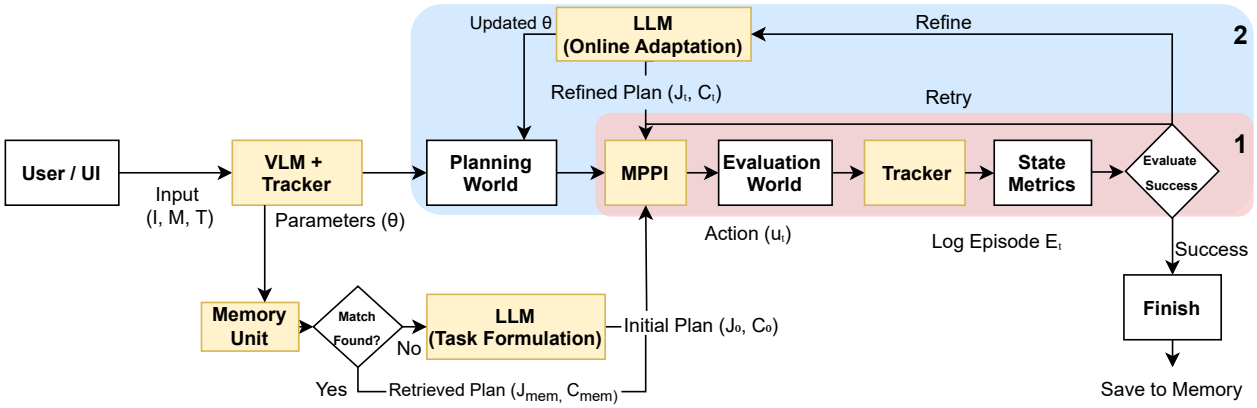


Fig. 3: The overall architecture of the **CoRAL** framework. Given an input image $I$, object models $M$ and task description $T$, the vision module extracts world parameters $\theta$. If the Memory Unit finds a similar successful experience, its retrieved plan $(J_{\text{mem}}, C_{\text{mem}})$ is used to guide the MPPI controller. Otherwise, the LLM (Task Formulation) module generates an initial plan $(J_0, C_0)$. The system then enters the main execution cycle, which is governed by two nested feedback loops labeled (1) and (2). **(1) The Inner Loop** is a high-frequency re-planning cycle. At each step, the MPPI, guided by the current plan, generates an action $u_t$ based on the latest 'State Metrics'. This loop ('Retry') continues until the task succeeds or a refinement is needed. **(2) The Outer Loop** is a low-frequency, high-level adaptation cycle. If the inner loop fails persistently, the 'Refine' path is taken, where the LLM (Online Adaptation) updates both the world model parameters ($\theta$) for the 'Planning World' and the strategic 'Refined Plan $(J_t, C_t)$' for the MPPI. Successful episodes are stored back into the Memory Unit.

example: in general, the LLM is free to introduce any cost terms constructible from the available state, pose, and action variables, and is not restricted to a fixed finite set of cost terms. Moreover, the LLM can solve the task in stages by defining separate cost functions and specifying conditions for transitioning between stages.

- **Initial Contact Strategy ($C_0$):** To bridge the gap between abstract task descriptions and continuous geometry, we employ the LLM as a *high-level semantic heuristic* to predict a *coarse region of interest* from the task description. The LLM outputs a nominal reference point $c_{\text{ref}}$ in the object's local frame and a spatial expansion factor $r$. A procedural helper then constructs an ellipsoid $\mathcal{E}(c_{\text{ref}}, \mathbf{A})$ centered at $c_{\text{ref}}$, with axes $\mathbf{A}$ aligned with the object's principal geometric axes and scaled by $r$. Crucially, this expansion serves as a spatial buffer that compensates for the imprecision of LLM-generated coordinates: even if $c_{\text{ref}}$ deviates from the ideal contact location, $r$ ensures the valid contact manifold $\mathcal{R}$ still covers the optimal region. The valid contact manifold is defined as $\mathcal{R} = \partial\mathcal{O} \cap \mathcal{E}(c_{\text{ref}}, \mathbf{A})$. We then sample candidate target points from this manifold to form the strategy set: $C_0 = \left\{ x_{\text{des}} \mid x_{\text{des}} \sim \text{Unif}(\mathcal{R}) \right\}$. Finally, a target $x_{\text{des}} \in C_0$ is integrated into the optimization not as a hard constraint, but as a *soft cost attractor* in $J_0$ (e.g., $w_{\text{attr}} \|p_{\text{eef}} - x_{\text{des}}\|^2$). This formulation creates a "stretched" cost landscape: it biases MPPI sampling toward the semantic region of interest while, due to the soft penalty, still allowing deviation to find the dynamically optimal contact point within that vicinity, improving robustness to perception noise.

### C. Reactive Planning and Execution (The Inner Loop)

The core of our system is a high-frequency, reactive execution cycle governed by the MPPI controller. This corresponds to the Inner Loop (1) in Figure 3. To address the latency challenges inherent in LLM inference, we implement a strict hierarchical control architecture:

- **Tier 1 (Hardware Level):** Joint impedance control running at 1kHz guarantees safety and compliance during physical interaction.
- **Tier 2 (Trajectory Level):** The MPPI planner runs at 10Hz, sampling trajectories based on the current cost function $J_t$.
- **Tier 3 (Reasoning Level):** The LLM operates asynchronously (∼1Hz) to update the cost structure periodically.

**MPPI Formulation:** The MPPI controller solves a stochastic optimal control problem at each timestep. Given a state-transition model $x_{t+1} = f(x_t, u_t) + \epsilon_t$, where $x_t$ is the state, $u_t$ is the target end-effector delta pose, and $\epsilon_t$ is system noise, the objective is to find $U = \{u_0, \ldots, u_{H-1}\}$ minimizing the expected total cost:

$$U^* = \arg\min_U \mathbb{E}\left[\phi(x_H) + \sum_{t=0}^{H-1} q(x_t, u_t)\right] \tag{3}$$

where $\phi(x_H)$ is a terminal state cost and $q(x_t, u_t)$ is the running cost at each step, directly defined by the LLM-generated cost function $J_0$ (see Eq. 2 for an example structure). MPPI approximates this optimization by:

1) Sampling $K$ control sequence perturbations $\delta U_k \sim \mathcal{N}(0, \Sigma)$ from a Gaussian distribution.

2) Creating $K$ rollout trajectories by applying the perturbed control sequences $V_k = U_{prev} + \delta U_k$ in the 'Planning World'.
3) Calculating the total cost $S(V_k)$ for each of the $K$ trajectories.
4) Computing an exponentially weighted average of the perturbations to update the control sequence:

$$U_{new} = U_{prev} + \sum_{k=1}^{K} w_k \delta U_k, \quad \text{where} \quad w_k = \frac{\exp\left(-\frac{1}{\lambda} S(V_k)\right)}{\sum_{j=1}^{K} \exp\left(-\frac{1}{\lambda} S(V_j)\right)} \tag{4}$$

Following the receding horizon principle, only the first action, $u_0$, of the newly optimized sequence $U_{new}$ is executed.

**Reactive Control Augmentation:** To achieve robustness against the inherent sim-to-real gap, we augment the nominal planned action with a real-time feedback term. The final control command $\nu_t$ sent to the robot is:

$$\nu_t = u_t + K_f \cdot (x_{\text{des}} - x_{\text{measured}}) \tag{5}$$

where $u_t$ is the action computed by MPPI, the error term is calculated from real-time sensors (e.g., force/torque, proprioception), and $K_f$ is a feedback gain matrix. This 'Retry' loop continues at a high frequency, constantly re-planning and correcting based on physical feedback.

### *D. Online Adaptation via LLM-driven Refinement (The Outer Loop)*

After a predefined number of attempts, a hyperparameter we denote as $N_{\text{retry}}$, the system triggers the low-frequency Outer Loop (2). This invokes the 'LLM (Online Adaptation)' module, which acts as a diagnostician and re-strategist.

The input to this module is the logged episode data $E_t$, which contains the history of states, actions, the contact strategies and cost functions that were used, and the estimated physical parameters that led to the failure. By analyzing this rich context, the LLM performs two critical functions:

1) **World Model Correction:** Acting as a system identification agent, the LLM refines the semantic priors. For example, if the robot pushes an object but the object moves less than predicted, the LLM can infer that its initial estimate of the object's mass was too low and output an 'Updated $\theta$'.
2) **Strategy Refinement:** The LLM can also alter the plan itself. It might change the weights of the cost function (e.g., prioritizing force control over position accuracy) or propose an entirely new contact strategy. This results in a 'Refined Plan $(J_t, C_t)$'.

This refined world model and plan are then fed back into the inner loop, allowing the system to learn from its failures and adapt its entire approach within a single task execution.

## IV. EXPERIMENTS

We conducted a series of experiments in a simulated environment, complemented by a real-world validation study, to rigorously evaluate the performance of CoRAL. Our evaluation is designed to answer four key research questions: **(RQ1)** How does CoRAL perform on complex, contact-rich manipulation tasks in a zero-shot setting compared to state-of-the-art baselines? **(RQ2)** How critical is each core component of our neuro-symbolic architecture—specifically the vision/language model role separation, the online refinement loop, and the memory unit—to the overall success? **(RQ3)** Can the system demonstrate robustness and adaptability by reasoning about and recovering from failures? **(RQ4)** Can CoRAL's performance be effectively carried to a real system?

**Testing Environments:** The simulated experiments were conducted in the evaluation world, implemented using ROBOSUITE library [35], which is based on the MUJOCO physics engine [26]. The robot is a simulated 7-DoF Franka Emika Panda arm with a parallel-jaw gripper. Sensory inputs include RGB-D images from a fixed camera, proprioceptive feedback, and force/torque data, which are provided by the robot's simulated sensors. In addition to our custom environments, two benchmark tasks from the LIBERO suite [18] were also incorporated for evaluation.

We additionally evaluate on a real Franka Emika Panda using the same sensing modalities (fixed-view RGB-D, proprioception, and force/torque), where force/torque readings are obtained from the robot's built-in actuator sensors. For real-world state estimation, we replace FOUNDATIONPOSE with a motion capture setup for simplicity.

The implementation details are included in supplementary material.

*a) Tasks and Evaluation Metrics:* We evaluated our framework on four challenging, contact- and force-critical manipulation tasks and two standard pick and place tasks, shown in Fig. 4, designed to be difficult for purely vision-based, collision-avoidant planners. Each task was performed 10 times with randomized initial object poses, object masses, surface friction coefficients and the object dimensions for the box and the board objects. The tasks are as follows: **T1: Push and Pick Cutting Board**, a multi-stage task testing pushing and reasoning about object parts and pose for grasping; **T2: Pick Box & T3: Pick and Place in Clutter**, standard pick-and-place tasks to establish a baseline; **T4: Push with Constant Force**, testing the reactive controller's ability to manage force feedback; **T5: Flip Box**, a dynamically complex maneuver; and **T6: Flip with Wall**, requiring multi-contact reasoning to use the wall as a fixture. We primarily use Success Rate (binary measure across 10 trials) to evaluate performance.

*b) Comparative Baselines:* We compare CoRAL against three state-of-the-art methods and four internal ablations. The **State-of-the-Art Baselines** consist of two categories: **1) End-to-End VLA Models:** We evaluate **OpenVLA-OFT** [12] and $\pi_{0.5}$ [3]. For these models, we rely on the officially released LIBERO-OBJECT checkpoint for pick-and-place tasks and the

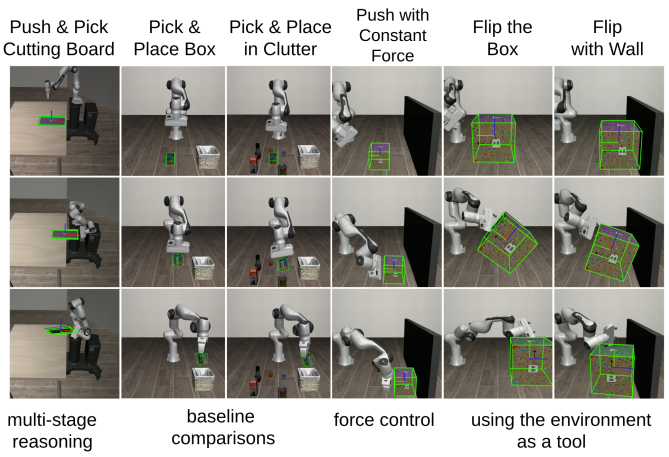


Fig. 4: CoRAL on six different tasks with the tracked pose overlay of the object of interest.

LIBERO-GOAL checkpoint for all other tasks. This setup tests CoRAL's zero-shot capabilities against powerful, pre-trained policies. **2) Cost-Generation Baseline:** We evaluate **L2R** [31], a method that uses LLMs to synthesize cost functions for an MPC controller. This baseline serves as a direct comparison for our neuro-symbolic planning approach but without the online adaptation mechanism. In addition, we include two **Human Expert-Designed Cost** baselines. In the *single-stage* variant, an expert manually designs a single MPPI running-cost for each task. In the *FSM* variant, the expert is allowed to construct an explicit finite-state machine with phase-specific costs (e.g., push–then–pick or push–then–flip). In both cases, the cost functions are tuned in a separate design environment and then evaluated *as-is* in our randomized test environment, providing an upper bound on what carefully engineered, task-specific objectives can achieve. Our **Ablation Baselines** are: **CoRAL (w/o Pose Tracking)**, which removes FoundationPose and relies on the VLM to estimate object poses, testing the criticality of a dedicated pose estimator; **CoRAL (w/o Memory)**, which removes the experience retrieval mechanism; **CoRAL (w/o Refinement)**, which disables the online adaptation loop; and **CoRAL (Unified VLM)**, which uses a single multimodal prompt for both perception and planning to test the importance of separating VLM/LLM roles.

### A. Results and Analysis

Table I presents a comprehensive overview of our experimental findings.

*1) State-of-the-Art Comparison (RQ1):* CoRAL significantly outperforms both state-of-the-art VLA baselines (OpenVLA-OFT, $\pi_{0.5}$) and the foundation-model-based planner baseline (L2R), particularly in tasks requiring sophisticated physical reasoning (T1, T4, T5, T6). While all baselines perform well on the simpler pick-and-place tasks (T2, T3), their performance degrades sharply on the more complex, contact-rich scenarios. This highlights distinct limitations in existing approaches. On one hand, end-to-end VLA policies prove insufficient for scenarios demanding explicit physical modeling, as they fail to generalize to non-obvious maneuvers like maintaining steady force (T4). On the other hand, the L2R baseline demonstrates the fragility of static code generation; it fails tasks like the wall-flip (T6: 1/10) because it generates fixed cost functions without explicitly formulating a contact strategy or employing an online refinement mechanism to handle physical deviations. In contrast, our framework excels by combining a dedicated contact strategy with an adaptive LLM that directly formulates and iteratively refines cost functions, enabling robust zero-shot execution across all dynamic interaction regimes.

*2) Comparison to Human-Designed Cost Functions:* The two human baselines approximate an upper bound from carefully engineered, task-specific objectives. As expected, the *Expert (FSM)* variant achieves the strongest overall performance, and the single-stage expert design remains competitive, particularly on simpler tasks such as T2–T4, where CoRAL largely matches but does not surpass its success rate (Table I). On more sequential and contact-heavy tasks (T1, T5, T6), CoRAL narrows the gap to the expert, achieving higher success rates than the single-stage baseline while remaining below the FSM upper bound. This shows that our LLM-based controller can recover much of the structure of expert-designed costs automatically, substantially reducing manual tuning effort while approaching expert-level performance on the hardest tasks.

*3) Ablation Study Analysis (RQ2):* Our ablation studies clearly demonstrate the necessity of each component in our architecture.

**The Synergy of Separated VLM/LLM Roles:** The *CoRAL (Unified VLM)* variant, which tasked a single VLM with both perception and planning, failed on nearly all complex tasks. This starkly illustrates our core hypothesis: separating the role of a VLM for perception from a dedicated LLM for strategy formulation is crucial for robust performance. The specialized modules provide more reliable and structured outputs for the planner.

**The Importance of Online Refinement:** The *w/o Refinement* variant showed a dramatic performance drop in multi-stage tasks like T1 (Push and Pick Board), with success falling from 5/10 to 0/10. In this task, the initial plan often failed because the VLM's initial friction estimate was slightly off, causing the board to slip during the pick. The full CoRAL framework, however, used the outer loop for the LLM to diagnose this from the physical outcome, refine the friction parameter in its world model, and successfully complete the task. This shows the system's ability to learn from failure.

**The Benefit of Experience Reuse:** The full framework *with Memory* consistently achieved the highest success rates. For instance, in T1 and T3, memory boosted the success rate from 2/10 to 5/10 and 9/10 to 10/10, respectively. By retrieving a successful "push-to-edge" strategy from a past experience, the system provided the planner with a superior initialization, accelerating convergence and leading to more robust solutions.

**The Criticality of a Dedicated Pose Estimator:** The *w/o*

TABLE I: Comparison against the baselines and ablation study. Performance is measured by success rate (x/10 trials).

| Method | T1: Push+Pick | T2: Pick+Place | T3: Clutter | T4: Const. Force | T5: Flip Box | T6: Flip w/ Wall |
|---|---|---|---|---|---|---|
| *State-of-the-Art Baseline* | | | | | | |
| OpenVLA-OFT [12] | 0/10 | 10/10 | 9/10 | 0/10 | 1/10 | 0/10 |
| $\pi_{0.5}$ [3] | 0/10 | 10/10 | 8/10 | 0/10 | 3/10 | 0/10 |
| L2R [31] | 0/10 | 10/10 | 9/10 | 5/10 | 4/10 | 1/10 |
| *Human Expert-Designed Cost Baselines* | | | | | | |
| Expert (single-stage) | 0/10 | 10/10 | 10/10 | 9/10 | 9/10 | 3/10 |
| Expert (FSM) | 8/10 | 10/10 | 10/10 | 10/10 | 10/10 | 9/10 |
| *Our Method (Ablation Study)* | | | | | | |
| **CoRAL (Ours)** | **5/10** | **10/10** | **10/10** | **9/10** | **9/10** | **7/10** |
| CoRAL (w/o Memory) | 2/10 | 10/10 | 9/10 | 9/10 | 7/10 | 5/10 |
| CoRAL (w/o Refinement) | 0/10 | 10/10 | 3/10 | 6/10 | 4/10 | 2/10 |
| CoRAL (Unified VLM) | 0/10 | 2/10 | 0/10 | 1/10 | 0/10 | 0/10 |
| CoRAL (w/o Pose Tracking) | 0/10 | 0/10 | 0/10 | 0/10 | 0/10 | 0/10 |

*Pose Tracking* ablation, which removed FoundationPose and relied solely on the VLM for pose estimation, resulted in a catastrophic failure across all tasks (0/10 success). The VLM, while powerful for semantic understanding, is ill-suited for the precision required by 6-DoF pose tracking through dynamic interactions. It frequently produced trivial or physically impossible pose estimations ("hallucinations") that rendered the planner's output useless. This result provides conclusive evidence that a dedicated, high-fidelity pose estimator is not merely beneficial but an essential component of our architecture, serving as the geometric foundation upon which all subsequent physical reasoning is built.

*4) Robustness Analysis (RQ3):* **Analysis of LLM-Guided Contact Strategy:** To isolate the contribution of the LLM's initial contact strategy ($C_0$), we conducted a targeted ablation on the challenging "Flip with Wall" task (T6). We compared the performance of our full framework against a variant where the LLM only provided the cost function ($J_0$), forcing the MPPI planner to discover useful contact points through its own sampling mechanism. Crucially, this variant effectively mimics the operational capability of standard foundation-model-based planner baseline like L2R [31], which rely solely on the optimizer to discover contact modes.

The guided trajectory (With Strategy, green) is direct and purposeful, immediately moving the end-effector to the correct corner of the box to initiate the flip (Fig. 5). In contrast, the unguided trajectory (Without Strategy, red) is chaotic and inefficient, exploring large, irrelevant portions of the workspace. The planning cost for the unguided agent remains high and erratic, indicating a constant struggle to find a viable plan. This difference is confirmed by the quantitative results: the guided approach was **83.9**$\%$ **more efficient** requiring fewer control steps (32 vs. 199 steps) and the end-effector traveled a **63.9**$\%$ **shorter path** (1.33m vs. 3.69m). This analysis provides clear evidence that the LLM's symbolic contact strategy is critical for transforming a computationally intractable, long-horizon contact problem into a solvable one by intelligently pruning the vast action search space.

**Robustness of Online Parameter Adaptation:** Beyond strategy and cost function refinement, CoRAL's 'Online Adaptation' module, driven by the LLM, exhibits a crucial ability to correct the agent's internal world model online. To demonstrate this, we ran the same experiment both in simulation and on the real system to verify that the adaptation mechanism remains effective under real-world unmodeled effects and sensing/actuation noise, intentionally initializing the *Planning World* with a severely overestimated mass (2.0kg vs. a ground truth of 0.25kg) and friction coefficient (0.9 vs. 0.5) for the cutting board. These initial biases represent a severe sim-to-real gap or a VLM hallucination.

Figure 6 shows the adaptation process. Given the execution history, the LLM's 'Online Adaptation' module identified that the board was not moving as expected despite high pushing force and updated the physical parameters. Through an iterative refinement process, it progressively adjusted its estimated mass and friction parameters. After several adaptation cycles, the agent's belief about both mass and friction converged remarkably close to their true values. This online correction of physical parameters is fundamental to the framework's robustness, allowing it to overcome initial environmental mischaracterizations and successfully execute contact-rich tasks that would otherwise fail due to a misaligned internal world model.

**Sensitivity to Pose and Physical-Parameter Estimates:** To further quantify the robustness of CoRAL to imperfect world-model initialization, we performed a sensitivity analysis on the "Push and Pick Cutting Board" task (T1). For physical parameter sensitivity, the cutting-board mass was uniformly sampled from $[0.4, 0.8]$kg, and the sliding friction coefficient from $[0.3, 0.6]$. Across 100 samples, the VLM-only estimates achieved a mean absolute error (MAE) of $0.29$kg for mass and $0.14$ for friction. These results support our design choice of treating VLM outputs as coarse semantic priors rather than accurate physical measurements. After four online refinement cycles, the errors decreased to $0.11$kg and $0.06$, respectively, improving the task success rate from $2/10$ to $5/10$.

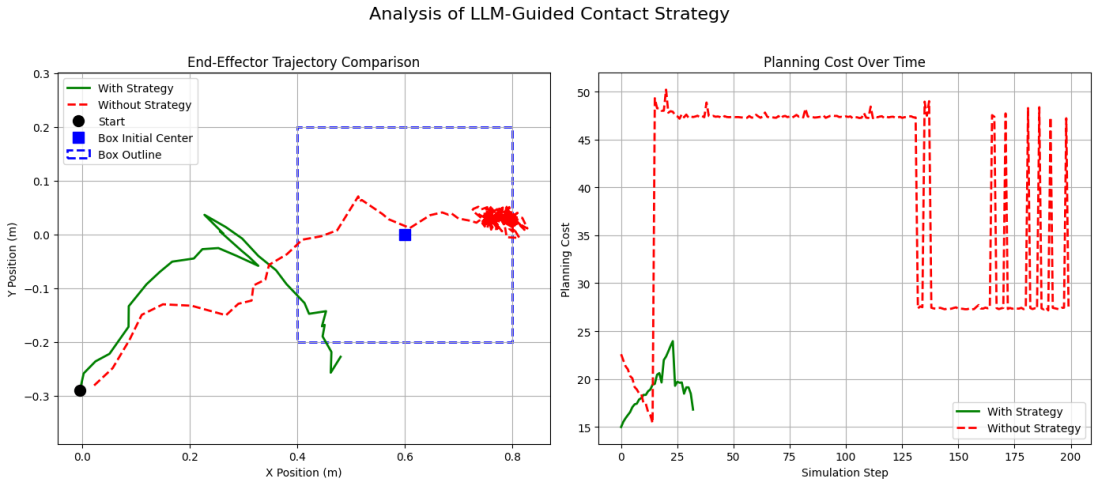


Fig. 5: Ablation of the LLM-guided contact strategy on the "Flip with Wall" task. **(Left)** The trajectory with the LLM's strategy (green) is direct and efficient, while the unguided trajectory (red) is erratic. **(Right)** The planning cost for the guided agent is significantly lower and more stable, indicating an easier optimization problem.

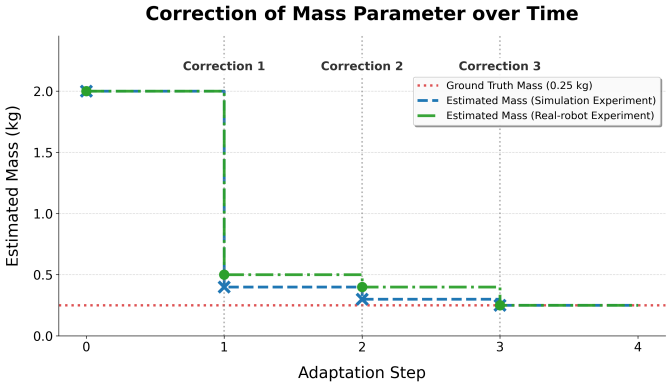


Fig. 6: Online parameter (mass) adaptation in simulation and real-world experiments.

For the same T1 task, Table II summarizes the effect of replacing estimated quantities with oracle information. Using ground-truth physical parameters alone did not further improve performance beyond the refined setting, suggesting that CoRAL can largely compensate for moderate mass and friction errors through replanning and online adaptation. In contrast, replacing the estimated object poses with ground-truth poses increased the success rate to $7/10$, and using both ground-truth poses and ground-truth physical parameters further improved it to $8/10$. This indicates that pose-estimation errors, especially during occlusions caused by direct end-effector contact, are the dominant bottleneck in this task. In our cutting-board trials, FoundationPose achieved a mean ADD error of 13.2mm over 10 episodes, while $88.2\%$ of frames had ADD below 5mm. The larger mean error is mainly caused by short tracking failures during contact-rich phases. Overall, physical-parameter errors primarily affect the magnitude of predicted object motion and can often be corrected through feedback and replanning, whereas pose failures directly corrupt the replanning state and are therefore more detrimental.

TABLE II: Cutting-board success rates under different pose and physical-parameter settings.

| **Condition** | **Success Rate** |
|---|---|
| VLM-estimated parameters, no parameter refinement | $2/10$ |
| VLM-estimated parameters, with parameter refinement | $5/10$ |
| Ground-truth physical parameters | $5/10$ |
| Ground-truth poses | $7/10$ |
| Ground-truth poses + ground-truth physical parameters | $8/10$ |

**Sequential Reasoning and Experience Reuse in the Cutting Board Task:** The "Push and Pick Cutting Board" task (T1) tests long-horizon sequential manipulation, requiring a stable push to expose the board followed by a precise grasp. As evidenced by Table I, this challenge highlights the importance of two core components: online adaptation and experience reuse.

First, long-horizon tasks are sensitive to model errors that accumulate over time. This is demonstrated by the *w/o Refinement* ablation, which failed entirely (0/10 success rate), mirroring the static L2R baseline [31]. While the initial plan was often sufficient for the push, slight inaccuracies in estimated friction caused the board to end in an unexpected pose, leading to a failed grasp. Our full model, however, leverages the outer loop to learn from the push outcome, allowing the 'LLM (Online Adaptation)' to refine friction estimates and update the plan for the subsequent pick.

Second, this task illustrates the benefit of the 'Memory Unit'. Including the memory module boosted the success rate from 2/10 to 5/10. This indicates that after a single completion, the system stores the successful interaction context (refined parameters and strategy) to provide a superior initialization for future attempts. This demonstrates that CoRAL can reuse prior successful experience, highlighting a clear path towards few-shot performance improvements as it gathers successful episodes.

**Explainability and Automated Failure Recovery** A key

advantage of our neuro-symbolic design is its inherent explainability, particularly during failure recovery. Unlike opaque end-to-end models, CoRAL can articulate *why* it failed and *what* it is doing to correct its plan. We demonstrate this with a scenario where the "Flip with Wall" task persistently fails, triggering the Outer Loop.

Instead of just outputting a new set of parameters, the 'LLM (Online Adaptation)' module provides a full natural language diagnosis of the failure and a detailed log of the corrective actions it is taking. The LLM provided a correct natural language diagnosis of a poorly weighted cost function and proceeded to adjust the specific weights to remedy the failure (see Supplementary Material).

**LLM Failure Modes:** While CoRAL avoids using the LLM as a direct controller, failures can still arise from the symbolic objectives it generates. We observed two main failure modes. First, the LLM occasionally produced highly imbalanced cost weights, e.g., $w_d : w_c = 1000 : 1$, causing MPPI to effectively ignore some objectives. We mitigated this by prompting the LLM to justify the relative importance of each term and keep weights within comparable numerical ranges, typically $[0.1, 10]$. Second, in multi-stage tasks such as T1 (*Push and Pick Cutting Board*) and T6 (*Flip with Wall*), the LLM sometimes produced ambiguous or incomplete stage-transition conditions, leading to premature or delayed phase switches; representative examples are provided in the Supplementary Material. These structural errors occurred in fewer than $10\%$ of trials but were more consequential than small weight errors. Importantly, CoRAL's modular design makes such failures interpretable: they appear as persistently high rollout costs or mismatches between the intended subgoal and observed state evolution, allowing the outer-loop refinement module to update the cost structure or stage logic.

*5) Real-World Validation (RQ4):* Finally, to validate the reliability of CoRAL under physical constraints that cannot be perfectly simulated (e.g., sensor noise, unmodeled friction, and calibration errors), we deployed the system zero-shot on the physical Franka Emika Panda robot. We evaluated the full suite of six tasks without any real-world fine-tuning (Fig. 1).

TABLE III: **Real-World Experimental Results.** CoRAL is evaluated on the physical robot across all six tasks without real-world fine-tuning.

| Task | Success Rate | Avg. Exec. Time (s) |
|---|---|---|
| T1: Push and Pick Cutting Board | 4/10 | $21.6 \pm 5.5$ |
| T2: Pick Box | 10/10 | $16.7 \pm 1.1$ |
| T3: Pick and Place in Clutter | 10/10 | $22.0 \pm 1.5$ |
| T4: Push with Constant Force | 9/10 | $11.7 \pm 1.9$ |
| T5: Flip Box | 7/10 | $9.2 \pm 2.6$ |
| T6: Flip with Wall | 6/10 | $25.3 \pm 4.9$ |

**Sim-to-Real Robustness:** As shown in Table III, CoRAL demonstrates strong sim-to-real transfer capabilities. While standard tasks (T2, T3) achieved perfect success rates, the framework also maintained robust performance on contact-critical tasks (T1, T4, T5). A notable observation was in the "Push and Pick" task (T1), where real-world surface friction varied significantly from the simulation. In these cases, the online adaptation loop (discussed in Section IV) successfully diagnosed the slippage and updated the friction parameters in real-time to salvage the trial.

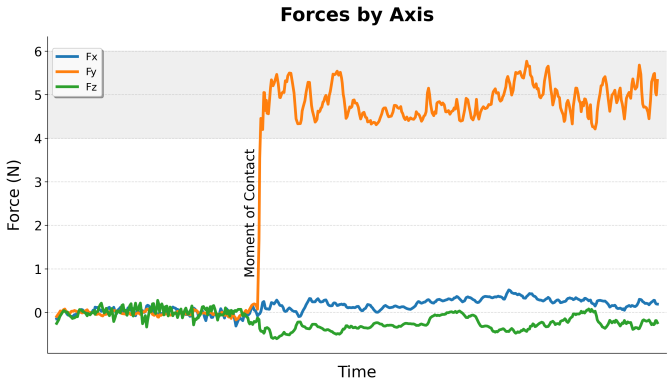


Fig. 7: **Real-World Force Regulation Profile (T4)** The plot shows the measured end-effector force over time during the "Push with Constant Force" task where the intended push direction is $+y$. The shaded region represents the target force range the LLM is instructed with. CoRAL successfully modulates the control actions to maintain contact forces within the desired bounds.

**Force Regulation:** The system's ability to handle active force constraints is highlighted in Task 4. Figure 7 shows the force profile from a real-world trial. Despite the inherent noise in the physical force/torque sensor, the MPPI controller—guided by the LLM's cost function—effectively regulated the interaction force within the target bounds (approx. 5N). This confirms that CoRAL's decoupled architecture not only plans for geometry but effectively closes the loop on force dynamics, bridging the gap between high-level semantic planning and low-level compliance.

## V. Limitations & Conclusion

In this paper, we introduced **CoRAL**, a novel framework that addresses the challenges of zero-shot, contact-rich manipulation. Our approach departs from conventional end-to-end paradigms by integrating foundation models with a reactive controller. Experiments on challenging tasks demonstrate that this modular, synergistic design enables the system to adapt to unseen scenarios without prior demonstrations, significantly enhancing both performance and explainability over monolithic approaches. While promising, the framework's performance is currently contingent on the fidelity of the vision-based world model and is subject to latency constrained by foundation model inference. These limitations and future research directions are discussed in detail in the supplementary material. We believe this hybrid paradigm—coupling large-scale, pre-trained knowledge with rigorous real-time control—is a promising direction for creating more capable and physically intelligent robotic agents.

## References


[1] Jianxin Bi, Kevin Yuchen Ma, Ce Hao, Mike Zheng Shou, and Harold Soh. Vla-touch: Enhancing vision-language-action models with dual-level tactile feedback. *arXiv preprint arXiv:2507.17294*, 2025.

[2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[3] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi$0. 5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2502.19645*, 2025.

[4] Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, 44(5):701–739, 2025.

[5] J Randall Flanagan, Miles C Bowman, and Roland S Johansson. Control strategies in object manipulation tasks. *Current opinion in neurobiology*, 16(6):650–659, 2006.

[6] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[7] Peng Hao, Chaofan Zhang, Dingzhe Li, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. Tla: Tactile-language-action model for contact-rich manipulation. *arXiv preprint arXiv:2503.08548*, 2025.

[8] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025.

[9] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, 2022.

[10] 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.

[11] Roland S Johansson and J Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10 (5):345–359, 2009.

[12] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[13] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, volume 270, pages 2098–2120. PMLR, 2025.

[14] Sung Soo Kim, Manuel Gomez-Ramirez, Pramodsingh H Thakur, and Steven S Hsiao. Multimodal interactions between proprioceptive and cutaneous signals in primary somatosensory cortex. *Neuron*, 86(2):555–566, 2015.

[15] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

[16] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.

[17] Yiyang Ling, Karan Owalekar, Oluwatobiloba Adesanya, Erdem Bıyık, and Daniel Seita. Impact: Intelligent motion planning with acceptable contact trajectories via vision-language models. *arXiv preprint arXiv:2503.10110*, 2025.

[18] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[19] Jason Jingzhou Liu, Yulong Li, Kenneth Shaw, Tony Tao, Ruslan Salakhutdinov, and Deepak Pathak. Factr: Force-attending curriculum training for contact-rich policy learning. *arXiv preprint arXiv:2502.17432*, 2025.

[20] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv: Arxiv-2310.12931*, 2023.

[21] Yecheng Jason Ma, William Liang, Hungju Wang, Sam Wang, Yuke Zhu, Linxi Fan, Osbert Bastani, and Dinesh Jayaraman. Dreureka: Language model guided sim-to-real transfer. In *Robotics: Science and Systems (RSS)*, 2024.

[22] 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.

[23] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[24] Ranjan Sapkota, Yang Cao, Konstantinos I Roumeliotis, and Manoj Karkee. Vision-language-action models:

Concepts, progress, applications and challenges. *arXiv preprint arXiv:2505.04769*, 2025.

[25] Muhammad Tayyab Khan and Ammar Waheed. Foundation model driven robotics: A comprehensive review. *arXiv e-prints*, pages arXiv–2507, 2025.

[26] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[27] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17868–17879, 2024.

[28] Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, 2017.

[29] Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. *arXiv preprint arXiv:2503.02881*, 2025.

[30] Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, et al. Forcevla: Enhancing vla models with a force-aware moe for contact-rich manipulation. *arXiv preprint arXiv:2505.22159*, 2025.

[31] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montserrat Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. Language to rewards for robotic skill synthesis. In *Conference on Robot Learning*, pages 374–404. PMLR, 2023.

[32] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

[33] Wentao Zhao, Jiaming Chen, Ziyu Meng, Donghui Mao, Ran Song, and Wei Zhang. VLMPC: Vision-language model predictive control for robotic manipulation. In *Robotics: Science and Systems (RSS)*, 2024.

[34] Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, et al. A survey on vision-language-action models: An action tokenization perspective. *arXiv preprint arXiv:2507.01925*, 2025.

[35] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.