RichMap: A Reachability Map Balancing Precision, Efficiency, and Flexibility for Rich Robot Manipulation Tasks

Yupu Lu[1], Yuxiang Ma[2], and Jia Pan[1⋆]

[1] School of Computing and Data Science, The University of Hong Kong, HKSAR
{luyp16,panj}@connect.hku.hk
[2] Massachusetts Institute of Technology, Cambridge, USA
yxma20@mit.edu

Abstract. This paper presents RichMap, a high-precision reachability map representation designed to balance efficiency and flexibility for versatile robot manipulation tasks. By refining the classic grid-based structure, we propose a streamlined approach that achieves performance close to compact map forms (e.g., RM4D) while maintaining structural flexibility. Our method utilizes theoretical capacity bounds on $\mathbb{S}^2$ (or $SO(3)$) to ensure rigorous coverage and employs an asynchronous pipeline for efficient construction. We validate the map against comprehensive metrics, pursuing high prediction accuracy ($> 98\%$), low false positive rates ($1\sim2\%$), and fast large-batch query (~15 μs/query). We extend the framework applications to quantify robot workspace similarity via maximum mean discrepancy (MMD) metrics and demonstrate energy-based guidance for diffusion policy transfer, achieving up to 26% improvement for cross-embodiment scenarios in the block pushing experiment.

Keywords: Reachability Map · Kinematics · Manipulation · Robot Similarity Analysis · Cross-Embodiment Policy Transfer.

1 Introduction

Reachability maps encode the set of reachable end-effector poses, serving as critical priors for robot manipulation tasks from base placement optimization [27] to motion planning [36]. The classical spatial-rotational grid structure [34, 35] has been widely adopted, but can be inefficient concerning the curse of dimensionality. To address this issue, other existing implementations sacrifice the simplicity for compactness [11, 22].

In this work, we propose RichMap, a reachability map framework designed to optimize the trade-off between precision, efficiency, and flexibility:

Effective theoretical analysis: We adopt a minimal kinematic assumption, a freely rotating wrist joint, and derive capacity bounds for orientation storage via geodesic distance on $\mathbb{S}^2$ (or $SO(3)$). This analysis guides parameter selection (cell

⋆ Corresponding author

size, angular threshold) to balance precision and storage, enabling reachability prediction comparable to compact 4D maps with low false positive rates.

Efficient construction framework: We design an asynchronous accelerated pipeline that decouples CPU-based forward kinematics sampling from GPU-based batched insertion. The framework scales to massive datasets (10^8 poses) while maintaining high throughput as the map densifies and achieving averaged microsecond-level query latencies for large batches.

Extended applications: The explicit separation of spatial and rotational components enables novel applications beyond classic reachability queries. We demonstrate robot workspace similarity quantification via cell-wise maximum mean discrepancy (MMD) metrics, and energy-based guidance for diffusion policy transfer, bringing challenging cross-embodiment cases to near-source performance levels ($\sim$90% vs. 95% source baseline) in the block pushing task.

2 Related Works

Reachability analysis is for evaluating workspace capabilities in robotics. In the joint space, early approaches utilized manipulability [32] to measure the determination of the manipulator posture in the workspace. This concept was later extended to account for joint limits and collisions [28, 26], providing local quality measures. Recent work also provides another way to analyze joint space dexterity given end-effector poses [31].

To provide global reachability information for the end-effector pose space, discrete map representations have been widely adopted. Capability maps [34] discretize the workspace into 3D grids to store reachability indices, such as the percentage of reachable orientations on a sphere [5, 20]. Higher-fidelity representations store detailed kinematic information within $SE(3)$ grids [4, 35], enabling precise feasibility checks. It can also integrate manipulability metrics for richer functional analysis [30]. However, the curse of dimensionality often limits the resolution or completeness of these maps.

Recent research focuses on optimizing storage and query efficiency. Analytical approaches utilize mathematical structures of the kinematic manifold to reduce data redundancy [11, 21, 22, 6]. Alternatively, learning-based methods approximate reachability boundaries using neural networks [15] to achieve continuous representations. Nonetheless, implicit representations and complex map structure will create difficulties in fast and wide applications. Our work addresses these limitations by proposing a balanced representation that maintains the simplicity and precision of explicit grid-based maps while leveraging theoretical guarantees and asynchronous generation pipeline to ensure high performance and flexibility.

3 Methodology

By theoretically analyzing position discretization and orientation coverage, we design the reachability map to efficiently encode the robot's workspace capabilities, enabling fast queries while maintaining high accuracy.

3.1 Problem Formulation

Consider a robot manipulator with forward kinematics function $f_k : \mathbb{R}^n \to SE(3)$, which maps a joint configuration $\mathbf{j} \in \mathbb{R}^n$ to an end-effector pose $\mathbf{T} \in SE(3)$ represented as 7-dimensional vectors $\mathbf{p} = [\mathbf{t}, \mathbf{q}]^\top$, where $\mathbf{t} = (x, y, z)$ specifies the position and the unit quaternion $\mathbf{q} = (q_w, q_x, q_y, q_z)$ encodes the orientation with $q_w \geq 0$ for simplicity (due to the antipodal identification $\mathbf{q} \equiv -\mathbf{q}$).

The reachability map aims to efficiently answer the query: given a target pose $\mathbf{p}^*$, determine whether there exists a collision-free joint configuration $\mathbf{j}^*$ such that $f_k(\mathbf{j}^*) \approx \mathbf{p}^*$ within some tolerance.

3.2 Grid-Based Workspace Discretization

To enable efficient spatial queries, we discretize the robot's workspace $\mathcal{W} \subset \mathbb{R}^3$ into a uniform 3D grid with cell size $\Delta \in \mathbb{R}^+$. The workspace bounds are defined as $\mathcal{W} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}]$. For any position $\mathbf{t}$, the corresponding grid cell index per dimension is computed as:

$$i_d = \left\lfloor \frac{t_d - t_{d,\min}}{\Delta} \right\rfloor, d \in \{x, y, z\}, \tag{1}$$

where indices are clamped to satisfy $0 \leq i_d < n_d$, with $n_d = \lceil (t_{d,\max} - t_{d,\min})/\Delta \rceil$ denoting the number of cells along dimension d. This discretization yields a total of $N_{\text{grid}} = n_x \cdot n_y \cdot n_z = O(n^3)$ grid cells, where n characterizes the workspace size relative to the cell resolution. By optimizing inserted poses, we can achieve a efficient balance between high spatial precision of $\Delta = 0.02$ m.

3.3 Orientation Coverage with Geodesic Distance

Within each grid cell, we store a compact set of reachable end-effector approach directions. Following the assumption in reachability analysis [22] that *the last wrist joint can rotate freely around 360 degrees with its axis aligned to the end-effector approach vector*, we represent orientations by the approach directions on the unit sphere $\mathbb{S}^2$ rather than full quaternions.

This reduction from $SO(3)$ to $\mathbb{S}^2$ significantly reduces storage requirements while maintaining sufficient orientation coverage for most manipulation tasks. But noted in Section 3.4, our pipeline actually supports progressively relaxing this assumption through rotation range tracking during map construction.

For two unit direction vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{S}^2$, the geodesic distance (great circle distance) on the sphere is:

$$d_{\mathbb{S}^2}(\mathbf{v}_1, \mathbf{v}_2) = \arccos(\mathbf{v}_1 \cdot \mathbf{v}_2), \tag{2}$$

where the dot product gives the cosine of the angle between the two directions.

Using this metric, we apply the following insertion criterion: a new pose with approach direction $\mathbf{v}$ is added to grid cell $\mathcal{C}(\mathbf{t})$ if and only if:

$$\min_{\mathbf{v}' \in \mathcal{V}_{\mathcal{C}(\mathbf{t})}} d_{\mathbb{S}^2}(\mathbf{v}, \mathbf{v}') \geq \theta, \tag{3}$$

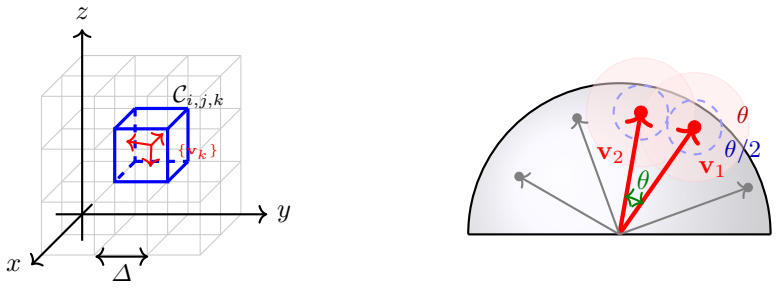

Fig. 1. Reachability map representation. Left: 3D workspace discretization with cell size Δ. Each cell stores approach direction vectors $\{\mathbf{v}_k\}$ (red arrows). Right: Orientation coverage via spherical cap packing. Directions $\mathbf{v}_1, \mathbf{v}_2$ maintain separation θ (green arc), with exclusive $\theta/2$-radius caps (blue dashed) ensuring non-overlapping packing.

where $\mathcal{V}_{\mathcal{C}(\mathbf{t})}$ denotes the set of directions already stored in the cell, and θ is the angular threshold. This criterion ensures that stored directions are well-distributed while avoiding redundancy.

To understand the storage requirements, we analyze the theoretical capacity of each cell. The insertion criterion requires that any two stored directions $\mathbf{v}_i, \mathbf{v}_j$ satisfy $d_{\mathbb{S}^2}(\mathbf{v}_i, \mathbf{v}_j) \geq \theta$. Consider an arbitrary direction $\mathbf{v}^*$ on $\mathbb{S}^2$: by the triangle inequality, its distances to any two stored directions must satisfy:

$$d_{\mathbb{S}^2}(\mathbf{v}^*, \mathbf{v}_i) + d_{\mathbb{S}^2}(\mathbf{v}^*, \mathbf{v}_j) \geq d_{\mathbb{S}^2}(\mathbf{v}_i, \mathbf{v}_j) \geq \theta. \tag{4}$$

Consequently, no direction can lie within distance $\theta/2$ of two distinct stored directions simultaneously. This property justifies assigning each stored direction an exclusive spherical cap of radius $\theta/2$, ensuring non-overlapping packing regions while respecting the minimum separation constraint.

The surface area of $\mathbb{S}^2$ is 4π, and an unit spherical cap of radius ψ on $\mathbb{S}^2$ has area $2\pi(1 - \cos\psi)$. Therefore, the maximum number of directions that can be packed with minimum separation θ per cell satisfies:

$$N_{\max} \cdot 2\pi \left(1 - \cos\frac{\theta}{2}\right) \leq 4\pi, \tag{5}$$

yielding

$$\overline{N}_{\max} \approx \frac{2}{1 - \cos(\theta/2)}. \tag{6}$$

For $\theta = 0.1$ rad ($\approx 5.7°$), this gives the upper bound $\overline{N}_{\max} \approx 1600$ directions.

In practical construction with robots such as UR5e or Franka Panda, the average number of stored directions per cell is around $K = 300$–500 and cell coverage rate 35–50%. For cell size $\Delta = 0.02$ m, this yields approximately 2×10^8 total stored poses, comparable to previous works and pursuing finer spatial resolution ($\Delta = 0.05$ m in RM4D [22] and Capability Map [35]).

Full quaternion representation. Naturally, the framework can store full $SO(3)$ quaternions. The geodesic distance corresponds to the rotation angle magnitude $d_{SO(3)}(\mathbf{q}_1, \mathbf{q}_2) = 2\arccos(|\mathbf{q}_1 \cdot \mathbf{q}_2|)$. The total volume of $SO(3)$ with the canonical metric is π^2. The volume of an unit ball with rotational radius ψ in $SO(3)$ is given by $V(\psi) = \pi(\psi - \sin\psi)$. Using sphere packing arguments with an exclusive cap radius of $\theta/2$, we obtain:

$$\overline{N}_{\max} \approx \frac{\pi^2}{\pi(\theta/2 - \sin(\theta/2))} = \frac{\pi}{\theta/2 - \sin(\theta/2)}. \tag{7}$$

With $\theta = 0.4$ rad ($\approx 22.9°$), this yields $\overline{N}_{\max} \approx 2360$ quaternions per cell. This captures complete orientation with a clear theoretical bound, which echoes the map setting (30°) used in the Capability Map [35].

3.4 Map Construction and Workflow Acceleration

We construct the reachability map through forward kinematics sampling, inserting poses only when they satisfy the geodesic distance criterion. The query operation locates the target cell, finds the nearest stored direction, and returns the associated joint configuration if within threshold θ, enabling dual purposes: fast reachability checking and approximate IK solving with $O(K)$ complexity per cell.

Algorithm 1: Asynchronous Reachability Map Construction

Input: Robot model, workspace $\mathcal{W}$, Δ, θ, target N_{target}
Output: Grid-based reachability map $\mathcal{G}$
Initialize empty grid $\mathcal{G}$; $N_{\text{inserted}} \leftarrow 0$;
while $N_{inserted} < N_{target}$ *and Inserted Rate* $>$ *Threshold* **do**
  $\{\mathbf{j}_i, \mathbf{p}_i\}_{i=1}^{B} \leftarrow$ Sample_FK_Batch();
  for *each cell c with new poses* $\{\mathbf{v}_i\}$ **do**
    Check $d_{\mathbb{S}^2}(\mathbf{v}_i, \mathbf{v}')$ vs old data (Phase 1);
    Resolve conflicts within new poses (Phase 2);
    Insert valid poses; $N_{\text{inserted}} \leftarrow N_{\text{inserted}} + |\text{valid}|$;
  end
end
return $\mathcal{G}$;

Asynchronous Pipeline Architecture. Targeting massive datasets (10^7–10^8 poses), sequential map building can take 10+ days. To address this, we design an asynchronous producer-consumer pipeline where multiple CPU worker processes generate collision-free poses in parallel while the main process executes GPU-accelerated batch insertions. Figure 2 illustrates this workflow:

CPU Generation Stage: Multiple worker processes (typically 4-16) operate independently and in parallel, each executing four sequential operations: (1)

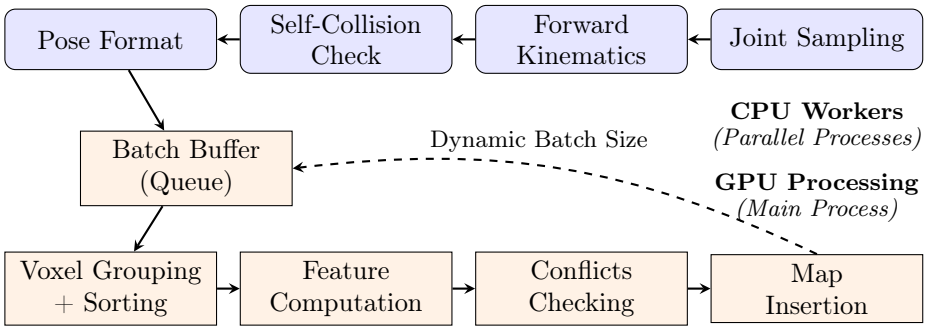

Fig. 2. Asynchronous pipeline architecture. Multiple CPU worker processes continuously generate collision-free poses in parallel while the main process executes GPU-accelerated batch insertions. The inter-process queue decouples the two stages, and batch size adjusts dynamically based on insertion throughput.

random joint sampling within robot-specific limits, (2) forward kinematics evaluation, (3) self-collision detection via physics simulation, and (4) pose formatting to $\mathbf{p}$ in SE(3) space. The workers share no state and communicate only through a thread-safe queue, enabling linear scaling with CPU core count while maintaining a steady stream of collision-free poses.

GPU Insertion Stage: The main process vectorizes batches (10^6–10^8 poses). As the map densifies and insertion rate drops, batch size automatically increases to maintain GPU utilization. This adaptive batching strategy ensures consistent throughput of the construction process.

Batched GPU Insertion. Sequential insertion becomes prohibitive as map density increases. Our batched strategy processes large chunks concurrently:

1. **Voxel Grouping:** Sort incoming poses by destination cell index (stable sort preserves temporal order), creating groups for batch processing.
2. **Feature Caching:** Compute and cache 3D axis vectors from quaternions in parallel, avoiding redundant computation during conflicts checks.
3. **Two-Phase Conflicts Detection:**
   - *Phase 1 (New vs Old):* Batch matrix multiplication (BMM) computes geodesic distances between new candidates and existing cell data.
   - *Phase 2 (New vs New):* Greedy selection with geodesic distances checking and stable sorting resolves conflicts within new candidates.

Rotation Range Tracking. The joint configurations sampled during generation enable progressive relaxation of the free wrist rotation assumption. For each inserted pose with approach direction $\mathbf{v}_i$, we maintain a set of observed wrist rotation angles $\Phi_i = \{[\phi^i_{1,\min}, \phi^i_{1,\max}], [\phi^i_{2,\min}, \phi^i_{2,\max}], \ldots\}$ based on joint configurations that share similar approach directions.

When a generated pose is rejected due to falling within the θ-neighborhood of existing pose i, we can augment the rotation set:

$$\Phi_i \leftarrow \Phi_i \cup \{[\phi_{\text{new,min}}, \phi_{\text{new,max}}]\}, \tag{8}$$

where $[\phi_{\text{new,min}}, \phi_{\text{new,max}}]$ is the wrist angle range of the rejected pose.

Section 4.1 shows the final insertion rate stabilizes around 6–7% due to orientation redundancy. This 15× rejection ratio provides substantial opportunities to refine rotation coverage: for each stored direction, we observe roughly 15 distinct wrist angles from rejected poses, progressively approaching complete $SO(3)$ representation.

3.5 Design Advantages

Our framework optimizes for high precision, efficiency, and flexibility properties, balancing the lightweight nature of RM4D [22] with the comprehensive utility of the Capability Map [35].

High precision, minimal assumptions, and theoretical capacity bounds. We adopt a single kinematic assumption—a freely rotating wrist joint—from RM4D to simplify the general $\mathbb{R}^3 + \mathbb{S}^2$ map structure. The following rigorous mathematical analysis of orientation capacity establishes a theoretical limit on the number of directions per cell, guiding settings selection for higher precision.

Furthermore, rotation range tracking (section 3.4) provides a way to progressively release this assumption, approaching $SO(3)$ coverage rather than directly discretize it.

Flexibility through spatial-rotational separation. We consciously elect to maintain a separated data structure (spatial grid and orientation set) typical of Capability Maps, rather than the highly compact representation of RM4D. Though incurring greater memory and computational costs, it affords the flexibility to empower diverse applications and extensions (Section 3.6).

Given the rapid expansion of modern hardware resources, we consider this cost a worthwhile trade-off for broader applicability and extensibility.

Efficiency via asynchronous pipeline. Our framework leverages an asynchronous producer-consumer pipeline to maximize throughput:

(1) This pipeline supports the decoupling of CPU-based FK sampling and GPU-based batched insertion. This parallelism ensures that map construction is limited only by the aggregate throughput, not sequential latency.
(2) FK evaluation is significantly faster than IK solving, and joint limits and collision constraints can be enforced during sampling without adjustments. Insertion is also accelerated through vectorized GPU processing.
(3) Joint configurations from sampling enable rotation range tracking. We can utilize the surplus of generated-but-rejected poses to refine orientation limits around stored approach directions.

3.6 Applications

The modular separation of spatial and rotational components enables several practical extensions.

Map Compression. For memory-constrained scenarios, an optional compression scheme is to represent each cell's orientation coverage using a single scalar value. Specifically, for each grid cell c, we generate M quasi-uniform samples $\{\mathbf{v}_j^s\}_{j=1}^M$ on $\mathbb{S}^2$ using Fibonacci lattice sampling and compute the coverage fraction as:

$$\rho_c = \frac{1}{M} \sum_{j=1}^{M} \mathbb{1} \left[\min_{\mathbf{v} \in \mathcal{V}_c} d_{\mathbb{S}^2}(\mathbf{v}_j^s, \mathbf{v}) < \theta \right], \tag{9}$$

where $\mathbb{1}[\cdot]$ denotes the indicator function. This compression reduces storage requirements from $O(N_{\text{grid}} \cdot K \cdot 3)$ floats for explicit direction storage to merely $O(N_{\text{grid}})$ floats for coverage fractions, which connects to implementations in [34, 4, 20].

Classic grasp-related tasks. As a general reachability representation of the capability map [35], our framework can naturally support classic tasks such as reachability-aware mobile base placement [27, 18, 17], feasible poses filtering for grasping [3, 16], and motion planning with reachability priors [36, 14, 23].

Robot similarity analysis. The grid-based representation facilitates quantitative comparison of workspace capabilities across different robot models. We employ a cell-wise similarity metric based on Maximum Mean Discrepancy (MMD) [9, 10] with a multi-scale Radial Basis Function (RBF) kernel. Given two robots with reachability grids $\mathcal{G}_A$ and $\mathcal{G}_B$ sharing identical spatial discretization, we compute for each corresponding cell pair c:

$$\text{MMD}^2(\mathcal{P}_A^c, \mathcal{P}_B^c) = \mathbb{E}_{AA} - 2\mathbb{E}_{AB} + \mathbb{E}_{BB}, \tag{10}$$

where $\mathbb{E}_{AA} = \mathbb{E}[k(\mathbf{p}_A, \mathbf{p}'_A)]$, $\mathbb{E}_{AB} = \mathbb{E}[k(\mathbf{p}_A, \mathbf{p}_B)]$, and $\mathbb{E}_{BB} = \mathbb{E}[k(\mathbf{p}_B, \mathbf{p}'_B)]$ denote the expected kernel similarities. The RBF kernel employs multi-scale bandwidth with $\gamma_m = 2^{m-M/2}$ for $m = 1, \ldots, M$ to capture similarities at different scales.

The resulting similarity grid $\mathcal{S}_{A \leftrightarrow B}$ quantifies workspace capability correspondence, with lower MMD values indicating greater functional similarity.

Learning-based Manipulation Tasks. We also integrate the robot similarity metric above into diffusion policy [7, 13]. This metric supports energy-based guidance, enabling applications in policy transfer, as demonstrated in Section 4.2.

4 Experiments

As discussed in Section 3.5, our reachability map is designed to achieve prediction performance comparable to the compact RM4D [22] while retaining the structural flexibility of the Capability Map [35]. To validate this design, we first benchmark our method against these two counterparts. We then demonstrate its practical utility through applications in robot similarity analysis and energy-based guidance for diffusion policy transfer.

4.1 Reachability Map Verification

Experimental Setup. We conduct experiments on four widely-used robot manipulators: Franka Panda, KUKA iiwa7, Universal Robots UR5e, and Kinova Gen3.

For verification, we generate 2 million test poses per robot with 50% reachable samples and 50% unreachable candidates. The latter comprises 10% poses outside the workspace radius, 10% below the ground plane, and 30% with random perturbations in free space. All candidates are verified with 100 solution attempts per pose using IKFlow, a flow-based IK solver that can rapidly sample and optimize batch IK solutions for a given end-effector pose [1]. A pose is labeled as feasible if at least one valid IK solution with no self-collision exists.

We construct reachability maps with workspace bounds of $[-1, 1] \times [-1, 1] \times [-1 + \triangle_z, 1 + \triangle_z]$ meters (0.2 or 0.3 adjusted for robot base offset) and geodesic distance threshold $\theta = 0.1$ rad. To investigate the effect of spatial resolution on prediction accuracy, we test two cell sizes: $\Delta = 0.05$ m (comparable to RM4D [22]) and $\Delta = 0.02$ m (corresponding to 15.6× more grid cells, $40^3 \rightarrow 100^3$). Map building stops when batch insertion rate drops below 1%. Every map can be built within 1 day with a superior consumer-level GPU.

Evaluation Metrics. We monitor map construction every 1 million poses inserted, evaluating four prediction metrics:

- **Accuracy:** Overall correct prediction rate
- **TPR (True Positive Rate):** Proportion of feasible poses correctly predicted as reachable
- **FPR (False Positive Rate):** Proportion of infeasible poses incorrectly predicted as reachable
- **IR (Insertion Rate):** Proportion of generated poses successfully inserted into the map
- **Query Time:** Average time (μs) per reachability query, averaged over a batch of test poses (2e6 if not specified).

Results. We first compare our method against two counterparts at $\Delta = 0.05$ m: RM4D [22], which uses compact 4D storage, and Capability Map (CM) [35], which stores full $SO(3)$ orientations (optimized with our framework, $\theta = 0.4$ rad to match insertion level). Table 1 summarizes the comparison.

Table 1. Comparison of Map Performance and Query Efficiency against Counterparts ($\Delta = 0.05$ m, $\theta = 0.1/0.4$ rad).

Method	Panda			IIWA7			UR5e			Kinova3		
	Ins.	Acc.	FPR	Ins.	Acc.	FPR	Ins.	Acc.	FPR	Ins.	Acc.	FPR
RM4D	1.1M	.980	.035	1.1M	.982	.035	1.5M	.984	.058	1.0M	.982	.045
CM	10.7M	.974	.052	9.8M	.972	.058	16.4M	.973	.082	9.1M	.972	.063
Ours	11.3M	.975	.037	10.2M	.978	.036	18.2M	.978	.063	10.0M	.978	.046
Query Efficiency (μs/query) with Varying Batch Sizes												
Method	Panda			IIWA7			UR5e			Kinova3		
	1e2	1e4	1e6	1e2	1e4	1e6	1e2	1e4	1e6	1e2	1e4	1e6
RM4D	41.9	49.0	45.6	56.2	49.7	45.2	40.7	43.5	46.4	47.8	44.9	46.0
CM	78.6	38.3	2.7	82.4	43.0	2.7	71.2	46.1	3.0	78.6	39.0	2.4
Ours	91.4	45.3	2.5	82.0	47.7	2.9	94.1	50.2	3.0	76.1	43.8	2.7

Table 2. Reachability Map Construction Results ($\theta = 0.1$ rad).

Robot	Δ (m)	Inserted	Occ.	$^{\text{Avg}}/_{\text{Cell}}$	Acc.	TPR	**FPR**	IR_{whole}	Time
Panda	0.05	11.3M	35.2%	503	0.975	0.982	0.037	6.4%	1.2
IIWA7	0.05	10.2M	43.5%	367	0.978	0.987	0.036	6.0%	1.3
UR5e	0.05	18.2M	52.7%	541	0.978	0.992	0.063	5.5%	1.3
Kinova3	0.05	10.0M	34.5%	454	0.978	0.991	0.046	5.4%	1.2
Panda	0.02	164.1M	32.8%	501	0.982	0.981	0.016	6.8%	13.5
IIWA7	0.02	145.1M	40.5%	358	0.984	0.984	0.015	6.7%	15.2
UR5e	0.02	260.0M	49.5%	526	0.984	0.989	0.028	7.1%	15.1
Kinova3	0.02	141.1M	32.2%	439	0.984	0.986	0.019	7.1%	13.2

The bottom section of Table 1 details the query speed across varying batch sizes. Simplified by theoretical assumptions, RM4D achieves comparable accuracy with $\sim$10$\times$ fewer poses and exhibits relatively constant query time due to CPU-based sequential processing. As a comparison, our GPU-accelerated implementation (Ours and CM) shows significant speedup for large batches ($\sim$2-3 μs/query for $N = 10^6$) despite 10$\times$ larger map size, highlighting the efficiency of our pipeline for bulk queries typical in sampling-based planning or learning. On the other hand, our method achieves **FPR** within 0.5% of RM4D (0.037 vs. 0.035 for Panda) while consistently outperforming CM. With prediction accuracy close to the compact map form (RM4D) and accelerated batched query speed, **our method offers a favorable balance between efficiency and precision**.

Table 2 further examines the effect of spatial resolution on our method. Comparing $\Delta = 0.05$ m and $\Delta = 0.02$ m reveals several key observations:

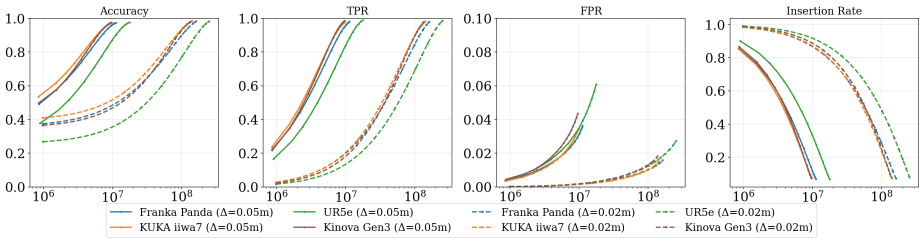

Fig. 3. Prediction metrics evolution during reachability map construction for four robots at two spatial resolutions ($\Delta = 0.05$ m solid lines, $\Delta = 0.02$ m dashed lines). All robots show rapid initial improvement followed by gradual saturation. Accuracy converges to $> 97\%$ for both resolutions, demonstrating reliable feasibility prediction. The finer resolution ($\Delta = 0.02$ m) achieves significantly lower FPR ($<3\%$) compared to coarser resolution ($\sim$4–7%), confirming that boundary approximation errors dominate false positives. Insertion rate decreases logarithmically as the map saturates, with both resolutions converging to similar final rates ($\sim$7%), indicating consistent orientation coverage density.

Spatial resolution effect on FPR. The most significant difference between $\Delta = 0.05$ m and $\Delta = 0.02$ m is the dramatic FPR reduction by about 2.5× improvement. This observation confirms our hypothesis that boundary approximation errors dominate FPR: finer cells better approximate the irregular reachability boundaries. It also underscores the importance of identifying optimal resolution parameters for balancing prediction accuracy and computational efficiency.

Map size scaling. Transitioning from $\Delta = 0.05$ m to $\Delta = 0.02$ m increases total inserted poses by $\sim$13–14× (e.g., Panda: 11.3M → 160.5M, UR5e: 20.5M → 260.0M). This scaling reflects both the 15.6× increase in grid cells and slightly varying occupancy rates across resolutions. Despite the larger map size, the average poses per cell remains relatively stable (Panda: 503 → 490, iiwa7: 367 → 358), indicating consistent orientation coverage density.

Robot-specific workspace characteristics. UR5e consistently exhibits the largest workspace coverage (occupancy $\sim$50–53%), which correlates with its higher FPR, as more boundary voxels accumulate approximation errors. Conversely, Panda and Kinova3 show lower occupancy ($\sim$32–35%), reflecting more constrained joint configurations.

Accuracy-resolution trade-off. While finer resolution ($\Delta = 0.02$ m) offers superior FPR and overall accuracy ($\sim$0.984 vs. $\sim$0.978), it requires $\sim$14× more storage and proportionally longer construction time. Query time also increases to $\sim$13–15 μs due to larger data/cache overheads, but remains 3× faster than the baseline RM4D. For applications tolerating FPR $<5\%$, $\Delta = 0.05$ m provides a cost-effective balance. Critical applications demanding minimal false positives should adopt $\Delta = 0.02$ m and insert more.

4.2 Similarity-Guided Diffusion Policy Transfer: Block Pushing Experiment

We leverage the robot similarity metric for energy-based guidance in diffusion policy [7, 13]. The discrete MMD grid in $\mathbb{R}^3$ space with boundaries $\mathcal{W}$ from Section 3.6 is transformed into a smooth energy landscape through three stages:

Stage 1: Seed initialization. Cells are classified as *seed cells* (high similarity, fixed at zero energy) or *unfilled cells* based on the MMD threshold:

$$E_c^{(0)} = \begin{cases} 0 & \text{if } s_c \leq \tau \quad \text{(seed cell, fixed throughout)} \\ \text{unfilled} & \text{otherwise} \end{cases} \tag{11}$$

where τ is the similarity threshold (set as the mean of all positive MMD values). Additionally, the central region near the robot base and any isolated zero-energy cells (with no filled neighbors) are smoothed as unfilled to avoid spurious seeds.

Stage 2: Wavefront energy propagation. Energy propagates outward from seed cells layer by layer. At each iteration t, the *boundary set* $\mathcal{B}^{(t)}$ consists of all currently unfilled cells that have at least one filled 6-connected neighbor. Each boundary cell is then updated:

$$E_c^{(t+1)} = \frac{1}{|\mathcal{N}_c^*|} \sum_{c' \in \mathcal{N}_c^*} E_{c'}^{(t)} + \delta, \quad \forall\, c \in \mathcal{B}^{(t)}, \tag{12}$$

where $\mathcal{N}_c^*$ is the subset of connected neighbors of c that already have assigned energy ($E_{c'} \geq 0$), and $\delta = 0.01$ is a fixed increment. Crucially, only cells in $\mathcal{B}^{(t)}$ are updated—seed cells and previously filled cells retain their values unchanged. Propagation **terminates** when $\mathcal{B}^{(t)} = \emptyset$, i.e., every cell has been assigned an energy value only once.

Stage 3: Smoothing. A 3D Gaussian filter smooths the energy landscape to yield a differentiable field $E : \mathbb{R}^3 \to \mathbb{R}$, where lower energy indicates higher robot similarity.

Following the energy-based model framework [33, 29], we interpret energy as unnormalized log-probability:

$$p(\mathbf{o_x}) \propto e^{-E(\mathbf{o_x})}, \quad \text{thus} \quad \nabla \log p(\mathbf{o_x}) = -\nabla E(\mathbf{o_x}), \tag{13}$$

where $\mathbf{o_x} \in \mathbb{R}^3$ represents the end-effector pose position. For a diffusion model generating robot trajectories with predicted noise $\epsilon(\mathbf{a}|\mathbf{o})$, we incorporate reachability guidance:

$$\bar{\epsilon}(\mathbf{a}|\mathbf{o}) = \epsilon(\mathbf{a}|\mathbf{o}) + \lambda \nabla E(\mathbf{o_x}), \tag{14}$$

where λ is the guidance strength. This formulation biases generated trajectories toward workspace regions with higher reachability similarity, enabling effective policy transfer.

Table 3. Success rates for policy transfer from xArm6 source to UR series robots over 9 models.

Robot	Direct Transfer		Guided Transfer		Improvement
	Success	Range	Success	Range	
xArm6 (source)	95.40%	92.94–98.00%	–	–	–
UR3	64.64%	62.82–71.88%	91.34%	87.84–95.96%	+26.70%
UR5	95.15%	92.86–98.98%	94.60%	89.88–98.98%	−0.54%
UR10	75.96%	74.70–82.74%	82.72%	77.64–86.78%	+6.76%
Average	**78.58%**	–	**89.56%**	–	**+10.97%**

Experimental Setup. To validate the practical effectiveness of reachability-guided policy transfer, we adopted the multi-modal block pushing experiment [8, 24]. The task involves manipulating two blocks to push them into two designated square target regions, with no requirement for color matching between blocks and targets. This challenging manipulation task tests both trajectory feasibility and task competence across robot morphologies.

Following standard evaluation practices, we train policies on the source robot (xArm6) and select the three best checkpoints from three independent training seeds (seeds 42, 43, 44), yielding nine models total for evaluation.

As for policy transfer, we conduct transfer experiments to three Universal Robots from the UR series with different scales: UR3 (compact), UR5 (medium), and UR10 (large). This selection covers significant variation in workspace sizes and kinematic structures, testing the robustness of similarity-based guidance.

Each trained policy is evaluated under two conditions: (1) *Direct transfer* ($\lambda = 0.0$): policy predictions used without modification, and (2) *Guided transfer*: energy-based guidance applied with varying strength $\lambda \in [0.1, 0.2, \cdots, 1.0]$.

Results. Table 3 summarizes the average success rates across nine trained models for direct transfer (baseline) and guided transfer. The reachability-guided approach achieves an average success rate of 89.56%, representing a 10.97% absolute improvement over direct transfer (78.58%). Notably, the improvement is highly robot-dependent: UR3 and UR10 show substantial gains (+26.70% and +6.76%), while UR5 exhibits minimal change (−0.54%).

Reachability similarity and transfer performance. The robot-dependent transfer outcomes strongly correlate with workspace reachability similarity patterns revealed in Figure 4. We identify a critical region near the robot base ($[-0.2, 0.2] \times [-0.2, -0.1]$ m, marked by red dashed rectangle) where kinematic differences between xArm6 and UR robots manifest most prominently.

UR3 and UR10 exhibit substantial reachability divergence from xArm6 in this region, with large MMD values (visualized as orange and red in the similarity field). When executing xArm6-trained trajectories, these robots frequently generate

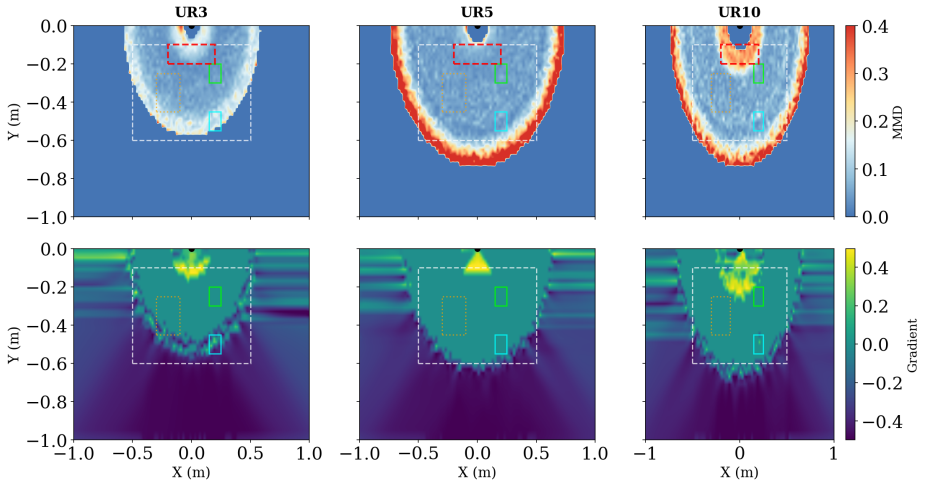

Fig. 4. Top view field visualizations across UR series transfers. Rows show similarity (top, via MMD) and gradient magnitude (bottom, $\|\nabla E\|$), columns show target robots. Similarity colormap: blue indicates low MMD (high similarity), red indicates high MMD (low similarity). The red dashed rectangle marks the *dangerous region* near the robot base where UR3 and UR10 exhibit low similarity (high MMD, orange/red), prone to generating self-constrained poses during direct transfer. In contrast, UR5 maintains high similarity (low MMD, blue) in this region, implying better direct policy transfer performance. Gradient fields (bottom) show how the direction and magnitude of energy increase, indicating how guidance pushes trajectories away from low-similarity regions. Rectangle annotations denote robot base (dot), table area (white dashed), initial block positions (orange dotted), and target regions (red and green solid).

poses in this region that lead to self-constraints (presented in the second row of Figure 5), explaining their reduced baseline success rates of 64.64% and 75.96%. The kinematic incompatibility from different robot structure makes these two robots struggle to replicate xArm6's actions near the base.

In contrast, UR5 maintains high reachability similarity throughout this critical region (blue coloring), indicating strong kinematic compatibility with xArm6. This alignment enables UR5 to execute source trajectories directly without modification, achieving 95.15% baseline success comparable to the source robot's 95.40%. The near-identical performance validates that reachability similarity serves as a reliable predictor of cross-embodiment transfer feasibility.

Energy-based guidance mechanism. The gradient field ∇E (Figure 4, bottom row) encodes the direction of increasing energy (decreasing similarity), providing spatial guidance to steer trajectories toward high-similarity regions. During diffusion denoising, this gradient is injected into the predicted noise according to Equation (14), biasing the iterative refinement process.

Figure 5 illustrates the progressive effect of guidance strength λ on UR3 policy execution. Without guidance ($\lambda = 0.0$), the robot follows xArm6's trajectory

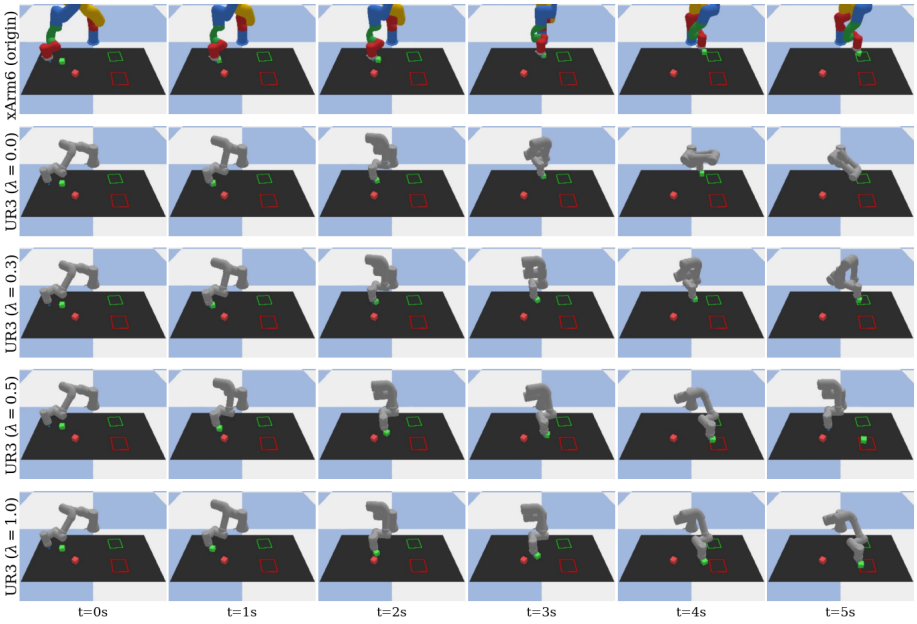

Fig. 5. Qualitative results showing the block pushing task execution. Top row depicts the xArm6 source policy execution. Subsequent rows show the same task transferred to UR3 with varying guidance strengths ($\lambda = 0.0$ direct transfer, $\lambda = 0.3, 0.5, 1.0$ with increasing guidance). The sequence spans the first 5 seconds at regular time intervals. The block is progressively pushed toward the farther side of the target regions as guidance strength increases, illustrating how reachability guidance influences the trajectory generation.

distribution, frequently entering the low-similarity region near the base and failing the task. As λ increases from 0.3 to 1.0, the gradient pushes trajectories away from this region with increasing strength.

The robot adapts by preferentially moving or pushing blocks toward the target region farther from the base, avoiding problematic arm configurations. This behavioral shift demonstrates how energy-based guidance exploits the multimodal property of diffusion policies—rather than forcing the robot to execute infeasible trajectories, it redirects the policy to explore alternative, kinematically feasible modes of task completion.

The improvements of UR3 and UR10 reflect the effects of bridging the kinematic gap between xArm6 and these robots. Conversely, UR5's minimal change (-0.54%) is expected—since the critical region already exhibits high similarity, guidance provides no additional benefit and may introduce minor perturbations that occasionally degrade performance.

5 Conclusion

This paper introduced RichMap, a grid-based reachability map that simplifies the classical spatial-rotational structure for efficient storage and insertion, while achieving higher precision through careful assumptions and theoretical analysis. Empirical validation confirms the design: across four robot manipulators, our map achieves comparable accuracy to RM4D (within 0.5% FPR) while consistently outperforming Capability Maps in false positive rates (25–35% FPR reduction), demonstrating that we can achieve performance close to compact forms while maintaining the flexibility of the grid-based structure. Moreover, our asynchronous pipeline and GPU acceleration enable large batched query speeds $\sim$18$\times$ faster than sequential counterparts. With finer spatial discretization (0.02 m), our method further reduces FPR to $\sim$1.6%. The modular structure supports extended applications, including robot similarity analysis and energy-based guidance for diffusion policy transfer. In a multimodal block pushing experiment, the guided policy transfer strategy demonstrates an average 11% improvement across three target robots, with particularly strong gains for kinematically dissimilar embodiments.

Limitations and Future Work. Our current similarity-based guidance operates only on 3D positions, which may be insufficient for orientation-sensitive tasks. Future work aims to explore more elegant integration of reachability maps in more challenging and general learning-based manipulation tasks, such as constructing full $SE(3)$ similarity, combining with other values (e.g., collision fields), trajectory planning within safe reachable set [2, 12], and leveraging motion similarity analysis [19, 25, 37] to capture dynamic properties based on static reachability. Neural network compression could also provide faster prediction and lighter memory usage while preserving the map's utility.

Acknowledgments. This work extensively involves GitHub Copilot for both code building and documentation drafting except bibliography. Careful line-by-line review, editing, adjustments, and reconstruction with steady buildup strategy for the main structure were adopted for correctness and clarity, supported by rigorous theoretical proofs. Detailed testing and cross-validation were performed to verify the reliability and reproducibility of all results related to reachability map construction.

This work was supported by the Natural Science Foundation of China (62461160309), the NSFC-RGC Joint Research Scheme (N_HKU705/24), Hong Kong RGC (GRF 17201025, GRF17200924). Yupu Lu is sponsored by the HKU Presidential PhD Scholarship.

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. Bansal, S., Chen, M., Herbert, S., Tomlin, C.J.: Hamilton-jacobi reachability: A brief overview and recent advances. In: 2017 IEEE 56th annual conference on decision and control (CDC). pp. 2242–2253. IEEE (2017)
3. Burget, F., Bennewitz, M.: Stance selection for humanoid grasping tasks by inverse reachability maps. In: 2015 IEEE International conference on robotics and automation (ICRA). pp. 5669–5674. IEEE (2015)
4. Cao, Y., Lu, K., Li, X., Zang, Y.: Accurate numerical methods for computing 2d and 3d robot workspace. International Journal of Advanced Robotic Systems **8**(6), 76 (2011)
5. Castelli, G., Ottaviano, E., Ceccarelli, M.: A fairly general algorithm to evaluate workspace characteristics of serial and parallel manipulators. Mechanics based design of structures and machines **36**(1), 14–33 (2008)
6. Cavelli, R.F., Cen Cheng, P.D., Indri, M.: Modeling the reachability space of robotic manipulators through ellipsoid equations. Journal of Intelligent & Robotic Systems **111**(3), 90 (2025)
7. Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. In: Proceedings of Robotics: Science and Systems (RSS) (2023)
8. Florence, P., Lynch, C., Zeng, A., Ramirez, O.A., Wahid, A., Downs, L., Wong, A., Lee, J., Mordatch, I., Tompson, J.: Implicit behavioral cloning. In: Conference on robot learning. pp. 158–168. PMLR (2022)
9. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: A kernel method for the two-sample-problem. Advances in neural information processing systems **19** (2006)
10. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. The journal of machine learning research **13**(1), 723–773 (2012)
11. Han, Y., Pan, J., Xia, M., Zeng, L., Liu, Y.J.: Efficient se (3) reachability map generation via interplanar integration of intra-planar convolutions. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 1854–1860. IEEE (2021)
12. Holmes, P., Kousik, S., Zhang, B., Raz, D., Barbalata, C., Johnson-Roberson, M., Vasudevan, R.: Reachable sets for safe, real-time manipulator trajectory design. In: Proceedings of Robotics: Science and Systems (RSS) (2020)
13. Janner, M., Du, Y., Tenenbaum, J., Levine, S.: Planning with diffusion for flexible behavior synthesis. In: International Conference on Machine Learning. pp. 9902–9915. PMLR (2022)
14. Jauhri, S., Peters, J., Chalvatzaki, G.: Robot learning of mobile manipulation with reachability behavior priors. IEEE Robotics and Automation Letters **7**(3), 8399–8406 (2022)
15. Jiang, L., Ren, J., Zhou, Z., Qu, Y., Lu, H., Wu, M.: Graph reinforcement learning-based reachability map for generalized mobile manipulation. IEEE Transactions on Cognitive and Developmental Systems (2025)
16. Jiang, P., Oaki, J., Ishihara, Y., Ooga, J., Han, H., Sugahara, A., Tokura, S., Eto, H., Komoda, K., Ogawa, A.: Learning suction graspability considering grasp quality and robot reachability for bin-picking. Frontiers in Neurorobotics **16**, 806898 (2022)
17. Kluge-Wilkes, A., Schmitt, R.H.: Mobile robot base placement for assembly systems: survey, measures and task clustering. In: Congress of the German Academic Association for Production Technology. pp. 439–447. Springer (2021)
18. Makhal, A., Goins, A.K.: Reuleaux: robot base placement by reachability analysis. In: 2018 second IEEE international conference on robotic computing (IRC). pp. 137–142. IEEE (2018)

19. Makondo, N., Rosman, B., Hasegawa, O.: Knowledge transfer for learning robot models via local procrustes analysis. In: 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids). pp. 1075–1082. IEEE (2015)
20. Porges, O., Lampariello, R., Artigas, J., Wedler, A., Borst, C., Roa, M.A.: Reachability and dexterity: Analysis and applications for space robotics. In: Workshop on advanced space technologies for robotics and automation-ASTRA (2015)
21. Quan, Y., Zhao, C., Lv, C., Wang, K., Zhou, Y.: The dexterity capability map for a seven-degree-of-freedom manipulator. Machines **10**(11), 1038 (2022)
22. Rudorfer, M.: Rm4d: A combined reachability and inverse reachability map for common 6-/7-axis robot arms by dimensionality reduction to 4d. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 7689–7695 (2025). https://doi.org/10.1109/ICRA55743.2025.11128095
23. Sandakalum, T., Ang Jr, M.H.: Motion planning for mobile manipulators—a systematic review. Machines **10**(2), 97 (2022)
24. Shafiullah, N.M., Cui, Z., Altanzaya, A.A., Pinto, L.: Behavior transformers: Cloning k modes with one stone. Advances in neural information processing systems **35**, 22955–22968 (2022)
25. Urain, J., Peters, J.: Generalized multiple correlation coefficient as a similarity measurement between trajectories. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1363–1369. IEEE (2019)
26. Vahrenkamp, N., Asfour, T.: Representing the robot's workspace through constrained manipulability analysis. Autonomous Robots **38**(1), 17–30 (2015)
27. Vahrenkamp, N., Asfour, T., Dillmann, R.: Robot placement based on reachability inversion. In: 2013 IEEE International Conference on Robotics and Automation. pp. 1970–1975. IEEE (2013)
28. Vahrenkamp, N., Asfour, T., Metta, G., Sandini, G., Dillmann, R.: Manipulability analysis. In: 2012 12th ieee-ras international conference on humanoid robots (humanoids 2012). pp. 568–573. IEEE (2012)
29. Xu, M., Geffner, T., Kreis, K., Nie, W., Xu, Y., Leskovec, J., Ermon, S., Vahdat, A.: Energy-based diffusion language models for text generation. arXiv preprint arXiv:2410.21357 (2024)
30. Xu, R., Luo, J., Wang, M.: Optimal grasping pose for dual-arm space robot cooperative manipulation based on global manipulability. Acta Astronautica **183**, 300–309 (2021)
31. Yao, H., Laha, R., Figueredo, L.F., Haddadin, S.: Enhanced dexterity maps (edm): A new map for manipulator capability analysis. IEEE Robotics and Automation Letters **9**(2), 1628–1635 (2023)
32. Yoshikawa, T.: Manipulability of robotic mechanisms. The international journal of Robotics Research **4**(2), 3–9 (1985)
33. Yu, L., Song, Y., Song, J., Ermon, S.: Training deep energy-based models with f-divergence minimization. In: International Conference on Machine Learning. pp. 10957–10967. PMLR (2020)
34. Zacharias, F., Borst, C., Hirzinger, G.: Capturing robot workspace structure: representing robot capabilities. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3229–3236. Ieee (2007)
35. Zacharias, F., Borst, C., Wolf, S., Hirzinger, G.: The capability map: A tool to analyze robot arm workspaces. International Journal of Humanoid Robotics **10**(04), 1350031 (2013)
36. Zhang, H., Sheng, Q., Sun, Y., Sheng, X., Xiong, Z., Zhu, X.: A novel coordinated motion planner based on capability map for autonomous mobile manipulator. Robotics and autonomous systems **129**, 103554 (2020)

37. Zielinska, T., Coba, G.: The measure of motion similarity for robotics application. Sensors **23**(3), 1643 (2023)