

# Elliptical K-Nearest Neighbors - Path Optimization via Coulomb's Law and Invalid Vertices in C-space Obstacles

Liding Zhang<sup>1</sup>, Zhenshan Bing<sup>1</sup>, Yu Zhang<sup>1</sup>, Kuanqi Cai<sup>1</sup>, Lingyun Chen<sup>1</sup>, Fan Wu<sup>1</sup>,  
Sami Haddadin<sup>1</sup>, Alois Knoll<sup>1†</sup>

**Abstract**—Path planning has long been an important and active research area in robotics. To address challenges in high-dimensional motion planning, this study introduces the Force Direction Informed Trees (FDIT\*), a sampling-based planner designed to enhance speed and cost-effectiveness in pathfinding. FDIT\* builds upon the state-of-the-art informed sampling planner, the Effort Informed Trees (EIT\*), by capitalizing on often-overlooked information in invalid vertices. It incorporates principles of physical force, particularly Coulomb's law. This approach proposes the elliptical  $k$ -nearest neighbors search method, enabling fast convergence navigation and avoiding high solution cost or infeasible paths by exploring more problem-specific search-worthy areas. It demonstrates benefits in search efficiency and cost reduction, particularly in confined, high-dimensional environments. It can be viewed as an extension of nearest neighbors search techniques. Fusing invalid vertex data with physical dynamics facilitates force-direction-based search regions, resulting in an improved convergence rate to the optimum. FDIT\* outperforms existing single-query, sampling-based planners on the tested problems in  $\mathbb{R}^4$  to  $\mathbb{R}^{16}$  and has been demonstrated on a real-world mobile manipulation task.

## I. INTRODUCTION

In the realm of robot motion planning, especially in high-dimensional spaces, sampling-based algorithms play a pivotal role in path planning [1]. These algorithms are rooted in the mathematical foundation of *random geometric graphs* (RGG) [2], probabilistic models representing randomly distributed networks in *configuration space* ( $C$ -space). In  $C$ -space, RGG position vertices randomly, extending into complex, higher-dimensional regions. Edges form between vertices within a specified distance, denoted as the graph's radius. Key features involve random vertex distribution in an  $n$ -dimensional space and edge creation based on proximity, typically measured using Euclidean metrics. The graph's connectivity hinges on the chosen radius, with a critical radius value determining connectivity or disconnectivity [2].

Motion planning, rooted in the A\* algorithm [3], has developed in various directions. Lifelong planning A\* (LPA\*) [4] addresses planning tasks in dynamic scenarios, while sampling-based planner Rapidly-exploring Random Trees (RRT) [5] explores nonconvex and high-dimensional spaces. RRT builds a tree from a starting point, aiming for the goal point. RRT-Connect [6] enhances efficiency by growing trees from both the start and goal positions. RRT\* [7] incrementally builds a tree using  $k$ -nearest [8] search, providing anytime solutions and provably converging to an optimal

<sup>1</sup>L. Zhang, Z. Bing, Y. Zhang, K. Cai, L. Chen, F. Wu, S. Haddadin and A. Knoll are with the Department of Informatics, Technical University of Munich, Germany. liding.zhang@tum.de

<sup>†</sup>The authors acknowledge the financial support by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for the Lighthouse Initiative KI.FABRIK (Phase 1: Infrastructure and the research and development program under grant no. DIK0249).

Corresponding author: Zhenshan Bing.

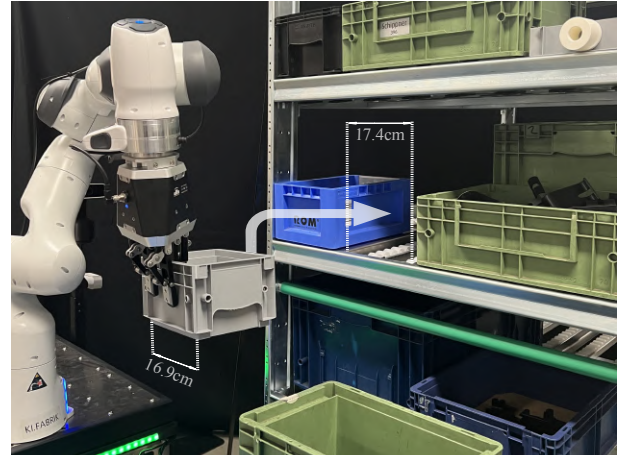


Fig. 1: Mobile manipulator robot during a real-time industry standardized ( $\pm 5$ mm) container-shelf arrangement task (Section VI-B).

solution. Informed RRT\* [9] simplifies the search and sampling set using elliptical concepts (*informed-sampling* [10]), thereby enhancing the convergence rate and the quality of the final solution during path optimization.

Batch Informed Trees (BIT\*) [11], [12] integrates Informed RRT\* and Fast Marching Trees (FMT\*) [8], combining single and multiple sampling techniques to compactly group states (*batch*) into implicit RGG. Its advanced version, ABIT\* [13], uses an inflation factor and a truncation factor to balance exploration and exploitation in a dense RGG approximation. Adaptively Informed Trees (AIT\*) [14], [15] and Effort Informed Trees (EIT\*) [14] employ an asymmetrical *forward-reverse* search method with sparse collision checks in the reverse search. These adaptations enhance the efficiency of exploration in motion planning algorithms. It has evolved from evaluating single-dimensional points to higher-dimensional regions. However, the common utilization of finding nearest neighbors for path rewiring is either mutual  $k$ -nearest or  $r$ -nearest [16] neighbors, those methods neglect information on obstacles and often abandon invalid vertices. The subsequent advancement involves exploring clusters of points or valuable regions, such as *invalid* vertices.

This paper introduces Force Direction Informed Trees (FDIT\*), an asymptotically optimal sampling-based planner that employs a force-direction method and elliptical  $k$ -nearest neighbors (elliptical-KNN) search to enhance point evaluation techniques. These methods use invalid samples in  $C$ -space obstacles to guide the search toward valuable areas, improving search efficiency in pathfinding.

FDIT\* enhances exploration by using invalid vertices and physical forces. It extends traditional  $k$ -nearest neighbors search through Coulomb's law-based force direction analysis, guiding the search and avoiding costly or infeasible areas.

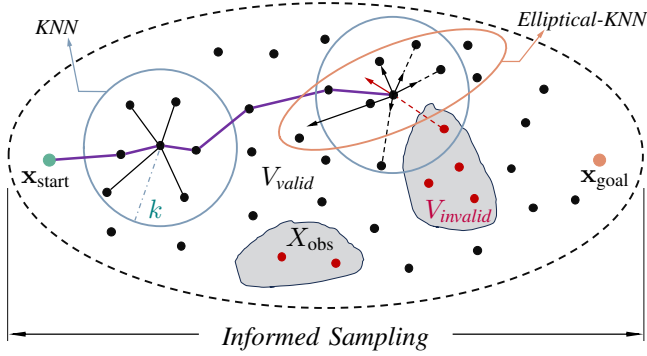


Fig. 2: The force directional elliptical  $k$ -nearest search is visualized. Black and red dots are sampled as valid and invalid vertices,  $k$  donates as the connection parameters, black thin lines as forward rewiring trees, magenta lines as the initial path, arrows indicate force directions and orange ellipses denote the area of elliptical-KNN in the forward search. Undesirable vertices are excluded from the neighbors, and relevant vertices are included in the elliptical-KNN.

This improves search efficiency and reduces path costs, especially in high-dimensional, multi-narrow passage scenarios. Demonstrated in Fig. 1, FDIT\* was tested on the DARKO robot for intralogistics tasks, showcasing efficient motion planning in confined environments. Its potential extends to automation and robotics, proving robust and effective in autonomous robotics and path planning.

The contributions of this work are summarized as follows:

- *Elliptical  $k$ -nearest neighbors*: FDIT\* defines nearest neighbors by summing forces influenced by valid and invalid samples around current states. This enhances the likelihood of approaching more explore-worthy regions.
- *Improved path quality*: FDIT\* incorporates force direction analysis based on Coulomb's law, resulting in more obstacle-awareness direction searches and up to 34.65% initial solution cost reduction in various test scenarios.
- *Real-world application in autonomous robotics*: FDIT\* was tested in an industry-standard shelf-container arrangement task, and it attained the highest success rate (80% under 1.0s) due to problem-specific exploration.

## II. BACKGROUND & PRELIMINARIES

In an RGG, the connections (i.e., edges) between states (i.e., vertices or sample points) depend on their relative geometric positions. Common RGGs have edges to a specific number of each state's nearest neighbors (a  $k$ -nearest graph [17]) or all neighbors within a particular distance (an  $r$ -nearest graph [18]). RGG theory establishes probabilistic relationships between the number and distribution of samples, the  $k$  or  $r$  defining the graph, and specific graph properties such as connectivity or relative cost through the graph [7].

### A. $R$ -Nearest Neighbors (RNN) Search

The connection radius  $r(q)$  is a dynamic parameter that scales according to the density of the state space and the sample size  $q$ . It is defined as:

$$r(q) := 2\eta \left( \left( 1 + \frac{1}{n} \right) \left( \frac{\lambda(X_{\hat{f}})}{\lambda(B_{1,n})} \right) \left( \frac{\log(q)}{q} \right) \right)^{\frac{1}{n}}, \quad (1)$$

where  $\eta$  is a normalization constant,  $n$  represents the dimensionality of the space,  $\lambda(\cdot)$  denotes the Lebesgue measure,

$X_{\hat{f}}$  is the informed set, and  $B_{1,n}$  is a unit ball in an  $n$ -dimensional space [14]. This formula ensures that  $r(q)$  adapts to the complexity of the space, enhancing the graph's connectivity without increasing computational effort [19].

### B. $K$ -Nearest Neighbors (KNN) Search

The number of nearest neighbors  $k(q)$ , crucial for defining the graph's sparsity, is determined as follows:

$$k(q) := \eta e \left( 1 + \frac{1}{n} \right) \log(q), \quad (2)$$

This formulation of  $k(q)$ , as suggested in [20], is designed to balance between thorough exploration of the state space and maintaining a manageable graph size. As the number of samples  $q$  increases,  $k(q)$  scales logarithmically, ensuring that each new node connects to a sufficient number of neighbors to maintain robustness in the graph structure.

### C. Coulomb's Law in Multi-Dimensional Space

Coulomb's law, foundational in classical electrodynamics, explains the force between two point charges in three-dimensional space [21]. The force is inversely proportional to the square of the distance between charges and directly proportional to the product of their magnitudes. Charges of the same type repel, while opposite charges attract. In an  $n$ -dimensional vacuum, the Coulomb's force magnitude between static charges is inversely proportional to the  $n-1$  power of the distance. Given two charges  $p$  and  $p'$  with magnitudes  $q$  and  $q'$ , and positions  $r$  and  $r'$ , the Coulomb force vector from  $q'$  to  $q$  is represented as:

$$\vec{F}(p, p') = \mathcal{E} q' \vec{r} = k_n \frac{qq'}{\|r - r'\|^{n-1}} \vec{r}, \quad (3)$$

where  $\mathcal{E}$  denotes the electric field intensity at  $p'$ ,  $k_n$  is the electrostatic constant in  $n$ -dimensional space, and  $\vec{r}$  is the unit vector from  $p'$  to  $p$ . The Coulomb force vector can be detailed in components as  $[f_1(p, p'), \dots, f_n(p, p')]$ , where  $n$  is the dimensionality of the field. The Coulomb forces are unique and independent of dimension for any two distinct charges. In an  $n$ -dimensional electrostatic field  $Q$  composed of charges, the Coulomb resultant force on a charge  $p$  is the sum of forces exerted by other charges, defined as:

$$\vec{F}_Q(p) := \sum_{p' \in Q \setminus p} \vec{F}(p, p') = [f_1(p), \dots, f_n(p)]. \quad (4)$$

where  $f_n(p)$  is the summation of the  $n$ -dimensional components of the Coulomb force on  $p$  by other charges. Coulomb forces and resultant forces provide the relative positioning and deviation of charge  $p$  within the electrostatic field.

For sampling-based planners, the Open Motion Planning Library (OMPL) [22], widely employed in benchmarking motion-planning algorithms, provides a comprehensive framework and tools for researchers to evaluate algorithms. Force Direction Informed Trees (FDIT\*) is integrated into the OMPL framework, the Planner-Arena benchmark database [23], and Planner Developer Tools (PDT) [24]

## III. PROBLEM FORMULATION

We define the optimal planning problem as outlined in [7]. In this context:

*Problem Definition 1 (Optimal Planning)*: Given a planning problem with state space  $X \subseteq \mathbb{R}^n$ , where  $X_{\text{obs}}$  denotes

---

**Algorithm 1:** get\_best\_ellipse\_k\_nearest\_samples

---

**Input :**  $\mathbf{x}$  - The current state,  $k$  - Number of nearest neighbors to find  
**Output:** Best set of  $k$  nearest neighbors within an ellipsoidal region around  $\mathbf{x}$

- 1  $F_{\mathcal{D}} \leftarrow$  an  $n$ -dimensional vector of all ones
- 2 ▷ initial force direction vector
- 3  $totalVertices \leftarrow 0$ ,  $invalidVertices \leftarrow 0$
- 4  $V_{ellipseNeighbors} \leftarrow$  ellipse\_neighbors( $\mathbf{x}, k, F_{\mathcal{D}}$ )
- 5  $\Phi \leftarrow$  calculate\_ratio( $V_{ellipseNeighbors}$ )
- 6 ▷ initialize charge ratio, skip loop when 90% NN is valid
- 7 **while**  $\Phi \geq 0.1$  **do** ▷ NN invalid samples less than 10%
- 8  $F_{\mathcal{D}} \leftarrow$   
compute\_force\_direction( $\mathbf{x}, V_{ellipseNeighbors}$ )
- 9  $V_{ellipseNeighborsNew} \leftarrow$  ellipse\_neighbors( $\mathbf{x}, k, F_{\mathcal{D}}$ )
- 10 ▷ recalculate NN based on the updated force direction
- 11 **foreach**  $\mathbf{x}_i \in V_{ellipseNeighborsNew}$  **do**
- 12  $totalVertices \leftarrow totalVertices + 1$
- 13 **if not** is\_valid( $\mathbf{x}_i$ ) **then**
- 14  $invalidVertices \leftarrow invalidVertices + 1$
- 15 **if**  $totalVertices > 0$  **then**
- 16  $\Phi \leftarrow invalidVertices/totalVertices$
- 17 ▷ non-negative charge ratio  $\in (0, 1]$
- 18 **else**
- 19  $\Phi \leftarrow 0$  ▷ no samples in nearest neighbor
- 20  $V_{ellipseNeighbors} \leftarrow V_{ellipseNeighborsNew}$
- 21 **foreach**  $\mathbf{x}_i \in V_{ellipseNeighbors}$  **do**
- 22 **if** is\_valid( $\mathbf{x}_i$ ) **then**
- 23  $V_{valid} \leftarrow V_{valid} \cup \{\mathbf{x}_i\}$
- 24 **return**  $V_{ellipseNeighbors}$

---

---

**Algorithm 2:** ellipse\_neighbors( $\mathbf{x}, F_{\mathcal{D}}$ )

---

**input :**  $\mathbf{x}$  - The current state,  $F_{\mathcal{D}}$  - Force direction vector  
**output :**  $V_{ellipseNeighbors}$  - The nearest neighbor vertices within the ellipsoidal region

- 1  $V_{ellipseNeighbors} \leftarrow$  ellipse\_nearest( $\mathbf{x}, k, F_{\mathcal{D}}$ )
- 2  $V_{ellipseNeighbors} \stackrel{\pm}{\leftarrow} V_{ellipseNeighbors} \cup \{\text{parent}_{\mathcal{F}}(\mathbf{x}) \cup \text{children}_{\mathcal{F}}(\mathbf{x})\} \setminus V_{ellipseNeighbors}$
- 3  $V_{ellipseNeighbors} \stackrel{\pm}{\leftarrow} \{X_i \in V_{ellipseNeighbors} | (X, X_i) \in E_{\mathcal{F}, invalid}\}$
- 4 **return**  $V_{ellipseNeighbors}$

---

states in collision with obstacles, and  $X_{\text{free}} = cl(X \setminus X_{\text{obs}})$  represents permissible states, where  $cl(\cdot)$  represents the *closure* of a set. The initial state is  $\mathbf{x}_{\text{start}} \in X_{\text{free}}$ , and the desired final states are in  $X_{\text{goal}} \subset X_{\text{free}}$ . A continuous map  $\sigma : [0, 1] \mapsto X$  represents a collision-free path, and  $\Sigma$  is the set of all nontrivial paths.

The optimal solution, denoted as  $\sigma^*$ , is the path minimizing a chosen cost function  $s : \Sigma \mapsto \mathbb{R}_{\geq 0}$ . This path connects the initial state  $\mathbf{x}_{\text{start}}$  to any goal state  $\mathbf{x}_{\text{goal}} \in X_{\text{goal}}$  through the free space:

$$\sigma^* = \arg \min_{\sigma \in \Sigma} \{s(\sigma) | \sigma(0) = \mathbf{x}_{\text{start}}, \sigma(1) \in \mathbf{x}_{\text{goal}}, \forall t \in [0, 1], \sigma(t) \in X_{\text{free}}\}. \quad (5)$$

where  $\mathbb{R}_{\geq 0}$  denotes non-negative real numbers. The cost of the optimal path is  $s^*$ .

Considering a discrete set of states,  $X_{\text{samples}} \subset X$ , represented as a graph with edges determined algorithmically by

---

**Algorithm 3:** ellipse\_nearest( $\mathbf{x}, X_{\text{all}}, F_{\mathcal{D}}, k$ )

---

**Input :**  $\mathbf{x}$  - The current state,  $X_{\text{all}}$  - All available vertices,  $F_{\mathcal{D}}$  - Force direction vector,  $k$  - The number of nearest valid vertices to find  
**Output:**  $V_{ellipseNearestValid}$  - The nearest  $k$  valid vertices within the ellipsoidal region,  $V_{\text{valid}}$  - All valid vertices,  $V_{\text{invalid}}$  - All invalid vertices

- 1 *Initialization-filter vertices within the ellipsoidal region and separate valid and invalid vertices*
- 2  $V_{\text{sortedValid}} \leftarrow \emptyset$ ,  $V_{\text{valid}} \leftarrow \emptyset$ ,  $V_{\text{invalid}} \leftarrow \emptyset$ ,  $V_{\text{ellipseNearest}} \leftarrow \emptyset$
- 3 **foreach**  $\mathbf{x}_i \in X_{\text{all}}$  **do**
- 4 **if** is\_within\_ellipse( $\mathbf{x}, \mathbf{x}_i, F_{\mathcal{D}}$ ) **then**
- 5 **if** is\_valid( $\mathbf{x}_i$ ) **then**
- 6  $V_{\text{valid}} \leftarrow V_{\text{valid}} \cup \{\mathbf{x}_i\}$
- 7  $V_{\text{invalid}} \leftarrow V_{\text{invalid}} \cup \{\mathbf{x}_i\}$
- 8  $V_{\text{sortedValid}} \leftarrow$  sort( $V_{\text{valid}}$ )
- 9  $V_{\text{ellipseNearestValid}} \leftarrow$  first  $k$  elements of  $V_{\text{sortedValid}}$
- 10 **return**  $V_{\text{ellipseNearestValid}}$

---

a transition function, we can characterize its properties using a probabilistic model, specifically implicit dense RGGs, i.e.,  $X_{\text{samples}} = \mathbf{x} \sim \mathcal{U}(X)$ , as discussed in [2].

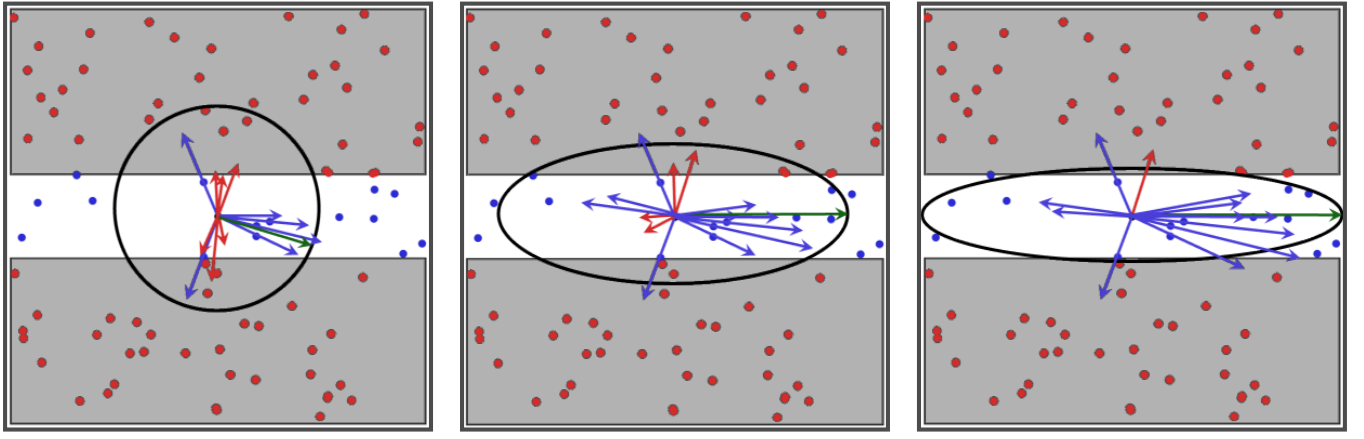
*Problem Definition 2 (Nearest Neighbors):* A fundamental primitive that frequently arises is the *nearest neighbors* (NN) problem. The problem is defined as follows: given a collection of  $n$  objects, build a data structure that, upon receiving an arbitrary query object, reports the dataset object that is most similar to the query. The concept of Randomized  $c$ -approximate RNNs, denoted as  $(c, R)$ -NN, is central in computational geometry. Consider a set  $D$  comprising points in a  $n$ -dimensional space  $\mathbb{R}^n$ , along with parameters  $R > 0$  and  $\delta > 0$ . The objective is to devise a data structure capable of, when presented with any query point  $q$ , identifying some  $cR$ -near neighbors of  $q$  within  $D$  with a probability of success no less than  $1 - \delta$ . Assuming  $\delta$  is a constant not close to 1 (e.g.,  $1/2$ ), success probability can be boosted by building and querying multiple data structure instances [20].

#### IV. FORCE DIRECTION INFORMED TREES (FDIT\*)

FDIT\* uses elliptical  $k$ -nearest neighbors and Coulomb's law-based force direction to guide the search process, steering it away from collisions and mimicking natural navigation patterns. A force computation formula (Alg. 4) and elliptical-KNN search are introduced to explore neighboring areas.

##### A. Notation

The state space of the planning problem is denoted by  $X \subseteq \mathbb{R}^n$ , where  $n \in \mathbb{N}$ . The start vertex is represented by  $\mathbf{x}_{\text{start}} \in X$ , and the goals are denoted by  $X_{\text{goal}} \subset X$ . The sampled states are denoted by  $X_{\text{sampled}}$ . The forward and reverse search trees are represented by  $\mathcal{F}$  and  $\mathcal{R}$ , respectively. The vertices and edges in these trees, denoted by  $V_{\mathcal{F}}$  and  $V_{\mathcal{R}}$ , are associated with valid states. The edges in the forward tree,  $E_{\mathcal{F}} \subseteq V_{\mathcal{F}} \times V_{\mathcal{F}}$ , represent valid connections between states. An edge comprises a source state,  $\mathbf{x}_s$ , and a target state,  $\mathbf{x}_t$ , denoted as  $(\mathbf{x}_s, \mathbf{x}_t)$ . An admissible estimate for the cost of a path is  $\hat{f} : X \rightarrow [0, \infty)$ . For sets  $A, B$ , and  $C$  with  $B, C$  being subsets of  $A$ , the notation  $B \stackrel{\pm}{\leftarrow} C$  denotes  $B \leftarrow B \cup C$ , and  $B \stackrel{-}{\leftarrow} C$  denotes  $B \leftarrow B \setminus C$ .



(a) Initial iteration with a circular search area, balancing negative and positive forces, with  $k = 12$ , negative = 6, positive = 6. (b) Subsequent iteration with an elliptical search area, influenced by the dominant resultant force, with  $k = 14$ , negative = 4, positive = 10. (c) Final iteration with a refined search area, following the optimized direction of the resultant force, with  $k = 12$ , negative = 1, positive = 11.

● Valid Samples (Positive) ● Invalid Samples (Negative) → Resultant Force

Fig. 3: Illustration of the progression of the search area refinement in FDIT\* on a narrow corridor benchmark scenario, it demonstrates how the algorithm incrementally focuses the search towards the most non-obstacle feasible regions, as the cumulative resultant force directs (Alg. 1). Red points represent invalid samples that fall within obstacles. Blue points signify valid samples within obstacle-free areas. Corresponding arrows indicate the force exerted by each point on the current position. The green arrow is the resultant force that guides the search towards feasible regions. The gray areas represent the obstacle block's walls, and the white spaces delineate the accessible areas for path planning.  $k$  represents the total number of nearest neighbors considered during the search.

*FDIT\*-specific Notation:* Force calculations are conducted through a formula incorporating elliptical search neighbors, denoted as  $V_{\text{ellipseNeighbors}}$ . Distinct electrical charges characterize samples within or outside obstacles: invalid charges  $V_{\text{invalid}}$  associated with energy  $\mathcal{E}_{\text{invalid}}$ , and valid charges  $V_{\text{valid}}$  with energy  $\mathcal{E}_{\text{valid}}$ . The charge ratio within  $V_{\text{ellipseNeighbors}}$  is denoted as  $\Phi$ , and the direction of the resultant force on each vertex is represented by  $F_{\hat{D}}$ . The electric field of nearest neighbor samples within  $V_{\text{ellipseNeighbors}}$  is denoted as  $Q_{\text{valid}}$ ,  $Q_{\text{invalid}}$ , respectively. Subsequent force calculations are executed on the acquired valid set to evaluate its quality.

$F_{\hat{D}}$  serving as a vector to store the force direction information and operates within the same  $\mathcal{C}$ -space dimension as the RGG. Spatial coordinates of valid and invalid sample points are respectively expressed as  $\mathbf{x}_{\text{positive}}$  and  $\mathbf{x}_{\text{negative}}$ . The parameter  $\rho_0$  remains constant in this context. The number of valid samples is denoted as  $m_{\text{valid}}$ , while  $m_{\text{invalid}}$  represents the count of invalid sample points. Electric charges of samples, denoted by  $q_1$  and  $q_2$ , correspond to the current vertex and a vertex from the surrounding set, both scalars. The attractive force exerted by valid sample points on the current vertex is represented by  $\vec{F}_{\text{attractive},i}$ , while  $\vec{F}_{\text{repulsive}}$  indicates the repulsive force exerted by invalid sample points. Magnetic fields vector generated by surrounding sample points and the energy of the current vertex within these magnetic fields are represented by  $\vec{\mathcal{E}}_{\text{attractive}}$  and  $\vec{\mathcal{E}}_{\text{repulsive}}$ , respectively.

The Euclidean distance between two samples is denoted as  $r_i$ , and the unit direction vector between two samples is represented by  $\hat{\mathbf{r}}_i$ . The magnitude and direction of the force exerted by the  $i^{\text{th}}$  vertex on the current vertex are denoted by  $\vec{F}_i$ . Sets such as  $V_{\text{ellipseNeighborsNew}}$ ,  $V_{\text{ellipseNeighborsValid}}$ , and  $V_{\text{sortedValid}}$  are utilized to store related vertices correspondingly. The non-negative scalar charge ratio  $\Phi$  represents the scale between valid and invalid vertices.

**Algorithm 4:** compute\_force\_direction( $\mathbf{x}$ ,  $V_{\text{ellipseNeighbors}}$ )

---

**input** :  $\mathbf{x}$  - The current state,  $V_{\text{ellipseNeighbors}}$  - A set of neighboring samples from ellipse neighbors  
**output** :  $F_{\hat{D}}$  - The resultant force direction vector

- 1  $F_{\hat{D}} \leftarrow \vec{0}$  ▷ initialize as a zero vector for safety
- 2 **foreach**  $\mathbf{x}_i \in V_{\text{ellipseNeighbors}}$  **do**
- 3      $r_i \leftarrow \text{distance}(\mathbf{x}, \mathbf{x}_i)$  ▷ compute the states distance
- 4      $\hat{\mathbf{r}}_i \leftarrow \frac{\mathbf{x}_i - \mathbf{x}}{r_i}$  ▷ compute the unit vector direction of  $\mathbf{x}_i$
- 5     **if** is\_valid( $\mathbf{x}_i$ ) **then**
- 6          $\vec{F}_{\text{attractive},i} \leftarrow \frac{Q_{\text{valid}}}{r_i^{n-1}} \hat{\mathbf{r}}_i$  ▷ attractive positive force
- 7     **else**
- 8          $\vec{F}_{\text{repulsive},i} \leftarrow -\frac{Q_{\text{invalid}}}{r_i^{n-1}} \hat{\mathbf{r}}_i$  ▷ repulsive negative force
- 9      $F_{\hat{D}} \leftarrow F_{\hat{D}} \pm \vec{F}_i$  ▷ add or minus depends on direction
- 10 **return**  $F_{\hat{D}}$

---

### B. Extension of Coulomb's Law to High Dimensions

Consider a  $n$ -dimensional Euclidean space  $\mathbb{R}^n$ . Let two vertex charges,  $q_1$  and  $q_2$ , be located in this space, separated by a distance  $r$ . The unit vector from  $q_1$  to  $q_2$  is denoted by  $\hat{\mathbf{r}}$ . The Coulomb force in high-dimension is proposed to be:

$$\vec{F} := \mathbf{k}_e \frac{q_1 q_2}{r^{n-1}} \hat{\mathbf{r}}, \quad (6)$$

where  $\vec{F}$  represents the force vector exerted by the charges on each other. The proportionality constant  $\mathbf{k}_e$  is analogous to Coulomb's constant in three dimensions, determining the strength of the force.  $n$  represents the dimensionality of the  $\mathcal{C}$ -space where these interactions occur, and  $r$  signifies the Euclidean distance between the charges.

### C. Elliptical $K$ -Nearest Neighbors Search Method

The elliptical-KNN search method deviates from the conventional spherical search area and instead utilizes an

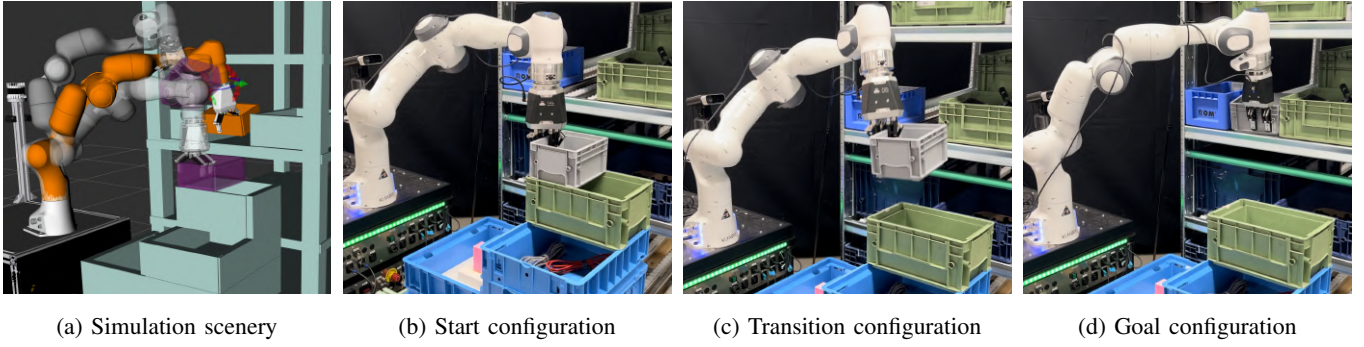


Fig. 4: Illustrates the simulation (a) and the real-world scenarios of DARKO robot for the industry shelf-arrangement task, (b) shows the start configuration of the arm to pick up the gray container from the pallet, (c) shows the transition configuration position of the task, (d) shows the goal configuration of the arm to place the container into the *narrow corridor* (tolerances  $\pm 5\text{mm}$ ) between two other containers.

elliptical or, more generally, an  $n$ -dimensional ellipsoidal search space [25]. The uniqueness of the elliptical search lies in its adaptive nature (Fig. 3), where the search area's shape dynamically aligns with the direction and magnitude of the Coulomb forces acting on the current state.

**Definition 1.** Given two vertices in an  $n$ -dimensional space,  $x$  and  $y$ , where  $x = (x_1, x_2, \dots, x_n)$  and  $y = (y_1, y_2, \dots, y_n)$ , the elliptical distance metric is defined as:

$$d_{\text{elliptical}}(x, y) := \sqrt{\sum_{i=1}^n \left( \frac{x_i - y_i}{v_i} \right)^2}, \quad (7)$$

where  $v_i$  represents the scaling factor for the  $i$ -th dimension, adjusting the importance and scale of different dimensions.

**Theorem 1.** (Dynamic selection for elliptical-KNN): Let  $P$  be a set of vertices in an  $n$ -dimensional space and  $q$  a query vertex. Assuming the distance metric is defined by the elliptical equation. The number of nearest neighbors,  $k$ , is dynamically determined by the function (2), then a vertex  $p \in P$  is an elliptical-KNN of  $q$  if and only if for all  $p' \in P$ ,  $p$  satisfies the following condition for a number less than or equal to  $k(q)$ :

$$\sum_{i=1}^n \left( \frac{q_i - p_i}{v_i} \right)^2 \leq \sum_{i=1}^n \left( \frac{q_i - p'_i}{v_i} \right)^2, \quad (8)$$

*Proof.* To prove that this selection optimizes the search process, we consider the continuous function  $f(q) = k(q)$  and its derivative with respect to  $q$ ,

$$k'(q) = \frac{df(q)}{dq} = \eta e \left( 1 + \frac{1}{n} \right) \frac{1}{q}. \quad (9)$$

This derivative indicates that as  $q$  increases, the rate of change of  $k(q)$  decreases, suggesting that  $k(q)$  grows logarithmically with  $q$ . The optimization comes from the fact that in denser regions of the space (higher  $q$ ), a minor increase in  $k$  is needed to capture a representative set of nearest neighbors due to the logarithmic relationship. This ensures the search is adaptive and efficient, focusing computational resources where they are most needed. By adjusting  $k$  dynamically, the search process can accommodate the varying density of vertices in different regions of the space, leading to an optimized balance between accuracy and computational cost. Hence, the dynamic selection of  $k$  based on  $q$  is shown to optimize the exploration under the defined conditions.  $\square$

In high-dimensional path planning, the elliptical-KNN search (Alg. 1, 2, and 3) becomes an ellipsoidal search, where the corresponding force component in that dimension weights each axis of the ellipsoid. This weighting inherently incorporates the relative information between the current state and other samples in the state space. The search becomes directional and informed. Prioritizing the search effort on areas more likely to yield an improved path quality.

The directionality of the search is a direct consequence of the force direction, which not only enhances the algorithm's efficiency by guiding it towards obstacle-free regions but also encodes valuable information about the state space's topology and obstacle distribution. This enables FDIT\* to navigate complex environments efficiently by leveraging force information for decision-making on search directions.

## V. FORMAL ANALYSIS

In this paper, we refer to Definition 24 from [7] to establish the concept of almost-sure asymptotic optimality.

### A. Almost-Sure Asymptotically Optimal Path

The FDIT\* algorithm utilizes an increasingly dense RGG approximation similar to EIT\*, in which the underlying graph almost certainly includes an asymptotically optimal path. The search in FDIT\* is also asymptotically resolution-optimal. Considering EIT\* is an almost-surely asymptotically optimal planner [24]. Therefore, FDIT\*'s RGG approximation almost certainly contains an optimal path.

### B. Elliptical-KNN vs. Traditional KNN

1) *Adaptive search area vs. fixed radius:* Traditional KNN search uses a fixed-radius spherical area. In contrast, FDIT\* employs an adaptive elliptical-KNN search, adjusting the ellipsoidal search space based on the net force direction and magnitude. This dynamic adaptation enables FDIT\* to focus the search more effectively in the direction indicated by the current state's force vector.

2) *Directional information and relative weighting:* The elliptical-KNN search incorporates directional information and relative weighting based on the force components in each dimension. In contrast, the traditional KNN search treats all regions equally, inadequate the capability to prioritize search directions based on the state space's dynamics or topology. The elliptical search inherently includes information about other vertices' relative positions to the current state, enhancing the search's relevance and efficiency.

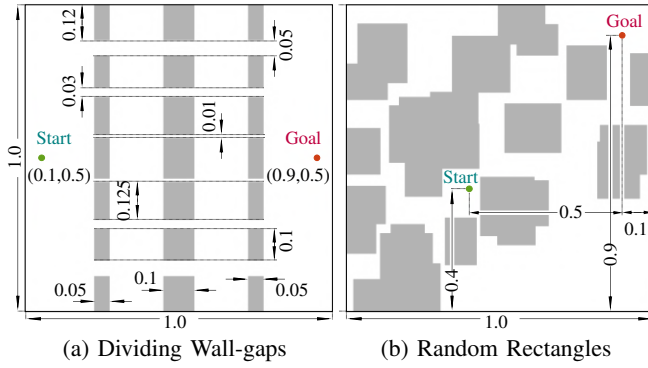


Fig. 5: Simulated planning problems were visualized using a 2D representation. The possible states, represented by  $X \subset \mathbb{R}^n$ , were restricted to a hypercube with a side length of 1.0 for both problems. Ten variations of the dividing wall-gaps and random rectangles experiment were tested, and the results are shown in Fig. 6.

3) *Search efficiency in high-dimensional spaces:* The elliptical-KNN search incorporates directional information and relative weighting based on force components in each dimension. Unlike traditional KNN search, which treats all regions equally, it can prioritize search directions based on the state space's dynamics and topology. This approach inherently includes information about other vertices' relative positions, enhancing the search's relevance and efficiency.

4) *Improved path optimization:* By configuring the NN search area with invalid vertices and the resultant force direction, the algorithm is probable to find valuable sampled states during the NN searching process, which form paths that are both feasible and optimal, particularly in environments with numerous obstacles and high-dimensional  $\mathcal{C}$ -space.

### C. Synthetic Electromagnetic Guidance Method

A virtual electromagnetic field is constructed with the target vertex as the center. The current vertex's real-time position coordinates are denoted as  $x$ , with the target vertex exerting an electromagnetic attraction or repulsion on the current vertex, represented as  $F_{\text{attractive}}(x)$ , and the electromagnetic potential energy denoted as  $\mathcal{E}_{\text{attractive}}(x)$ . Surrounding vertices in obstacles exert an electromagnetic repulsive force on the current vertex, represented as  $F_{\text{repulsive}}(x)$ , with the electromagnetic repulsive potential energy denoted as  $\mathcal{E}_{\text{repulsive}}(x)$ . The electromagnetic potential energy at the current state's position  $x$ ,  $\mathcal{E}_{\text{attractive}}(x)$ , is expressed as follows:

$$\mathcal{E}_{\text{attractive}}(x) = \mathbf{k}_a \cdot \frac{1}{|x - x_{\text{positive}}|^2}, \quad (10)$$

where  $\mathbf{k}_a$  is the Coulomb's charge coefficient related to the positive force, and  $x_{\text{positive}}$  denotes the location of the positive (valid) vertex. The electromagnetic force (Alg. 4, line 6) exerted on the current vertex at position  $x$ ,  $F_{\text{attractive}}(x)$ , is:

$$F_{\text{attractive}}(x) = \mathbf{k}_a \cdot \frac{1}{|x - x_{\text{positive}}|^3} \cdot (x - x_{\text{positive}}), \quad (11)$$

The electromagnetic repulsive potential energy at the current vertex's position  $x$  is represented as  $\mathcal{E}_{\text{repulsive}}(x)$ , where  $\mathbf{k}_r$  is the charge coefficient related to the negative vertex,  $x_{\text{negative}}$  is the negative (invalid) vertex's position, and  $\rho_0$  is the range within which the negative vertex can exert

electromagnetic repulsion on the current vertex, is:

$$\mathcal{E}_{\text{repulsive}}(x) = \begin{cases} -\mathbf{k}_r \cdot \frac{1}{|x - x_{\text{negative}}|^2} & , |x - x_{\text{negative}}| \leq \rho_0 \\ 0 & , |x - x_{\text{negative}}| > \rho_0 \end{cases}, \quad (12)$$

The electromagnetic repulsive force on the current vertex at position  $x$ ,  $F_{\text{repulsive}}(x)$ , is:

$$F_{\text{repulsive}}(x) = \begin{cases} -\mathbf{k}_r \cdot \frac{(x - x_{\text{negative}})}{|x - x_{\text{negative}}|^3} & , |x - x_{\text{negative}}| \leq \rho_0 \\ 0 & , |x - x_{\text{negative}}| > \rho_0 \end{cases}. \quad (13)$$

Through these adjustments, we have constructed a virtual potential field model based on Coulomb's law suitable for path planning of the current vertex. Considering the interactions between charges, including attraction and repulsion, this model enables effective navigation of the current vertex in an environment containing positive and negative vertices. The following forms for electromagnetic potential energy and force expressions in high-dimensional state spaces are proposed:

$$\vec{\mathcal{E}}_{\text{attractive},i}(\mathbf{x}) = \mathbf{k}_e \sum_{i=1}^{m_{\text{valid}}} \frac{q_1 q_{2,i}}{\|\mathbf{x} - \mathbf{x}_{\text{positive},i}\|^{n-1}}, \quad (14)$$

$$\vec{\mathcal{E}}_{\text{repulsive},j}(\mathbf{x}) = -\mathbf{k}_e \sum_{j=1}^{m_{\text{invalid}}} \frac{q_1 q_{2,j}}{\|\mathbf{x} - \mathbf{x}_{\text{negative},j}\|^{n-1}}, \quad (15)$$

where  $m_{\text{valid}}$  and  $m_{\text{invalid}}$  are the number of valid and invalid sample vertices in  $\mathcal{C}$ -space, respectively.

$$\vec{F}_{\text{attractive},i}(\mathbf{x}) = \mathbf{k}_e \sum_{i=1}^{m_{\text{valid}}} \frac{q_1 q_{2,i}}{\|\mathbf{x} - \mathbf{x}_{\text{positive},i}\|^n} (\mathbf{x}_{\text{positive},i} - \mathbf{x}), \quad (16)$$

$$\vec{F}_{\text{repulsive},j}(\mathbf{x}) = -\mathbf{k}_e \sum_{j=1}^{m_{\text{invalid}}} \frac{q_1 q_{2,j}}{\|\mathbf{x} - \mathbf{x}_{\text{negative},j}\|^n} (\mathbf{x}_{\text{negative},j} - \mathbf{x}). \quad (17)$$

In these formulas:

- $\mathbf{k}_e$  is a proportionality constant to Coulomb's constant.
- $q_1$  is the charge of the current state,  $q_{2,i}$  and  $q_{2,j}$  are the charges of the  $i$ -th valid sample vertex and the  $j$ -th invalid sample vertex, respectively.
- $\mathbf{x}$  is the position vector of the current state,  $\mathbf{x}_{\text{positive},i}$  and  $\mathbf{x}_{\text{negative},j}$  are the position vectors of the  $i$ -th valid vertex and the  $j$ -th invalid vertex, respectively.
- $n$  is the dimensionality of the  $\mathcal{C}$ -space.

In contemplating the combined influence of numerous sample vertices on the present condition, these expressions prove versatile within high-dimensional state spaces.

## VI. EXPERIMENTAL RESULTS

FDIT\* was evaluated against several existing algorithms, including different versions of RRT-Connect, Informed RRT\*, BIT\*, AIT\*, ABIT\*, and EIT\* from the Open Motion Planning Library (OMPL). Tests were conducted in simulated environments ranging from  $\mathbb{R}^4$  to  $\mathbb{R}^{16}$  and real-world manipulation scenarios using an Intel i7 3.90 GHz processor with 32GB of LPDDR3 3200 MHz memory. The main goal was to minimize the median initial path length ( $\mathcal{C}_{\text{init}}^{\text{med}}$ ) over 100 runs. For all planners, the RGG constant  $\eta$  was set to 1.1, and the rewiring factor to 1.001. RRT-based algorithms used

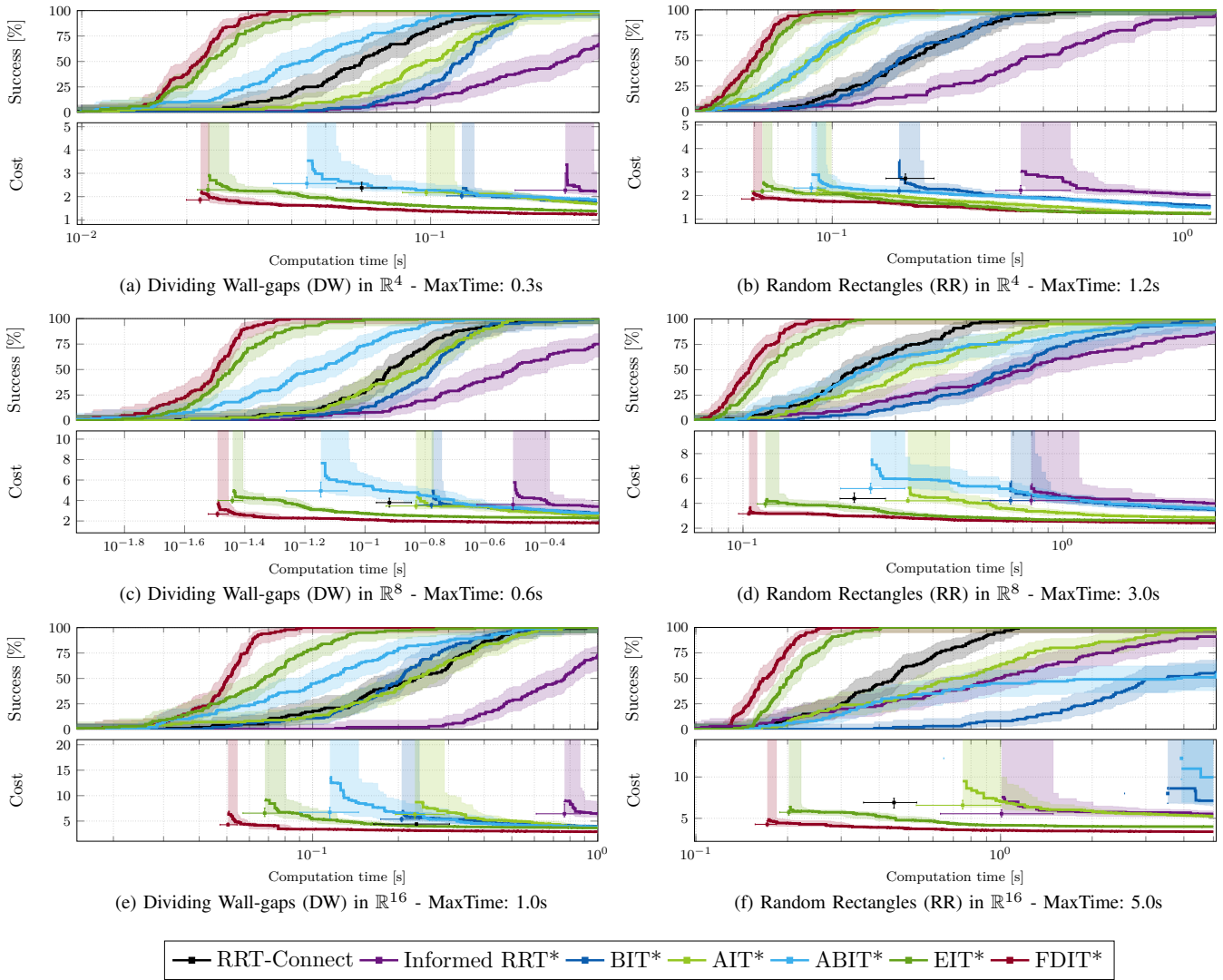


Fig. 6: Detailed experimental results from Section VI-A are presented above. Fig. (a), (c), and (e) depict test benchmark dividing wall gaps outcomes in  $\mathbb{R}^4$ ,  $\mathbb{R}^8$  and  $\mathbb{R}^{16}$ , respectively. Panel (b) showcases ten random rectangle experiments in  $\mathbb{R}^4$ , while panels (d) and (f) demonstrate in  $\mathbb{R}^8$  and  $\mathbb{R}^{16}$ . In the cost plots, boxes represent solution cost and time, with lines showing cost progression for planners (unsuccessful runs have infinite costs). Error bars provide nonparametric 99% confidence intervals for solution cost and time.

a 5% goal bias, with maximum edge lengths adjusted for space dimensionality. FDIT\*, BIT\*, AIT\*, ABIT\*, and EIT\* sampled 200 states per batch, utilizing Euclidean distance and effort heuristics with graph pruning. Unlike the others, FDIT\* employed elliptical-KNN for NN search and rewiring.

FDIT\*'s elliptical-KNN search method dynamically adjusted the search regions. The planner's adaptive mechanism optimizes the NN search areas and rewires at any time.

### A. Experimental Tasks

The planners were subjected to testing across three distinct problem domains:  $\mathbb{R}^4$ ,  $\mathbb{R}^8$ , and  $\mathbb{R}^{16}$ . In the first scenario, a constrained environment resembling a dividing wall with several narrow gaps was simulated, allowing valid paths in multiple general directions for non-intersecting solutions (Fig. 5a). Each planner underwent 100 runs, and the computation time for each anytime asymptotically optimal planner is demonstrated in the labels, with varying random seeds. The overall success rates and median path lengths for all planners are depicted in Fig. 6a, 6c, and 6e.

TABLE I: Benchmarks evaluation comparison (Fig. 6)

	EIT*			FDIT*			$c_{med}^{init} \uparrow$ (%)
	$t_{med}^{init}$	$c_{med}^{init}$	$c_{med}^{final}$	$t_{med}^{init}$	$c_{med}^{init}$	$c_{med}^{final}$	
RR - $\mathbb{R}^4$	0.0632	2.1817	1.2286	0.0593	1.8532	1.2173	15.06
RR - $\mathbb{R}^8$	0.1175	3.9503	2.6092	0.1043	<b>3.1590</b>	2.4199	20.03
RR - $\mathbb{R}^{16}$	0.2019	5.7269	3.9973	0.1715	<b>4.2738</b>	<b>3.3797</b>	<b>25.37</b>
DW - $\mathbb{R}^4$	0.0230	2.2950	1.3820	0.0218	1.8600	1.2332	18.95
DW - $\mathbb{R}^8$	0.0362	4.0013	2.2889	0.0323	<b>2.6774</b>	<b>1.8085</b>	<b>33.34</b>
DW - $\mathbb{R}^{16}$	0.0679	6.5623	3.6617	<b>0.0505</b>	<b>4.2887</b>	<b>2.9037</b>	<b>34.65</b>

In the second test environment, random widths were assigned to *axis-aligned hyperrectangles*, which were generated within the  $\mathcal{C}$ -space arbitrarily (Fig. 5b). Unique random rectangle problems were created for each dimension of the  $\mathcal{C}$ -space, and each planner was run 100 times for every instance. Fig. 6b, 6d, and 6f depict the success rates and median path costs within the computation time.

As observed in Table I, there's a median cost improvement across varied benchmark scenarios, correlating with dimensionality. In the case of the DW- $\mathbb{R}^{16}$  scenario, the initial median solution cost exhibits a reduction of up to 34.65%.

## B. Path Planning for DARKO

FDIT\* demonstrated its effective elliptical-KNN techniques during an industry container and multi-layer shelf arrangement tasks (Fig. 1 and Fig. 4). DARKO (8-DoF) is a mobile manipulation robotic platform designed to tackle challenges within intralogistics supermarket cells. Inserting industry containers into narrow spaces is difficult due to component standardization. The task is to place the industry standard container between two larger containers on the shelf. Due to the tolerance scope between containers being  $\leq 5\text{mm}$ , planning a collision-free feasible path is challenging. This container-shelf arrangement task aims to optimize path length and success rate. All planners had 1.0 seconds to address this problem. Over 15 trials, FDIT\* was 80% successful with a median solution cost of 10.9225. EIT\* was 73.3% successful and had a median solution cost of 14.5726. AIT\* was 60% successful with a median solution cost of 16.7038, and ABIT\* was 66.67% successful with a median solution cost of 16.4719. FDIT\* consistently optimizes the elliptical-KNN by integrating the information of *invalid vertices* in these  $\mathcal{C}$ -space obstacles. This allows FDIT\* to become more problem-specific and achieve higher success rates in completing tasks. The detailed behavior of DARKO can be viewed in <https://youtu.be/s7CHNLbVo6k>.

## VII. DISCUSSION & CONCLUSION

In this paper, we introduce FDIT\*, an advanced sampling-based planner using elliptical-KNN search, which improves on traditional KNN by including invalid vertices and calculating force directions with Coulomb's law. This method enhances path planning algorithms. Determining vertex charges  $q_{2,i}$  and  $q_{2,j}$  is crucial, as it affects their influence on the current state  $q_1$ . Currently, all vertices have the same charge, but practical applications may require different charges based on potential values. Future research will focus on this quantification. In tests with the DARKO robot, FDIT\* demonstrated its adaptability by achieving short path lengths and quickly generating initial solutions, especially in narrow environments. This showcases its practical effectiveness.

In conclusion, FDIT\* uses invalid vertices and physical force dynamics to focus search regions and reduce costs. In the future, we could further test FDIT\* in bio-inspired robots [26], consider incorporating human acceptability and comfort into planning [27], and investigate reinforcement learning-based methods [28] for robot motion planning.

## REFERENCES

- [1] K. Cai, W. Chen, D. Dugas, R. Siegwart, and J. J. Chung, "Sampling-based path planning in highly dynamic and crowded pedestrian flow," *IEEE Transactions on Intelligent Transportation Systems*, 2023. I
- [2] M. Penrose, *Random geometric graphs*. OUP Oxford, 2003, vol. 5. I, 4
- [3] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. I
- [4] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning a\*," *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93–146, 2004. I
- [5] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001. I
- [6] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 995–1001 vol.2. I
- [7] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011. I, II, 4, V
- [8] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 883–921, 2015. I
- [9] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *2014 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2014. I
- [10] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Informed sampling for asymptotically optimal path planning," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 966–984, 2018. I
- [11] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (bit\*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 3067–3074. I
- [12] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Batch informed trees (bit\*): Informed asymptotically optimal anytime search," *The International Journal of Robotics Research*, vol. 39, no. 5, 2020. I
- [13] M. P. Strub and J. D. Gammell, "Advanced bit (abit): Sampling-based planning with advanced graph-search techniques," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 130–136. I
- [14] M. P. Strub and J. D. Gammell, "Adaptively informed trees (ait\*) and effort informed trees (eit\*): Asymmetric bidirectional sampling-based path planning," *The International Journal of Robotics Research*, vol. 41, no. 4, pp. 390–417, 2022. I, II-A
- [15] M. P. Strub and J. D. Gammell, "Adaptively informed trees (ait): Fast asymptotically optimal path planning through adaptive heuristics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3191–3198. I
- [16] M. Kleinbort, O. Salzman, and D. Halperin, "Collision detection or nearest-neighbor search? on the computational bottleneck in sampling-based motion planning," *Springer Proceedings in Advanced Robotics*, p. 624–639, 2020. I
- [17] F. Xue and P. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, p. 169–181, Mar. 2004. II
- [18] E. N. Gilbert, "Random plane networks," *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, no. 4, pp. 533–543, 1961. II
- [19] K. Solovey and M. Kleinbort, "The critical radius in sampling-based motion planning," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 266–285, 2020. II-A
- [20] A. Andoni, "Nearest neighbor search: the old, the new, and the impossible," Ph.D. dissertation, Massachusetts Institute of Technology, 2009. II-B, 4
- [21] A. Lenard, "Exact statistical mechanics of a one-dimensional system with coulomb forces," *Journal of Mathematical Physics*, vol. 2, no. 5, pp. 682–693, 1961. II-C
- [22] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012. II-C
- [23] M. Moll, I. A. Sucan, and L. E. Kavraki, "Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 96–102, 2015. II-C
- [24] J. D. Gammell, M. P. Strub, and V. N. Hartmann, "Planner developer tools (pdt): Reproducible experiments and statistical analysis for developing and testing motion planners," in *Proceedings of the Workshop on Evaluating Motion Planning Performance (EMPP), IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022. II-C, V-A
- [25] S. Ohmori, "A predictive prescription using minimum volume k-nearest neighbor enclosing ellipsoid and robust optimization," *Mathematics*, vol. 9, no. 2, p. 119, 2021. IV-C
- [26] Z. Bing, A. Rohregger, F. Walter, Y. Huang, P. Lucas, F. O. Morin, K. Huang, and A. Knoll, "Lateral flexion of a compliant spine improves motor performance in a bioinspired mouse robot," *Science Robotics*, vol. 8, no. 85, 2023. VII
- [27] K. Cai, W. Chen, C. Wang, S. Song, and M. Q.-H. Meng, "Human-aware path planning with improved virtual doppler method in highly dynamic environments," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 1304–1321, 2022. VII
- [28] Z. Bing, D. Lerch, K. Huang, and A. Knoll, "Meta-reinforcement learning in non-stationary and dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3476–3491, 2023. VII