# Look as You Leap: Planning Simultaneous Motion and Perception for High-DoF Robots

Qingxi Meng[1], Emiliano Flores[1], Carlos Quintero-Peña[1], Peizhu Qian[1], Zachary Kingston[1], Shannan K. Hamlin[3], Vaibhav Unhelkar[2], and Lydia E. Kavraki[2]

***Abstract*—Most common tasks for robots in dynamic spaces require that the environment is regularly and actively perceived. The perception task considered in this work can represent a broad range of robot perception objectives, including object detection, human activity recognition, and human face detection. For example, a service robot may need to continuously detect and localize a target object during manipulation, while an assistive robot in a healthcare setting may need to maintain reliable perception of a human face or activity for interaction and safety monitoring. These tasks impose perception constraints on the robot motion. However, solving motion and perception tasks simultaneously is challenging, as these objectives often impose conflicting requirements. Furthermore, while robots must react quickly to changes in the environment, directly evaluating the quality of perception (e.g., object detection confidence) is often expensive or infeasible at runtime. This problem is especially important in human-centered environments, such as homes and hospitals, where effective perception is essential for safe and reliable operation. In this work, we address the challenge of solving motion planning problems for high-degree-of-freedom (DoF) robots from a start to a goal configuration with continuous perception constraints under both static and dynamic environments. Our solution is a GPU-parallelized perception-score-guided probabilistic roadmap planner with a neural surrogate model (PS-PRM). Unlike existing active perception-, visibility-aware or learning-based planners, our work jointly considers perception tasks and constraints when searching for a solution to the motion planning problem. Our method uses a neural surrogate model to approximate perception scores, incorporates them into a roadmap-based solution, and leverages GPU parallelism to enable efficient online replanning in dynamic settings. We demonstrate that our planner, evaluated on high-DoF robots, outperforms RL- and trajectory-optimization-based baseline methods in both static and dynamic environments in both simulation and real-robot experiments.**

***Index Terms*—Motion and Path Planning, Constrained Motion Planning, Nonholonomic Motion Planning, Reactive and Sensor-Based Planning**

## I. Introduction

Achieving mobile manipulation (e.g., navigating to a table to grasp a tool) alongside perception tasks (e.g., tracking an object or a human) remains highly challenging. In this work, we address the problem of solving motion planning problems for high degree-of-freedom (DoF) robots from a start to a goal configuration while optimizing a given perception task in both static and dynamic environments. The perception task considered in this work can represent a broad range of perception objectives, including object detection or tracking, human activity recognition, and face detection. We define the perception quality as a task-dependent numerical measure of how well the perception task is achieved from a given continuous robot trajectory. Examples include object detection confidence scores, human pose estimation accuracy, face detection confidence, or object tracking reliability metrics. A fundamental challenge emerges: the objectives of motion and perception often conflict. The perception tasks could impose non-trivial constraints on robot motion, as performance varies with object shape, occlusion, distance, and viewing angle [1–3]. For example, achieving reliable perception may require the robot to move closer to a target or adopt specific viewing angles, which can conflict with motion objectives such as maintaining clearance from obstacles, respecting kinematic limits, or computing/optimizing for a short and smooth path. At the same time, planning for high-degree-of-freedom (DoF) robots with perception constraints must account for the complex, nonlinear relationship between the robot's configuration and the camera pose.

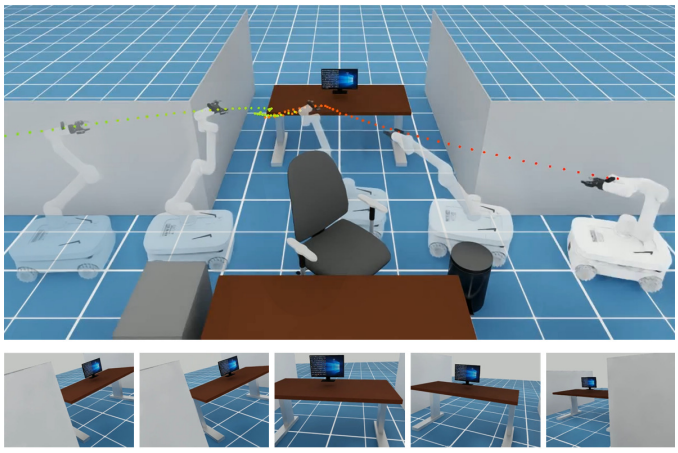


Figure 1. A UR5 on a differential-drive base moves through a cluttered environment while tracking a monitor with its wrist-mounted camera. **Top:** the robot's trajectory (fading right to left) as it maintains a high detection rate. **Bottom:** camera views for the configurations shown above.

As an example, consider the situation in Fig. 1. A mobile manipulator with a wrist-mounted camera must move from a start to a goal configuration through a cluttered environment while observing a monitor at the end of the hallway. The robot must reach its goal with a trajectory that avoids occlusions

This work was supported in part by NSF 2326390 and NSF CCF-2336612. (Corresponding author: Lydia E. Kavraki, Qingxi Meng.)

[1]Qingxi Meng, Emiliano Flores, Carlos Quintero-Peña, Peizhu Qian, and Zachary Kingston are with the Department of Computer Science, Rice University, Houston, TX 77005 USA (e-mail: qm15@rice.edu).

[2]Vaibhav Unhelkar and Lydia E. Kavraki are with the Department of Computer Science, Rice University, Houston, TX 77005 USA, and also with the Ken Kennedy Institute, Rice University, Houston, TX 77005 USA (e-mail: vaibhav.unhelkar@rice.edu; kavraki@rice.edu).

[3]Shannan K. Hamlin is with the Houston Methodist Academic Institute, Houston, TX 77030 USA (e-mail: shamlin@houstonmethodist.org).

and achieves high tracking confidence.

The planning challenges are further compounded in dynamic environments, where objects and people move. In such settings, robots must continuously adapt to changes while achieving high perception quality [4–7]. For example, a key motivation of this work comes from human-centered environments such as nurse training settings [8, 9], where robotic tutors are used to support skill acquisition. In such an environment [10], the robot must continuously monitor a trainee to detect compliance with sterile techniques and provide timely feedback while simultaneously moving through the environment. Fig. 7 and Fig. 12 illustrate this challenge. In these settings, the robot must continuously track nurses' faces while navigating through a cluttered clinical environment. However, naively evaluating perception performance (e.g., detection confidence) for perception-aware planning is prohibitively expensive in real environments [11] because it would require physically placing the sensor at a large number of distinct poses throughout the environment. This motivates the need for efficient mechanisms to deliver perception quality and support fast, responsive replanning in high-dimensional spaces.

Existing works that consider both motion and perception are not designed for the problem examined in this paper. Informative path planners [12–15] aim to find paths that maximize information gain while respecting constraints such as path length or budget, but they often become computationally expensive, especially when the problem size grows. Perception-aware path planners are typically designed for low-DoF systems [16–19] or rely on overly simplified perception models [20], such as keeping a point of interest centered in the field of view. Active perception approaches [21–29] focus on selecting discrete viewpoints to maximize scene understanding, rather than ensuring consistent perception quality along a motion trajectory. Moreover, most of these approaches are not designed to operate in dynamic environments where objects or humans may move and where continuous adaptation is needed. As a result, they lack the responsiveness required for real-time perception-aware planning in practical settings.

In this paper, we propose **Perception-Score**-guided **Probabilistic Roadmap** Planning (PS-PRM): a GPU-parallelized, roadmap-based planner that explicitly incorporates the estimated quality of a given perception task (e.g., object detection confidence score) into motion planning for high-degree-of-freedom (DOF) robots. The perception score is predicted by a neural surrogate model trained to approximate perception quality for specific tasks. A key aspect of our formulation is that perception quality is modeled in SE(3) space rather than in robot-specific configuration space. By basing the surrogate perception model on the camera pose and its relative transform to the target, it becomes robot- and environment-agnostic, enabling direct reuse across different high-DoF platforms with similar sensor setups. Fast forward kinematics (FK) allows this model to still be directly and efficiently linked to the configuration space and enables better approximation of continuous perception performance through interpolation between configurations. This design choice allows our planner to generalize beyond a single robot and environment, a property we demonstrate in our experiments. PS-PRM jointly optimizes for both motion and perception quality by associating each configuration with an estimated perception score and using these scores to guide roadmap construction and path selection. To efficiently approximate perception quality during planning, our neural surrogate model is trained on data from various objects and human targets that are typically present in the specific environment the robot is expected to operate. To account for occlusions in cluttered environments, we incorporate a ray-casting-based pipeline that dynamically adjusts perception estimates. Finally, to enable fast replanning in dynamic environments, we develop a GPU-parallelized framework that executes collision checking, FK, and perception evaluation in batch. These components allow our method to scale to high-dimensional planning problems while achieving high perception quality in both static and dynamic scenarios.

In this work, we make the following key contributions:

- A new problem formulation for motion planning with high degree-of-freedom (DoF) robots that plans from a start to a goal configuration while optimizing perception quality in both static and dynamic environments.
- A perception score estimation pipeline that uses a neural surrogate model and ray casting to efficiently approximate perception quality while accounting for occlusions in cluttered scenes, extracting more complex and computationally expensive information than landmarks and informative regions.
- A planner that uses the neural surrogate model and a roadmap-based approach.
- A batch-processing infrastructure that accelerates collision checking, FK and perception evaluation using GPU parallelism, enabling real-time replanning.
- Extensive simulation and real-robot experiments on four different high-DoF robot systems, demonstrating that PS-PRM consistently improves both perception performance and planning efficiency compared to baseline methods.

## II. Problem Statement

We consider a robot with $k$ controllable joints and an onboard camera with a limited and known field of view. The robot's configuration space is $\mathcal{Q} \subseteq \mathbb{R}^k$. The robot is located in a workspace $\mathcal{W} \subseteq \mathbb{R}^3$ that contains objects and people of interest that need to be monitored, as well as obstacles to be avoided for safety. The robot has access to the pose and geometry of all obstacles in $\mathcal{W}$ and the positions of the centroids of objects of interest. As the robot moves, the objects or humans of interest may also move, and we assume the robot has real-time access to their poses. The subset of all robot configurations that are collision-free is denoted by $\mathcal{Q}_{free} \subseteq \mathcal{Q}$. The motion planning problem is to find a path $\sigma : [0, 1] \to \mathcal{Q}_{free}$ that assigns a collision-free configuration to each input parameter, starting from $\sigma(0) = q_{start}$ and ending in a goal region $\sigma(1) \in \mathcal{Q}_{goal}$.

We additionally define a motion cost function $c_m : \Sigma_{\mathcal{Q}} \to \mathbb{R}^+$, representing the length of a path. To consider perception quality, we define a perception score function $p_s : \Sigma_{\mathcal{Q}} \to [0, 1]$ which quantifies the performance of a perception task along $\sigma$. Based on this perception score, a perception cost is defined as $c_p = 1 - p_s$, with which we can define a composite cost

combining motion and perception quality $c(\sigma) = c_m(\sigma) + c_p(\sigma)$.

We denote the set of all paths in $\mathcal{Q}$ as $\Sigma_{\mathcal{Q}}$, and all collision-free paths as $\Sigma_{\mathcal{Q}_{\text{free}}}$. The central problem to solve is to find a collision-free path $\sigma^* : [0, 1] \to \mathcal{Q}_{\text{free}}$ that minimizes the composite cost function $c(\sigma) = c_m(\sigma) + \alpha \cdot c_p(\sigma)$, where the trade-off between motion and perception cost is governed by a scalar weight $\alpha \geq 0$. The optimal path is given by $\sigma^* = \arg\min_{\pi \in \Sigma_{\mathcal{Q}_{\text{free}}}} [c_m(\sigma) + \alpha \cdot c_p(\sigma)]$, if such a path exists.

In our problem setting, minimizing $c_m$ implies shorter, better paths, while minimizing $c_p$ is implicitly maximizing the perception score through the path. For example, in an object tracking task, the perception score may correspond to the average tracking confidence score obtained as the robot moves along the path. Our problem formulation seeks to minimize a weighted sum of motion and perception costs, aiming to find a path that is both short and achieves good perception performance.

## III. Related Work

In this section, we discuss three categories of related problems examined in this paper: informative path planning, perception-aware path planning, and active perception. While all of these approaches consider perception and motion, they address fundamentally different problem formulations compared to ours.

### A. Informative Path Planning

There have been many works on the problem of informative or information-rich path planning [30–32]. The general problem is to maximize a metric of information quality (e.g., probability of locating a target, accuracy of the inspection, or quality of the survey) while satisfying constraints on energy, time or perception. The informative motion planning problem of maximizing information gathered subject to a constraint is particularly challenging because it typically requires searching over a large and complex space of possible trajectories. Such problems have been shown to be NP-hard [33], or even PSPACE-hard [34], depending on the form of the objective function and the space of possible trajectories. Several general solvers for robotic information gathering problems utilize combinatorial optimization techniques to search over a discrete grid. The recursive greedy algorithm [33] is one example that achieves bounded performance for submodular objective functions. Branch and bound techniques have also been proposed that only require the objective function to be monotonic [35]. Another approach is to utilize a finite-horizon solver that solves the problem for only a portion of the budget at a time [36]. However, the resulting solvers typically require computation exponential in the size of the problem instance due to the large blowup in the search space. Thus, they are mostly shown with 2D environments and with very few obstacles.

Another related line of work is planning under uncertainty or planning in information spaces, which focuses on selection of trajectories that minimize the localization uncertainty. This problem has generally been solved with Partially Observable Markov Decision Processes (POMDPs) or through graph-search in the belief space [37]. While these approaches are well-established, their computational complexity grows exponentially in the number of possible actions and observations.

To overcome these issues, the use of sampling-based motion planning algorithms, such as the RRT [38] and the probabilistic roadmap (PRM) [39], has increased enormously in past years. Such algorithms have the advantage of quickly exploring a space of interest to achieve a feasible solution. Variants of sampling-based motion planning have also been proposed to maximize some metric of information quality. The prior algorithms have been shown to solve related problems for certain objective functions, including information-rich RRTs [40] using Fisher information matrices and belief-space RRTs [41].

Unlike these works, this paper specifically focuses on selecting a trajectory that balances high perception quality with short motion length. Furthermore, these prior algorithms often focus on 2D environments [31, 32, 40, 41] and their information metrics do not apply to the perception task we consider in this work where the approximate analytical model is infeasible.

### B. Perception-aware Path Planning

Perception-aware path planning has been studied extensively for Unmanned Aerial Vehicles (UAVs) and Micro Air Vehicles (MAVs) because a perception-aware component is especially important when flying in dynamic environments. Furthermore, a consistent detection of the moving obstacles is necessary to obtain a good estimate of their locations and prediction of their future trajectories.

Specifically, our work relates to recent work on optimizing motion and perception tasks for UAVs and MAVs in applications such as quadrotor flying [16–19], under dynamic environments [12–15] and for scenes with semantic classes [20]. In most cases, the perception task aims at maintaining visibility of landmarks and informative regions for visual odometry [20], reducing the velocity of the projection of points of interest on the camera plane to avoid blurred motion [16, 42] or guaranteeing the visibility of obstacles for safety [12, 15, 43]. Further relevant work addresses the problem of having a UAV record or chase a target [44–47] For example, [46] focused on tracking a moving target with a downward-facing camera, while [44] proposed a way to follow a moving target while avoiding other static obstacles in the environment.

For UAVs, most methods formulate the problem as optimization using Model Predictive Control (MPC) [12, 15, 16, 43] or Reinforcement Learning (RL) [13, 18, 19, 48]. MPC-based methods jointly optimize perception and motion for agile UAV navigation using numerical optimization, without obstacles [16] or for static obstacles and occlusions [43]. For example, the work [20] steer the UAV towards informative areas using a path planner while keeping landmarks in the camera field of view through optimization. RL-based methods create policies that directly map from sensor measurements to controls while keeping visibility of regions of interest. The policy learner has access to privileged information during training, *e.g.,* by using

Table I
Comparison with active perception and perception-aware path planning. **Fixed Goal**: task specifies a fixed goal configuration. **Cont. Perc.**: perception constraints enforced along the whole trajectory, not just at waypoints. **General Perc.**: supports general perception objectives beyond a single task category. **High-DoF**: applicable to high-degree-of-freedom robots.

| Method | Fixed Goal | Cont. Perc. | General Perc. | High-DoF |
|---|---|---|---|---|
| Ours | ✓ | ✓ | ✓ | ✓ |
| Active Perception | ✗ | ✗ | ✓ | ✓ |
| Perception-aware Planning | ✓ | ✓ | ✗ | ✗ |

MPC [49], a planner [18] or a state-based trained expert [19]. Another line of work focuses on using the UAV or MAV for autonomous aerial cinematography or low altitude traffic surveillance that can reason about both geometry and scene context in real-time [50, 51].

Unlike these works, we consider high-DoF systems (e.g., mobile robots with an onboard camera controlled by multiple joints) with kinematic constraints on environments with many obstacles, where optimization is likely to get stuck in a local minima. As such, we focus on planning simultaneously for efficient motion and perception quality on the high-dimensional space of robots such as mobile manipulators instead of UAVs or MAVs. Furthermore, unlike most of the perception-aware planning works that aim at maintaining visibility of landmarks and informative regions for visual odometry, we address more complex perception tasks which are significantly more computationally expensive to evaluate.

### C. Active Perception

Also referred as view planning, active perception considers applications such as object [52] and scene [53] reconstruction, object recognition [54], pose estimation [55], and active mapping [24], with single robot or multiple robots [56]. In these tasks, perception is typically incorporated into the path planning process by selecting optimal viewpoints to maximize the performance of the specific problem [21–23]. One of the goals of active perception is active localization, which seeks to compute control actions and trajectories that minimize the pose estimation uncertainty. Most active localization works have been in the context of simultaneous localization and mapping (SLAM) or exploration [24, 29]. Depending on the sensor used, they can be classified into range-based (e.g. RGB-D cameras) [57–59] or vision-based [25–29].

Viewpoint planning or next best view planning (NBV) is a subset of active perception where the sensor pose or a sequence of poses is planned to maximize the information gain, i.e., minimize the entropy or uncertainty about the state of the environment or target objects, subject to constraints such as obstacle avoidance and movement cost. The challenge of next best view planning is proposing and selecting views from which a scene can be efficiently observed. The quality of an observation can be quantified by its accuracy (i.e., how closely the captured data resembles the actual scene) and completeness (i.e., what proportion of the scene has been observed). Observation accuracy primarily depends on the sensor capabilities but can be improved by considering the scene texture and geometry. The completeness of an observation is determined by the coverage obtained from captured views. The efficiency of a NBV approach is quantified by the time and travel distance required to obtain an observation. For detail oriented tasks, especially at the object level, such as active recognition, pose estimation [60], mapping or reconstruction [61, 62], manipulators or mobile manipulators are typically used with attention-driven next best view planning.

Recently, learning-based active perception [63–67] has become popular across robotics. These approaches primarily formulate perception as an information-gathering problem, finding viewpoints that reduce uncertainty in mapping, object state estimation, or task-relevant observations. For example, [63] formulates damage inspection as a POMDP and uses deep reinforcement learning to select next sensing poses that maximize crack-segmentation quality during underwater structural inspection. The authors in [64] propose an asynchronous active vision–action framework that serially trains a next-best-view camera policy and a next-best-pose gripper policy for manipulation tasks under occlusion, but the perception policy is still defined in a low-dimensional camera-space rather than full robot configuration space. Several mapping-driven approaches [65, 66] focus on mobile robots or aerial platforms, where active perception guides viewpoint selection to improve scene reconstruction fidelity. [65] uses confidence-driven next-best-views to actively densify Gaussian-splatting maps, while [66] develops a real-time navigation pipeline in the maps that couples safe corridor planning with fast vision-based localization. Broader surveys such as [67] further emphasize how vision-based learning in drones centers around improving perception for navigation, mapping, and control, typically in 3-DoF mobile systems.

Unlike learning-based active perception methods which choose perception-oriented intermediate goals or viewpoints, our planner operates with a fixed motion planning goal. In addition, active perception is largely decoupled from whole-body motion planning and focuses on viewpoint selection for mapping or manipulation in low-dimensional action spaces. This is different from our objective of considering perception constraints directly within high-DoF motion planning, which requires sensing quality, occlusion reasoning, and task performance to be jointly evaluated in the robot's full configuration space. In Table I, we summarize the different objectives of our method, active perception and perception-aware path planning.

### D. Implicit Neural Representations/Neural Surrogate Model

A surrogate model is an engineering approach employed when directly measuring or computing an outcome of interest is either infeasible or prohibitively expensive. In such cases, an approximate model of that outcome is used instead [68]. With the rise of deep learning, neural or deep learning-based surrogate models have been applied in numerous domains, including earth science [69], image processing [70], fluid dynamics [71], and power grids [72]. In robotics and perception, recent advances in machine learning have produced effective implicit neural representations of spatial information, such as Neural Radiance Fields (NeRFs) [73] and Signed Distance Fields [74]. These representations have been used for tasks ranging from learning multi-object dynamics [75] and defining

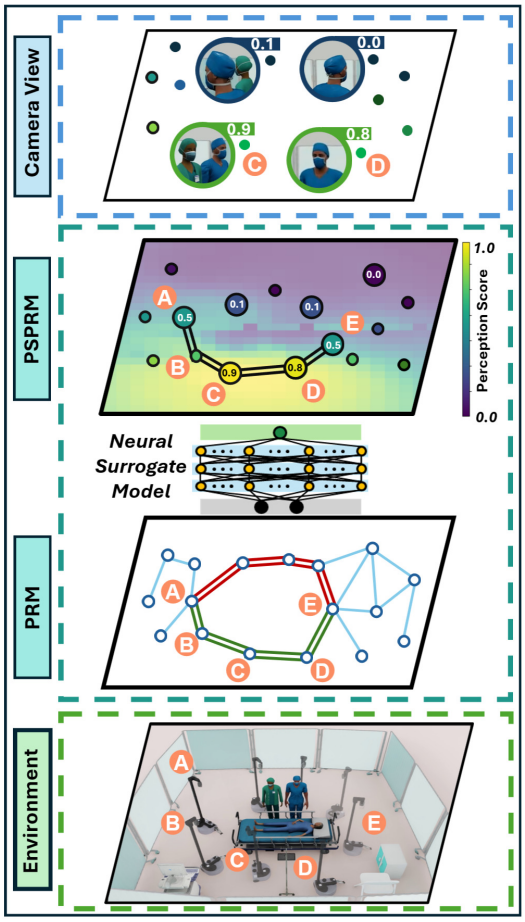



Figure 2. The PS-PRM pipeline. **Bottom:** environment with candidate configurations. **Middle:** PRM generation, neural surrogate scoring, and occlusion checking run in parallel to select the optimal path. **Top:** camera views confirming differing perception scores.

manipulation planning constraints [76], to enabling reactive robot manipulation [77] and vision-based navigation [78]. They offer compact, efficient storage [78], a continuous geometric representation [76–78], and can be learned directly from sensor data [79], making them particularly appealing for planning. Moreover, recent years have seen neural representation models trained on broad datasets demonstrate impressive potential for generalizable manipulation [80–83]. For instance, neural object descriptors (NODs) [81] have emerged as a powerful method for extracting dense, part-level features that generalize across object instances. Building on these directions, our work trains a neural model to map robot configurations directly to the desired perception metric (e.g., object detection confidence) as it is infeasible and very expensive to compute the perception score for different robot configurations in the real world.

## IV. Overview of PS-PRM

We propose PS-PRM, a perception-constrained roadmap-based planning framework for high-DoF robots that jointly optimizes motion feasibility and task-dependent perception quality. PS-PRM is designed to operate in both static and dynamic environments, enabling robots to maintain high-quality perception of task-relevant objects or humans while performing navigation or manipulation tasks.

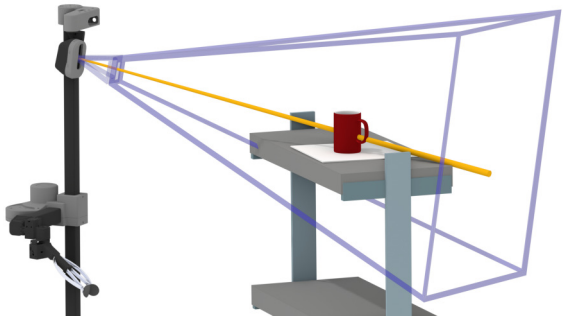


Figure 3. The camera model: line of sight (orange) and camera frustum (purple), shown against a cup intersecting each.

Fig. 2 illustrates the overall PS-PRM pipeline, including roadmap construction, perception score estimation, and path selection based on combined motion and perception costs. At a high level, PS-PRM constructs a PRM over a manifold defined by object-centered visibility constraints, such as line of sight or camera frustum alignment. Each configuration in the roadmap is scored based on its estimated perception quality, using a learned neural surrogate model. To handle cluttered scenes and inter-object occlusions, we further refine the perception score using a ray-casting-based occlusion checking module. Finally, PS-PRM selects the path that maximizes perception quality while minimizing motion cost. We choose PRM over tree-based planners in this work because it simplifies the underlying optimization problem and allows for more efficient node sampling and evaluation in batch, which can be efficiently parallelized on the GPU.

In dynamic environments, where objects or people may move, PS-PRM supports rapid replanning through a GPU-accelerated pipeline that updates perception scores, re-evaluates roadmap validity, and generates new plans in real time. The system is designed to operate efficiently at scale by batching key operations such as forward kinematics, surrogate model inference, and collision checking in environments where radical changes are not expected.

The following sections describe the three main components of PS-PRM in detail. Sec. IV-A introduces the neural surrogate model used to estimate perception scores. Sec. IV-B describes the construction of the perception-score-guided planner. Sec. IV-C presents the GPU-parallelized pipeline.

### A. Neural Surrogate Perception Score Models

In real-world environments, it is infeasible to directly evaluate perception scores for all robot configurations due to the time and effort required to physically reposition the robot and collect sensor data at scale. As a result, simulation offers a practical alternative: scoring each configuration by rendering the robot's camera view and passing it through a perception model. However, this approach is still computationally intensive. Each evaluation involves placing the robot in the simulated scene, generating a photorealistic rendering, and running the task-specific perception model to obtain a score. As demonstrated in our experiments, rendering high-quality images is time-consuming, especially when evaluating a large number of robot configurations. Moreover, perception scores

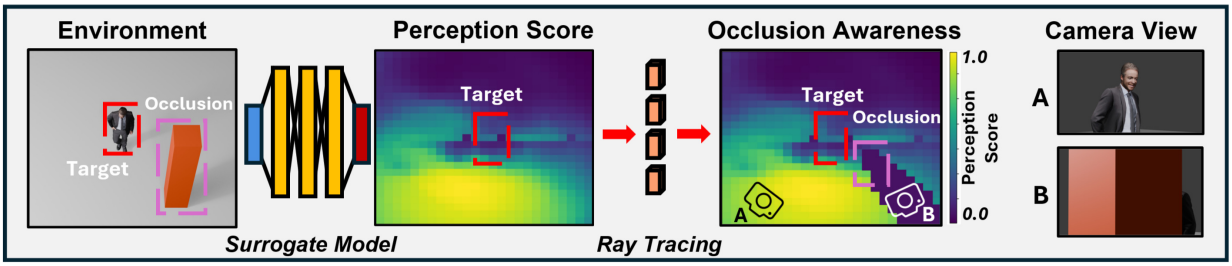



Figure 4. Perception-aware evaluation via neural surrogate model and occlusion-aware ray tracing. Left to right: environment with target (human) and occluder (orange box); surrogate-predicted perception score field; ray-traced occlusion refinement; camera views at points A and B, showing lower scores under occlusion.

are environment-dependent; any change in the object requires re-scoring previously scored configurations.

To address these challenges, we propose a neural surrogate model inspired by [84, 85] that approximates perception scores without requiring explicit rendering. Alg. 2 outlines the full pipeline. Given a robot configuration and the target object label, the surrogate model predicts the expected perception performance (e.g., detection confidence for a cup). We first apply FK to map the configuration from $\mathcal{Q}$ to the robot's camera pose. The neural model is then trained to map these camera transforms to perception scores obtained from labeled, photorealistic simulations. Specifically, we generate a dataset by rendering views of objects or humans from varied camera poses in simulation and evaluating each image with a perception model. The surrogate is implemented as a multilayer perceptron (MLP) [86] that takes the relative pose between camera and object, along with an object encoding, and outputs the corresponding perception score. Trained on this dataset, the surrogate enables fast and scalable evaluation during planning. Importantly, because the input is based on camera-to-object pose rather than robot-specific joint values, the model generalizes across robots and environments with similar perception setups. This robot-agnostic formulation is central to our approach: by operating in SE(3) camera space rather than robot configuration space, the surrogate can be reused across different platforms without retraining.

To address object inter-occlusion in cluttered indoor environments, we build upon techniques similar to those in [87, 88] by performing ray-casting towards surrounding objects. This step determines whether the object of interest is occluded by checking for intersections between each cast ray and the Axis-Aligned Bounding Boxes (AABB) of nearby objects. For a new robot configuration, we obtain an occlusion score in addition to the perception score estimated by the neural surrogate model. This occlusion score is computed as the ratio of rays that ultimately do not intersect the bounding box of the target object, an indicator that something else is blocking the view. If the occlusion score exceeds a chosen threshold, we set the perception score of the neural surrogate model to zero for that configuration. Fig. 4 shows a detailed overview of this occlusion checking step.

Fig. 5 illustrates the overall neural surrogate perception score model. This two-tier approach ensures that the neural surrogate model also accounts for occlusion in an efficient, fast manner, since ray-casting against bounding boxes can be highly parallelized. In practice, relying on bounding boxes provides a reasonable approximation of object geometry, though it may not perfectly capture complex shapes or partial occlusions. Nevertheless, for many motion-planning tasks in cluttered environments, it strikes a good balance between speed and accuracy.

### B. Perception Score-Guided Planner

To enable robots to maintain a high score for a given perception task throughout their motion, the PS-PRM planning process consists of two key stages: (1) sampling configurations that satisfy visibility constraints, and (2) constructing, scoring, and planning over a PRM built from these configurations. The first stage ensures that each sampled configuration allows the robot to keep the target object or person within view. The second stage integrates perception quality scores into the roadmap and performs graph-based planning to find a path that balances motion efficiency and perception quality. We describe each stage in detail below.

*1) Sampling with Visibility Constraints:* To maximize perception scores along the planned motion, the robot's onboard camera must continuously keep the object or person of interest in sight. In this work, we consider two commonly used visibility constraints for maintaining such monitoring, as illustrated in Fig. 3: line of sight and camera frustum. In the line-of-sight model, the robot's optical axis is aligned with the centroid of the target, effectively keeping the target centered on the image plane (*e.g.*, using a pinhole camera model) [12, 16]. In the camera frustum model, the objective is to keep the target within the camera's viewing volume, represented as a rectangular pyramid or frustum [89]. These visibility constraints define a manifold within the robot's free configuration space consisting of configurations where the target remains in view.

More formally, we define the set of all robot configurations that enable the robot to monitor the object or person of interest to lie in a manifold $M$ implicitly represented as the zero-level set defined by $h_M(q) = 0$ where $q \in \mathcal{Q}$, $h_M(q) : \mathbb{R}^k \to \mathbb{R}^l$, and $l$ is the intrinsic dimensionality of the constraint ($l \leq k$). For the line-of-sight model, $h_M(q)$ is the distance between the centroid of the object and the line of sight from the robot's camera. For the camera frustum model, we model $h_M(q)$ as $\max(0,\ \mathrm{sd}_f(q, p))$ where $\mathrm{sd}_f$ is the signed distance between the object $p$ and the surface of the robot's camera frustum when the robot is at configuration $q$.

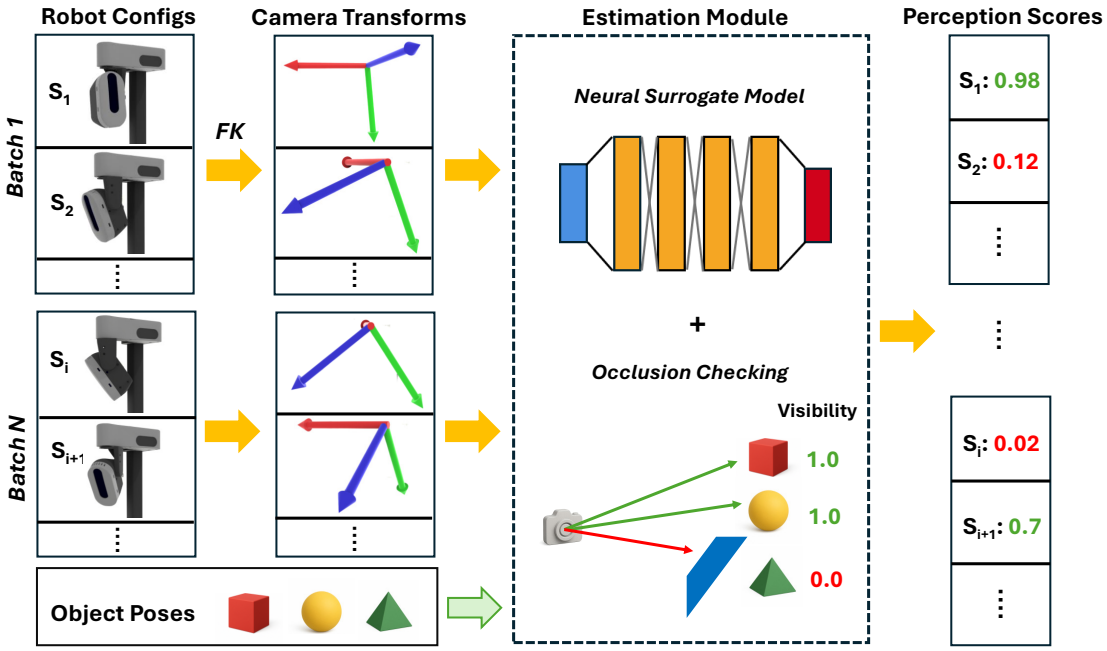



Figure 5. Parallel perception-score evaluation for a batch of robot configurations. Camera poses are computed via FK; occlusion checking and neural surrogate scoring run concurrently; outputs combine into a final perception score (0–1) per configuration.

For high-DoF robots, computing motion that considers the aforementioned object-centered models can be achieved using sampling-based motion planning (SBMP) [90] with manifold constraints [91] where planning happens on a subspace of the free configuration space defined by a manifold. Specifically, in the construction of our proposed planner shown in Alg. 1, PS-PRM first constructs a PRM with states and transitions that satisfy the manifold constraints, using either line-of-sight or camera frustum. The PRM is defined $G = (V, E)$, where $V$ is the set of configurations and $E$ the set of edges connecting them. The neighbor nodes in the PRM are constrained to lie on the same manifold and are selected based on their geodesic proximity within a certain distance. To sample from the manifold, PS-PRM first selects a robot configuration from $\mathcal{Q}_{free}$ and then applies the projection operator to map it onto the surface of $M$. This involves iteratively retracting the point towards a minimum of the constraint function and solving a linear system of equations at each step. Configurations that cannot be projected into valid in-manifold configurations are discarded, and the algorithm resamples until a valid configuration is found or a timeout is reached. This process ensures that the object is placed either within the line of sight or inside the camera frustum.

While enforcing visibility constraints helps keep the target object or person within the robot's sight, this alone is not sufficient for many real-world perception tasks. Our experiments show that visibility-based approximations often fall short in practice as they lack semantic and geometrical information specific to the objects and the task. This affects in complex environments where factors such as object shape (e.g., human faces looking clearer closer and in front of the person), occlusions, background clutter, and lighting conditions significantly affect perception performance [1–3]. To address these limitations, our proposed planner incorporates task-aware perception scores to guide planning toward configurations that support more reliable perception.

*2) PRM construction and scoring:* While the visibility constraints provide a bound on necessary conditions for the perception task, they do not consider the actual performance of the task. To remedy this, PS-PRM expands the Probabilistic Roadmap planner (PRM) [39], by incorporating perception scores into the graph and using them during planning.

After PS-PRM builds the manifold-constrained PRM, the edges initially encode only the distance between configuration nodes under a given metric (e.g., Euclidean distance from state to state). This is used as the motion cost, normalized to facilitate its later combination with the perception cost. PS-PRM then assigns perception scores to the edges to account for the perception task. Since each perception score is calculated discretely for each state, thus ignoring the continuous path taken from node to node, PS-PRM also uses interpolated intermediate configurations to compute a normalized perception score for an edge, $c_p(q_u, q_v) \in [0, 1]$. PS-PRM performs robot-space dependent interpolation between the $q_u$ and $q_v$ to generate $N$ valid intermediate configurations, $q_1, q_2, \ldots, q_N$. It then calculates the perception scores corresponding to these intermediate configurationsand computes their normalized perception score $c_p(q_1), c_p(q_2), \ldots, c_p(q_N)$. Next, PS-PRM calculates the average normalized perception score of the two configurations connected by an edge and their intermediate configurations to obtain the normalized perception score associated with the edge $c_p(q_u, q_v)$:

$$c_p(q_u, q_v) = \frac{c_p(q_u) + c_p(q_v) + \sum_{i=1}^{N} c_p(q_i)}{N + 2} \quad (1)$$

Finally, PS-PRM linearly combines the normalized perception score and normalized motion cost for an edge to determine

a new edge weight. While other nonlinear combinations are explored, the linear formulation offered a good balance between interpretability and empirical performance. Since the perception score is intended to be maximized while the motion cost is intended to be minimized, we subtract the normalized perception score and add a constant 1 to ensure that the edge weight remains non-negative, within the range $[0, 2]$. The combination is defined as follows, where $\alpha \in [0, 1]$ is a factor that balances the perception score and motion cost:

$$c(q_u, q_v) = 1 - \alpha \cdot c_p(q_u, q_v) + (1 - \alpha) \cdot c_m(q_u, q_v) \quad (2)$$

After the PRM is constructed, PS-PRM connects the start and goal to the roadmap and applies an A* search [92] with an admissible heuristic to find a path from $q_{start}$ to $\mathcal{Q}_{goal}$ for the robot to follow.

Next, to guarantee the optimality of the A* search, we show that our A* heuristic remains admissible under the modified edge-cost formulation. Since consistency implies admissibility [93], it is sufficient to prove that our heuristic is consistent. We define our A* heuristic for a PRM $G = (V, E)$ as a "hop"-based function that lower-bounds the remaining motion and perception costs:

$$h(q) \;=\; H_{\min} \cdot c^{\min}, \quad (3)$$

where $H_{\min}(q)$ denotes the hop distance, i.e., the minimum number of edges required to reach the goal from the configuration $q$ obtained via a shortest-path search, and $c^{\min}$ represents the minimum edge cost over the entire roadmap:

$$c^{\min} = \min_{(u,v)\in E} c(q_u, q_v). \quad (4)$$

For any edge $(u, v) \in E$, a feasible path from $u$ to the goal is to first take $(u, v)$ and then follow a shortest path from $v$ to the goal. Thus, we have:

$$H_{\min}(q_u) \;\le\; 1 + H_{\min}(q_v). \quad (5)$$

Therefore, our heuristic function $h(\cdot)$ satisfies:

$$\begin{aligned} h(q_u) &= H_{\min}(q_u) \cdot c^{\min} && \text{Eq. (3)} \\ &\le (1 + H_{\min}(q_v)) \cdot c^{\min} && \text{Eq. (5)} \\ &\le c(q_u, q_v) + h(q_v) && \text{Eqs. (3) and (4)} \end{aligned}$$

which is exactly the condition of a consistent heuristic. Thus, we show that our A* heuristic remains consistent and admissible under the modified edge cost formulation.

This formulation only requires nonnegativity of edge costs, without assuming a specific form such as Euclidean distance for motion cost. It applies broadly, e.g., when $c_m$ is defined as the trajectory length, energy, or control effort, and $c_p$ is a non-negative normalized neural perception score. In this paper, we demonstrate experiments on object detection and tracking, as well as human face detection and activity recognition, though the approach can be generalized to other perception tasks.

Since our problem is formulated as a sampling-based motion planning problem, probabilistic completeness [93] is an important property. We therefore show that probabilistic completeness still holds after introducing our modified edge-cost formulation. Specifically, we show that our perception-weighted PRM formulation does not invalidate the standard PRM completeness. This holds because the reweighting affects only the edge costs, not the set of feasible configurations or connections. Assume a PRM constructs a graph $G = (V, E)$ over samples from the collision-free configuration space $Q_{\text{free}}$. The PRM completeness proof on Theorem 7.4.1 from [93], does not depend on edge cost function, only on samples and connections. Let $\sigma$ denote any feasible path with clearance $\delta > 0$. With a probability exponentially increasing with the number of sampled states, PRM will include vertices $\{q_1, \ldots, q_n\}$ such that consecutive vertices are within a local planner's connection radius and $\sigma = (q_1, \ldots, q_n)$ will be found. Since our composite edge cost $c(u, v)$ shown in Eq. (2) is nonnegative and finite for all $(u, v) \in E$, the reweighting does not eliminate any edges or feasible connections. Consequently, PS-PRM remains probabilistically complete: if a feasible path exists in $Q_{\text{free}}$, the probability that PS-PRM finds a finite-cost path approaches one as the number of samples increases.

**Algorithm 1:** Perception Score-Guided PRM Construction (PS-PRM)

**Input:** Environment $\mathcal{E}$, Robot Model $\mathcal{R}$, Object of Interest $\mathcal{O}$
**Output:** Roadmap $\mathcal{G} = (V, E)$ with perception-aware edge weights
Initialize empty graph $\mathcal{G} = (V, E)$
$N \leftarrow$ number of samples
**for** $i = 1$ to $N$ **do**
  Sample $q_i$ in free space $\mathcal{C}_{\text{free}}$
  Project $q_i$ onto the manifold using constraint projection
  **if** $h_M(q_i)$ is less than a threshold **then**
    Add $q_i$ to $V$
  **end if**
**end for**
**for** each $q_u \in V$ **do**
  **for** each $q_v$ within geodesic distance $\delta$ **do**
    Normalize the original distance-based edge weight between $q_u$ and $q_v$ to obtain the motion cost $c_m(q_u, q_v)$.
    Generate $N$ intermediate configurations $q_1, \ldots, q_N$
    Compute normalized perception score $c_p(q_u, q_v)$ using Eq. (1)
    Compute edge weight $c(q_u, q_v)$ using Eq. (2)
    Add edge $(q_u, q_v)$ to $E$ with weight $c(q_u, q_v)$
  **end for**
**end for**
**return** $\mathcal{G}$

### C. GPU-Parallelized Replanning Pipeline

The proposed perception scoring methods are highly parallelizable. Since the perception scores are computed using a neural surrogate model specifically optimized for GPU inference, we are motivated to implement the entire planning pipeline on the GPU. For collision checking, PS-PRM evaluates the validity of sampled configurations with hardware acceleration: instead of checking each obstacle individually, the algorithm performs batched collision checking on all obstacles simultaneously. Furthermore, both FK and surrogate model inference are performed in batches and executed in

**Algorithm 2:** Neural Surrogate Model for Perception Score Estimation

**Input:** Robot Configuration $q$, Object $\mathcal{O}$
**Output:** Perception Score $c_p(q)$
Compute camera transform $T_c$ using forward kinematics
Query neural surrogate model: $c_p(q) \leftarrow \text{NN}(T_c, \mathcal{O})$
Perform ray-casting to check occlusions
**if** occlusion score $> \tau$ **then**
    Set $c_p(q) \leftarrow 0$
**end if**
**return** $c_p(q)$

the GPU. This node-level parallelization accelerates roadmap construction and perception-scoring and enables real-time replanning in high-dimensional spaces. PS-PRM can therefore operate effectively in dynamic environments where the object of interest, goals or obstacles may move during execution. In such cases, the planner updates node perception scores in real time and replans the path as needed to adapt to the new environment. Fig. 5 shows a detailed overview of the GPU-accelerated process.

As the robot executes its planned trajectory, it continuously monitors whether the object has moved. If movement is detected (*e.g.*, by the onboard camera), PS-PRM invokes the parallelized pipeline to re-project the existing nodes and update their perception scores.

Nodes that cannot be projected into valid configurations or that collide in the new environment are pruned from the roadmap along with their associated edges. After pruning and re-scoring, the planner re-evaluates the current trajectory by computing a new average perception score; if that score falls below a specified threshold, it triggers a replanning step to find a more suitable path from the robot's current position to the goal region. If the remaining nodes fail to produce a safe path to the goal, PS-PRM partially rebuilds the roadmap by sampling additional nodes while retaining any valid ones, thus avoiding a complete reconstruction. This leverages each step based on their computational needs, while building a PRM can be performed in seconds, re-evaluating nodes' collisions and scores can be done at higher frequencies of several times a second. Through this GPU-accelerated parallelization, PS-PRM rapidly adapts to dynamic changes in the environment and maintains both collision-free and perception-optimized trajectories with reduced computation time.

In this work, we primarily consider small to medium target or obstacle motions, where changes in the environment are gradual relative to the planner's replanning frequency. Under these conditions, the proposed planner can update trajectories online by replanning fast enough to adapt to dynamic targets, obstacles, and goals. Specifically, Sec. V-F demonstrates the feasibility of our system. Large or abrupt target motions are not the primary focus of this work. When the target dynamics exceed the planner's replanning capability (e.g., rapid, discontinuous, or adversarial movements), the problem becomes fundamentally one of high-frequency tracking or reactive control rather than perception-constrained motion planning, which is outside the scope of this paper.

## V. Experimental Results

In this section, we evaluate our proposed methods in simulation and real-robot experiments using the Hello Robot's Stretch 2 [94], Fetch robot, and a UR5. We use PyBullet [95] for collision checking, OMPL's Python bindings [96] for planning, and Isaac Sim [97] for simulation and visualization. All experiments were conducted on an Intel i7-12700K CPU and an NVIDIA GeForce RTX4090 GPU. We consider both object detection/tracking, human face detection and activity recognition as perception tasks. For the robot's camera parameters in simulation experiments, we use a resolution of 640 $\times$ 480 pixels, a field of view of 42.5 degrees, and a clipping range of 0.3 meters for the near plane and 10 meters for the far plane.

### A. Evaluated Methods

We implement four baseline methods as follows:

- **Baseline 1 Post:** This baseline method initially performs standard motion planning without accounting for any constraints, and then post-processes the robot's free joints (*e.g.,* camera pan and tilt) along the resulting path to align the robot's line of sight with the centroid of the object or person as described in details in Sec. IV-B1. This is achieved through a simple geometric analysis given the position and orientation of the robot at each configuration with respect to the monitoring object or person.
- **Baseline 2 Rejection:** This baseline method extends a standard sampling-based planner by modifying its feasibility function to ensure that sampled configurations remain within the robot's camera frustum.
- **Baseline 3 Manifold-line:** This baseline method is an implementation of a manifold constrained motion planner using the distance between the centroid of the object and the line of sight from the robot's camera's as $h_M(q)$ as described in details in Sec. IV-B1.
- **Baseline 4 Manifold-frustum:** This baseline method is an implementation of the manifold constrained motion planner described in Sec. IV-B1 using $h_M(q) = \max(0, \text{sd}_f(q, p))$ where $\text{sd}_f$ is the signed distance between the object $p$ and the surface of the robot's camera frustum when the robot is at configuration $q$.
- **Baseline 5 Visibility-Aware TrajOpt:** This method draws elements from related trajectory optimization work [98], modeling the problem as a minimization of costs to reach the goal configuration, generate a smooth and kinematically feasible path and keep the object within view of the camera. Additional constraints are used to generate a non-holonomic trajectory for the mobile base and to ensure collision-free paths.
- **Baseline 6 RL-Based Goal-Conditioned Active Perception:** This method formulates perception-aware navigation problem as a goal-conditioned active perception problem and learns a policy via Proximal Policy Optimization (PPO) [99]. This baseline is implemented based on elements from the works [63, 64, 100, 101], where an agent jointly optimizes mobility and observability of a target object, and by standard goal-conditioned navigation [102].

Table II

Planning and perception results for the Stretch robot in a simulated home environment, averaged over 2,500 planning problems. "Preprocess/Plan./Total Time" are pre-planning, planning, and combined durations; "Path Len." and "Succ. Rate" are motion planning metrics; "Det. Rate," "Conf. Score," and "Track Rate" are object-tracking metrics.

| Method | Motion Planning | | | | | Object Tracking | | |
|---|---|---|---|---|---|---|---|---|
| | Preprocess Time (s) | Plan. Time (s) | Total Time (s) | Path Len. (rad) | Succ. Rate | Det. Rate | Conf. Score | Track Rate |
| Post | - | 0.051 | 0.051 | 6.277 | 99.7% | 20.9% | 0.531 | 57.9% |
| Rejection | - | 1.264 | 1.264 | 10.379 | 98.9% | 29.7% | 0.599 | 74.0% |
| Manifold-frustum | - | 0.495 | 0.495 | 11.141 | 98.8% | 41.0% | 0.597 | 80.9% |
| Manifold-line | - | 2.859 | 2.859 | 7.478 | 90.1% | 48.9% | 0.482 | 67.4% |
| Visibility-Aware TrajOpt | - | 0.152 | 0.152 | 5.62 | 93.0% | 35.7% | 0.561 | 64.4% |
| RL-Based Active Perception | 169.8 | 0.404 | 170.204 | 21.75 | 57.3% | 75.5% | 0.634 | 80.0% |
| PS-PRM-frustum | 291.7 | 0.014 | 291.714 | 16.688 | 96.4% | 73.7% | 0.761 | 89.8% |
| PS-PRM-line | 341.5 | 0.018 | 341.518 | 15.007 | 95.7% | **80.1%** | **0.792** | **91.0%** |
| Neural-PS-PRM-frustum | 32.08 | 0.017 | 32.097 | 15.814 | 97.1% | 72.8% | 0.759 | 89.8% |
| Neural-PS-PRM-line | 31.79 | 0.021 | 31.811 | 13.518 | 96.3% | 79.6% | 0.771 | 90.4% |

The first four baseline methods are tree-based planners derived from RRT-Connect, and therefore they do not require preprocessing time.

The fifth baseline is a Visibility-Aware Trajectory Optimization planner, based on a drone-oriented related world [98]. Trajectories are modeled as a set of discrete points over which costs are calculated, accounting for joint position and velocity constraints, collision avoidance and angle of vision via forward kinematics. The optimization is modeled as a least squares problem and solved through Levenberg-Marquardt.

For the RL-Based Goal-Conditioned Active Percpetion baseline, the agent observes the current robot configuration, goal configuration, monitored object position, and episode progress, and outputs continuous configuration-space actions. Invalid actions are rejected using OMPL collision checking and penalized during training. The reward combines perception quality, motion cost, goal progress, collision penalties, and sparse goal completion rewards, encouraging the agent to jointly optimize navigation efficiency and target observability. Perception quality is approximated using a geometry-based proxy derived from forward kinematics, adapted from the objective function used in the [63]. Similar to the works [63, 64], the policy is parameterized by a shared actor-critic MLP with layers $[256, 256, 128]$ and trained using PPO with learning rate $3\times10^{-4}$, rollout length 512, batch size 64, and $10^5$ environment interaction steps. Training is performed on randomized start and goal configurations sampled from the environment, and the learned policy is evaluated deterministically for up to 200 steps per query.

We implement four versions of our approach PS-PRM, as described in Sec. IV, varying along two key dimensions: the perception model (line-of-sight vs. camera frustum) and whether a neural surrogate is used for scoring.

- **PS-PRM-line:** Uses a line-of-sight model to estimate perception quality, with direct evaluation (no neural surrogate).
- **PS-PRM-frustum:** Uses a camera frustum model to estimate perception quality, with direct evaluation (no neural surrogate).
- **Neural-PS-PRM-line:** Uses a line-of-sight model, with perception quality predicted by a neural surrogate model.
- **Neural-PS-PRM-frustum:** Uses a camera frustum model, with perception quality predicted by a neural surrogate model.

Table III

In-distribution benchmark and out-of-distribution robustness of the neural surrogate model: estimation error (RMSE, MAE), ranking (Spearman's $\rho$), and detection classification (Accuracy, Precision, Recall, F1), under standard evaluation plus two OOD conditions (occlusion, dim lighting).

| Method | Estimation Error | | Ranking | Detection Classification | | | |
|---|---|---|---|---|---|---|---|
| | RMSE ↓ | MAE ↓ | Spearman $\rho$ ↑ | Acc. ↑ | Prec. ↑ | Rec. ↑ | F1 ↑ |
| **Surrogate Model** | 0.154 | 0.085 | 0.641 | 93.1% | 74.4% | 65.5% | 69.7% |
| *Out-of-Distribution Robustness Analysis* | | | | | | | |
| Occluded (11.8% of valid configs) | 0.258 | 0.151 | — | — | — | — | — |
| Dim Lighting ($L/L_{\text{ref}} = 0.75$) | 0.049 | 0.034 | 0.641 | — | — | — | — |
| Dim Lighting ($L/L_{\text{ref}} = 0.50$) | 0.105 | 0.073 | 0.641 | — | — | — | — |
| Dim Lighting ($L/L_{\text{ref}} = 0.25$) | 0.175 | 0.122 | 0.641 | — | — | — | — |

For both **PS-PRM-line** and **PS-PRM-frustum**, we select $N = 5$ and $\alpha = 0.75$ in our experiments to strike a balance between action cost and perception performance. It is important to note that our planner is not highly sensitive to these parameters as long as they remain within a reasonable range. Through experimentation, we found that small variations in $N$ and $\alpha$ do not significantly impact the overall performance.

For both **Neural-PS-PRM-line** and **Neural-PS-PRM-frustum**, we set the number of samples to $N = 5$ and the weight parameter to $\alpha = 0.8$ in our experiments, aiming to balance the trade-off between action cost and perception performance. To support this framework, we train a neural surrogate model that predicts perception scores for the relevant tasks.

Specifically, in our experiments, we use the object detection confidence score as the perception score for object detection and tracking tasks, as described in Sec. V-C, Sec. V-G, and Sec. V-H. For human tracking tasks, such as those discussed in Sec. V-D and Sec. V-F, we use the confidence score of the human face detector RetinaFace [103] from Deepface [104] as the perception score.

The neural surrogate model is implemented as a Multilayer Perceptron (MLP) [86], consisting of a shared architecture with five fully connected layers, each with 256 hidden units and ReLU (Rectified Linear Unit) activation. The input to the network includes the relative pose (comprising both position and orientation) between the camera and the object, as well as an object encoding that indicates which object the robot is monitoring. We use a one-hot encoding to represent the object being monitored. The network outputs the predicted

perception score corresponding to that camera transform and object pose.

This model is trained on a dataset composed of a diverse set of commonly encountered objects and humans from the COCO dataset [105], enabling generalization across a wide range of realistic detection scenarios. We generate a dataset of perception scores as the objects or human are viewed from different camera transform in Issac Sim. Specifically, we sample a set of 50000 camera transforms where the object or human are in the field of view uniformly at random. For each camera transform, we render the images and evaluate the perceptions scores using a perception model with the relevant perception tasks. Then, we train our model on the collected dataset for 200 epochs with an Adam [106] optimizer, learning rate of $1 \times 10^{-4}$, and batch size of 512.

For the dynamic experiments in Sec. V-F, all evaluations are conducted using the GPU implementation, as the CPU-based version is not able to run and replan at a rate for dynamic scenarios. This highlights the necessity of fast replanning when targets, obstacles, or goals are moving, and demonstrates the practical advantage of GPU parallelism in enabling real-time perception-constrained planning.

### B. Perception Scoring Benchmark

We evaluate the neural surrogate model from three complementary perspectives: confidence estimation accuracy, ranking consistency, and binary detection performance. Estimation quality is quantified using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) between the predicted confidence scores and the YOLO-based ground truth, capturing absolute prediction error. Since the planner primarily depends on relative perception quality rather than exact confidence values, we additionally evaluate ranking consistency using Spearman's rank correlation coefficient $\rho$, which measures how well the model preserves the ordering of camera poses according to perceptual quality. Finally, we assess binary perception score prediction using standard classification metrics (accuracy, precision, recall, and F1 score) with a confidence threshold of $0.75$.

As shown in Table III, the surrogate model achieves low estimation error, indicating accurate confidence prediction, while also maintaining a strong positive rank correlation, demonstrating its ability to preserve perceptual ordering across camera viewpoints. In addition, the high classification accuracy and balanced precision–recall performance indicate that the model reliably distinguishes high-quality perception configurations from lower-quality ones. Collectively, these results suggest that the surrogate provides both accurate and decision-relevant perception estimates, making it well suited for perception-guided motion planning.

To further evaluate generalization, we examine the surrogate model under two out-of-distribution conditions. For occlusion, we sample 500 valid robot configurations in the hospital scene and verify line-of-sight visibility using PyBullet raycasting. Among these samples, $11.8\%$ of configurations contain object occlusion caused by surrounding furniture. Because the surrogate model operates solely on camera–object relative pose and does not explicitly encode geometric visibility, it systematically overestimates the perception score for occluded configurations, with a mean error of $0.151$ and an RMSE of $0.258$. Nevertheless, the planner's collision-free constraint implicitly mitigates part of this issue by excluding configurations in which the robot itself intersects with obstacles. As a result, the remaining $88.2\%$ of sampled configurations maintain a good performance to the object.

For varying illuminance, the surrogate model is structurally invariant to lighting changes because its inputs are purely geometric. To characterize the expected degradation under low-light conditions, we model the confidence variation analytically using the power-law relationship $\hat{c}(L) = c_{\text{nom}}(L/L_{\text{ref}})^{0.6}$ [107], while the surrogate prediction remains fixed at $c_{\text{nom}}$. Under uniform illuminance scaling, the model fully preserves ranking consistency (Spearman $\rho = 0.641$), since the relative ordering of camera viewpoints remains unchanged. However, the absolute estimation error increases as illuminance decreases. The surrogate remains within the in-distribution error range ($\text{RMSE} < 0.154$) when illuminance is above approximately $46\%$ of the training condition, beyond which overestimation becomes increasingly significant.

These results indicate that the surrogate model generalizes well under moderate environmental variation and maintains reliable ranking behavior across different viewing conditions. However, its performance degrades under severe occlusion and extreme low-light conditions due to the absence of explicit visibility and lighting representations in the model input. These limitations motivate future work on occlusion-aware and illuminance-conditioned surrogate models to further improve robustness in challenging real-world environments.

The ray-tracing accuracy is also tested, comparing the visibility rate given by our proposed system using a single ray test, giving a score of 1 if the ray hits its target AABB and 0 otherwise. An environment with a perception target (a human) similar to Fig. 4 is evaluated with a wall, a thin wall (partial occlusion) and a fence-like wall (so target is visible through). The test was evaluated against a YOLO-based boolean detection with a $0.75$ confidence threshold. The single-ray test proved to be a conservative estimator in the fence and thin wall environments, reporting an $90.1\%$ and $85.1\%$ visibility rate compared to YOLO's $97.0\%$ and $95.0\%$, as the ray is easily intercepted by thin geometric features despite the target being actually visible through gaps. Conversely, it slightly overestimated visibility in solid occlusion ($54.5\%$ estimated vs. $49.0\%$ ground truth detection), likely due specific angles and target-specific inaccuracies, such as a human being more easily distinguishable if the face or silhouette are visible. Overall, the single-ray test approximated the ground truth within a $10\%$ margin, which allowed our use for it as a computationally efficient occlusion estimator.

### C. Planning Benchmark Results

For simulation experiments, we create an environment using objects commonly found in a home such as a table and a cabinet (shown in Fig. 6). Additionally, we introduce five different objects (a cup, a suitcase, a potted plant, a fork, and a TV) common in a home for the robot to detect and track, further mimicking real-world tasks.

We generate $2,500$ motion planning problems by randomly sampling collision-free start and goal configurations for the base, while adjusting the pan and tilt joints so that the robot is directly facing the tracking object. Specifically, we sample 500 problems for each of five objects, with five variations in the object's poses within the environment for each set.

Table II presents results across 2,500 planning problems, evaluating metrics such as planning time, path length, success rate. We use YOLOv9 [108] and DeepSORT [109] for real-time perception performance assessment. It is measured by per-frame object detection rate, average confidence score of object detection, and tracking rate.

For the perception task of object detection and tracking, all methods except **PS-PRM-frustum**, **PS-PRM-line**, **Neural-PS-PRM-frustum**, **Neural-PS-PRM-line**, and **RL-Based Active Perception** exhibit detection rates below 50% and confidence scores under 0.6. This highlights the importance of our method in directly integrating perception scores into the planning process. In contrast, both **PS-PRM-frustum**, **PS-PRM-line**, **Neural-PS-PRM-frustum**, and **Neural-PS-PRM-line** achieve an object detection rate above 70%, a confidence score exceeding 0.75, and a tracking rate around 90%, with **PS-PRM-line** delivering the best object tracking performance among all methods.

The **RL-Based Active Perception** baseline achieves relatively strong perception performance, obtaining a $75.5\%$ detection rate and $80.0\%$ tracking rate, substantially outperforming classical motion-planning baselines such as Post, Rejection, and Manifold-based approaches. However, this improved observability comes at the cost of significantly longer trajectories and reduced planning reliability. In particular, the average path length of **RL-Based Active Perception** is $21.75$ rad, which is $44.9\%$ longer than **PS-PRM-line** and $60.9\%$ longer than **Neural-PS-PRM-line**. This behavior suggests that the learned policy prioritizes maintaining favorable viewpoints over efficient goal-reaching behavior. Furthermore, the success rate drops to only $57.3\%$, compared to over $95\%$ for all variants of **PS-PRM** and Neural-**PS-PRM**, indicating that the learned policy frequently fails to reach the goal region before timeout or becomes trapped in locally suboptimal behaviors. Another limitation of **RL-Based Active Perception** is the significant training overhead required for each environment and object-monitoring scenario. The method requires approximately $169.8$ s of training/preprocessing time per scene, whereas the learned surrogate variants Neural-**PS-PRM-frustum** and Neural-**PS-PRM-line** require only around 32 s, representing approximately a $5.3\times$ reduction in preprocessing time. Although the RL baseline achieves competitive detection performance, its confidence score ($0.634$) remains lower than **PS-PRM-line** ($0.792$) and Neural-**PS-PRM-line** ($0.771$). This gap highlights an important distinction between the methods: the RL policy is only indirectly encouraged to maintain observability through reward shaping and a geometric proxy, whereas our approach explicitly optimizes a neural surrogate model trained directly on the downstream perception task.

In contrast, the **Visibility-Aware Trajectory Optimization** keeps a high reliability and is even the best method for shorter paths ($5.62$rad). However, compared with the PS-PRM and RL methods, its lack of semantic understanding through actual perception information results in a reduced detection rate and confidence. Furthermore, given it uses a simple initialization for the optimization problem (a linear interpolation from start to goal configurations) it has a reduced success rate in comparison to the PS-PRM methods, as the robot might still find itself in collision or be unable to solve for constraints (such as not moving in the y-axis for a non-holonomic robot).

It should be noted that constructing the constrained PRM with perception-based edge scores, using approximately 3,000 nodes across different monitored objects and scenes (in this experiment, 5 different objects and 5 locations each), takes on average 341.5 seconds for **PS-PRM-line** and 291.7 seconds for **PS-PRM-frustum**. Once the PRM is built, subsequent planning is very fast, as demonstrated in Table II.

The preprocessing times for **PS-PRM-frustum** and **PS-PRM-line** are relatively long due to the high computational cost of rendering the scene to compute perception scores. In contrast, **Neural-PS-PRM-frustum** and **Neural-PS-PRM-line** achieve significantly shorter preprocessing times by using a lightweight neural surrogate model to directly estimate perception scores, thereby avoiding costly rendering and computationally intensive perception models. This approach is also advantageous in dynamic scenarios, as perception scores can be efficiently updated when the monitored object or human moves.

Notably, the performance of **Neural-PS-PRM-frustum** and **Neural-PS-PRM-line** remains comparable to that of **PS-PRM-frustum** and **PS-PRM-line**. Specifically, **Neural-PS-PRM-frustum** and **Neural-PS-PRM-line** are approximately ten times faster than the version without the neural surrogate model, while maintaining comparable performance, achieving detection rates within $1\%$ and confidence scores within $0.01$ of the original. Interestingly, the resulting path lengths are also slightly shorter. This suggests that access to ground-truth perception scores may not be essential, and that a well-trained surrogate model can provide sufficiently accurate estimates to achieve a comparable or even better balance between perception quality and motion efficiency.

Fig. 6 illustrates a path generated by **PS-PRM-line**. demonstrating the observations made at two different stages of the path. Notably, it selects a trajectory that stays slightly farther from the monitored object, resulting in a better viewing angle at the cost of a slightly longer path. We notice that the confidence score is sometimes higher when detecting the back of the screen than the front. This occurs because the back of the monitor is dark and high-contrast relative to the background, whereas the front is lighter and contains glossy or reflective regions. In general, detection confidence produced by a perception model can vary due to multiple factors, including the distribution of training data, color and texture contrast, lighting conditions, surface reflectance, and viewpoint-dependent features. As the learned perception score is trained across a wide range of environments, viewpoints, and object configurations, these artifacts are mitigated and smoothed.

These results demonstrate that integrating perception scores into motion planning, especially through a neural surrogate model, enables efficient planning while maintaining high

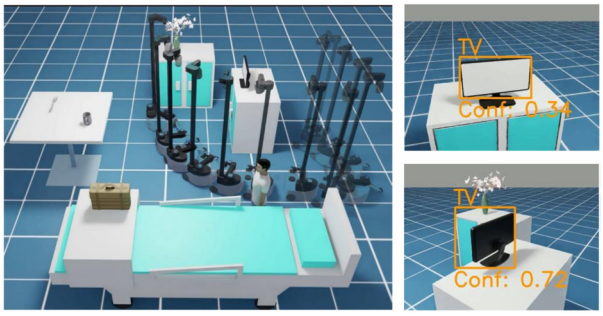



Figure 6. Example **PS-PRM** trajectory for the simulation experiment in Sec. V-C. Robot traces and camera images are sampled along the path, fading in as the robot nears the goal. Snapshots show YOLO confidence scores and ground-truth bounding boxes.

perception quality. This provides a scalable and effective solution for real-time perception-aware robot motion.

### D. Human Tracking in a Nursing Setting

As shown in Fig. 7, we create a nursing environment with two nurses with masks and one patient. The robot is required to plan a path from the start to the goal position while aiming to continuously detect some key features of the nurses. In this experiment, we consider the face detection task, and we use RetinaFace [103] for real-time face detection performance assessment. For this multi-person task, we use the midpoint between the two individuals as the monitoring point and the average of their face detection scores to construct the PRM and assess performance. In this experiment, we use the Fetch robot and plan the motion of its base, torso, and camera to perform the assigned task.

For the specific scenario depicted in Fig. 7, we evaluate both **Manifold-line** and **Neural-PS-PRM-line** over 100 independent runs. Each run of **Manifold-line** is given a maximum of 60 seconds to compute a solution. For **Neural-PS-PRM-line**, we construct a PRM and score its nodes and edges by incorporating the face detection confidence score into the planning process with a trained neural surrogate model.

Table IV summarizes the average performance over all runs, reporting metrics such as planning time, path length, face detection rate (for detecting at least one nurse or both nurses), and average face detection confidence score. We observe that **Manifold-line** tends to choose a shorter path by moving behind the nurses, as shown in Fig. 7, which often leads to poor visibility of their faces. In contrast, **Neural-PS-PRM-line** consistently favors slightly longer and cluttered paths that provide better visibility of the nurses' faces.

As shown in Table IV, **Neural-PS-PRM-line** significantly outperforms **Manifold-line** in both face detection rate and average confidence score. This improvement comes at the cost of increased path length and planning time. Specifically, **Neural-PS-PRM-line** achieves over 98% success in detecting at least one nurse and over 63% for detecting both nurses, while the baseline method fails entirely in the two-face detection task. An example of the routes selected by our method and the baseline is shown in Fig. 7.

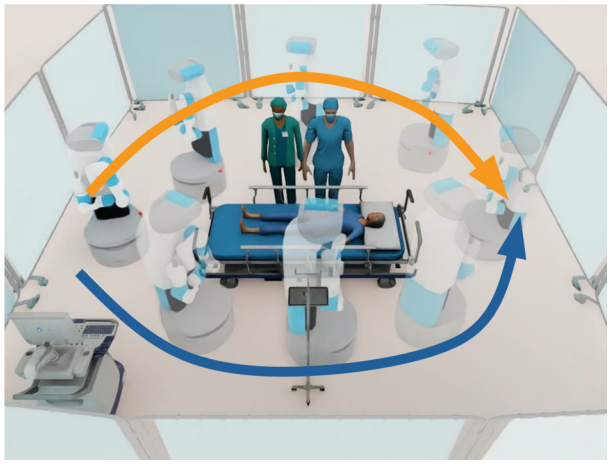


Figure 7. Simulated nursing environment: the robot plans start-to-goal while tracking both masked nurses. **Manifold-line** takes a shorter path (orange) with poor visibility; **Neural-PS-PRM-line** takes a longer path (blue) with better detection.

Table IV
Comparison of **Manifold-line** and **Neural-PS-PRM-line** in a simulated nursing environment. "Plan. Time" includes PRM construction for **Neural-PS-PRM-line**. "Det. Rate (1/2 face)" is the detection rate for one or both nurses; "Conf. Score" is the average face-detection confidence.

| Method | Motion Planning | | Human Tracking | | |
|---|---|---|---|---|---|
| | Plan. Time (s) | Path Len. (rad) | Det. Rate (1 face) | Det. Rate (2 faces) | Conf. Score |
| Manifold-line | 5.31 | 6.89 | 31.53% | 0.00% | 0.49 |
| Neural-PS-PRM-line | 32.91 | 11.51 | 98.40% | 63.29% | 0.81 |

### E. Conflicting target placements

Fig. 8 illustrates a scenario with conflicting target placements, where the planner must balance multiple perception objectives. In this experiment, the two nurses are spatially separated such that maintaining high-quality perception of both targets simultaneously is not feasible. To resolve this conflict, the planner does distance-based target-selection. As the Stretch robot moves from left to right, it initially prioritizes tracking Nurse 1, and shifts attention toward Nurse 2 as the robot approaches closer, reflecting the changing importance of the perception objectives. Given the requirement of constraint satisfaction for this problem, the constraint tolerance is increased for this experiment to work, allowing the camera to aim away from one object into another during planning.

Similarly, in Fig. 9 the conflict arises from the complex collision environment the robot has to move in, with a narrow passage between the start and goal configuration. The distance-based priority metric induces the change in the current tracked object, while the sampling-based planning allows to successfully solve the narrow passage movement.

### F. Planning in a Dynamic Environment

As shown in Fig. 10, we construct a dynamic environment containing a human, a cabinet, a shelf, and other common household objects such as a sofa, table, and chair. The robot is tasked with planning a path from its initial position to reach one of two goal locations (the cabinet or the shelf), while continuously detecting key facial features of the human. This

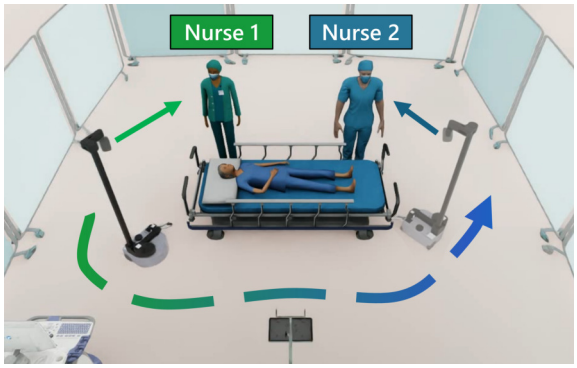



Figure 8. Distance-based priority under conflicting perception objectives: with the two nurses spatially separated, the Stretch robot shifts attention from Nurse 1 to Nurse 2 as it moves left to right.

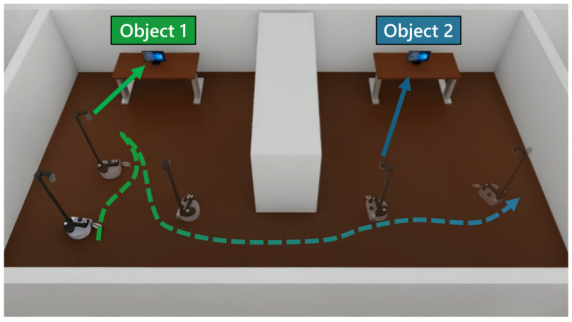



Figure 9. Planning through a narrow passage: distance-based priority shifts which object the camera focuses on as the robot crosses.

experiment focuses on a face detection and tracking task, utilizing RetinaFace [103] to assess real-time face detection performance. All trials are conducted using the Stretch 2 robot. During execution, the human moves and rotates from side to side, requiring the robot to dynamically select which goal to pursue in order to maintain high face detection performance.

For the specific scenario illustrated in Fig. 10, we evaluate both **Manifold-line** and **Neural-PS-PRM-line** over 100 independent runs. Each **Manifold-line** trial is given a maximum of 60 seconds to compute a plan. In contrast, **Neural-PS-PRM-line** constructs a PRM and assigns perception-aware scores using a trained neural surrogate model based on face detection confidence. Unlike **Manifold-line**, which does not account for the human's dynamic movements, **Neural-PS-PRM-line** supports online replanning during execution as described in Sec. IV-C. Table V presents the average results across all runs, evaluating metrics such as planning time, path length, face detection rate, and average confidence score.

As expected, **Manifold-line** consistently selects a single goal and results in very low face detection performance. In contrast, **Neural-PS-PRM-line** initially chooses the path toward the shelf on the left, as the human begins facing that direction. During execution, the robot replans in real time and switches to the path toward the cabinet on the right as the human turns, thereby maintaining better visibility of the face. As shown in Table V, **Neural-PS-PRM-line** significantly outperforms **Manifold-line** in both face detection rate and average confidence score, while also demonstrating successful online replanning. An example of the routes selected by our method is shown in Fig. 10.

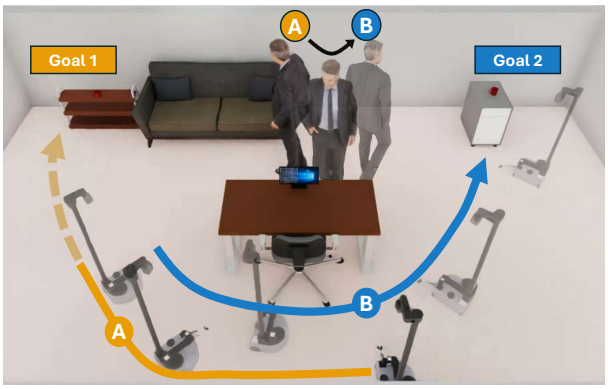



Figure 10. Dynamic planning for human face detection: as the human rotates, the Stretch 2 robot replans from the shelf (Goal 1, orange) to the cabinet (Goal 2, blue) to preserve visibility. **Neural-PS-PRM-line**'s online replanning achieves much higher detection rates than static planning.

Table V
Comparison of **Manifold-line** and **Neural-PS-PRM-line** in a dynamic, human-aware scenario. "#Plans" is the average number of replans during execution; "Det. Rate" and "Conf. Score" are face-detection rate and average confidence.

| | Motion Planning | | | Face Detection | |
|---|---|---|---|---|---|
| Method | Plan. Time (s) | #Plans | Path Len. (rad) | Det. Rate | Conf. Score |
| Manifold-line | 0.17 | 1 | 5.48 | 74.13% | 0.55 |
| Neural-PS-PRM-line | 2.49 | 2.12 | 10.53 | 90.53% | 0.73 |

In summary, these experiments confirm that our approach **PS-PRM** is effective for planning simultaneous perception and motion for high-DoF robots in practical human-robot interaction applications in dynamic environments.

### *G. Mobile Manipulator Planning in Blind Spot Alley*

In Fig. 1, we illustrate another scenario where a mobile manipulator (a UR5 robot mounted on a differential-drive base) moves from a start configuration at the right-side corner to a goal configuration at the left-side corner, navigating through a cluttered environment. During this motion, the robot must maintain high perception performance, measured by its ability to continuously track a monitor placed at the far end of the cross lane. We use **Neural-PS-PRM-frustum** in this case to plan full-body motion for the robot, ensuring that the monitor remains consistently detected with high confidence scores. As shown in Fig. 1, the robot continuously adjusts its arm to maintain a clear view of the monitor with its wrist-mounted camera, while simultaneously navigating its differential-drive base to avoid obstacles. The accompanying camera views confirm that the monitor is successfully detected throughout the entire motion. This capability is broadly applicable, enabling the robot to perceive blind spots in the environment to avoid collisions or continue tracking critical objects or humans.

### *H. Real Robot Experiments*

*1) Static Object Tracking Task:* In this experiment, as shown in Fig. 11, we set up an environment where the robot must choose a path, either left or right around the table, to reach the other side. Simultaneously, the robot is tasked with continuously detecting the cup placed on the table. The left path is shorter, but visibility of the cup is poor due to occlusion by the monitor.

The right path is longer, as it is partially blocked by the chair, but offers better visibility of the cup.

We compare **PS-PRM-line** (blue path) with **Manifold-line** (orange path). As illustrated in Fig. 11, **PS-PRM-line** takes a longer route to avoid occlusion by the monitor, ensuring continuous detection of the cup, while the **Manifold-line** opts for the shorter path, which leads to the cup being occluded. This is further supported by the robot's camera view images (displayed sequentially from top left to bottom right), showing that **PS-PRM-line** detects the cup in significantly more frames compared to **Manifold-line**.

*2) Dynamic Human Tracking Task:* In this experiment, illustrated in Fig. 12, we demonstrate the deployment of our system in a realistic nurse training scenario. The environment includes a nurse engaged in her workspace, surrounded by typical elements such as work tables, a chair, a platform, a garbage can, and two cabinets that serve as potential goal locations where the robot needs to grab some tools for the nurse. The task requires the Stretch 2 robot to navigate to one of these locations while maintaining reliable face detection of the nurse. As the nurse rotates during the task, the robot must adapt to changes in facial visibility. Our system initially plans a path to the left cabinet (Goal 1) when the nurse faces left, but dynamically replans to the right cabinet (Goal 2) as she turns right, preserving optimal visibility. This experiment highlights how **Neural-PS-PRM-line** enables evaluating trajectories on real-time and replanning, a critical requirement in nurse training and collaborative medical environments.

*3) Human Tracking with Moving Obstacle and Goal:* In this experiment (illustrated in Fig. 13), we demonstrate our system's performance in a realistic, dynamic scenario. The environment contains a seated person reading a book, surrounded by two cabinets that serve respectively as an obstacle and a goal location the robot must reach. The task requires the Stretch 2 robot to reach a configuration near the goal cabinet while maintaining reliable face tracking of the person. As the person naturally rotates during the task, the robot must adapt to changes in their pose and the obstacle locations. When the person initially faces right, the robot selects a path on the right side; as the person turns left, the system dynamically replans and shifts to a left-side path to preserve optimal perception. We then introduce additional environmental changes: the obstacle cabinet is moved mid-execution, forcing the robot to replan in real time to avoid collision, and we further demonstrate that the robot can plan toward a moving goal as the goal cabinet changes position. For experiment setup, we use an OptiTrack motion capture system to track the positions and orientations of both the cabinets and the human; this could be replaced with alternative pose tracking solutions.

By exploiting the biased sampler and triggering replanning via movement thresholds (determined by collision paddings), the planner achieves execution rates of up to 2Hz. It also generated an average of 8 trajectories per run, more than the 3 environment changes. This highlights the capability to plan multiple times on single scene alterations. Overall, this experiment shows how Neural-PS-PRM-line enables collision- and perception-aware online replanning, supporting continuous coordination between motion and perception in the presence of moving obstacles and shifting perception target and motion goal locations.

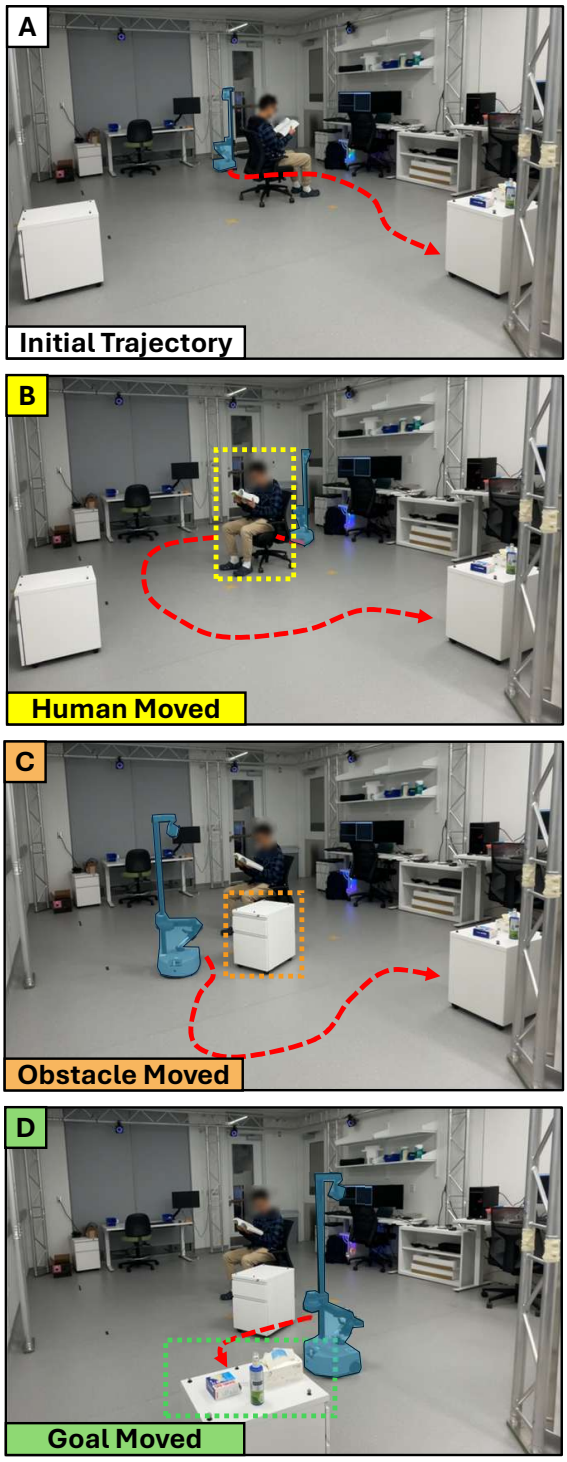



Figure 13. Dynamic human, obstacle, and goal movement: the Stretch 2 robot (blue) switches paths as the person rotates (B), replans around a moved obstacle (C), and replans again as the goal cabinet moves (D).

*4) Manipulator Grasping Task with Human Tracking Constraints:* In this experiment (illustrated in Fig. 14), we demonstrate our system's performance on a real-robot grasping task using a UR5 manipulator. The environment contains a seated person reading a book, and the robot is required to grasp an object while continuously monitoring the person's

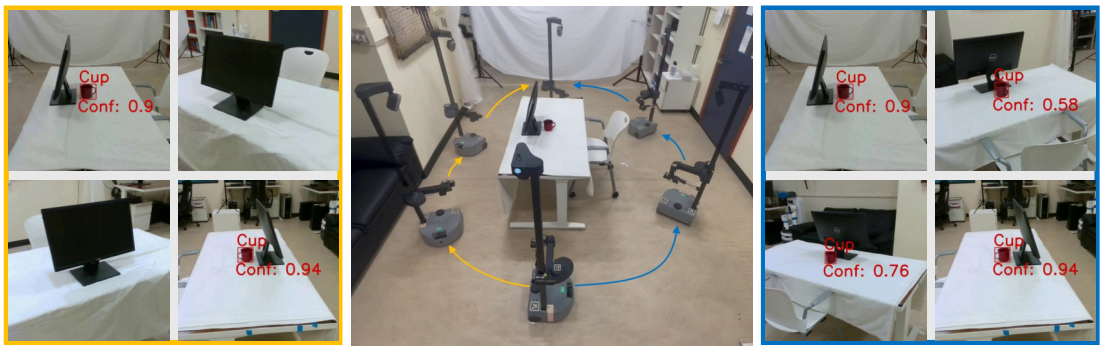



Figure 11. Real-robot cup-detection experiment: motion paths for **Manifold-line** (orange) and **PS-PRM-line** (blue), with camera views for each shown top-to-bottom, left-to-right.

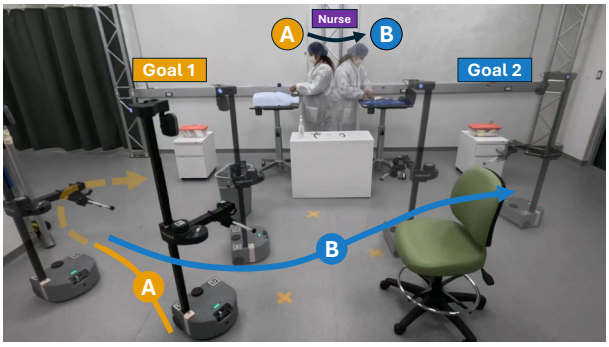



Figure 12. Real-robot face-detection experiment: as the nurse rotates, the Stretch 2 robot replans from the left cabinet (Goal 1, orange) to the right cabinet (Goal 2, blue) to preserve visibility, using **Neural-PS-PRM-line**'s online replanning.

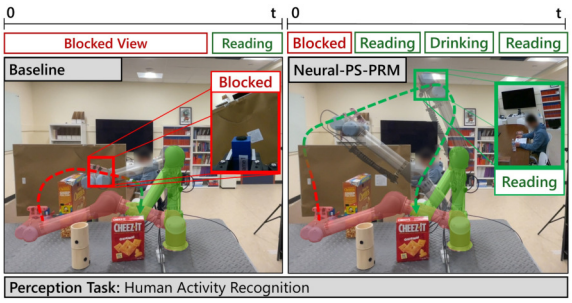



Figure 14. Perception-constrained planning for human activity recognition on a 6-DoF UR5. Without perception costs (short path), the person is largely occluded; the **PS-PRM** trajectory detours to maintain visibility. Trajectories are dotted, red when occluded and green when visible.

activity. A wooden board introduces additional occlusion that can partially block the wrist-mounted camera view. The robot's task is to move from left to right while maintaining human activity detection for as long as possible.

During the execution, the person alternates between reading a book, picking up a water bottle to drink, and then continuing to read. Human activity recognition is performed using a vision–language–based activity detection model [110]. As shown in the left subfigure, the baseline method **PS-PRM-line**, which does not incorporate occlusion reasoning, selects the shortest path to the goal. However, this trajectory fails to observe the person's activity for most of the execution and, in particular, misses the drinking action. In contrast, as shown in the right subfigure, **Neural-PS-PRM-line** plans a longer trajectory that moves above the wooden board, maintaining visibility of the person throughout most of the motion. As a result, it successfully captures the drinking activity and accurately detects the time intervals during which the person is reading.

This real-robot experiment demonstrates the practical applicability of our method to manipulation tasks that require human awareness under occlusion.

In summary, these real-robot experiments show that **PS-PRM** is suitable for tasks requiring simultaneous perception and motion in dynamic environments.

## VI. Discussion

In this work, we address the problem of planning robot motions for a high-degree-of-freedom (DoF) robot that effectively achieves a given perception task while the robot and the perception target move in a dynamic environment. We propose a GPU-parallelized perception-score-guided probabilistic roadmap planner with a neural surrogate model (PS-PRM): a roadmap-based planner that explicitly incorporates the estimated quality of a perception task into motion planning for high-DoF robots. Our method uses a learned model to approximate perception scores and leverages GPU parallelism to enable efficient online replanning in dynamic settings.

Our extensive experiments across multiple platforms, including the UR5 robot, the Fetch robot, and Hello Robot's Stretch 2 robot, demonstrate the consistent benefits of PS-PRM over conventional planning pipelines. In a range of realistic environments, including cluttered home settings, blind-spot navigation, and dynamic human-robot interaction scenarios, PS-PRM significantly improves perception outcomes while maintaining or modestly increasing motion costs. For instance, we observe at least a 1.6× improvement in object detection rates in home settings and over a 2.1× improvement in face detection rates in crowded nursing environments. Moreover, the neural surrogate model enables planning that is approximately 10× faster than approaches relying on ground-truth perception pipelines, while achieving comparable detection accuracy and sometimes even shorter paths.

Our case studies highlight the practical value of our method in diverse perception-aware planning tasks: in the

UR5 scenario, the arm continuously reorients to maintain a clear view of a monitor despite occlusions; in the Fetch robot experiments, the planner actively selects longer but more perceptually informative paths through nurse-populated corridors; and in dynamic human-robot scenarios with the Stretch robot, the system successfully replans on-the-fly to maintain face visibility despite human movement and rotation.

While our work focuses on object and face detection, the underlying framework is general and applicable to other perception-driven tasks such as semantic mapping, affordance detection, or visual servoing. PS-PRM serves as a bridge between motion generation and high-level perception, enabling robots to make informed decisions that account for both geometry and visibility constraints.

This direction opens several promising avenues for future research. First, while the neural surrogate model significantly reduces computation time and is robot- and environment-agnostic, it is currently trained offline using a fixed set of perception tasks, object categories, and environmental conditions. Future work could explore domain adaptation, continual learning, or occlusion- and illuminance-aware surrogate models to improve robustness under out-of-distribution scenarios and varying real-world environments. Second, incorporating uncertainty estimates from the surrogate model may further enhance robustness, particularly in ambiguous or noisy scenes. Third, there remains opportunities for further GPU parallelization, such as incorporating ideas from [111] to further accelerate PRM construction and query. An interesting extension of this work is to explore how the framework can be adapted to a broader range of perception tasks and application domains, such as semantic segmentation or 3D reconstruction.

Finally, real-world deployments in assistive and collaborative environments, such as smart homes and healthcare, will require consideration of social and ethical factors. For example, in nursing applications, robots may need to detect faces, interpret gestures, and maintain situational awareness while respecting privacy. Although this work is focused on training simulations, a natural extension is to integrate privacy-preserving constraints into the planning process [112], allowing robots to intelligently navigate and perceive while adhering to user-centered constraints.

In summary, this work demonstrates the effectiveness of perception-guided planning for high-DoF robots in both static and dynamic environments. By jointly reasoning about motion and perception, PS-PRM offers a flexible and generalizable foundation for building more capable, context-aware robotic systems.

## REFERENCES


[1] T. Yin, X. Zhou, and P. Krahenbuhl. “Center-based 3d object detection and tracking”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 11784–11793.

[2] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah. “Deep learning-based human pose estimation: A survey”. In: *ACM Computing Surveys* 56.1 (2023), pp. 1–37.

[3] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. “Image segmentation using deep learning: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 44.7 (2021), pp. 3523–3542.

[4] T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges. “Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization”. In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1696–1703.

[5] L. Blasi, E. D’Amato, M. Mattei, and I. Notaro. “Path planning and real-time collision avoidance based on the essential visibility graph”. In: *Applied Sciences* 10.16 (2020), p. 5613.

[6] M. Piazza, M. Piccinini, S. Taddei, and F. Biral. “Mptree: A sampling-based vehicle motion planner for real-time obstacle avoidance”. In: *IFAC-PapersOnLine* 58.10 (2024), pp. 146–153.

[7] T. Mercy, W. Van Loock, and G. Pipeleers. “Real-time motion planning in the presence of moving obstacles”. In: *2016 European Control Conference (ECC)*. IEEE. 2016, pp. 1586–1591.

[8] S. M. Aloush and F. A. Alsaraireh. “Nurses’ compliance with central line associated blood stream infection prevention guidelines”. In: *Saudi medical journal* 39.3 (2018), p. 273.

[9] C. Quintero-Pena, P. Qian, N. M. Fontenot, H.-M. Chen, S. K. Hamlin, L. E. Kavraki, and V. Unhelkar. “Robotic Tutors for Nurse Training: Opportunities for HRI Researchers”. In: *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2023, pp. 220–225.

[10] P. Qian, F. Bajraktari, C. Quintero-Peña, Q. Meng, S. Hamlin, L. Kavraki, and V. Unhelkar. “ASTRID: A Robotic Tutor for Nurse Training to Reduce Healthcare-Associated Infections”. In: *Proceedings of the 2025 Robotics: Science and System*. 2025.

[11] R. Singh, J. Liu, K. Van Wyk, Y.-W. Chao, J.-F. Lafleche, F. Shkurti, N. Ratliff, and A. Handa. “Synthetica: Large Scale Synthetic Data for Robot Perception”. In: *arXiv preprint arXiv:2410.21153* (2024).

[12] J. Tordesillas and J. P. How. “PANTHER: Perception-Aware Trajectory Planner in Dynamic Environments”. In: *IEEE Access* 10 (2022), pp. 22662–22677.

[13] A. Singla, S. Padakandla, and S. Bhatnagar. “Memory-Based Deep Reinforcement Learning for Obstacle Avoidance in UAV With Limited Environment Knowledge”. In: *IEEE Transactions on Intelligent Transportation Systems* 22.1 (2021), pp. 107–118.

[14] H. Masnavi, A. K. Singh, and F. Janabi-Sharifi. “Differentiable-Optimization Based Neural Policy for Occlusion-Aware Target Tracking”. In: *arXiv preprint arXiv:2406.14639* (2024).

[15] H. Masnavi, J. Shrestha, M. Mishra, P. B. Sujit, K. Kruusamäe, and A. K. Singh. “Visibility-Aware Navigation With Batch Projection Augmented Cross-Entropy Method Over a Learned Occlusion Cost”. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 9366–9373.

[16] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza. “PAMPC: Perception-Aware Model Predictive Control for Quadrotors”. In: *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* 2018.

[17] B. Ichter, B. Landry, E. Schmerling, and M. Pavone. “Perception-Aware Motion Planning via Multiobjective Search on GPUs”. In: *Robotics Research*. Ed. by N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti. Cham: Springer International Publishing, 2020, pp. 895–912.

[18] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza. “Learning high-speed flight in the wild”. In: *Science Robotics* 6.59 (2021), eabg5810.

[19] Y. Song, K. Shi, R. Penicka, and D. Scaramuzza. “Learning Perception-Aware Agile Flight in Cluttered Environments”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 1989–1995.

[20] L. Bartolomei, L. Teixeira, and M. Chli. “Perception-aware Path Planning for UAVs using Semantic Segmentation”. In: *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* 2020, pp. 5808–5815.

[21] R. Bajcsy. “Active perception”. In: *Proceedings of the IEEE* 76.8 (1988), pp. 966–1005.

[22] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. “Active vision”. In: *International journal of computer vision* 1 (1988), pp. 333–356.

[23] S. Soatto. “Actionable information in vision”. In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE. 2009, pp. 2138–2145.

[24] L. Liu, S. Fryc, L. Wu, T. L. Vu, G. Paul, and T. Vidal-Calleja. “Active and interactive mapping with dynamic gaussian process implicit surfaces for mobile manipulators”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3679–3686.

[25] A. J. Davison and D. W. Murray. “Simultaneous localization and map-building using active vision”. In: *IEEE transactions on pattern analysis and machine intelligence* 24.7 (2002), pp. 865–880.

[26] T. A. Vidal-Calleja, A. Sanfeliu, and J. Andrade-Cetto. "Action selection for single-camera SLAM". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 40.6 (2010), pp. 1567–1581.

[27] C. Mostegel, A. Wendel, and H. Bischof. "Active monocular localization: Towards autonomous monocular exploration for multirotor mavs". In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 3848–3855.

[28] S. A. Sadat, K. Chutskoff, D. Jungic, J. Wawerla, and R. Vaughan. "Feature-rich path planning for robust navigation of MAVs with mono-SLAM". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 3870–3875.

[29] D. Morilla-Cabello, J. Westheider, M. Popović, and E. Montijano. "Perceptual factors for environmental modeling in robotic active perception". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 4605–4611.

[30] K. H. Low, J. M. Dolan, and P. Khosla. "Adaptive multi-robot wide-area exploration and mapping". In: *Proceedings of the 7th International Joint Conference on Autonomous agents and Multiagent systems-Volume 1*. 2008, pp. 23–30.

[31] G. A. Hollinger and G. S. Sukhatme. "Sampling-based robotic information gathering algorithms". In: *The International Journal of Robotics Research* 33.9 (2014), pp. 1271–1287.

[32] J. Rückin, L. Jin, and M. Popović. "Adaptive informative path planning using deep reinforcement learning for uav-based active sensing". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 4473–4479.

[33] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser. "Efficient informative sensing using multiple robots". In: *Journal of Artificial Intelligence Research* 34 (2009), pp. 707–755.

[34] J. H. Reif. "Complexity of the mover's problem and generalizations". In: *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*. IEEE Computer Society. 1979, pp. 421–427.

[35] J. Binney and G. S. Sukhatme. "Branch and bound for informative path planning". In: *2012 IEEE international conference on robotics and automation*. IEEE. 2012, pp. 2147–2154.

[36] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias. "Efficient multi-robot search for a moving target". In: *The International Journal of Robotics Research* 28.2 (2009), pp. 201–219.

[37] B. Bonet and H. Geffner. "Planning with incomplete information as heuristic search in belief space". In: *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*. 2000, pp. 52–61.

[38] S. M. LaValle, J. J. Kuffner, and Jr. "Rapidly-Exploring Random Trees: Progress and Prospects". In: *Algorithmic and Computational Robotics: New Directions*. 2000, pp. 293–308.

[39] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". In: *IEEE transactions on Robotics and Automation* 12.4 (1996), pp. 566–580.

[40] D. Levine, B. Luders, and J. How. "Information-rich path planning with general constraints using rapidly-exploring random trees". In: *AIAA Infotech@ Aerospace 2010*. 2010, p. 3360.

[41] A. Bry and N. Roy. "Rapidly-exploring random belief trees for motion planning under uncertainty". In: *2011 IEEE international conference on robotics and automation*. IEEE. 2011, pp. 723–730.

[42] H. Lu, Q. Zong, S. Lai, B. Tian, and L. Xie. "Real-Time Perception-Limited Motion Planning Using Sampling-Based MPC". In: *IEEE Transactions on Industrial Electronics* 69.12 (2022), pp. 13182–13191.

[43] B. Penin, P. R. Giordano, and F. Chaumette. "Vision-Based Reactive Planning for Aggressive Target Tracking While Avoiding Collisions and Occlusions". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3725–3732.

[44] B. Penin, P. R. Giordano, and F. Chaumette. "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3725–3732.

[45] B. Penin, R. Spica, P. R. Giordano, and F. Chaumette. "Vision-based minimum-time trajectory generation for a quadrotor UAV". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 6199–6206.

[46] J. Thomas, J. Welde, G. Loianno, K. Daniilidis, and V. Kumar. "Autonomous flight for detection, localization, and tracking of moving targets with a small quadrotor". In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1762–1769.

[47] G. Li, A. Tunchez, and G. Loianno. "PCMPC: Perception-constrained model predictive control for quadrotors with suspended loads using a single camera and IMU". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 2012–2018.

[48] J. Tordesillas and J. P. How. "Deep-PANTHER: Learning-Based Perception-Aware Trajectory Planner in Dynamic Environments". In: *IEEE Robotics and Automation Letters* 8.3 (2023), pp. 1399–1406.

[49] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 528–535.

[50] R. Bonatti, W. Wang, C. Ho, A. Ahuja, M. Gschwindt, E. Camci, E. Kayacan, S. Choudhury, and S. Scherer. "Autonomous aerial cinematography in unstructured environments with learned artistic decision-making". In: *Journal of Field Robotics* 37.4 (2020), pp. 606–641.

[51] I. Bozcan and E. Kayacan. "Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 8504–8510.

[52] M. Mendoza, J. I. Vasquez-Gomez, H. Taud, L. E. Sucar, and C. Reta. "Supervised learning of the next-best-view for 3d object reconstruction". In: *Pattern Recognition Letters* 133 (2020), pp. 224–231.

[53] T. Yi, D. Zhang, L. Luo, and J. Luo. "View planning for grape harvesting based on active vision strategy under occlusion". In: *IEEE Robotics and Automation Letters* 9 (2024), pp. 2535–2542.

[54] E. Johns, S. Leutenegger, and A. J. Davison. "Pairwise decomposition of image sequences for active multi-view recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3813–3822.

[55] R. Zeng, Y. Wen, W. Zhao, and Y.-J. Liu. "View planning in robot active vision: A survey of systems, algorithms, and applications". In: *Computational Visual Media* 6 (2020), pp. 225–245.

[56] F. Sukkar, G. Best, C. Yoo, and R. Fitch. "Multi-robot region-of-interest reconstruction with Dec-MCTS". In: *2019 International conference on robotics and automation (ICRA)*. IEEE. 2019, pp. 9101–9107.

[57] H. J. S. Feder, J. J. Leonard, and C. M. Smith. "Adaptive mobile robot navigation and mapping". In: *The International Journal of Robotics Research* 18.7 (1999), pp. 650–668.

[58] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. "Information based adaptive robotic exploration". In: *IEEE/RSJ international conference on intelligent robots and systems*. Vol. 1. IEEE. 2002, pp. 540–545.

[59] A. Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy. "Estimation, planning, and mapping for autonomous flight using an RGB-D camera in GPS-denied environments". In: *The International Journal of Robotics Research* 31.11 (2012), pp. 1320–1343.

[60] N. Atanasov, B. Sankaran, J. Le Ny, G. J. Pappas, and K. Daniilidis. "Nonmyopic view planning for active object classification and pose estimation". In: *IEEE Transactions on Robotics* 30.5 (2014), pp. 1078–1090.

[61] Y. Li and Z. Liu. "Information entropy-based viewpoint planning for 3-D object reconstruction". In: *IEEE Transactions on Robotics* 21.3 (2005), pp. 324–337.

[62] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa. "Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects". In: *Journal of Real-Time Image Processing* 10 (2015), pp. 611–631.

[63] W. Tang and M. R. Jahanshahi. "Active perception based on deep reinforcement learning for autonomous robotic damage inspection". In: *Machine Vision and Applications* 35.5 (2024), p. 110.

[64] G. Wang, H. Li, S. Zhang, D. Guo, Y. Liu, and H. Liu. "Observe Then Act: Asynchronous Active Vision-Action Model for Robotic Manipulation". In: *IEEE Robotics and Automation Letters* (2025).

[65] L. Jin, X. Zhong, Y. Pan, J. Behley, C. Stachniss, and M. Popović. "Activegs: Active scene reconstruction using gaussian splatting". In: *IEEE Robotics and Automation Letters* (2025).

[66] T. Chen, O. Shorinwa, J. Bruno, A. Swann, J. Yu, W. Zeng, K. Nagami, P. Dames, and M. Schwager. "Splat-nav: Safe real-time robot navigation in gaussian splatting maps". In: *IEEE Transactions on Robotics* (2025).

[67] J. Xiao, R. Zhang, Y. Zhang, and M. Feroskhan. "Vision-based learning for drones: A survey". In: *IEEE Transactions on Neural Networks and Learning Systems* (2025).

[68] S. Razavi, B. A. Tolson, and D. H. Burn. “Review of surrogate modeling in water resources”. In: *Water Resources Research* 48.7 (2012).

[69] T. Weber, A. Corotan, B. Hutchinson, B. Kravitz, and R. Link. “Deep learning for creating surrogate models of precipitation in Earth system models”. In: *Atmospheric Chemistry and Physics* 20.4 (2020), pp. 2303–2317.

[70] C. Xiao, O. Leeuwenburgh, H.-X. Lin, and A. Heemink. “Conditioning of deep-learning surrogate models to image data with application to reservoir characterization”. In: *Knowledge-Based Systems* 220 (2021), p. 106956.

[71] P. Du, X. Zhu, and J.-X. Wang. “Deep learning-based surrogate model for three-dimensional patient-specific computational fluid dynamics”. In: *Physics of Fluids* 34.8 (2022).

[72] A. Hamid, D. Rafiq, S. A. Nahvi, and M. A. Bazaz. “Deep learning assisted surrogate modeling of large-scale power grids”. In: *Sustainable Energy, Grids and Networks* 34 (2023), p. 101031.

[73] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[74] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. “Deepsdf: Learning continuous signed distance functions for shape representation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 165–174.

[75] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint. “Learning multi-object dynamics with compositional neural radiance fields”. In: *Conference on robot learning*. PMLR. 2023, pp. 1755–1768.

[76] D. Driess, J.-S. Ha, M. Toussaint, and R. Tedrake. “Learning models as functionals of signed-distance fields for manipulation planning”. In: *Conference on robot learning*. PMLR. 2022, pp. 245–255.

[77] M. Koptev, N. Figueroa, and A. Billard. “Neural joint space implicit signed distance functions for reactive robot manipulator control”. In: *IEEE Robotics and Automation Letters* 8.2 (2022), pp. 480–487.

[78] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager. “Vision-only robot navigation in a neural radiance world”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 4606–4613.

[79] G. S. Camps, R. Dyro, M. Pavone, and M. Schwager. “Learning Deep SDF Maps Online for Robot Navigation and Exploration”. In: *arXiv preprint arXiv:2207.10782* (2022).

[80] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei. “Voxposer: Composable 3d value maps for robotic manipulation with language models”. In: *arXiv preprint arXiv:2307.05973* (2023).

[81] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. “Neural descriptor fields: Se (3)-equivariant object representations for manipulation”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 6394–6400.

[82] M. Shridhar, L. Manuelli, and D. Fox. “Cliport: What and where pathways for robotic manipulation”. In: *Conference on robot learning*. PMLR. 2022, pp. 894–906.

[83] B. Wen, W. Lian, K. Bekris, and S. Schaal. “You only demonstrate once: Category-level manipulation from single visual demonstration”. In: *arXiv preprint arXiv:2201.12716* (2022).

[84] C. Quintero-Pena, W. Thomason, Z. Kingston, A. Kyrillidis, and L. E. Kavraki. “Stochastic implicit neural signed distance functions for safe motion planning under sensing uncertainty”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 2360–2367.

[85] K. Long, K. M. B. Lee, N. Raicevic, N. Attasseri, M. Leok, and N. Atanasov. “Neural Configuration-Space Barriers for Manipulation Planning and Control”. In: *arXiv preprint arXiv:2503.04929* (2025).

[86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[87] B. Xu, A. Tao, H. Thomas, J. Zhang, and T. D. Barfoot. “MakeWay: Object-Aware Costmaps for Proactive Indoor Navigation Using LiDAR”. In: *arXiv preprint arXiv:2408.17034* (2024).

[88] J. Gao, C. Gu, Y. Lin, Z. Li, H. Zhu, X. Cao, L. Zhang, and Y. Yao. “Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing”. In: *European Conference on Computer Vision*. Springer. 2024, pp. 73–89.

[89] K. Sung, P. Shirley, and S. Baer. *Essentials of interactive computer graphics: concepts and implementation*. CRC Press, 2008.

[90] A. Orthey, C. Chamzas, and L. E. Kavraki. “Sampling-based motion planning: A comparative review”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 7 (2023), pp. 285–310.

[91] Z. Kingston, M. Moll, and L. E. Kavraki. “Sampling-based methods for motion planning with constraints”. In: *Annual review of control, robotics, and autonomous systems* 1 (2018), pp. 159–185.

[92] P. E. Hart, N. J. Nilsson, and B. Raphael. “A formal basis for the heuristic determination of minimum cost paths”. In: *IEEE transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107.

[93] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, and W. Burgard. *Principles of robot motion: theory, algorithms, and implementations*. MIT press, 2005.

[94] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich. “The design of stretch: A compact, lightweight mobile manipulator for indoor human environments”. In: *IEEE Int. Conf. Robot. Autom.* IEEE. 2022, pp. 3150–3157.

[95] E. Coumans and Y. Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning*. http://pybullet.org. 2016.

[96] I. A. Sucan, M. Moll, and L. E. Kavraki. “The open motion planning library”. In: *IEEE Robotics & Automation Magazine* 19.4 (2012), pp. 72–82.

[97] NVIDIA. *NVIDIA Isaac Sim*. https://developer.nvidia.com/isaac-sim. [Accessed 08-Aug-2024]. 2022.

[98] Q. Wang, Y. Gao, J. Ji, C. Xu, and F. Gao. “Visibility-aware Trajectory Optimization with Application to Aerial Tracking”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 5249–5256.

[99] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. “Proximal Policy Optimization Algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[100] D. Jayaraman and K. Grauman. “Learning to Look Around: Intelligently Exploring Unseen Environments for Semantic Embodied Question Answering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[101] P. Mirowski, M. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, A. Zisserman, and R. Hadsell. “Learning to Navigate in Cities Without a Map”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2018.

[102] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. “Target-driven Visual Navigation in Indoor Scenes Using Deep Reinforcement Learning”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017.

[103] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou. “Retinaface: Single-stage dense face localisation in the wild”. In: *arXiv preprint arXiv:1905.00641* (2019).

[104] S. Serengil and A. Ozpinar. “A Benchmark of Facial Recognition Pipelines and Co-Usability Performances of Modules”. In: *Journal of Information Technologies* 17.2 (2024), pp. 95–107.

[105] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft coco: Common objects in context”. In: *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*. Springer. 2014, pp. 740–755.

[106] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[107] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel. “Benchmarking robustness in object detection: Autonomous driving when winter is coming”. In: *arXiv preprint arXiv:1907.07484* (2019).

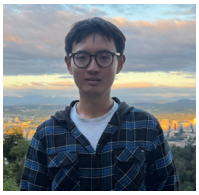
[108] C.-Y. Wang and H.-Y. M. Liao. “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information”. In: *arXiv preprint arXiv:2402.13616* (2024).

[109] N. Wojke and A. Bewley. “Deep Cosine Metric Learning for Person Re-identification”. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 748–756.

[110] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. “Efficient Memory Management for Large Language Model Serving with PagedAttention”. In: *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*. 2023.

[111] M. Akinc, K. E. Bekris, B. Y. Chen, A. M. Ladd, E. Plaku, and L. E. Kavraki. “Probabilistic roadmaps of trees for parallel computation of multiple query roadmaps”. In: *Robotics Research. The Eleventh International Symposium: With 303 Figures*. Springer. 2005, pp. 80–89.

[112] R. Shome, Z. Kingston, and L. E. Kavraki. “Robots as AI Double Agents: Privacy in Motion Planning”. In: *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* 2023, pp. 2861–2868.



**Qingxi Meng** received the B.S. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2020, and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2022. He is currently working toward the Ph.D. degree in computer science with Rice University, Houston, TX, USA.

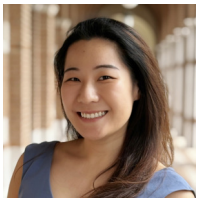
His research interests include perception-aware motion planning, GPU-accelerated planning algorithms, and robot perception for human-robot interaction in healthcare environments.

**Emiliano Flores** received the B.Sc. degree in robotics engineering from Tecnológico de Monterrey, Mexico, in 2025. He is currently working toward the Ph.D. degree in computer science with Rice University, Houston, TX, USA, under the supervision of Prof. Lydia E. Kavraki.

His research interests include hardware-accelerated, perception- and task-aware motion planning, optimization-based methods and long-horizon planning.

**Carlos Quintero-Peña** received the B.S. degree in electronic engineering and the M.Sc. degree in electronic and computer engineering from Universidad de los Andes, Bogotá, Colombia, in 2009 and 2010, respectively, and the Ph.D. degree in computer science from Rice University, Houston, TX, USA, in 2024, under the supervision of Prof. Lydia E. Kavraki. He was a recipient of a Fulbright Scholarship in 2019.

His research interests include motion planning for high-degree-of-freedom robots, optimization-based planning under sensing uncertainty, and human-in-the-loop robot planning.



**Peizhu Qian** received the B.S. degree in computer science and mathematics from Simmons College, Boston, MA, USA, in 2019, and the Ph.D. degree in computer science from Rice University, Houston, TX, USA, in 2025, under the supervision of Prof. Vaibhav Unhelkar.

She is currently an Assistant Professor of computer science with the University of Houston, Houston, TX, USA. Her research interests include human-robot interaction, human-centered AI, explainable AI, and healthcare robotics, including intelligent robotic tutors for clinical training.

**Zachary Kingston** received the B.S., M.S., and Ph.D. degrees in computer science from Rice University, Houston, TX, USA, in 2016, 2017, and 2021, respectively, under the supervision of Prof. Lydia E. Kavraki.

From 2021 to 2024, he was a Postdoctoral Researcher and Lab Manager with the Kavraki Lab at Rice University. He is currently an Assistant Professor of computer science with Purdue University, West Lafayette, IN, USA, where he leads the Computational Motion, Manipulation, and Autonomy (CoMMA) Lab. His research interests include sampling-based and manifold-constrained motion planning, GPU-accelerated planning, and planning for high-dimensional and soft robotic systems.

**Shannan K. Hamlin** received the B.S.N. degree from Houston Christian University, Houston, TX, USA, and the M.S.N. and Ph.D. degrees from the University of Texas Health Science Center at Houston, Houston, TX, USA.

She is an Associate Professor of Nursing with the Houston Methodist Academic Institute, where she also serves as Chair of Nursing and Director of the Center for Nursing Research, Education and Practice. She is a board-certified Adult-Gerontology Acute Care Nurse Practitioner and a Fellow of the American College of Critical Care Medicine. Her research interests include hemodynamics and hemorheology in critically ill patients.

**Vaibhav Unhelkar** received the Ph.D. degree in autonomous systems from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2020.

He is an Assistant Professor of Computer Science at Rice University, Houston, TX, USA, where he directs the Teaming Lab. His research lies at the intersection of human modeling, explainable AI, and human-robot interaction, aiming to advance human-machine teaming. In collaboration with domain experts, he has developed robotic assistants for assembly and clinical environments, which work with and alongside humans.

**Lydia E. Kavraki** (Fellow, IEEE) received the Ph.D. degree in computer science from Stanford University, Stanford, CA, USA, in 1995.

She is the Kenneth and Audrey Kennedy University Professor of Computing with Rice University, Houston, TX, USA, and Director of the Ken Kennedy Institute. She coauthored *Principles of Robot Motion* (MIT Press, 2005), pioneered sampling-based motion planning, and leads development of the Open Motion Planning Library. Her research interests include motion planning, human-robot collaboration, and robotics/AI methods for computational biomedicine.

Dr. Kavraki is a Fellow of AAAI, AIMBE, ACM, and AAAS, and a Member of the National Academies of Engineering, Sciences, and Medicine.