

B^{*}: Efficient and Optimal Base Placement for Fixed-Base Manipulators

Zihang Zhao , Graduate Student Member, IEEE, Leyao Cui , Graduate Student Member, IEEE, Sirui Xie , Saiyao Zhang , Zhi Han , Lecheng Ruan , and Yixin Zhu 

Abstract—Proper base placement is crucial for task execution feasibility and performance of fixed-base manipulators, the dominant solution in robotic automation. Current methods rely on pre-computed kinematics databases generated through sampling to search for solutions. However, they face an inherent trade-off between solution optimality and computational efficiency when determining sampling resolution—a challenge that intensifies when considering long-horizon trajectories, self-collision avoidance, and task-specific requirements. To address these limitations, we present B^{*}, a novel optimization framework for determining the optimal base placement that unifies these multiple objectives without relying on pre-computed databases. B^{*} addresses this inherently non-convex problem via a two-layer hierarchical approach: The outer layer systematically manages terminal constraints through progressively tightening them, particularly the base mobility constraint, enabling feasible initialization and broad solution space exploration. Concurrently, the inner layer addresses the non-convexities of each outer-layer subproblem by sequential local linearization, effectively transforming the original problem into a tractable sequential linear program (SLP). Comprehensive evaluations across multiple robot platforms and task complexities demonstrate the effectiveness of B^{*}: it achieves solution optimality five orders of magnitude better than sampling-based approaches while

maintaining perfect success rates, all with reduced computational overhead. Operating directly in configuration space, B^{*} not only solves the base placement problem but also enables simultaneous path planning with customizable optimization criteria, making it a versatile framework for various robotic motion planning challenges. B^{*} serves as a crucial initialization tool for robotic applications, bridging the gap between theoretical motion planning and practical deployment where feasible trajectory existence is fundamental.

Index Terms—Base placement, sequential optimization.

I. INTRODUCTION

FIXED-BASE manipulators continue to dominate robotic automation due to their stability, precision, and payload capacity [1], [2], despite emerging advances in mobile and humanoid platforms [3], [4]. Supported by well-established commercial offerings, they remain central to both industrial applications and research advances. This progress is further accelerated by sophisticated algorithms, from global motion planning [5], [6], [7], [8], [9] to local motion generation [10], [11], [12], which autonomously generate joint trajectories for diverse tasks.

The effectiveness of these algorithms, however, depends critically on well-defined Cartesian trajectories within the manipulator’s workspace [13]. For fixed-base manipulators, this workspace is determined by the base placement, making proper base placement crucial to successful trajectory execution. The significance of base positioning extends beyond basic reachability considerations—it directly influences the manipulator’s ability to maintain continuous joint configurations between via points, avoid self-collision, and optimize performance throughout complete motion sequences.

This placement problem presents complex challenges due to the highly non-convex nature of robot workspaces, characterized by discontinuities, singularities, and regions of varying dexterity [14], [15]. As illustrated in Fig. 1, optimal base placement is highly sensitive to both the robot’s kinematic structure and specific task requirements—a configuration that minimizes trajectory length for one task may severely limit the robot’s capabilities for another.

Current base placement methods face a fundamental limitation: they rely on pre-computed kinematic databases generated through discretized sampling of the robot’s workspace. These databases—storing either direct mappings between end-effector and base positions or collections of inverse kinematics (IK) solutions (see Section II)—create an unavoidable trade-off. High-resolution sampling produces better solutions but severely impacts computational efficiency; low-resolution sampling

Received 16 April 2025; accepted 22 August 2025. Date of publication 2 September 2025; date of current version 8 September 2025. This article was recommended for publication by Associate Editor H. Chen and Editor C.-B. Yan upon evaluation of the reviewers’ comments. This work was supported in part by the National Science and Technology Major Project under Grant 2022ZD0114900, in part by the National Natural Science Foundation of China under Grant 62376031, in part by Beijing Nova Program, in part by the State Key Lab of General AI at Peking University, in part by the PKU-BingJi Joint Laboratory for Artificial Intelligence, and in part by the National Comprehensive Experimental Base for Governance of Intelligent Society, Wuhan East Lake High-Tech Development Zone. (Zihang Zhao, Leyao Cui, and Sirui Xie contributed equally to this work.) (Corresponding author: Yixin Zhu.)

Zihang Zhao is with the Institute for Artificial Intelligence, Peking University, Beijing 100871, China, also with the School of Psychological and Cognitive Sciences Peking University, Beijing 100871, China, also with the Beijing Key Laboratory of Behavior and Mental Health, Peking University, Beijing 100871, China, and also with the LeapZenith AI Research Shanghai 201707, China.

Leyao Cui and Saiyao Zhang are with the University of Chinese Academy of Sciences, Beijing 100049, China, and also with the the School of Psychological and Cognitive Sciences, Peking University, Beijing 100871, China.

Sirui Xie and Yixin Zhu are with the Institute for Artificial Intelligence, Peking University, Beijing 100871, China, and with the School of Psychological and Cognitive Sciences, Peking University, Beijing 100871, China, and also with the Beijing Key Laboratory of Behavior and Mental Health, Peking University, Beijing 100871, China (e-mail: yixin.zhu@pku.edu.cn).

Zhi Han is with the University of Chinese Academy of Sciences, Beijing 100049, China.

Lecheng Ruan is with the College of Engineering, Peking University, Beijing 100871, China.

Data is available online at <https://bstar-planning.github.io>

This article has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2025.3604741>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3604741

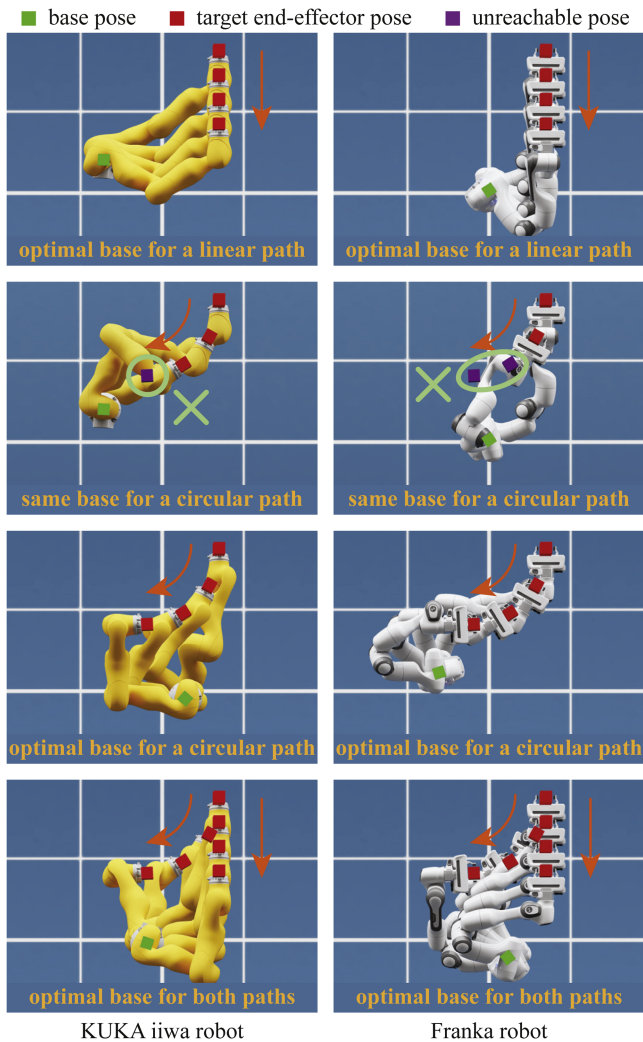


Fig. 1. Varied optimal base placement across robotic platforms and task requirements. To demonstrate this, we present several examples using two widely adopted manipulators—the KUKA iiwa (left) and Franka (right) robots. Our examples encompass four key scenarios: (i) optimal base placement for a linear path, (ii) a circular path using the base placement optimized for the linear path (showing unreachable poses), (iii) optimal base placement for the circular path, and (iv) optimal base placement that accommodates both paths. This comparison illustrates the necessity of an efficient method to generate the optimal base placement for each scenario.

offers speed but compromises solution quality and feasibility. This limitation is intrinsic to the sampling-based paradigm.

Furthermore, as task complexity grows—such as requiring more via points in a path, incorporating self-collision checks, or meeting additional task-specific constraints—the scalability of these methods is further compromised. This results in a significant challenge: balancing precision and computational speed becomes increasingly difficult regardless of search algorithm sophistication. These limitations motivate the development of a new framework that can address these shortcomings and provide a more efficient and scalable solution for base placement optimization.

We present B^* , an optimization framework that overcomes these limitations by unifying path-wide feasibility, self-collision avoidance, and task-specific requirements into a continuous optimization formulation. To tackle this highly complex optimization, B^* employs two-layer hierarchical strategies:

- Outer layer with progressive constraint tightening: The terminal optimization problem is temporarily reformulated into a more tractable form through systematic constraint relaxation, particularly the base mobility constraint. Initially, the fixed-base constraint is relaxed by introducing three additional (DoFs) to model the base, akin to a mobile manipulator. This relaxation ensures feasible initialization through guaranteed IK solutions while enabling the exploration of a broader solution space. Constraints are then gradually enforced, using the explored solutions as effective initial conditions for subsequent optimization.
- Inner layer with sequential linearization: To address the inherent non-convexities within each sub-problem generated by the outer layer, an iterative process of local linearization is employed. This approach approximates the non-convex problem by constructing a sequence of locally linear sub-problems that are computationally tractable. Combined with the progressive constraint-tightening strategy of the outer layer, this hierarchical framework effectively transforms the originally highly non-convex optimization problem into a tractable sequential linear program (SLP).

We quantitatively validate B^* through comprehensive testing in four manipulator types and 2400 randomly generated paths of varying complexity levels. Operating directly in configuration space without relying on pre-computed databases, B^* achieves: (i) perfect success rates across all test cases, even for complex trajectories with multiple via-points and self-collision constraints; (ii) solution optimality five orders of magnitude better than sampling-based methods, particularly in challenging scenarios requiring precise positioning; and (iii) reduced computational overhead despite higher precision.

These results establish B^* as a versatile framework for optimal base placement, effectively bridging the gap between theoretical motion planning and practical deployment. By operating directly in configuration space, our approach not only solves the base placement problem but also enables simultaneous path planning with customizable optimization criteria. This capability opens new possibilities for unified trajectory and base optimization in robotic manipulation, particularly in applications requiring high precision and efficiency.

Our contributions are twofold:

- A novel configuration-space optimization framework that determines optimal base placement without precomputed databases, ensuring solution optimality while maintaining high success rates and computational efficiency.
- Comprehensive evaluation across multiple manipulators (6-DoF to 7-DoF) and varying path complexities, demonstrating the framework’s versatility and robustness in diverse scenarios.

The remainder of this paper is organized as follows: Section II examines current base placement methods and their limitations in detail. Section III presents the theoretical framework of B^* , including our novel optimization approach. Section IV provides comprehensive experimental validation across various robotic platforms and task complexities. Section V analyzes key performance characteristics and discusses future research directions. Section VI summarizes our findings and contributions.

II. RELATED WORK

A proper base placement is fundamental for fixed-base manipulators to execute tasks successfully. Research in this field has evolved through several stages of increasing complexity,

starting with simple reachability verification and progressing to sophisticated trajectory-aware optimization.

Initial studies concentrated on single-point position reachability [16], leveraging the concept of reachability maps (RMs). These maps were precomputed by systematically sampling the kinematic relationships between the base and end-effector, expressed through homogeneous transformation matrices [14], [17], [18]. While this foundational approach proved effective for basic positioning, researchers recognized the need for optimization criteria, leading to the integration of manipulability metrics for base placement selection [19], [20]. The limitations of static reachability maps in dynamic environments prompted the development of adaptive reachability maps (RMs) [21]. Building upon these advances, researchers expanded the framework to address more sophisticated requirements, incorporating base orientation considerations [17], [22] and multi-point reachability analysis [23], [24]. Recent advances in learning-based methodologies have enabled neural networks to function as efficient approximators for complex kinematic mappings, substantially enhancing computational performance for single-point reachability analysis [25].

Despite significant advances in reachability mapping, a fundamental limitation remained: while these methods could verify point-wise reachability, they could not guarantee the feasibility of continuous trajectories between reachable points. This limitation became particularly apparent in practical applications where two points might be individually reachable yet connected by no viable path due to kinematic constraints or environmental obstacles. This crucial insight led to a paradigm shift in representing kinematic relationships, replacing the traditional homogeneous transformation matrix approach with an IK-based formulation. This transition enabled a more comprehensive evaluation of both point reachability and path feasibility within the robot's configuration space [26], [27].

However, even IK-based approaches remain constrained by their reliance on pre-computed sampling-based databases and search algorithms for optimal base placement. This dependence on discrete sampling creates an inherent trade-off: higher sampling resolution provides better precision but demands greater computational resources and memory capacity, while lower resolution sacrifices precision for efficiency.

In contrast to existing approaches, this paper introduces B^* , a novel optimization-based framework for determining optimal base placement. Unlike traditional methods that depend on pre-computed databases, B^* operates directly within the robot's configuration space. Moreover, compared to learning-based methods, it offers broader applicability to manipulators with arbitrary configurations. Through its unified optimization formulation, B^* achieves significant improvements in computational efficiency while maintaining high success rates and solution optimality across diverse task scenarios.

III. THE B^* METHOD

We present B^* , our optimization framework for fixed-base manipulator placement. We first formalize the mathematical foundations of optimal base placement (Section III-A), then introduce a two-layer optimization strategy that decomposes this non-convex problem into tractable sub-problems (Section III-B).

A. Problem Formulation

We formulate the base placement problem for fixed-base manipulators, intended to be mounted on a flat surface (whether horizontal or slanted), as an optimization problem. For a robotic manipulator mounted on a fixed base, we seek to compute the optimal base placement $\mathbf{q}^b = [x^b, y^b, \theta^b]^T \in SE(2)$ ($SE(k)$ denotes special Euclidean group in k dimensions) that enables successful task execution. The manipulation task comprises an ordered sequence of desired end-effector poses in $SE(3)$:

$$\mathbf{x}_{1:t} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t] \in \mathbb{R}^{t \times 6}, \quad (1)$$

where each $\mathbf{x}_i \in \mathbb{R}^6$ encodes position and orientation parameters at time step i . These task poses can be provided directly by a human operator or generated via interpolation methods for smooth and continuous motion between successive steps.

For any candidate base placement \mathbf{q}^b , the manipulator must achieve feasible joint configurations $\mathbf{q}_{1:t}^m = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_t] \in \mathbb{R}^{t \times n}$ to reach each target pose \mathbf{x}_i , where n is degree of freedom (DoF) of the manipulator. The optimization problem requires finding both an optimal base placement and a sequence of valid joint configurations that satisfy the manipulator's kinematic constraints. This leads to the following formulation:

$$\begin{aligned} & \text{minimize} && f(\mathbf{q}^b, \mathbf{q}_{1:t}^m) \\ & \text{subject to} && g_i(\mathbf{q}^b, \mathbf{q}_{1:t}^m) = 0, \quad i = 1, 2, \dots, n_{\text{eq}} \\ & && h_i(\mathbf{q}^b, \mathbf{q}_{1:t}^m) \leq 0, \quad i = 1, 2, \dots, n_{\text{ineq}} \end{aligned} \quad (2)$$

where f is a scalar objective function, and g_i and h_i represent equality and inequality constraints, respectively. We detail these objectives and constraints below.

Objective Function: The primary requirement in base placement optimization is satisfying reachability constraints while maintaining flexibility in objective function formulation. We present two common formulations. The simplest is the feasibility-only approach:

$$f(\mathbf{q}^b, \mathbf{q}_{1:t}^m) = 1, \quad (3)$$

which transforms the optimization into a feasibility test, seeking any valid solution that satisfies all constraints.

For tasks requiring efficient execution, we can further minimize path length:

$$f(\mathbf{q}^b, \mathbf{q}_{1:t}^m) = \sum_{i=1}^{t-1} \|\mathbf{q}_{i+1}^m - \mathbf{q}_i^m\|. \quad (4)$$

We utilize non-differentiable ℓ_1 penalties, although smooth ℓ_2 penalties produce comparable results [9]. The inclusion of the ℓ_1 term elegantly reformulates the problem into an equivalent linear programming task with linear constraints. By introducing auxiliary slack variables z_i with constraints $z_i \geq x_i$, $z_i \geq -x_i$ and $z_i \geq 0$, each absolute value term $|x_i|$ is transformed into a linear program equivalent. This choice also ensures compatibility with a wide range of optimization solvers [28].

Equality Constraints: The primary equality constraints ensure the end-effector reaches all targeted poses. Let $\psi(\mathbf{q}^b, \mathbf{q}_i^m) : SE(2) \times \mathbb{R}^n \rightarrow \mathbb{R}^6$ denote the forward kinematics mapping from the base and joint configuration to end-effector pose. We enforce exact pose achievement through:

$$g_1 : \psi(\mathbf{q}^b, \mathbf{q}_i^m) = \mathbf{x}_i, \quad i = 1, 2, \dots, t. \quad (5)$$

Inequality Constraints: In addition to achieving each pose, the manipulator must respect several physical and safety constraints. First, base placement $\mathbf{q}^b \in SE(2)$ must remain within allowable position and orientation ranges determined by working conditions. Let \mathbf{q}^{b^m} and \mathbf{q}^{b^M} denote the lower and upper bounds, respectively:

$$h_1 : \mathbf{q}^{b^m} \preceq \mathbf{q}^b \preceq \mathbf{q}^{b^M}, \quad (6)$$

where \preceq represents an element-wise “less than or equal to” relationship between vectors.

Next, each joint configuration \mathbf{q}_i^m must remain within its feasible range, bounded by mechanical joint limits. With lower and upper bounds denoted as \mathbf{q}^{m^m} and \mathbf{q}^{m^M} , we impose:

$$h_2 : \mathbf{q}^{m^m} \preceq \mathbf{q}_i^m \preceq \mathbf{q}^{m^M}, \quad i = 1, 2, \dots, t. \quad (7)$$

Finally, collision-free motion is enforced through:

$$h_3 : \text{sd}(\mathbf{q}^b, \mathbf{q}_i^m) \geq 0, \quad i = 1, 2, \dots, t, \quad (8)$$

where the signed distance $\text{sd}(\cdot)$ as defined in [29] represents the minimal translation distance required to alter the spatial relationship between objects. This formulation addresses self-collision avoidance and can be extended to obstacle avoidance by incorporating obstacle geometry information.

B. Two-Layer Optimization of B*

The optimization problem presented above is highly non-convex and challenging to solve directly. We address this complexity through a hierarchical two-layer optimization structure. Specifically, the inner layer manages local non-convexity through iterative convex approximations of the original problem. The outer layer employs a novel approach by initially treating the base as three additional DoFs, analogous to a mobile manipulator. This formulation ensures solution feasibility while enabling comprehensive exploration of the solution space. By progressively increasing penalties on base movement, the algorithm converges to a fixed base configuration, effectively balancing thorough solution space exploration with fixed-base constraint satisfaction.

Outer Layer—Base Relaxation: The complexity of optimization stems largely from finding an initially feasible solution under fixed base constraints. We address this through a strategic relaxation approach.

We first solve a relaxed problem by treating the fixed base as mobile, transforming the timestep-invariant base configuration \mathbf{q}^b into a time-varying sequence $\mathbf{q}_{1:t}^b$, effectively introducing three additional DoF per timestep. To ensure convergence to a fixed base configuration, we introduce a base movement penalty with an iteration-dependent coefficient $\mu(j)$ that increases with outer loop iteration j .

The modified objective function $f'(\mathbf{q}_{1:t}^b, \mathbf{q}_{1:t}^m; \mu(j))$ becomes:

$$f'(\mathbf{q}_{1:t}^b, \mathbf{q}_{1:t}^m; \mu(j)) = f(\mathbf{q}_{1:t}^b, \mathbf{q}_{1:t}^m) + \mu(j) \sum_{i=1}^t \|\mathbf{q}_i^b - \bar{\mathbf{q}}^b\|_1, \quad (9)$$

where $\bar{\mathbf{q}}^b$ is the arithmetic mean of base poses across all steps. We employ ℓ_1 penalties over ℓ_2 as they more effectively drive the second term to zero with matching coefficients $\mu(j)$, ensuring convergence to a stable configuration. The coefficient $\mu(j)$ progressively increases across outer loop iterations, gradually enforcing the fixed base constraint.

All constraints involving the fixed base \mathbf{q}^b must be reformulated for the relaxed problem. Each instance of \mathbf{q}^b in the original constraints is replaced with its time-indexed counterpart \mathbf{q}_i^b at the corresponding timestep i . For example, the base limits constraint becomes:

$$h_1 : \mathbf{q}^{b^m} \preceq \mathbf{q}_i^b \preceq \mathbf{q}^{b^M}, \quad i = 1, 2, \dots, t. \quad (10)$$

These constraints are subsequently transformed into penalty terms to accommodate potentially infeasible initial conditions, ensuring the algorithm can start from arbitrary initial states while maintaining numerical stability.

Inner Layer—Linearization: Even after relaxation, the problem’s inherent non-convexity poses significant challenges for optimization. Drawing inspiration from sequential programming [30], we address this through iterative convex approximations within trust regions.

Given a non-convex function $\phi(x)$, we construct its convex approximation $\phi_c(x)$ through first-order Taylor expansion around the current point x_0 :

$$\phi_c(x) = \phi(x_0) + \dot{\phi}(x_0)(x - x_0), \quad (11)$$

where $\dot{\phi}(x_0)$ represents the first-order derivative at x_0 . The trust region size adapts based on approximation quality—expanding when the approximation performs well and contracting when it poorly represents the original function [30].

This two-layer framework combines base relaxation with local linear approximation to transform the non-convex optimization into a series of tractable linear programs (LPs). The outer iterations broadly explore the solution space for suitable initialization, while inner iterations efficiently traverse the current non-convex space to determine a high-quality solution specifying fixed base placement and joint configurations.

IV. EVALUATIONS

We evaluate B* through comprehensive simulation studies assessing its effectiveness in determining optimal fixed-base manipulator poses across diverse tasks. Our evaluation comprises two components: First, Section IV-B provides qualitative insights into B*’s operational principles through representative examples that illustrate its behavior across different scenarios. Second, Section IV-C presents detailed quantitative comparisons against baseline methods using extensive randomly generated test cases of varying complexity.

A. Implementation Details

B* optimizes for minimal-length task completion paths as formalized in Eq. (4). The process initializes using an IK solution incorporating 3 additional base DoFs. For the first point x_1 , random initialization ensures comprehensive solution space exploration. Subsequent points are initialized using the preceding point’s IK solution to promote trajectory continuity.

The optimization problem is implemented using the COPT solver [31]. Given the established correlation between initialization quality and sequential programming performance [9], B* employs up to 10 retry attempts with different initializations when needed.

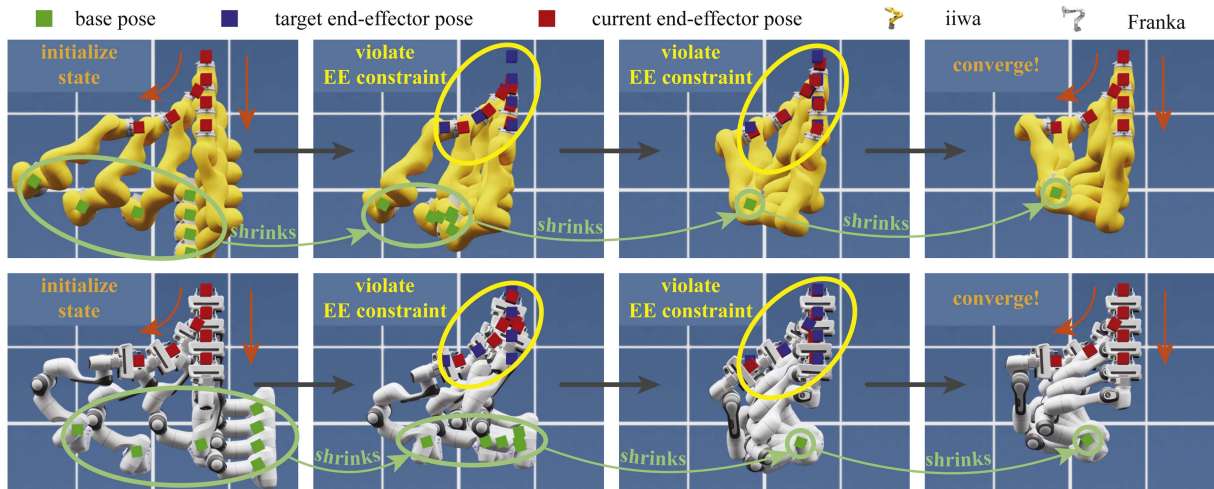


Fig. 2. Visualization of B^* 's optimization process over multiple iterations. Each row shows a different test scenario, demonstrating the method on both the KUKA iiwa and Franka robots performing compound trajectories. Starting from an initial configuration with relaxed base constraints (leftmost column), the optimization progressively constrains the base position through increasing penalty coefficients (middle columns), until converging to a single fixed base placement (rightmost column). During intermediate steps, some constraints, such as the end-effector (EE) constraint (Eq. (5)), may temporarily violate the target states as the algorithm balances between base convergence and task constraints. More examples and full optimization sequences are available in the Supplementary Video S1.

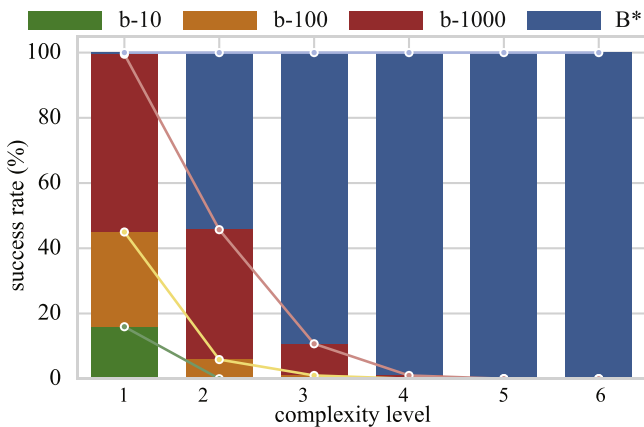


Fig. 3. Success rate comparison between baseline methods and B^* . While baseline methods with more IK samples show higher success rates at low complexity levels, their performance deteriorates sharply as the complexity level l increases. With just 10 samples (b-10), the baseline struggles even at level 1, while 1000 samples (b-1000) maintain moderate success through level 2 before failing. In contrast, B^* maintains 100% success rate across all complexity levels l (see Eq. (14)).

B. Qualitative Results

We evaluated B^* through experiments on a representative trajectory type, incorporating ground collision avoidance constraints. The type comprises complex compound paths that combine linear and circular segments, requiring 64 uniformly spaced end-effector poses. To demonstrate generalizability, we tested B^* on two industrial manipulators with distinct kinematic structures and workspace characteristics: the Franka Emika Panda and KUKA LBR iiwa.

Fig. 2 captures the optimization progression through four key iterations for each task category. To maintain visual clarity, we show selected configurations—7 poses for compound paths. Starting from the leftmost column with initial IK solutions using relaxed base poses (Section IV-A), the optimization proceeds through intermediate stages where increasing penalty

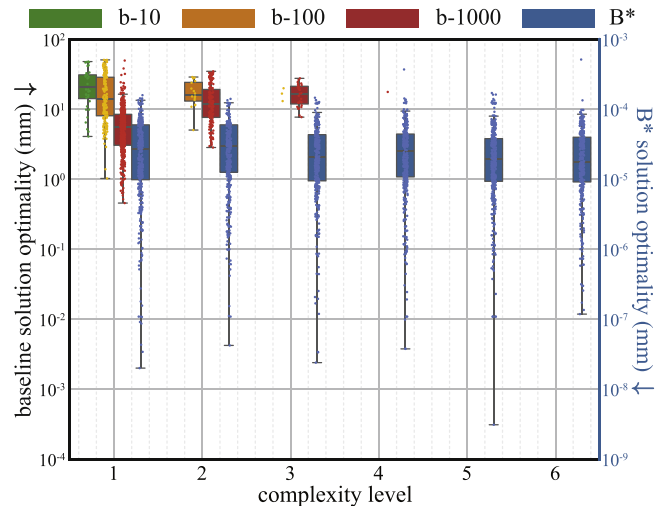


Fig. 4. Solution optimality (ATE) comparison between baseline methods (left axis) and B^* (right axis) across complexity levels. Baseline methods, even with increased IK samples, achieve only millimeter-level precision due to sampling resolution limits. B^* achieves five orders of magnitude better precision by operating in continuous configuration space. Box plots show error distributions: boxes indicate interquartile range (IQR), center lines represent medians, and whiskers extend to $1.5 \times \text{IQR}$ beyond quartiles.

coefficients drive convergence toward a fixed base. During this process, the robot may temporarily deviate from target end-effector poses due to our soft constraint formulation. The rightmost column shows the final solution: a single, collision-free base placement that enables the robot to reach all desired end-effector poses successfully.

C. Quantitative Results

We conducted extensive quantitative evaluations comparing B^* against baseline approaches across varying task complexities and robot platforms. Our evaluation framework consists of four key components: (i) a systematic test data generation process

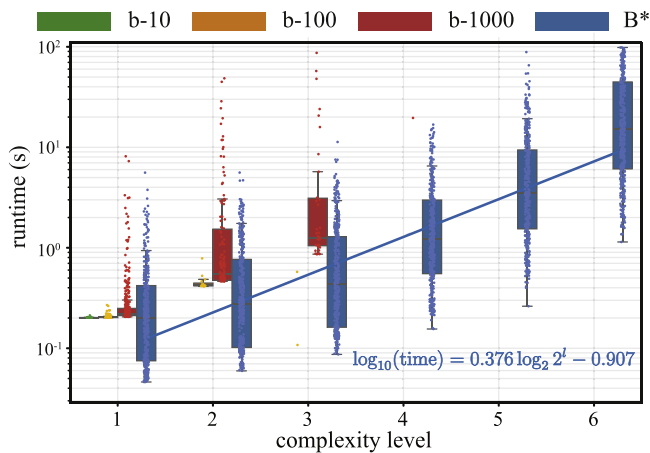


Fig. 5. Runtime comparison between baseline methods and B* across complexity levels. While baseline methods show increased runtime with higher IK numbers, B* demonstrates superior efficiency with linear scaling (fit shown: $\log_{10}(\text{time}) = 0.376 \log_2 2^l - 0.907$). Box plots in the same format as Fig. 4 illustrate runtime distributions, with outliers above 100 s excluded for visualization clarity.

covering diverse workspace configurations, (ii) implementation of established baseline methods using pre-computed IK databases, (iii) comprehensive performance metrics measuring success rate, solution optimality, and computational efficiency, and (iv) controlled testing environment for fair comparison.

Large-Scale Test Data Generation: To ensure rigorous evaluation, we developed a systematic approach for generating test datasets that comprehensively span each manipulator’s workspace across varying complexity levels. Our data generation process consists of three key components.

The first component focuses on initial configuration sampling. We generate the start pose \mathbf{x}_1 by uniformly sampling the joint space to achieve comprehensive coverage:

$$\mathbf{x}_1 = \psi(\mathbf{q}^b, \mathbf{q}_1^m), \quad \mathbf{q}_1^m \sim \mathcal{U}(\mathbf{q}^{mM}, \mathbf{q}^{mM}). \quad (12)$$

\mathbf{q}^b merely accounts for a planar transformation in $SE(2)$ without affecting workspace characteristics. Therefore, we set \mathbf{q}^b to zero vector without losing generality.

The second component generates workspace-valid pose sequences. To maintain kinematic feasibility, we construct subsequent poses through incremental perturbations in joint space. Each pose builds upon its predecessor through small displacements ξ_j :

$$\mathbf{x}_k = \psi\left(\mathbf{q}^b, \mathbf{q}_1^m + \sum_{j=1}^{k-1} \xi_j\right), \quad \xi_j \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad (13)$$

where $\boldsymbol{\mu} \in \mathbb{R}^n$ has all entries equal to 0.01 rad and $\boldsymbol{\Sigma} = \text{diag}(0.005^2) \in \mathbb{R}^{n \times n}$. These parameters were empirically selected for effective workspace exploration. We validate trajectory feasibility using NVIDIA Isaac Sim’s Luna module.

The third component implements multi-level complexity scaling. For systematic evaluation across different scales, we generate pose sequences of varying lengths based on complexity level l :

$$t = 2^l, \quad l \in \mathbb{N}^+. \quad (14)$$

We applied this framework to four widely-used manipulators: Franka Emika Panda, Kinova Gen3, KUKA LBR iiwa, and Universal Robots UR10, covering both 6-DoF and 7-DoF manipulators. For each robot-complexity combination ($l = 1, \dots, 6$), we generated 100 distinct sequences, yielding 2400 comprehensive test cases.

Baseline Methods: We evaluated B* against search-based approaches utilizing pre-computed IK solution databases. These baseline methods treat the base position’s three planar DoF as additional joints in the IK formulation. For each target point in a sequence, we employ CuRobo [32] to generate γ collision-free IK solutions comprising both manipulator joint configurations and base positions. Using this IK dataset, the baseline methods search for feasible joint paths allowing point-to-point movement while maintaining the base position within a confined space. We implement this search using a breadth-first algorithm to ensure global optimality for fair comparison with the optimization-based method B* in solution optimality. To investigate the impact of solution sampling density, we evaluated three baseline configurations: b-10, b-100, and b-1000, corresponding to $\gamma \in \{10, 100, 1000\}$ IK solutions per target point.

Metrics: To comprehensively evaluate performance across methods, we employ three complementary metrics that assess different aspects of solution quality and efficiency:

- *Success rate:* As our primary effectiveness metric, success rate evaluates algorithmic reliability using well-defined binary criteria. For B*, success is determined by optimization convergence to a valid solution. Baseline methods must satisfy two conditions for success: maintaining base position variations within specified tolerances (0.01 m for translations and 0.05 rad for rotation), and completing the search within a 10 min time limit.
- *Solution optimality:* To rigorously assess solution optimality, we quantify base path deviation using the absolute trajectory error (ATE) [33]. This metric computes the root-mean-squared error of Lie algebra components across all pairwise base placement combinations.
- *Runtime:* Computational scalability serves as a crucial criterion for practical deployment in complex scenarios. We analyze time efficiency by measuring total computation times across varying complexity levels. For B*, this encompasses the complete runtime of all optimization attempts. For baseline methods, we include both the time required for IK solution generation and the subsequent optimal base placement search process.

Test Environment: The implementation of all methods uses Python as the primary programming language, with all computations performed exclusively on the CPU using an AMD Ryzen 9 5950X processor supported by 64 GB of RAM.

Results: Our experimental evaluation reveals clear performance patterns across different methods. As shown in Figs. 3 and 4, baseline approaches are effective for simple scenarios but degrade significantly with increasing sequence length. While using more IK solutions improves both success rate and solution optimality, this comes at a substantial computational cost. This trade-off is evident in Fig. 5, where higher numbers of IK solutions result in longer runtime due to increased search complexity.

B*, in contrast, achieves optimal performance across all robots and complexity levels, maintaining consistent success rates as demonstrated in Fig. 3. Through direct configuration space optimization, B* overcomes sampling resolution limitations, achieving five orders of magnitude improvement in solution

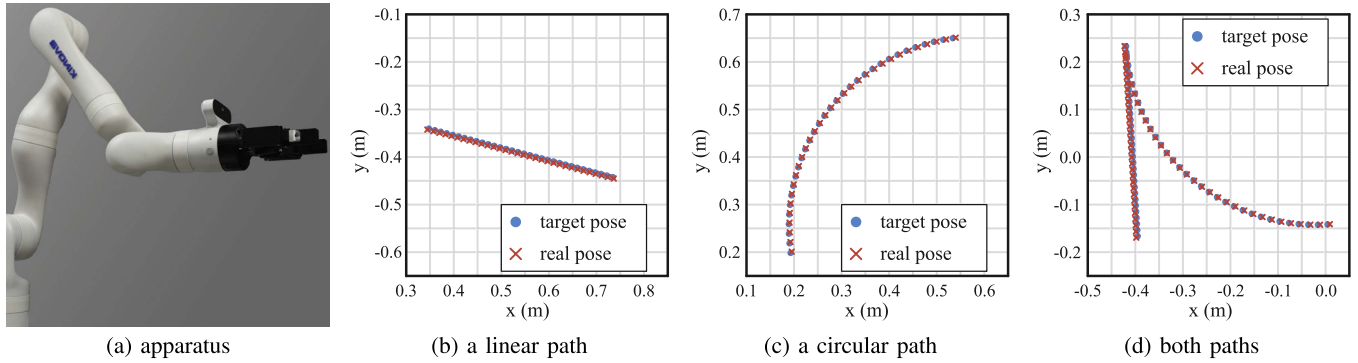


Fig. 6. Real-world validation. (a) The effectiveness of B^* is demonstrated on a 7-DoF Kinova robot in a real-world setting. With accurate kinematic and collision models, the joint path generated by B^* is successfully executed, achieving the target pose. This is illustrated through the execution of a linear path (b), a circular path (c), and a compound path (d). The complete execution process is available at the Supplementary Video S2.

TABLE I
SUB-PROCESS RUNTIME OF B^* IN MILLISECONDS

Level	Initialization	Inner Layer	Outer Layer
1	6.255 ± 3.561	20.516 ± 3.898	79.350 ± 115.193
2	9.216 ± 5.130	25.491 ± 4.870	132.652 ± 188.241
3	13.665 ± 4.725	35.049 ± 6.178	219.044 ± 304.350
4	24.507 ± 8.162	60.584 ± 10.931	451.903 ± 564.226
5	46.386 ± 14.869	114.614 ± 21.725	1032.060 ± 1191.717
6	96.314 ± 42.203	269.816 ± 65.030	2838.154 ± 3123.153

optimality (Fig. 4). These substantial gains come with improved computational efficiency— B^* maintains faster completion times than baseline methods while scaling linearly with scenario complexity (Fig. 5) with detailed sub-process run time in Table I.

V. DISCUSSIONS AND FUTURE WORK

Our study highlights B^* 's effectiveness in efficiently determining optimal base placements for manipulation tasks. Traditional approaches relying on pre-computed kinematic databases show a clear dependence on the number of IK solutions (Figs. 3 and 4). While infinite sampling could theoretically achieve perfect results, this is impractical due to significant computational and storage overhead, especially in the breadth-first search used in baseline methods to guarantee optimality. Although alternative search methods like depth-first or heuristic search can reduce resource demands, they compromise optimality. More importantly, regardless of the search method used, these approaches fail to address the fundamental issue of sampling resolution inherent in sampling-based methods.

B^* tackles these challenges with a fundamentally different approach: direct optimization in the robot's configuration space. This method achieves a five-order-of-magnitude improvement in solution optimality over baseline methods while maintaining a perfect success rate and linear computational scalability. Beyond base placement optimization, the B^* also serves as a path planner with customizable cost functions, enabling the optimization of additional criteria, such as minimal path length, as demonstrated in this study.

B^* incorporates constraints such as joint limits, ground collisions, and self-collisions into the optimization. With accurate kinematic and collision models, its results are transferable to real-world applications. To validate this, we applied B^* to a

TABLE II
OF INITIALIZATION RETRIES ACROSS COMPLEXITY LEVELS

level	1	2	3	4	5	6
#	1.1 ± 0.3	1.1 ± 0.3	1.1 ± 0.4	1.3 ± 0.8	1.4 ± 1.0	2.0 ± 2.0

physical 7-DoF Kinova robot with target poses shown in Fig. 1. For convenience and without compromising rigor, we transform the world coordinate frame to align with the base poses calculated by B^* in each scenario. The generated joint configurations were directly executed on the physical robot, with end-effector poses verified through the Kinova official interface. The results, presented in Fig. 6, confirm that all configurations are executable and achieve the desired target poses. The complete execution process and detailed application scenarios are provided in the Supplementary Video S2 and Supplementary Video S3, respectively.

However, the current deployment of the B^* exhibits limitations in initialization, which is critical for optimization convergence in highly non-convex problems. Even with advanced convexification techniques, initialization remains an open question in the broader optimization field [15]. In this study, we demonstrate the effectiveness of the B^* using random initialization, achieving notable success even with a 7-DoF manipulator and 64 target end-effector poses. However, task complexity increases the number of required initialization retries, as shown in Table II. This escalation arises from the growing number of optimization variables, which impacts performance in two ways: it prolongs the time required to solve individual optimization instances and increases the total solution time due to additional retries, as illustrated in Fig. 5. In more severe scenarios, like highly constrained environments, the random initialization could lead to failure.

Given these challenges, developing a more sophisticated initialization method for B^* emerges as a promising direction for future work. Moreover, while this study focuses primarily on reachability and collision constraints, future work will explore incorporating additional real-world constraints, such as forces exerted at the end-effector, mounting limitations, and dynamic obstacles in the environment. These extensions will enhance B^* 's applicability to real-world scenarios.

VI. CONCLUSION

In this paper, we present B*, a novel optimization framework that improves both computational efficiency and solution optimality for base placement determination in fixed-base manipulators, outperforming traditional sampling-based methods. B* employs a two-layer optimization strategy: an outer layer for relaxed base exploration and an inner layer that addresses non-convex constraints using iterative local linearization. This approach effectively tackles challenges associated with long-horizon tasks, task-specific requirements, and self-collision avoidance. Furthermore, by operating directly in the configuration space, B* eliminates reliance on precomputed kinematic databases, providing a scalable and versatile solution for a wide range of robotic applications.

ACKNOWLEDGMENT

The author would like to thank Dr. Chi Zhang (BIGAI), Yuyang Li (PKU), and Youqian Peng (PKU) for useful discussions.

REFERENCES

- [1] M. I. A. Ferreira and S. R. Fletcher, *The 21st Century Industrial Robot: When Tools Become Collaborators*. Berlin, Germany: Springer, 2022.
- [2] Z. Zhao et al., "Embedding high-resolution touch across robotic hands enables adaptive human-like grasping," *Nature Mach. Intell.*, vol. 7, no. 6, pp. 889–900, 2025.
- [3] P. Arm et al., "Scientific exploration of challenging planetary analog environments with a team of legged robots," *Sci. Robot.*, vol. 8, no. 80, 2023, Art. no. ead e9548.
- [4] S. Dafarra et al., "iCub3 avatar system: Enabling remote fully immersive embodiment of humanoid robots," *Sci. Robot.*, vol. 9, no. 86, 2024, Art. no. eadh 3834.
- [5] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*. New York, NY, USA: ACM, 2022, pp. 287–290.
- [6] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. TSSC-4, no. 2, pp. 100–107, Jul. 1968.
- [7] I. A. Sucas, M. Moll, and L. E. Kavraki, "The open motion planning library," *RA-M*, vol. 19, no. 4, pp. 72–82, 2012.
- [8] M. Zucker et al., "Chomp: Covariant hamiltonian optimization for motion planning," *The Int. J. Robot. Res.*, vol. 32, no. 9/10, pp. 1164–1193, 2013.
- [9] J. Schulman et al., "Motion planning with sequential convex optimization and convex collision checking," *The Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.
- [10] Y. Karayiannidis, C. Smith, F. E. V. Barrientos, P. Ögren, and D. Kragic, "An adaptive control approach for opening doors and drawers under uncertainties," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 161–175, Feb. 2016.
- [11] A. O'Neill et al., "Open X-embodiment: Robotic learning datasets and Rt-X models: Open X-embodiment collaboration⁰," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 6892–6903.
- [12] Z. Zhao et al., "Tac-Man: Tactile-informed prior-free manipulation of articulated objects," *IEEE Trans. Robot.*, vol. 41, pp. 538–557, 2025.
- [13] S. R. Buss and J. -S. Kim, "Selectively damped least squares for inverse kinematics," *J. Graph. Tools*, vol. 10, no. 3, pp. 37–49, 2005.
- [14] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing Robot Workspace Structure: Representing Robot Capabilities," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 3229–3236.
- [15] S. Shetty, T. Lombrone, T. Loew, and S. Calinon, "Tensor train for global optimization problems in robotics," *Int. J. Robot. Res.*, vol. 43, no. 6, pp. 811–839, 2024.
- [16] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 1970–1975.
- [17] T. Birr, C. Pohl, and T. Asfour, "Oriented Surface Reachability Maps for Robot Placement," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 3357–3363.
- [18] M. Rudorfer, "RM4D: A combined reachability and inverse reachability map for common 6-/7-axis robot arms by dimensionality reduction to 4D," 2024, arXiv:2410.06968.
- [19] F. Burget and M. Bennewitz, "Stance selection for humanoid grasping tasks by inverse reachability maps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 5669–5674.
- [20] F. Reister, M. Grotz, and T. Asfour, "Combining navigation and manipulation costs for time-efficient robot placement in mobile manipulation tasks," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9913–9920, Oct. 2022.
- [21] Y. Yang, V. Ivan, Z. Li, M. Fallon, and S. Vijayakumar, "iDRM: Humanoid motion planning with realtime end-pose selection in complex environments," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2016, pp. 271–278.
- [22] J. Dong and J. C. Trinkle, "Orientation-based reachability map for robot base placement," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 1488–1493.
- [23] F. Paus, P. Kaiser, N. Vahrenkamp, and T. Asfour, "A combined approach for robot placement and coverage path planning for mobile manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 6285–6292.
- [24] J. Xu, K. Harada, W. Wan, T. Ueshiba, and Y. Domae, "Planning an efficient and robust base sequence for a mobile manipulator performing multiple pick-and-place tasks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 11018–11024.
- [25] T. Sandakalun, N. X. Yao, and M. H. Ang, "Inv-reach net: Deciding mobile platform placement for a given task," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2022, pp. 127–133.
- [26] S. Oswald, P. Karkowski, and M. Bennewitz, "Efficient coverage of 3D environments with humanoid robots using inverse reachability maps," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2017, pp. 151–157.
- [27] T. Weingartshofer, C. Hartl-Nesic, and A. Kugi, "Optimal TCP and robot base placement for a set of complex continuous paths," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 9659–9665.
- [28] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.
- [29] C. Ericson, *Real-Time Collision Detection*. Boca Raton, FL, USA: CRC Press, 2004.
- [30] J. Nocedal and S. J. Wright, *Numerical Optimization*. Berlin, Germany: Springer, 1999.
- [31] D. Ge, Q. Huangfu, Z. Wang, J. Wu, and Y. Ye, "Cardinal optimizer (copt) user guide," 2022, arXiv:2208.14314.
- [32] B. Sundaralingam et al., "Curobo: Parallelized collision-free robot motion generation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 8112–8119.
- [33] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.