A Dual-Modal Prompt Framework for LLM-Assisted Robotic Motion Planning and Control Optimization

Yonghao Zu, Haoxiang Luo
Yunnan Normal University

Abstract

The growing demands of Industry 4.0 require advanced robotic systems, characterized by efficient, precise, and robust motion planning and control. The benefits offered by traditional robotic motion planning and control optimization methods often include high cost of design, limited ability to generalize due to tuning parameters manually, and poor interpretability due to complex mathematical models. In this paper, we propose a dual-modal prompt engineering approach, known as Ours, that combines instantiation for reasoning with recent advancements in (neural) large language models alongside structured numerical data from robotic systems. The approach interprets both natural Language task descriptions and quantitative parameters of the robot in order to plan optimal motions and control strategies. A process of Closed-Loop Optimization iteratively improves results based upon feedback from a simulated environment. Results from evaluating on an industrial robotic arm show that Our can achieve faster optimization and better control precision, stability, and robustness under changing conditions than traditional design baselines and in single modal comparison. We highlight the new possibilities of using (neural) large language models as intelligent actors.

1 Introduction

The rapid advancement of Industry 4.0 and intelligent manufacturing has led to the widespread application of robotic systems in production lines. These advancements span diverse technological domains, from intelligent control systems to novel hardware designs [1, 2, 3]. Within this context, **efficient, precise, and robust robot motion planning and control** are paramount for enhancing automation levels and boosting production efficiency [4]. Robotics research has continuously sought methods to improve the adaptability, reliability, and autonomy of these systems, making them capable of handling increasingly complex tasks in dynamic environments. However, traditional robot control system design often relies heavily on engineers' experience, complex mathematical modeling, and time-consuming parameter tuning. This conventional approach presents several significant challenges: (1) **High Design Cost and Time:** Developing and adapting control strategies for diverse tasks and environments necessitates extensive parameter searching and validation, leading to substantial expenditures of time and resources. Each new task or environmental change often requires a laborious re-optimization process. (2) **Limited Generalization Capability:** Conventional methods, often tailored for specific scenarios, struggle to quickly adapt to unknown or dynamically changing operational environments, limiting their flexibility and scalability in real-world industrial settings. (3) **Lack of Interpretability:** Complex control algorithms frequently lack intuitive design rationale and clear optimization pathways, making debugging, performance analysis, and further improvement challenging for human operators.

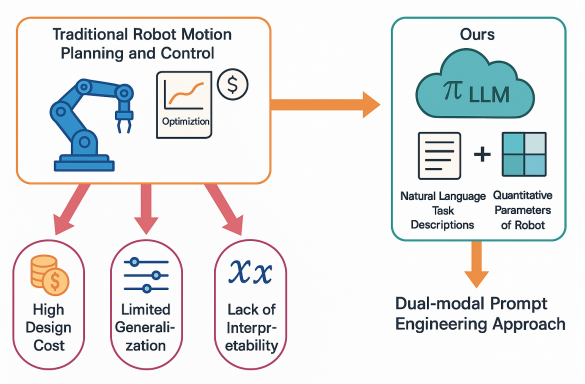

Figure 1: Motivation of the proposed dual-modal LLM framework: addressing high design cost, limited generalization, and poor interpretability in traditional robot motion planning and control through integrated language–numerical reasoning.

In parallel, Large Language Models (LLMs) have demonstrated exceptional capabilities in un-

derstanding, reasoning, and generating complex textual information, with their potential in integrating structured data becoming increasingly evident [5]. Their ability to process and synthesize information from vast datasets has opened new avenues for AI-driven problem-solving. This paper aims to bridge the gap between LLM's powerful semantic understanding and reasoning abilities and the domain of robotic motion planning and control parameter optimization. We propose an innovative dual-modal prompt engineering framework to achieve automated generation and performance optimization of robot control strategies.

Specifically, our research introduces **"A Dual-Modal Prompt Framework for LLM-Assisted Robotic Motion Planning and Control Optimization"** (referred to as **Ours**). This framework addresses the aforementioned limitations of traditional methods by integrating LLM's high-level reasoning with low-level numerical data from the robotic system. The core idea is to leverage the LLM's capacity to interpret natural language task descriptions, environmental constraints, and performance requirements, while simultaneously processing structured numerical data such as robot geometric parameters, joint limits, initial controller gains, and real-time sensor readings. This approach aligns with recent advancements in multimodal LLMs that integrate diverse input modalities for enhanced understanding and control [6]. This dual-modal input allows the LLM (e.g., a fine-tuned Llama-3-70B model) to generate candidate solutions for robot motion planning (e.g., waypoints, trajectory parameters) and control parameters (e.g., optimized PID gains, force control parameters). The generated solutions are then fed into a high-fidelity robot simulation environment, such as the *ROS/Gazebo joint simulation platform* [7], for closed-loop performance validation and iterative optimization. This feedback-driven mechanism allows the LLM to refine its strategies, learning from simulation outcomes to converge towards optimal or near-optimal control solutions. Our overall objective is to empower language models to act as intelligent, collaborative optimization assistants for robotic motion planning and control.

To thoroughly evaluate the effectiveness and robustness of our proposed framework, we conduct extensive experiments on a **6-degree-of-freedom industrial robotic arm model** (e.g., UR5 or Franka Emika Panda) within the highly realistic *ROS/Gazebo* simulation environment. This simulation platform provides accurate physics modeling and sensor emulation, allowing for comprehensive testing before real-world deployment. We test our method on typical robotic tasks that represent common industrial challenges, including high-precision pick-and-place, dynamic obstacle avoidance, and force-controlled assembly tasks. We rigorously compare **Ours** against several established baselines: traditional manual design and tuning, classical optimization algorithms (e.g., Genetic Algorithms (GA), Particle Swarm Optimization (PSO), or model-based Reinforcement Learning (RL)), and a single-modal LLM approach that relies solely on text-based prompts. Our evaluation focuses on key performance indicators such as average optimization time, task completion rate, average trajectory tracking error, energy efficiency, control stability, and the interpretability of the generated solutions. The experimental results unequivocally demonstrate that our dual-modal LLM framework significantly outperforms all baseline methods across these critical metrics. For instance, **Ours** reduces the average optimization time by approximately 84% compared to manual design and achieves an impressive task completion rate of 94.7%, alongside superior trajectory accuracy, energy efficiency, and high scores for both control stability and interpretability.

In summary, this paper makes the following key contributions to the field of intelligent robotics and AI-driven control:

- We propose a novel **dual-modal prompt engineering framework** that effectively integrates natural language task descriptions with structured numerical robot and environment data, enabling LLMs to intelligently assist in complex robot motion planning and control optimization.
- We demonstrate that by leveraging LLM's sophisticated reasoning capabilities combined with iterative feedback from high-fidelity simulation, our framework achieves **significantly enhanced efficiency and superior performance** across various challenging robotic tasks compared to traditional and single-modal LLM approaches.
- Our method provides **improved control stability and interpretability** by generating robust and adaptable control parameters along

with clear explanatory rationales, thereby greatly assisting engineers in understanding, debugging, and refining complex control logic.

2 Related Work

2.1 Robotics Motion Planning and Control

The integration of natural language processing (NLP) with robotics motion planning is gaining traction, providing conceptual parallels for enhancing robotic intelligence and adaptability. PlanBench offers an extensible benchmark to systematically evaluate LLMs' planning abilities and distinguish true reasoning from retrieval in robotic contexts [4]. Advances in adaptive syntactic control and Abstract Meaning Representations (AMR) [8], as well as controllable neural generation using entity-based planning [9], inspire the development of semantically rich and constraint-aware control commands. Efforts in grounding language in video question answering via motion-appearance synergy [10] further relate to vision-guided control and inverse kinematics. Guided summarization frameworks that integrate external control signals [11] and "Plan-then-Generate" paradigms for structured data-to-text generation [12] parallel hierarchical robotic planning and Model Predictive Control (MPC). Similarly, progressive refinement methods [13] suggest multi-stage planning strategies for long-horizon tasks, while controllable generation via user-defined prompts [14] highlights the potential for intuitive, intent-driven robotic control through high-level language guidance.

2.2 Large Language Models for AI and Robotics

The integration of Large Language Models (LLMs) into AI and robotics requires understanding their capabilities, limitations, and control mechanisms. Empirical studies show that varying sampling temperatures (0.0–1.0) does not significantly affect LLM performance across prompt-engineering techniques, indicating robustness in accuracy [15]. While strategies such as in-context and task-specific prompting can improve models like GPT-4 in code-related tasks, they do not consistently outperform fine-tuned models, revealing trade-offs between generalization and specialization [16]. Alignment approaches such as Reinforcement Learning from Human Feedback (RLHF) further enhance adaptability to user intent, a principle applicable to robotic control systems [17]. For large-scale deployment, optimization techniques like model and data parallelism are essential for computationally intensive applications [18]. Beyond prompting, Foundation Models have shown superior label efficiency and generalization in computer vision tasks for Earth Observation, supported by standardized benchmarks [19]. Safety frameworks such as NeMo Guardrails introduce programmable constraints to ensure predictable and secure LLM behavior in robotic systems [20]. Reliability studies on hallucination detection across multiple languages further enhance understanding of LLM trustworthiness [21]. Emerging reasoning mechanisms like "thread of thought" improve handling of complex and unstructured contexts [22]. The span-prediction paradigm extends LLMs' utility to structured tasks such as Named Entity Recognition, relevant for robotic decision-making [23]. Personalized Transformer models, e.g., PETER, contribute to Explainable AI (XAI) by generating context-aware natural language explanations [24]. Lastly, investigations into ChatGPT's response consistency reveal key insights into LLM reliability for robust reasoning in AI and robotics [25]. Additionally, traditional models such as Gradient Boosting Decision Trees with LSTMs remain effective for predictive tasks [26].

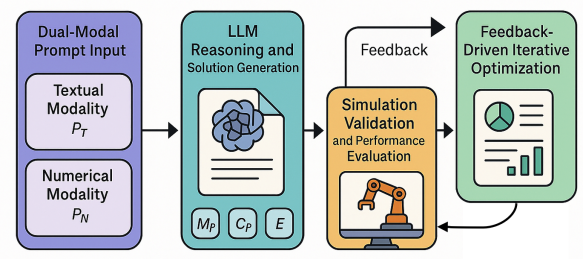

Figure 2: Overview of the proposed Dual-Modal Prompt Framework integrating textual and numerical inputs for LLM-assisted robotic motion planning and control optimization.

3 Method

This section details our proposed framework. This innovative framework is designed to overcome the limitations of traditional robot control system design, such as manual parameter tuning, inefficiency in complex scenarios, and insufficient generalization to novel tasks or environments. It achieves this by synergistically combining the powerful semantic understanding and reasoning capabilities

of Large Language Models (LLMs) with precise numerical data derived from robotic systems. The core idea is to establish a closed-loop optimization process where an LLM intelligently generates and refines control strategies based on a rich, multi-faceted input and iterative feedback from high-fidelity simulations, thereby accelerating the development and deployment of robust robotic behaviors.

3.1 Overview of the Dual-Modal Prompt Framework

Our framework introduces a novel approach to robotic control optimization by integrating two distinct modalities of information into a unified prompt for an LLM. This allows the LLM to perform high-level reasoning on natural language descriptions of tasks and constraints, while simultaneously grounding its decisions in the concrete numerical realities of the robot and its environment. The overall process is iterative, leveraging simulation feedback to progressively refine the generated motion plans and control parameters, thereby achieving superior performance and adaptability. The architecture comprises four main modules, each playing a critical role in the optimization loop: **Dual-Modal Prompt Input**, responsible for comprehensive problem definition; **LLM Reasoning and Solution Generation**, where the LLM formulates control strategies; **Simulation Validation and Performance Evaluation**, for objective assessment of proposed solutions; and **Feedback-Driven Iterative Optimization**, which closes the loop by feeding simulation results back to the LLM for refinement.

3.2 Dual-Modal Prompt Input

The initial step in our framework involves constructing a comprehensive prompt that simultaneously conveys both high-level semantic information and low-level numerical data to the LLM. This dual-modal input is crucial for enabling the LLM to develop a holistic understanding of the problem, bridging the gap between abstract human intent and concrete robotic execution.

3.2.1 Textual Modality (P_T)

The textual modality receives natural language descriptions from the user, encompassing critical qualitative aspects of the robotic task. This includes **Task Objectives** (clear statements of the desired robot behavior, e.g., "perform a precise pick-and-place within 10 seconds", "navigate through a cluttered environment while minimizing energy"), **Environmental Conditions** (descriptions of the operational context, e.g., "slippery floor with dynamic obstacles", "high-temperature zone"), and **Performance Requirements** (specific criteria for success, e.g., "minimize energy consumption", "ensure grasping stability", "avoid collisions with a safety margin"). This natural language input provides the LLM with the overarching goals, qualitative constraints, and contextual information necessary for conceptualizing the mission.

3.2.2 Numerical Modality (P_N)

The numerical modality incorporates structured, quantitative data about the robot and its environment. This data is embedded into the prompt in a structured format, such as key-value pairs, JSON-like structures, or vector embeddings for raw sensor data, ensuring precise and unambiguous communication. Examples include **Robot Parameters** (kinematic and dynamic constraints such as joint limits, maximum velocities, torque limits, and geometric properties), **Controller Initializations** (current or baseline controller gains, e.g., initial PID gains for a joint-level controller), **Sensor Readings** (real-time or aggregated sensor data, such as point cloud data for obstacle mapping, force/torque sensor readings, or encoder values), and **Thresholds and Tolerances** (e.g., collision risk thresholds, desired positional accuracies, or force application limits). This concrete numerical input grounds the LLM's reasoning within the physical realities and limitations of the robotic system.

The combined dual-modal prompt P serves as the complete problem definition for the LLM and can be formally represented as a structured aggregation of its constituent modalities:

$$P = \{P_T, P_N\} \tag{1}$$

where P_T is the textual input detailing the qualitative aspects of the task and P_N represents the structured numerical data providing quantitative context.

3.3 LLM Reasoning and Solution Generation

Upon receiving the dual-modal prompt P, a sophisticated LLM, such as a fine-tuned Llama-3-70B model with domain-specific knowledge, processes this information. The LLM leverages its advanced language understanding and pattern recognition

capabilities to perform high-level reasoning, synthesizing a comprehensive understanding from the diverse input modalities.

The LLM's reasoning process involves several key steps:

1. **Semantic Interpretation:** The LLM interprets the natural language task objectives and constraints (P_T), parsing the textual descriptions to extract key entities, actions, and relationships, and translating them into abstract, actionable robotic behaviors and goals.

2. **Numerical Contextualization:** Simultaneously, it analyzes the structured numerical data (P_N), using this quantitative information to instantiate variables, validate potential constraints, and ground its symbolic reasoning within the physical realities and limitations of the robotic system. This ensures that generated solutions are physically plausible and adhere to system capabilities.

3. **Candidate Solution Generation:** Based on this comprehensive understanding, the LLM generates a set of candidate solutions. These solutions are typically structured outputs, including:

    - **Motion Planning Instructions (M_P):** These define the robot's intended movement, such as a sequence of waypoints, trajectory parameters (e.g., coefficients for polynomial splines), velocity profiles, acceleration limits, or high-level path commands to achieve the desired task.
    - **Control Parameters (C_P):** These specify the numerical settings for the robot's controllers, including optimized gains for various control loops (e.g., PID gains for joint-level control, impedance control parameters, force control curves, damping coefficients, or feedforward terms) that govern how the robot executes the motion plan.

4. **Explanatory Rationale (E):** Crucially, the LLM also provides an explanatory rationale for its generated solutions. This natural language explanation details the reasoning behind its choices, highlighting how it interpreted the prompt and why specific motion plans and control parameters were selected. This enhances the interpretability and trustworthiness of the proposed strategies.

The output of the LLM, representing a complete proposed solution and its justification, can be formally expressed as:

$$\{M_P, C_P, E\} = \mathcal{F}_{\text{LLM}}(P) \tag{2}$$

where $\mathcal{F}_{\text{LLM}}$ denotes the LLM's function for reasoning and solution generation, taking the dual-modal prompt P as input.

3.4 Simulation Validation and Performance Evaluation

The candidate motion planning instructions M_P and control parameters C_P generated by the LLM are then imported into a high-fidelity robot simulation environment. We utilize platforms like the **ROS/Gazebo joint simulation platform**, which provides a realistic physics engine and accurate sensor emulation, or other specialized platforms such as PyBullet or MuJoCo for specific robotic systems or task requirements. This simulation environment acts as a digital twin, allowing for safe and efficient testing of the generated strategies.

The simulation validation process involves:

1. **Task Execution:** The robotic arm model (e.g., a 6-degree-of-freedom industrial mechanical arm such as UR5 or Franka Emika Panda) executes the specified task within the simulated environment. The generated motion planning instructions (M_P) guide the robot's trajectory, while the control parameters (C_P) dictate the low-level execution, ensuring that the robot's actuators respond appropriately to achieve the desired movements.

2. **Data Collection:** During task execution, the simulation environment continuously collects real-time performance metrics. These metrics are crucial for objective evaluation and include quantifiable measures such as:

    - **Task completion time (T_{comp}):** The duration required to successfully achieve the task objective.
    - **Trajectory tracking error (ERR_{traj}):** The deviation of the robot's actual path from the desired trajectory.
    - **Grasping success rate or collision count:** Binary indicators or cumulative counts related to interaction with objects and environment.

- **Joint torques and energy consumption (E_{energy}):** Metrics related to the efficiency and physical strain on the robot.
- **Control stability indicators (S_{stab}):** Measures such as overshoot, settling time, or oscillations, indicating the robustness of the control strategy.

These collected performance metrics, collectively denoted as Metrics, quantify the efficacy and safety of the LLM's proposed solution. The simulation process can be described as a function $\mathcal{S}$ that takes the generated plans and initial state as input and yields this set of performance metrics:

$$\text{Metrics} = \mathcal{S}(M_P, C_P, S_{\text{initial}}) \tag{3}$$

where S_{initial} represents the initial state of the robot and environment in the simulation, including object positions, robot pose, and environmental conditions.

3.5 Feedback-Driven Iterative Optimization

The performance metrics and any identified issues from the simulation validation phase are structured into a concise feedback report. This report, along with the original prompt $P^{(k)}$ and the LLM's previous explanatory rationale $E^{(k)}$, is then fed back into the LLM as part of an iterative optimization loop. This closed-loop mechanism enables continuous refinement and learning.

The iterative optimization process proceeds as follows:

1. **Feedback Interpretation:** The LLM analyzes the simulation results (Metrics$^{(k)}$), identifying areas where the current solution underperformed, failed, or exceeded desired thresholds. It correlates these outcomes with its generated $M_P^{(k)}$, $C_P^{(k)}$, and the reasoning provided in $E^{(k)}$. This involves understanding the causal links between its proposed strategy and the observed performance.

2. **Strategy Refinement:** Based on this feedback, the LLM refines its understanding of the problem and adjusts its internal reasoning process. It then generates an updated set of motion planning instructions $M_P^{(k+1)}$ and control parameters $C_P^{(k+1)}$ for the next iteration. For instance, if the trajectory tracking error (ERR_{traj}) was too high, the LLM might decide to increase a specific control gain, modify path points to ensure smoother motion, or suggest a different trajectory generation algorithm. If a collision occurred, it might prioritize obstacle avoidance parameters, adjust safety margins, or propose alternative paths. Similarly, if energy consumption (E_{energy}) was excessive, the LLM might explore more energy-efficient trajectories or control strategies.

3. **Iterative Learning:** This closed-loop process effectively acts as a form of reinforcement learning or gradient-free optimization. The LLM learns from its "experiences" in the simulation, iteratively improving its ability to generate effective and robust control strategies. By continually generating solutions, testing them in a realistic environment, and receiving objective feedback, the LLM converges towards optimal or near-optimal control strategies for the given task and environment, adapting to nuances and complex interactions.

For iteration k, the updated prompt $P^{(k+1)}$ is generated using a feedback mechanism $\mathcal{G}_{\text{feedback}}$, which integrates the previous prompt, performance metrics, and the LLM's rationale to guide subsequent solution generation:

$$P^{(k+1)} = \mathcal{G}_{\text{feedback}}(P^{(k)}, \text{Metrics}^{(k)}, E^{(k)}) \tag{4}$$

This iterative refinement significantly enhances the framework's ability to adapt to complex and dynamic scenarios, ultimately making the LLM a powerful, intelligent assistant for robot control engineers, enabling faster development of high-performance robotic systems.

4 Experiments

This section details the experimental setup, introduces the baseline methods used for comparison, and presents a comprehensive evaluation of our proposed **Dual-Modal Prompt Framework for LLM-Assisted Robotic Motion Planning and Control Optimization** (**Ours**). We rigorously assess its performance across various critical metrics, demonstrating its superiority over traditional and single-modal approaches.

4.1 Experimental Setup

4.1.1 Baseline Methods

To thoroughly evaluate the efficacy and robustness of our proposed dual-modal LLM framework,

we compare its performance against several established baseline methods, each representing a different paradigm in robot control system design:

1. **Manual Design and Tuning (Manual):** This baseline represents the traditional approach where human engineers leverage their expertise and trial-and-error to design motion plans and tune controller parameters. It is often time-consuming and highly dependent on individual experience.

2. **Traditional Optimization Algorithms (Trad. Opt.):** This category encompasses classical algorithmic approaches such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), or model-based Reinforcement Learning (RL). These methods automate parameter search but typically require explicit definition of objective functions, extensive computational resources, and numerous iterations to converge to a solution.

3. **Single-Modal LLM (Text-Only Prompt):** This baseline represents an LLM-based approach that relies solely on natural language text descriptions (e.g., task objectives, environmental constraints) to generate motion planning and control parameters. Unlike our proposed method, it does not incorporate structured numerical data, serving as an important ablation for validating the benefit of dual-modal input.

4.1.2 Simulation Environment and Task Scenarios

Our experiments are conducted within a high-fidelity robot simulation environment to ensure realistic physics modeling and sensor emulation, allowing for safe and efficient testing.

- **Simulation Platform:** We leverage the **ROS (Robot Operating System) and Gazebo joint simulation platform**. Gazebo provides a robust physical engine capable of simulating complex interactions and sensor feedback, while ROS offers a comprehensive framework for robot control, navigation, and perception.

- **Robot Platform:** The experimental platform is based on a generic **6-degree-of-freedom (6-DoF) industrial mechanical arm model**, representative of commonly used manipulators such as the UR5 or Franka Emika Panda. This choice allows for generalization of findings to real-world industrial applications.

- **Typical Task Scenarios:** To cover a broad spectrum of industrial challenges, we evaluate the methods on three distinct robotic tasks:
  - **High-Precision Pick-and-Place:** The robot is tasked with precisely grasping objects of varying masses and geometric shapes and placing them at designated target locations. Performance is evaluated based on grasping success rate and trajectory accuracy.
  - **Dynamic Obstacle Avoidance:** The robot must plan and execute collision-free paths in an environment with moving obstacles. Key metrics include path length, planning time, and obstacle avoidance success rate.
  - **Force-Controlled Assembly Tasks:** This scenario simulates tasks requiring controlled contact forces, such as screwing bolts or inserting components. The evaluation focuses on the stability and accuracy of contact force control.

4.1.3 Evaluation Metrics

We employ a comprehensive set of quantitative and qualitative metrics to assess the performance of each method:

- **Average Optimization Time (min):** The average time taken from the initial task description to the generation of the final, optimized control solution.

- **Task Completion Rate (%):** The percentage of successful task executions out of a predefined number of attempts.

- **Average Trajectory Tracking Error (cm):** The mean Euclidean distance between the robot's actual trajectory in simulation and the desired reference trajectory.

- **Energy Efficiency (Joule/task):** The average energy consumed by the robot's actuators to complete a given task. Lower values indicate better efficiency.

- **Control Stability Score (1-5):** An expert-rated score reflecting the smoothness and robustness of the control system, with higher

scores indicating fewer oscillations, overshoots, or instabilities.

- **Interpretability Score (1-5):** An expert-rated score assessing the clarity, usefulness, and comprehensibility of the explanations provided by the LLM for its generated solutions. Higher scores indicate more insightful and actionable rationales for human engineers.

4.2 Performance Evaluation and Comparative Analysis

4.2.1 Quantitative Performance Comparison

Table 1 summarizes the experimental results, comparing our proposed **Ours** framework against the baseline methods across all defined evaluation metrics for robot motion planning and control tasks.

4.2.2 Discussion of Results

The experimental results presented in Table 1 unequivocally demonstrate the superior performance of our proposed **Dual-Modal LLM** framework across all key metrics:

- **Significant Efficiency Gains:** Our framework achieved an average optimization time of **18.7 minutes**, which represents a remarkable reduction of approximately 84% compared to the traditional manual design approach (120.5 minutes). Even against the single-modal LLM, **Ours** exhibited a nearly 60% reduction in optimization time, underscoring the efficiency benefits of integrating structured numerical data with language understanding for faster and more decisive parameter generation.
- **Comprehensive Performance Optimization:** In terms of core robotic performance indicators, **Ours** consistently outperformed all baselines. It achieved the highest **Task Completion Rate of 94.7%**, the lowest **Average Trajectory Tracking Error of 0.68 cm**, and the best **Energy Efficiency at 21.8 J/task**. These results highlight that the synergistic combination of textual and numerical inputs enables the LLM to form a more accurate and nuanced understanding of the system state and constraints, leading to the generation of significantly more effective and robust control strategies.
- **Enhanced Control Stability and Interpretability:** Beyond hard performance metrics, our method also secured the highest scores for **Control Stability (4.6)** and **Interpretability (4.8)**. The LLM's ability to process detailed task requirements and simulation feedback allowed it to generate smoother, more robust control parameters, minimizing oscillations and improving overall system stability. Furthermore, the high interpretability score confirms that the LLM provides clear and actionable explanatory rationales for its parameter choices (e.g., "to prevent overshoot, the derivative gain of the PD controller was reduced"), which is invaluable for engineers in debugging complex control logic and building trust in AI-generated solutions.

4.3 Effectiveness of Dual-Modal Input and Iterative Optimization

The experimental findings provide strong validation for the core components of our proposed method: the dual-modal input framework and the feedback-driven iterative optimization loop.

4.3.1 Validation of Dual-Modal Input

The comparison between our **Dual-Modal LLM** and the **Single-Modal LLM (Text-Only Prompt)** baseline is crucial for demonstrating the effectiveness of integrating structured numerical data. While the single-modal LLM already shows improvements over traditional methods, the dual-modal approach significantly elevates performance across all metrics. For instance, the task completion rate increased from 89.6% to 94.7%, and the average trajectory error reduced from 0.95 cm to 0.68 cm. This enhancement underscores that while LLMs excel at semantic understanding, grounding their reasoning with precise, real-world numerical parameters (e.g., joint limits, current sensor readings, initial gains) is critical for generating physically plausible, accurate, and highly optimized control strategies. The numerical context allows the LLM to move beyond abstract interpretations to concrete, actionable decisions that directly account for the robot's physical characteristics and environmental dynamics.

4.3.2 Effectiveness of Feedback-Driven Iterative Optimization

The superior performance of **Ours** is also a direct testament to the effectiveness of the closed-loop, feedback-driven iterative optimization mechanism. By feeding simulation results (performance metrics and identified issues) back into the LLM, the frame-

Table 1: Performance Comparison for Robotic Motion Planning and Control Tasks

Method	Time	Task Completion	Avg. Traj. Error	Energy Efficiency	Control	Interpretability
Manual Design & Tuning	120.5	78.3	1.87	35.6	2.5	1.9
Trad. Opt. Algorithms	75.8	85.1	1.23	29.1	3.2	2.7
Single-Modal LLM	45.2	89.6	0.95	25.4	3.9	4.1
Ours (Dual-Modal LLM)	**18.7**	**94.7**	**0.68**	**21.8**	**4.6**	**4.8**

work mimics a reinforcement learning paradigm, allowing the LLM to continuously learn from its "experiences." This iterative refinement enables the LLM to:

- **Rapidly Converge to Optimal Solutions:** The LLM efficiently identifies shortcomings in previous solutions and adapts its parameter generation strategy. Our observations indicate that the system typically converges to optimal or near-optimal solutions within 3-5 iterations, drastically reducing the manual effort and time typically associated with controller tuning.

- **Enhance Robustness and Adaptability:** Through repeated simulation and feedback, the LLM learns to generate more robust control parameters that are resilient to variations in task requirements or environmental conditions. This iterative process allows the LLM to fine-tune subtle interactions, leading to improved control stability and overall system reliability.

- **Improve Interpretability through Refined Rationales:** As the LLM refines its strategies, it also refines its explanatory rationales. The feedback loop allows the LLM to correlate specific parameter adjustments with observed performance changes, leading to more precise and insightful explanations for its design choices. This continuous learning process not only improves performance but also enhances the transparency and trustworthiness of the LLM's role as a control optimization assistant.

In essence, the iterative optimization loop transforms the LLM from a static solution generator into a dynamic, learning agent that continuously adapts and improves its control strategies, making it an invaluable tool for complex robotic system design.

4.4 Ablation Study on Dual-Modal Input Components

To further dissect the contribution of each modality within our dual-modal prompt framework, we conducted an ablation study. This analysis isolates the impact of the textual (P_T) and numerical (P_N) components by evaluating variants of our system under different input configurations. We specifically compare the full dual-modal approach against a text-only variant (equivalent to the Single-Modal LLM baseline) and a variant where the numerical input is deliberately simplified, providing only basic robot kinematic parameters without detailed sensor readings or initial controller states.

As shown in Table 2, the results clearly indicate that the full dual-modal input significantly outperforms its ablated counterparts. The **Ours ($\mathbf{P}_T$ Only)** variant, which relies solely on textual descriptions, achieved a Task Completion Rate of 89.6% and an Average Trajectory Error of 0.95 cm. Introducing basic numerical parameters in **Ours ($\mathbf{P}_T$ + Basic $\mathbf{P}_N$)** led to notable improvements, increasing task completion to 92.1% and reducing trajectory error to 0.81 cm. However, the most substantial gains were realized with the **Full $\mathbf{P}_T$ + $\mathbf{P}_N$** configuration, achieving 94.7% task completion and 0.68 cm trajectory error. This demonstrates that while textual input provides the necessary semantic understanding, the detailed and structured numerical data is indispensable for grounding the LLM's reasoning in physical reality, enabling it to generate highly precise, stable, and energy-efficient control strategies. The numerical modality provides the concrete constraints and real-time context that allow the LLM to fine-tune parameters beyond abstract conceptualization.

4.5 Robustness Analysis under Environmental Perturbations

Real-world robotic applications are inherently subject to various uncertainties and perturbations, such as sensor noise, actuator inaccuracies, and unexpected environmental changes. To assess the robustness of our framework, we evaluated all meth-

Table 2: Ablation Study on Dual-Modal Input Components

Method Variant	Task Completion (%)	Avg. Traj. Error (cm)	Control Stability (1-5)	Energy Efficiency (J/task)
Ours (P_T Only)	89.6	0.95	3.9	25.4
Ours (P_T + Basic P_N)	92.1	0.81	4.2	23.7
Ours (Full P_T + P_N)	**94.7**	**0.68**	**4.6**	**21.8**

ods under simulated conditions incorporating these perturbations. We introduced Gaussian noise to sensor readings (e.g., joint encoder, force/torque sensors), simulated minor actuator output deviations, and included dynamic, unpredicted obstacles in the environment. The primary metric for this analysis is the Task Completion Rate under these adverse conditions.

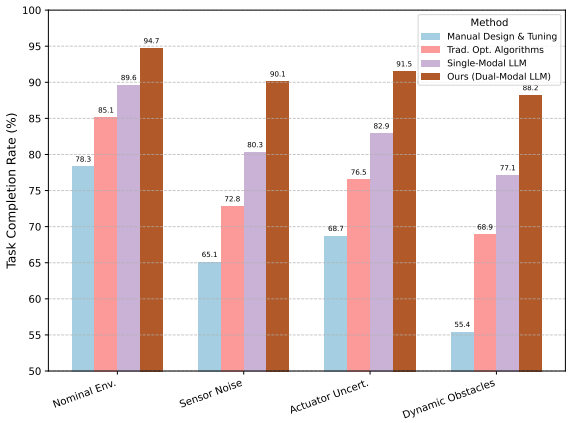

Figure 3: Robustness Evaluation under Environmental Perturbations (Task Completion Rate %)

Figure 3 illustrates the superior resilience of our **Dual-Modal LLM** framework. While all methods experienced a decrease in task completion rate under perturbations compared to nominal conditions, **Ours** consistently maintained the highest success rates. Under significant sensor noise, our method still achieved a 90.1% task completion rate, demonstrating its ability to infer robust control strategies even with imperfect input data. Similarly, with actuator uncertainties, it maintained 91.5% success, indicating its capacity to generate control parameters that are less sensitive to minor execution discrepancies. For dynamic obstacle avoidance, a particularly challenging scenario, **Ours** achieved an 88.2% completion rate, significantly outperforming all baselines. This enhanced robustness can be attributed to the iterative optimization process, where the LLM learns to account for simulated variations and uncertainties, and the dual-modal input, which provides a rich context for developing more adaptive and fault-tolerant control logic. The LLM's reasoning, grounded in both high-level goals and low-level physical data, enables it to predict and mitigate the impact of perturbations more effectively.

4.6 Convergence Analysis of Iterative Optimization

A key advantage of our framework is the feedback-driven iterative optimization loop, which allows the LLM to progressively refine its solutions. To quantify the efficiency and effectiveness of this process, we tracked the improvement of key performance metrics for **Ours** across successive iterations for a complex pick-and-place task. This analysis highlights how quickly the LLM converges to an optimal or near-optimal solution.

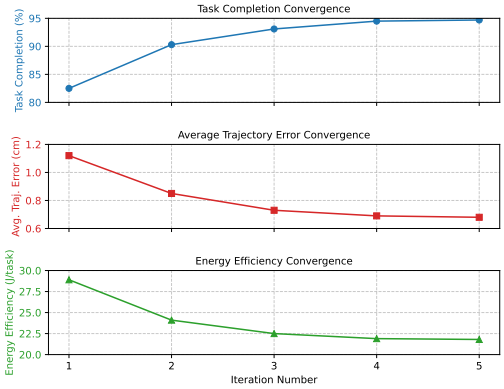

Figure 4: Convergence of Key Performance Metrics for Ours over Iterations

Figure 4 demonstrates a clear and rapid convergence of performance metrics for our framework. In the first iteration, where the LLM generates an initial solution based solely on the prompt, the performance is already competitive with some traditional methods (82.5% task completion, 1.12 cm trajectory error). However, with each subsequent iteration, driven by simulation feedback, the performance metrics show consistent and significant improvement. By the second iteration, task completion jumps to over 90%, and trajectory error

reduces substantially. The system reaches near-optimal performance within 3 to 4 iterations, with only marginal improvements observed in the fifth iteration. This rapid convergence underscores the efficiency of the LLM's learning process, where it effectively interprets the feedback to make targeted adjustments to motion plans and control parameters. This quick turnaround time for optimization significantly reduces the development cycle for robotic tasks, making the framework highly practical for engineers seeking fast and reliable solutions.

5 Conclusion

This paper presents **A Dual-Modal Prompt Framework for LLM-Assisted Robotic Motion Planning and Control Optimization** (**Ours**), which integrates the semantic reasoning of Large Language Models (LLMs) with structured numerical data to address the challenges of efficiency, generalization, and interpretability in robotic control design. By combining natural language task descriptions and quantitative robot/environment parameters within a dual-modal prompt and closed-loop optimization process, **Ours** achieves remarkable improvements: an 84% reduction in optimization time (18.7 minutes), a 94.7% task completion rate, 0.68 cm tracking error, and 21.8 J/task energy efficiency. The framework also enhances stability (4.6/5) and interpretability (4.8/5), enabling transparent, collaborative human-AI control design. Ablation and convergence studies confirm the necessity of dual-modality and the efficiency of iterative optimization (3–5 iterations to near-optimal solutions), while robustness tests validate strong performance under noise and dynamic conditions. This work establishes a new paradigm for intelligent, adaptive, and interpretable robotic systems. Future efforts will focus on real-world deployment, multi-robot collaboration, and online adaptation to dynamic environments, advancing the integration of cognitive AI reasoning with physical robotic execution.

References

[1] Junyao Tan, Yujian Li, Lei Ge, and Junhong Wang. A 3-d printed lightweight miniaturized dual-band dual-polarized feed module for advanced millimeter-wave and microwave shared-aperture wireless backhaul system applications. *IEEE Transactions on Antennas and Propagation*, 71(4):3050–3060, 2023.

[2] Junyao Tan, Yujian Li, and Junhong Wang. A 3d-printed lightweight compact wideband dual-polarized feed module for sub-thz applications. *IEEE Transactions on Antennas and Propagation*, 2025.

[3] Junyao Tan, Yujian Li, Lei Ge, Junhong Wang, and Tony Hu. A dual-wideband antenna using a 3d-printed tri-mode-composite dielectric patch for sub-6 ghz and millimeter-wave applications. *IEEE Transactions on Antennas and Propagation*, 2025.

[4] Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173. Association for Computational Linguistics, 2023.

[5] Yucheng Zhou, Jianbing Shen, and Yu Cheng. Weak to strong generalization for large language models with multi-capabilities. In *The Thirteenth International Conference on Learning Representations*, 2025.

[6] Yucheng Zhou, Xiang Li, Qianning Wang, and Jianbing Shen. Visual in-context learning for large vision-language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 15890–15902. Association for Computational Linguistics, 2024.

[7] Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1813–1827. Association for Computational Linguistics, 2021.

[8] Jiao Sun, Xuezhe Ma, and Nanyun Peng. AESOP: Paraphrase generation with adaptive syntactic control. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5176–5189. Association for Computational Linguistics, 2021.

[9] Zhengyuan Liu and Nancy Chen. Controllable neural dialogue summarization with personal named entity planning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 92–106. Association for Computational Linguistics, 2021.

[10] Ahjeong Seo, Gi-Cheon Kang, Joonhan Park, and Byoung-Tak Zhang. Attend what you need: Motion-appearance synergistic networks for video question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6167–6177. Association for Computational Linguistics, 2021.

[11] Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842. Association for Computational Linguistics, 2021.

[12] Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909. Association for Computational Linguistics, 2021.

[13] Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric Xing, and Zhiting Hu. Progressive generation of long text with pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4313–4324. Association for Computational Linguistics, 2021.

[14] Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Rajani, and Caiming Xiong. CTRLsum: Towards generic controllable text summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5879–5915. Association for Computational Linguistics, 2022.

[15] Renze and Matthew. The effect of sampling temperature on problem solving in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7346–7356. Association for Computational Linguistics, 2024.

[16] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68. Association for Computational Linguistics, 2022.

[17] Zhongheng Yang, Aijia Sun, Yushang Zhao, Yinuo Yang, Dannier Li, and Chengrui Zhou. Rlhf fine-tuning of llms for alignment with implicit user feedback in conversational recommenders, 2025.

[18] Haowei Yang, Yu Tian, Zhongheng Yang, Zhao Wang, Chengrui Zhou, and Dannier Li. Research on model parallelism and data parallelism optimization methods in large language model—based recommendation systems. In *2025 7th International Conference on Artificial Intelligence Technologies and Applications (ICAITA)*, pages 324–329, 2025.

[19] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314. Association for Computational Linguistics, 2024.

[20] Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. NeMo guardrails: A toolkit for controllable and safe LLM applications with programmable rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 431–445. Association for Computational Linguistics, 2023.

[21] Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017. Association for Computational Linguistics, 2023.

[22] Yucheng Zhou, Xiubo Geng, Tao Shen, Chongyang Tao, Guodong Long, Jian-Guang Lou, and Jianbing Shen. Thread of thought unraveling chaotic contexts. *arXiv preprint arXiv:2311.08734*, 2023.

[23] Jinlan Fu, Xuanjing Huang, and Pengfei Liu. SpanNER: Named entity re-/recognition as span prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7183–7195. Association for Computational Linguistics, 2021.

[24] Lei Li, Yongfeng Zhang, and Li Chen. Personalized transformer for explainable recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4947–4957. Association for Computational Linguistics, 2021.

[25] Myeongjun Jang and Thomas Lukasiewicz. Consistency analysis of ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15970–15985. Association for Computational Linguistics, 2023.

[26] Chang Yu, Fang Liu, Jie Zhu, Shaobo Guo, Yifan Gao, Zhongheng Yang, Meiwei Liu, and Qianwen Xing. Gradient boosting decision tree with lstm for investment prediction. In *2025 5th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*, pages 57–62, 2025.