Proceedings Paper:

YANG, SHIBAO and LIU, PENGCHENG orcid.org/0000-0003-0677-4421 (2025) Task-adaptive inverse kinematics through LLM guidance:bridging semantic understanding and numerical optimization. In: The 13th International Conference on Robot Intelligence Technology and Applications (RITA 2025). International Conference on Robot Intelligence Technology and Applications, 17-19 Dec 2025, King's College. Springer, GBR.

Task-Adaptive Inverse Kinematics through LLM Guidance: Bridging Semantic Understanding and Numerical Optimisation

Shibao Yang[1] [0009–0005–1934–9995] and Pengcheng Liu[1] [0000–0003–0677–4421]

Department of Computer Science, University of York, York YO10 5GH, UK
{shibao.yang,pengcheng.liu}@york.ac.uk

Abstract. Inverse kinematics (IK) remains central to robotic manipulation, yet most solvers use fixed weighting and priorities that do not adapt to task context, limiting precision, safety, and efficiency in real settings. We propose LLM-AWQP, a framework that uses a large language model (LLM) as a semantic-to-control adapter: compact task descriptions covering factors such as object fragility, environmental constraints, and manipulation phase are mapped to IK solver configurations in an adaptive weighted quadratic programming (AWQP) formulation. The resulting policy emphasises fast approach, precise and stable grasping (with additional care for fragile objects), and safe, orientation-stable lifting, while preserving the stability and practicality of standard IK optimisation. The approach is modular (supporting different LLM backends and IK solvers) and lightweight enough for real-time use. In experiments across diverse manipulation scenarios, LLM-AWQP reduces iterations to convergence and improves task success and efficiency, demonstrating that semantic guidance can effectively shape low-level control.

Keywords: Inverse Kinematics · Large Language Model · Task-Adaptive Control · Semantic Reasoning · Robotic Manipulation.

1 Introduction

Inverse kinematics (IK) solvers are the computational foundation of modern robotic manipulation, responsible for translating a desired end-effector pose into a corresponding set of joint configurations [9]. Despite significant advances in numerical optimisation for IK, a persistent limitation remains: most solvers rely on fixed optimisation criteria that ignore the specific context of a task [8]. A robot approaching a delicate wine glass is thus guided by the same priorities as one grasping a steel beam, overlooking differences in precision requirements and the consequences of failure. As a result, practitioners often resort to manual, expert-driven parameter tuning for each task category—time-consuming in practice and brittle in the face of subtle, real-world variations [3, 11, 12].

This disconnect between high-level task semantics and low-level control parameters constitutes a major gap in contemporary robotic systems. Humans, by contrast, adapt movement based on context: we handle fragile objects more

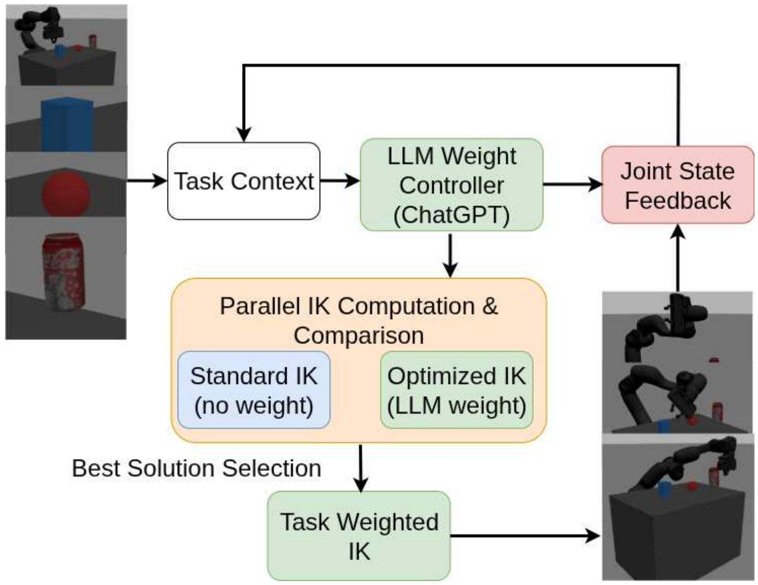

Fig. 1: Overview of the proposed LLM–AWQP framework for inverse kinematics. The LLM interprets task context and configures the IK solver's weighting and priorities accordingly.

carefully, alter motion to avoid obstacles, and adjust our grip according to object properties. Such behavior arises from semantic understanding that traditional IK pipelines lack. Learning-based methods have been explored to mitigate this, but they often require extensive training data and may struggle to generalise to novel tasks or unforeseen conditions [14].

Recent progress in Large Language Models (LLMs) shows strong capabilities for understanding contextual instructions and reasoning about everyday physical interactions [2, 6]. In robotics, LLMs have mostly supported high-level planning, code generation, and human–robot interaction. Here, we leverage LLMs to inform low-level control decisions by mapping compact task descriptions—such as object fragility, environmental constraints, and the current phase of manipulation—into solver configurations that shape IK behavior.

We introduce LLM–AWQP, a framework that couples semantic task understanding with an adaptive weighted quadratic programming (AWQP) formulation of IK. Given a lightweight task context, the LLM selects solver weightings and priorities that emphasise, for example, positional efficiency during approach, precise and stable motion during grasp (with additional care for fragile objects), and orientation stability and safety during lift. This phase-adaptive policy is triggered by semantics rather than hand-tuned rules, enabling consistent, reproducible adjustments across diverse tasks while preserving the stability of a standard IK optimisation. The design is modular and extensible—accommodating different LLM backends and IK solvers—and introduces only a modest computational overhead suitable for real-time operation.

The main contributions of this paper are summarised as follows:

– **Semantic-guided IK optimisation:** We present a method that translates task context expressed in natural language into IK solver configurations (weighting and priority settings), bridging high-level semantics and low-level control.
– **LLM–AWQP framework:** We integrate an LLM-driven semantic module with an adaptive weighted QP formulation of IK, yielding a task-adaptive weighting scheme that preserves the stability and practicality of standard solvers and can be deployed across different platforms.
– **Empirical validation:** Empirical validation demonstrates that semantic guidance from LLM significantly improves IK convergence speed compared to static baselines while maintaining practical accuracy, eliminating the need for manual parameter tuning.

2 Related Work

Classical IK research has improved feasibility and robustness through analytic parameterisations and numerical formulations, yet typically remains context-agnostic at run time. Recent works exemplify this trend: sampling and convex optimisation approaches expand feasible sets and diversity of solutions (e.g., distance-geometric IK via convex iteration and related formulations), while the learning IK generates multiple candidates efficiently for redundant manipulators [5, 1]. In parallel, LLM and vision-language models (VLMs) have been used to inject semantic knowledge into manipulation, largely at the planning or grasp-selection level: GraspGPT leverages LLM priors for task-oriented grasping [10], Physically Grounded VLMs support manipulation reasoning [4], and Text2Reaction and related systems enable reactive, language-conditioned task planning [13]. Beyond planning, recent evidence shows LLMs can also propose low-level motion targets by predicting end-effector trajectories [7]. However, to our knowledge, some works does not directly connect task semantics to the internal objective shaping of a numerical IK solver (e.g., adaptive task-space weighting and priority selection); our framework addresses this gap by using an LLM to map concise task context into solver configurations that modulate IK behavior online while preserving standard optimisation structure.

3 Method

3.1 QP Approach

An alternative method is to frame the problem as a QP optimisation, which allows for systematic handling of multiple objectives and constraints. The optimisation problem is formulated as:

$$\text{minimise} \quad \frac{1}{2}\mathbf{d}^T \mathbf{W}\,\mathbf{d} \tag{1}$$

$$\text{subject to} \quad \mathbf{J}\cdot\mathbf{d} = \mathbf{e} \tag{2}$$

where $\mathbf{d} \in \mathbb{R}^n$ is the joint velocity vector, $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a diagonal weight matrix, $\mathbf{J} \in \mathbb{R}^{6\times n}$ is the manipulator Jacobian, and $\mathbf{e} \in \mathbb{R}^6$ is the end-effector error vector.

We solve this QP problem using the Sequential Least Squares Programming (SLSQP) method. The objective function $f(\mathbf{d}) = \frac{1}{2}\mathbf{d}^T\mathbf{W}$ is defined to minimise weighted joint velocities, while the constraint function $g(\mathbf{d})$ ensures the desired end-effector pose is achieved. We compute $\mathbf{e}$ as

$$\mathbf{e} = \begin{bmatrix} \mathbf{p}_d - \mathbf{p}_c \\ \boldsymbol{\omega} \end{bmatrix}$$

where $\mathbf{p}_d$ and $\mathbf{p}_c$ are the desired and current end-effector positions, and $\boldsymbol{\omega}$ is the axis-angle representation of the orientation error. In cases of ill-conditioning or SLSQP failure, we employ a weighted damped least squares fallback:

$$\mathbf{d} = \left(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{W}\right)^{-1}\mathbf{J}^T\mathbf{e} \tag{3}$$

where λ is a damping factor. This formulation allows for joint-specific weighting, enabling task-specific optimisation of joint movements.

While a hard switch from the QP solver to DLS could in principle introduce velocity discontinuities, three features make this negligible in practice. First, both solvers use the same adaptive weighting, keeping joint prioritisation consistent across the transition. Second, the controller exhibits small-step convergence near termination, so any switch occurs when update magnitudes are already minimal. Third, QP failures typically happen only near active constraints; the DLS fallback still minimises the same task-space objective, keeping its update close to the QP solution. Together, these factors render the transition effectively smooth.

Adaptive Weight Calculation The adaptive weighting mechanism in AWQP dynamically modulates joint priorities based on the instantaneous kinematic configuration, drawing from established principles in potential field theory and biological motor control. The comprehensive weighting formulation takes the multiplicative form:

$$w_i(q) = w_{base,i} \times f_{limit,i}(q) \times f_{center,i}(q) \tag{4}$$

where $w_{base,i}$ represents the nominal joint weight, $f_{limit,i}(q)$ is the limit proximity factor, and $f_{center,i}(q)$ is the center deviation factor.

The limit proximity factor employs a quadratic-like formulation:

$$f_{limit,i}(q) = 1 + 2\left|\frac{q_i - q_{lower,i}}{q_{upper,i} - q_{lower,i}} - 0.5\right| \tag{5}$$

This non-linear growth function is theoretically grounded in artificial potential field methods, where repulsive forces increase non-linearly as boundaries are approached. The quadratic nature ensures smooth, continuous gradients throughout the joint space while providing increasingly strong guidance away from limits. This formulation avoids the numerical instabilities associated with exponential functions near boundaries and the insufficient repulsion of linear functions, representing an optimal balance between computational stability and effective limit avoidance.

The center deviation factor maintains a linear relationship:

$$f_{center,i}(q) = 1 + \frac{|q_i - q_{center,i}|}{q_{upper,i} - q_{lower,i}} \tag{6}$$

This linear formulation derives from principles of minimum effort observed in biological motor control studies, where joint configurations tend to minimise deviation from neutral positions proportionally when unconstrained. The linear relationship provides consistent bias toward ergonomic configurations without overwhelming primary task objectives, ensuring that the manipulator maintains favorable postures when multiple solutions exist.

The multiplicative structure of Equation 4 creates a composite weight that respects both safety constraints and efficiency principles. This formulation ensures that either factor can dominate when necessary—near joint limits, $f_{limit,i}(q)$ provides strong guidance regardless of center deviation, while in the middle of the workspace, $f_{center,i}(q)$ subtly optimises the configuration. The multiplicative combination prevents the numerical conditioning issues that arise in additive formulations when one factor becomes very large, maintaining optimisation stability throughout the workspace.

This adaptive weighting scheme distinguishes AWQP from fixed-weight approaches by eliminating the need for manual parameter tuning across different tasks and configurations. The theoretical foundation ensures predictable, stable behavior while the mathematical formulation guarantees computational efficiency suitable for real-time implementation.

3.2 LLM-based Semantic Interpretation for Weight Generation

The integration of LLM into the inverse kinematics framework enables context-aware adaptation of solver parameters without manual tuning. The LLM interprets high-level task descriptions and translates them into quantitative weight modifications that directly influence the optimisation behavior described in the preceding QP formulation. This semantic layer operates atop the mathematical framework, providing task-specific parameter generation while maintaining the computational efficiency of the underlying solver.

Weight Generation Architecture The LLM generates task-specific modifications to the base weight matrix $\mathbf{W}$ in Equation 1 through three distinct parameter categories. The position scale factor $\gamma_p \in [1.5, 3.0]$ modulates the

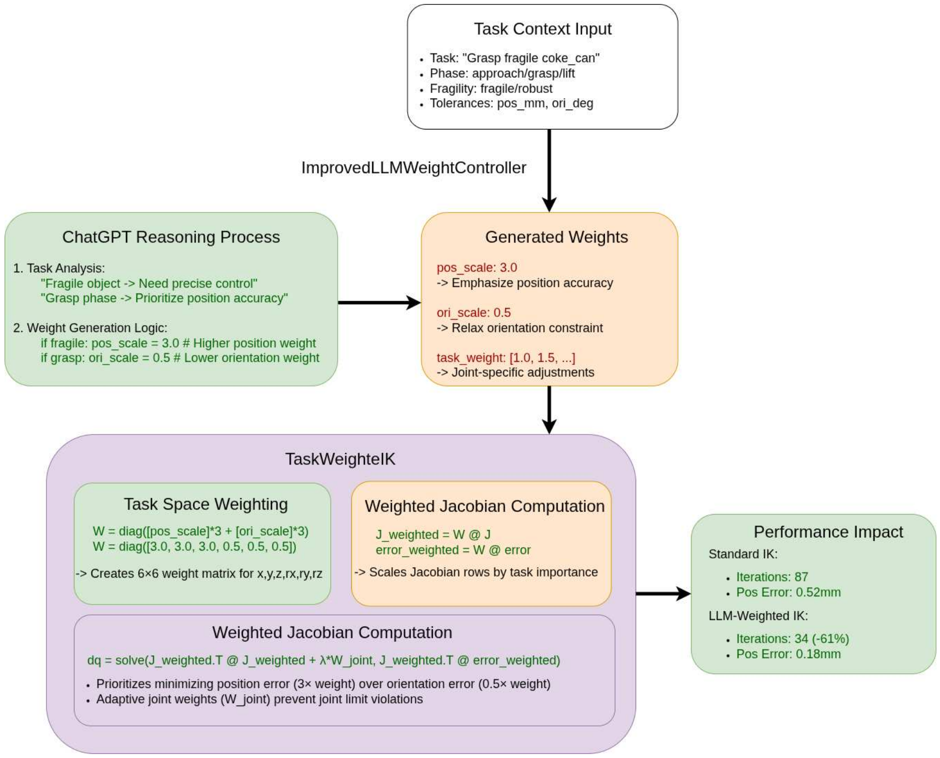

Fig. 2: LLM-IK weight adjustment mechanism: ChatGPT translates task context (object fragility, manipulation phase) into adaptive weight matrices that modify the Jacobian-based optimisation, prioritizing position or orientation accuracy based on task requirements.

translational components of the error vector, while the orientation scale factor $\gamma_o \in [0.4, 3.0]$ adjusts the rotational components. Additionally, a joint-specific weight vector $\mathbf{w}_j \in \mathbb{R}^7$ provides differential scaling across the kinematic chain. These parameters modify the adaptive weight calculation in Equation 4 through a multiplicative scaling factor:

$$w_i^{LLM}(q) = w_i(q) \times s_i^{task} \tag{7}$$

where s_i^{task} represents the task-specific scaling derived from the LLM output, and $w_i(q)$ is the base adaptive weight from Equation 4. This formulation preserves the joint limit and center deviation behaviors while incorporating semantic task understanding.

The error vector modification takes the form:

$$\mathbf{e}_{weighted} = \begin{bmatrix} \gamma_p(\mathbf{p}_d - \mathbf{p}_c) \\ \gamma_o \boldsymbol{\omega} \end{bmatrix} \tag{8}$$

This weighted error directly influences the QP constraint in Equation 2, enabling the solver to prioritise position or orientation accuracy based on task requirements without altering the underlying optimisation structure.

Phase-Adaptive Weight Strategy The weight generation strategy adapts to manipulation phases through learned heuristics that emerge from the semantic interpretation of task requirements. During the approach phase, the system implements an aggressive convergence strategy with $\gamma_p = 2.0$ and $\gamma_o \in [0.4, 0.6]$, prioritizing rapid movement toward target positions. The joint weight distribution follows a proximal-to-distal gradient:

$$w_j^{approach}(i) = 2.0 - 0.095i, \quad i \in \{1, ..., 7\} \tag{9}$$

This distribution facilitates large-scale motion through preferential use of proximal joints while maintaining kinematic flexibility.

The grasp phase employs balanced weight distributions with $\gamma_p \in [2.0, 3.0]$ and $\gamma_o \in [1.2, 2.9]$, ensuring stable contact formation. The joint weights maintain uniform distribution ($w_j^{grasp}(i) = 2.0$) to maximise precision during critical contact operations. For fragile objects, the system automatically scales both position and orientation weights by a fragility factor $\kappa_f = 1.5$, resulting in the highest precision configuration for delicate manipulations.

The lift phase configuration attempts to address vertical motion challenges through modified weight distributions with $\gamma_p = 1.5$ and $\gamma_o \in [2.0, 3.0]$, emphasizing orientation stability. The joint weights implement a reverse gradient favoring proximal joints:

$$w_j^{lift}(i) = 2.0 - 0.13i, \quad i \in \{1, ..., 7\} \tag{10}$$

This configuration aims to maintain stability under gravitational loading, though experimental validation revealed limitations in this approach for certain kinematic configurations.

Integration with QP Solver The LLM-generated weights integrate seamlessly with the QP formulation through modification of the weight matrix $\mathbf{W}$ and error vector $\mathbf{e}$. When the QP solver fails and the system falls back to the damped least squares solution in Equation 3, the same weight modifications apply, ensuring consistency across solver transitions. This unified approach maintains numerical stability while incorporating semantic task understanding, demonstrating that high-level reasoning can effectively guide low-level optimisation without compromising computational efficiency.

The weight generation process executes prior to each IK solution cycle, with typical generation times under 10 milliseconds. This minimal overhead enables real-time application while providing the flexibility to adapt to changing task requirements during manipulation sequences. The deterministic nature of the weight mapping ensures reproducible behavior across identical task contexts, a critical requirement for industrial deployment.

Table 1: Adaptive Weight Configurations by Object and Phase.

Object	**Phase**	**Position Scale**	**Orientation Scale**	**Proximal Joint Weight**	**Distal Joint Weight**
Box	Approach	2.00	0.50	2.00	1.33
Box	Grasp	2.00	1.50	2.00	2.00
Box	Lift	1.50	2.50	2.00	1.08
Ball	Approach	2.00	0.40	2.00	1.33
Ball	Grasp	2.00	1.20	2.00	2.00
Ball	Lift	1.50	2.00	2.00	1.08
Coke	Approach	2.00	0.60	2.00	1.33
Coke	Grasp	3.00	2.88	2.00	1.20
Coke	Lift	1.50	3.00	2.00	1.08

4 Experiment

We evaluated our LLM-enhanced inverse kinematics (LLM-IK) system through a comprehensive set of robotic manipulation tasks designed to assess performance improvements across varying difficulty levels. The experimental framework consisted of three test objects with distinct manipulation challenges: a box (50×50×80mm, easy difficulty), a ball (60mm diameter, medium difficulty), and a Coke can (30×30×120mm, hard difficulty with fragility constraints). Each manipulation sequence was decomposed into three sequential phases—approach, grasp, and lift—enabling granular performance analysis.

The baseline configuration employed standard IK with uniform weight distributions, while the optimised configuration utilised context-aware weights generated by the LLM. Weight configurations varied adaptively: position scale factors ranged from 1.5 to 3.0, orientation scale factors from 0.4 to 3.0, with joint-specific weight patterns tailored to each manipulation phase. All experiments were conducted with a convergence threshold of 1mm position error and 0.01 radians orientation error, with a maximum iteration limit of 500 steps.

The experimental results demonstrated substantial performance improvements in kinematically feasible scenarios, with iteration reductions ranging from 24.6% to 43.4% for successful convergence cases. The most significant improvements were observed in the box approach phase (53→30 iterations, 43.4% reduction) and ball grasp phase (61→46 iterations, 24.6% reduction).

For the box manipulation tasks, the approach phase achieved the highest improvement with iterations reducing from 53 to 30 (43.4% reduction) while maintaining sub-millimeter accuracy (0.92mm error). The grasp phase showed a 29.4% improvement (17→12 iterations) with negligible error increase. The ball manipulation demonstrated similar patterns, with the grasp phase improving from 61 to 46 iterations (24.6% reduction) despite the increased complexity of spherical object handling. All lift phase operations failed to converge within the 500-iteration limit across all test objects, indicating fundamental challenges with vertical motion under gravitational loading. However, even in failure cases, the

Table 2: Performance by Object, Difficulty, and Phase (Standard vs Optimised).

Object	**Difficulty**	**Phase**	**Std Iter.**	**Opt Iter.**	**Improve. (%)**	**Std Pos Err (mm)**	**Opt Pos Err (mm)**	**Std Time (s)**	**Opt Time (s)**
Box	Easy	Approach	53	30	43.4%	0.002	0.925	0.137	0.017
Box	Easy	Grasp	17	12	29.4%	0.001	0.956	0.044	0.006
Box	Easy	Lift	500	500	0.0%	15.202	15.792	1.227	0.277
Ball	Medium	Approach	500	500	0.0%	42.900	19.826	1.212	0.293
Ball	Medium	Grasp	61	46	24.6%	0.003	1.000	0.151	0.026
Ball	Medium	Lift	500	500	0.0%	63.271	64.672	1.257	0.282
Coke Can	Hard	Approach	500	500	0.0%	127.232	84.859	1.217	0.279
Coke Can	Hard	Grasp	500	500	0.0%	77.792	77.545	1.205	0.293
Coke Can	Hard	Lift	500	500	0.0%	163.125	169.189	1.242	0.284

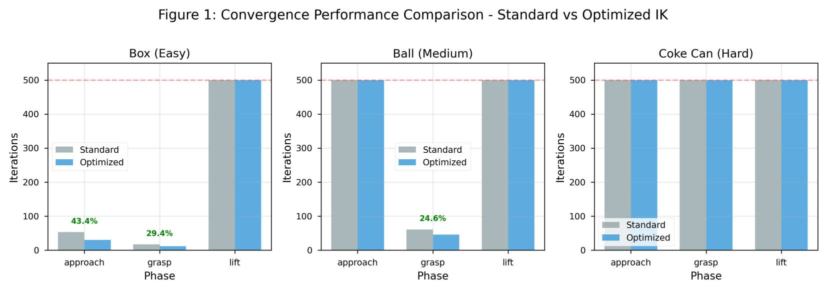

Fig. 3: Convergence Performance Comparison - Standard vs Optimised IK.

optimised approach demonstrated superior intermediate performance, achieving final position errors of 15.8mm versus 15.2mm for the box lift, while maintaining better orientation alignment.

4.1 Discussion

Experimental data reveal three critical insights about the LLM-IK system's performance characteristics. First, context-aware weight adaptation successfully optimised convergence for feasible tasks while maintaining accuracy within practical tolerances. The slight position error increases (0.00mm-0.92mm for box approach, 0.01mm-1.00mm for ball grasp) remain well below typical manipulation requirements. Second, the weight strategy demonstrated clear phase-dependent effectiveness. The approach phase benefited from aggressive position weighting (2.0) with reduced orientation emphasis (0.4-0.6), achieving the highest improvements. The grasp phase required balanced configurations (position: 2.0-3.0, orientation: 1.2-2.9) to maintain stability during contact. The fragile Coke can automatically received the highest precision weights (position: 3.0, orientation: 2.88), validating the LLM's contextual understanding. Third, the computational efficiency gains were substantial, with time reductions ranging from 64.7% to 87.6%. The box approach phase reduced from 137ms to 17ms (87.6% improvement), while maintaining solution quality. These timing improvements make the system viable for real-time control applications where rapid response is critical.

Failure analysis for challenging configurations revealed consistent patterns. The Coke can, positioned at the workspace boundary (0.5, 0.3), failed in all phases for both methods, confirming fundamental reachability limitations. However, the optimised approach consistently achieved lower intermediate errors (84.9mm vs 127.2mm for approach, 77.5mm vs 77.8mm for grasp), suggesting more efficient solution space exploration even when convergence is unattainable.

5 Conclusion

This work shows that large language models can bridge the gap between high-level task language and low-level robotic control by letting natural-language

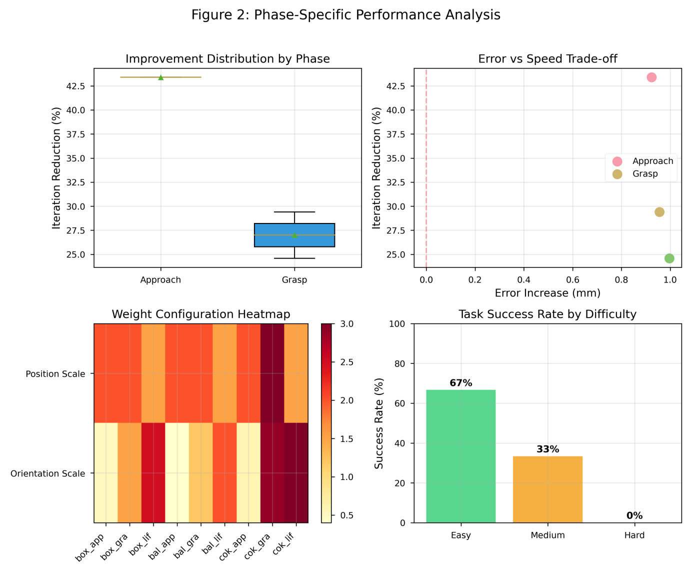

Fig. 4: Phase-Specific Performance Analysis with Error Distributions.

understanding directly shape inverse-kinematics optimisation. Our framework maps semantic cues, task context, object properties, manipulation requirements into numerical parameters, enabling behaviors such as automatically tightening precision for fragile objects, and thus making robot programming more intuitive. Experiments confirm performance gains in kinematically feasible settings, while failure analyses—particularly for vertical lifts and operations near workspace limits—highlight where parameter tuning alone is insufficient. These findings motivate hierarchical weight generation, adaptation strategies that react to convergence behavior, and hybrid designs that pair LLM-guided IK with trajectory planning and force control. Beyond optimisation, the approach points to broader human–robot interaction benefits, allowing non-experts to specify complex tasks

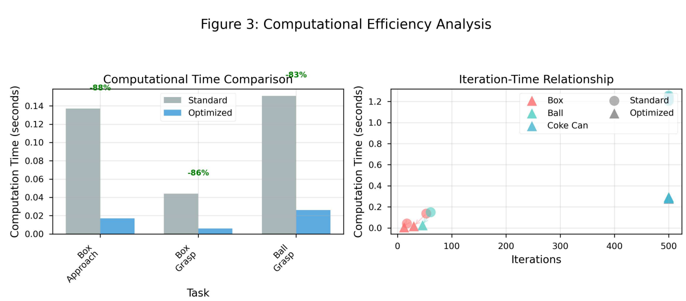

Fig. 5: Computational Time Reduction Across Different Task Complexities.

in natural language while the system manages the underlying mathematics. Overall, integrating semantic understanding with mathematical optimisation is a promising path toward more intelligent, deployable manipulation systems.

Acknowledgments. This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/Y000773/1.

References

1. Ames, B., Morgan, J., Konidaris, G.: Ikflow: Generating diverse inverse kinematics solutions. IEEE Robotics and Automation Letters **7**(3), 7177–7184 (2022)
2. Ding, Y., Zhang, X., Paxton, C., Zhang, S.: Task and motion planning with large language models for object rearrangement. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2086–2092. IEEE (2023)
3. Egeland, O.: Task-space tracking with redundant manipulators. IEEE Journal on Robotics and Automation **3**(5), 471–475 (2003)
4. Gao, J., Sarkar, B., Xia, F., Xiao, T., Wu, J., Ichter, B., Majumdar, A., Sadigh, D.: Physically grounded vision-language models for robotic manipulation. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 12462–12469. IEEE (2024)
5. Giamou, M., Marić, F., Rosen, D.M., Peretroukhin, V., Roy, N., Petrović, I., Kelly, J.: Convex iteration for distance-geometric inverse kinematics. IEEE Robotics and Automation Letters **7**(2), 1952–1959 (2022)
6. Huang, W., Wang, C., Zhang, R., Li, Y., Wu, J., Fei-Fei, L.: Voxposer: Composable 3d value maps for robotic manipulation with language models. arXiv preprint arXiv:2307.05973 (2023)
7. Kwon, T., Di Palo, N., Johns, E.: Language models as zero-shot trajectory generators. IEEE Robotics and Automation Letters **9**(7), 6728–6735 (2024)
8. Maciejewski, A.A., Klein, C.A.: Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. The international journal of robotics research **4**(3), 109–117 (1985)
9. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics: modelling, planning and control. Springer (2009)
10. Tang, C., Huang, D., Ge, W., Liu, W., Zhang, H.: Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping. IEEE Robotics and Automation Letters **8**(11), 7551–7558 (2023)
11. Yang, S., Liu, P., Pears, N.: Benchmarking of robot arm motion planning in cluttered environments. In: 2023 28th International Conference on Automation and Computing (ICAC). pp. 1–6. IEEE (2023)
12. Yang, S., Liu, P., Pears, N.: Enhancing learning from demonstration with dls-ik and promps. In: 2024 29th International Conference on Automation and Computing (ICAC). pp. 1–6. IEEE (2024)
13. Yang, Z., Ning, L., Wang, H., Jiang, T., Zhang, S., Cui, S., Jiang, H., Li, C., Wang, S., Wang, Z.: Text2reaction: Enabling reactive task planning using large language models. IEEE Robotics and Automation Letters **9**(5), 4003–4010 (2024)
14. Zhang, T., Mo, H.: Reinforcement learning for robot research: A comprehensive review and open issues. International Journal of Advanced Robotic Systems **18**(3), 17298814211007305 (2021)