

SE(3) Neural Potential Fields for 6-DoF Trajectory Planning Directly from Images Without Explicit 3D Reconstruction

Jeffrey Eiyike¹, Masoud Ataei¹, Elvis Gyaase¹, and Vikas Dhiman²

Abstract—Reaching a 6-DoF grasp pose in clutter requires a collision-free trajectory, conventionally obtained by reconstructing the scene in 3D and planning inside that reconstruction, at the cost of its accuracy and compute. Potential fields learned directly from images remove that dependency but inherit the classical weakness of artificial potential fields: where attractive and repulsive gradients cancel, the descent grazes the obstacle instead of going around it, and can stall short of the goal. We present an SE(3) neural potential field learned from posed RGB images and supervised with a *navigation function*, the geodesic distance to the grasp through free space recovered from those same images during training, which removes both failures. On two tabletop scenes, from obstacle-blocked starts executed on a UR10, the field converges within 3 cm of the grasp from every start and every path it executes is collision-free against the ground-truth geometry, against 25 % and 0 % under image supervision alone; mean clearance rises from under a centimeter to 8.6–8.8 cm and arm-link contacts fall from 20.6–50.4 % to 2.7–5.5 % of executed configurations. Executed grasp success is 90.0 % and 40.0 % on the two scenes, the residual failures being refusals of the Cartesian executor rather than of the field. Planning takes about 2 s against 67–133 s for RRT* on a reconstruction of the same images, though under a common offline harness the two are comparable: the deployed margin is the cost of collision-checking a dense reconstruction, not planner complexity.

I. INTRODUCTION

Motion planning for manipulation in unstructured scenes relies on a 3D understanding of the workspace [1], [2], obtained from depth sensors [3], multi-view reconstruction [4], [5], or implicitly by vision-to-action policies [6]. Depth is costly and incomplete [7], reconstruction is noisy and slow [8], and end-to-end policies are hard to interpret [9]. We take a middle road: inspired by neural radiance fields [5], we train a network directly from posed images to represent a *neural potential field* (NPF) over end-effector poses whose negative gradient is the plan. No explicit 3D model is built; the field only has to be accurate along the trajectories the gripper will follow, and it remains interpretable as a cost landscape.

This material is based upon work supported by the US Department of Energy (DOE), Office of Critical Minerals and Energy Innovation, Advanced Materials and Manufacturing Office under CPS Agreement 35863, and Oak Ridge National Laboratory/University of Maine SM²ART program with research and res (Copy)ources used at the Advanced Structures and Composites Center (ASCC), a University of Maine research center.

¹Jeffrey Eiyike, Masoud Ataei and Elvis Gyaase are with the Department of Electrical and Computer Engineering, University of Maine, Orono, Maine, 04469, USA {jeffrey.eiyike, masoud.ataei, elvis.gyaase}@maine.edu

²Vikas Dhiman is a Researcher with the Department of Electrical and Computer Engineering, University of Maine, Orono, Maine, 04469, USA vikas.dhiman@maine.edu

The 3D field is supervised through a novel 3D-to-2D projection (Sec. IV-B) that projects its gradient into each training view and compares it with 2D potential gradients derived from instance segmentation (Sec. IV-A). Such image-derived attraction/repulsion targets, however, inherit the classical weakness of artificial potential fields: summed attractive and repulsive gradients cancel near obstacles, so the descent grazes surfaces rather than detouring around them, and can stall at spurious minima. We therefore additionally supervise the field with a *navigation function* the geodesic distance to the grasp through the configuration-space free set recovered from the same images (Sec. IV-F) which has no minimum other than the goal, and distill it into the field with point-sampled losses. Our contributions are: (i) an SE(3) neural potential field learned from images, defined through three rigidly attached gripper points so that position and orientation are planned jointly; (ii) a 3D-to-2D potential-gradient projection that trains it from segmentation-derived 2D fields; (iii) navigation-function supervision built from a visual hull of the training views, which removes stalls and spurious minima by construction and needs no depth, mesh or CAD model; and (iv) executed Gazebo trials on the cluttered eight-object scene: from 10 obstacle-blocked start poses the navigation-function NPF reaches the grasp pose on 90.0 % of them, with 100.0 % of its executed paths collision-free against the ground-truth objects at 8.8 cm mean clearance, and plans each query in 1.9 ± 0.8 s; the same field trained from images alone is collision-free on 25.0 % of its paths, while RRT* needs 67.4 ± 1.8 s per query on a reconstruction of those images (132.9 ± 13.8 s on ground-truth geometry) and reaches 50.0 % (80.0 %) of the starts.

II. RELATED WORK

This paper lies at the intersection of potential-field methods, neural scene representations, and grasp and motion planning. We briefly review these three lines of work and position our approach with respect to each.

A. Potential Fields and Navigation Functions

Artificial potential fields [10] turn planning into gradient descent on a goal attractor plus obstacle repulsors and remain attractive for reactive control [11], but the summed field has spurious local minima and requires hand-tuned weights, usually addressed with perturbations or hybrid escape strategies [12]. Navigation functions [13] remove the minima by construction on a known free space. We learn the field from images and supervise it with a navigation function computed

from image-derived geometry, so the local-minimum problem is addressed at the target rather than at run time.

B. Planning in Neural Scene Representations

NeRFs have been used as planning substrates by treating rendered density as a collision cost [14], as a Poisson point process yielding probabilistically safe voxels [15], or as a source of reachable sets [16]; grasp-oriented variants add object-level reasoning [17], [18] or language grounding [19], [20], [1]. All keep an explicit geometric proxy between the field and the planner. We instead use the NeRF machinery only as a differentiable projection operator: the learned quantity is the SE(3) planning cost itself, and no density threshold, occupancy map, or mesh is extracted at planning time.

C. Grasp and Motion Planning

Sampling-based planners such as RRT* are general but take tens of seconds on reconstructed geometry, optimization-based planners need accurate models and initialization, and learned policies answer in milliseconds but need task-specific data. Recent work couples grasp selection with motion generation [1], [2], [21], [22], [23] or predicts discrete grasps end-to-end [24]. Our method forgoes reconstruction and yields the entire SE(3) trajectory from a single learned field.

III. BACKGROUND

A. Neural Radiance Field (NeRF)

NeRF [5] represents a scene by a density $\sigma_\theta(\mathbf{x})$ and a view-dependent color $\mathbf{c}_\theta(\mathbf{x}, \hat{\mathbf{d}})$. Along a ray $\mathbf{r}(t) = \mathbf{o} + t\hat{\mathbf{d}}$ sampled at $t_1 < \dots < t_N$, the color is rendered by quadrature,

$$\hat{C}(\mathbf{r}) \approx \sum_{i=1}^N T_i \alpha_i \mathbf{c}_i, \quad (1)$$

with transmittance $T_i = \exp(-\sum_{j<i} \sigma_j \delta_j)$, opacity $\alpha_i = 1 - e^{-\sigma_i \delta_i}$, and $\delta_i = t_{i+1} - t_i$. The networks are trained with the photometric loss $\mathcal{L}_c = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \|\hat{C}(\mathbf{r}) - C(\mathbf{r})\|_2^2$ over all rays $\mathcal{R}$ of the training images.

B. SE(3) Potential Fields

A pose is (x, q) with $x \in \mathbb{R}^3$ and a unit quaternion $q \in S^3$. Since q and $-q$ represent the same orientation, orientation distances use $d_q(q_1, q_2) = 2 \arccos(|q_1^\top q_2|)$ and poses are compared with $d_{SE(3)} = \|x - x^*\| + c d_q(q, q^*)$, $c = 0.1 \text{ m/rad}$. The potential network never consumes q directly: the pose is expanded into three rigidly attached gripper points, the center and the two finger tips, using the fixed finger offsets o_l, o_r supplied with the grasp,

$$p_c = x, \quad p_l = x + R(q) o_l, \quad p_r = x + R(q) o_r, \quad (2)$$

and a shared point-wise multilayer perceptron (MLP) ψ_θ with sinusoidal encoding gives the pose potential

$$\phi_\theta(x, q) := \psi_\theta(p_c) + \psi_\theta(p_l) + \psi_\theta(p_r). \quad (3)$$

Gradients with respect to x and q follow by automatic differentiation through the rigid coupling, so the field can act differently on each finger while preserving the gripper geometry.

IV. METHOD

Given posed RGB images of a scene, the robot segments the goal object and the obstacles with an instance detector, converts each mask into a 2D potential (attractive for the goal, repulsive for obstacles; Sec. IV-A), and trains one SE(3) neural potential field with a NeRF-like architecture whose rendered gradient must match these 2D fields (Sec. IV-B). The NeRF color loss regularizes geometry, three SE(3) goal losses establish the grasp pose as the minimum (Sec. IV-D), and a navigation function distilled from the visual hull of the same views removes spurious minima (Sec. IV-F). Although the color loss is retained, no reconstruction is extracted: planning follows the learned field directly (Sec. IV-G).

A. Observed Potential Field in 2D

For each image I_i , we segment the goal object and the obstacle objects using YOLOSeg, producing binary segmentation masks $\{\mathbf{M}_o\}_{o=1}^O = \text{YOLOSeg}(I_i)$ with $\mathbf{M}_o \in \{0, 1\}^{H \times W}$ for all $o \in \{1, \dots, O\}$. Among the segmented objects, one is the goal object, with ID g . Using each segmentation mask, we heuristically construct a 2D potential field for each object. The constants β_g, α_r and the repulsive decay are fixed once and shared across all scenes and objects (values in supplementary material); unlike classical APF deployments, no per-scene or per-object tuning is performed. Each per-object field has the resolution of the input image, $\phi_{2D,o} \in \mathbb{R}^{H \times W}$, as do the combined field and its gradients. For the goal object, we construct an attractive potential field that increases with the negative inverse of the signed distance field, $d(\mathbf{u}, \mathbf{M}_g)$,

$$\phi_{2D,g}(\mathbf{u}) := \begin{cases} -\beta_g / (d(\mathbf{u}, \mathbf{M}_g) + \epsilon), & \text{if } d(\cdot) > 0 \\ -\beta_g / \epsilon, & \text{if } d(\cdot) \leq 0. \end{cases} \quad (4)$$

where g is the goal object id, and β_g is a hyperparameter. Here the signed distance field $d(\mathbf{u}, \mathbf{M})$ is computed using the euclidean distance transform implemented in the [FastGeodis](#) library [25],

$$d(\mathbf{u}, \mathbf{M}) := \begin{cases} \min_{\mathbf{v} | \mathbf{M}[\mathbf{v}] = 1} \|\mathbf{u} - \mathbf{v}\|_2 & \text{if } \mathbf{M}[\mathbf{u}] = 0 \\ -\min_{\mathbf{v} | \mathbf{M}[\mathbf{v}] = 0} \|\mathbf{u} - \mathbf{v}\|_2 & \text{otherwise,} \end{cases} \quad (5)$$

where the mask $\mathbf{M}[\mathbf{v}] = 1$ for all points $\mathbf{v}$ inside the object. For all other obstacle objects, the repulsive potential field decays exponentially with the distance,

$$\phi_{2D,o}(\mathbf{u}) := \alpha_r \exp(-\beta_o d(\mathbf{u}, \mathbf{M}_o)), \quad \forall o \neq g, \quad (6)$$

where α_r and β_o are hyperparameters. Finally, the total observed 2D potential field is given by adding the potential fields of all objects, $\phi_{2D}(\mathbf{u}) := \sum_{o=1}^O \phi_{2D,o}(\mathbf{u})$.

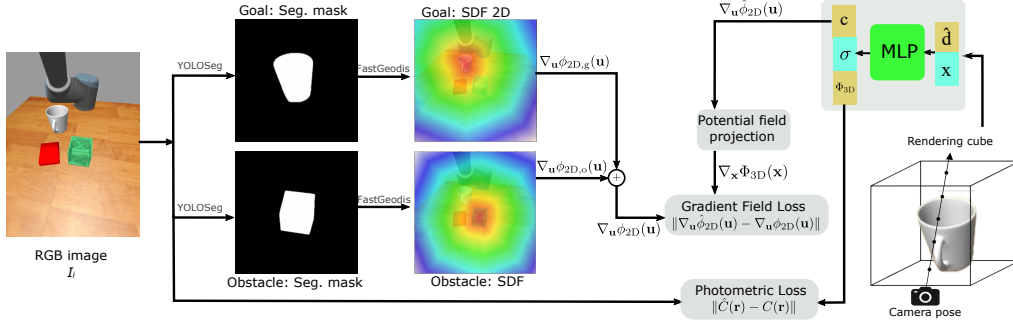


Fig. 1: Training pipeline. Instance masks of each view become 2D signed-distance potentials (SDFs; attractive for the goal and repulsive for obstacles) whose gradients supervise the SE(3) potential network: a pose is expanded to the three gripper points, ψ_θ is evaluated at each, and the projected 3D gradients are aggregated along rays by density-weighted rendering. Color, projected-gradient, and SE(3) goal losses train the field; the navigation-function losses of Sec. IV-F then refine it by point-sampled distillation, without volume rendering.

B. Potential Field Projection

Each image only constrains the *relative* attraction and repulsion of the end effector, so we relate the gradients of the 3D and 2D fields rather than their values. As in color rendering, the 3D gradient is integrated along rays with the opacity weights of Eq. (1); the query orientation is fixed to the target q^* , so every sample $\mathbf{x}_i = \mathbf{r}(t_i)$ is expanded to the three gripper points, and

$$\nabla_{\mathbf{u}} \hat{\phi}_{2D}(\mathbf{u}) := \sum_{i=1}^N T_i \alpha_i w_{\text{aniso}} \sum_{k \in \{c, l, r\}} \text{proj}_{2D}(\nabla_{p_k} \phi_{\theta, 3D}), \quad (7)$$

where $\text{proj}_{2D}(\mathbf{y}) = \pi(K^c R_w \mathbf{y})$ projects a world-frame gradient through the intrinsics and rotation of the camera with $\pi(\mathbf{v}) = \mathbf{v}_{1:2}/\mathbf{v}_3$, and $w_{\text{aniso}} = \exp(\beta(1 - (\hat{\mathbf{d}}^\top \bar{\mathbf{g}}_i)^2))$ emphasizes components perpendicular to the ray, where the image is informative. The projected-gradient loss is

$$\mathcal{L}_g = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \|\nabla_{\mathbf{u}} \hat{\phi}_{2D}(\mathbf{u}) - \nabla_{\mathbf{u}} \phi_{2D}(\mathbf{u})\|_2^2. \quad (8)$$

Supervising the projected gradient rather than the value makes the target invariant to the unobservable per-view offset of ϕ .

C. Target Grasp Pose

The 6-DoF target (x^*, q^*) and the finger offsets come from the AnyGrasp detector [26] on the training views and are stored as fixed, non-learnable buffers of the model, which removes the goal drift of the earlier position-only formulation and anchors the field to a physically meaningful grasp.

D. SE(3) Goal-Aware Losses

Three losses make $\phi_{\theta, 3D}$ a potential well centered at (x^*, q^*) . They sample poses from $\mathcal{B}$: positions from $\mathcal{N}(x^*, s_p^2 I)$, $s_p = 0.5$ m, and orientations in a geodesic ball of radius $s_q = \pi$ around q^* (20 % within 0.2 rad for local refinement). The rotation-vector gradient $\nabla_r \phi$ is the tangent-space projection of the quaternion gradient, and orientation is coupled to position with $s = 0.2$ m/rad.

Goal Potential Loss: This loss pins the potential to zero at the target pose:

$$\mathcal{L}_{\text{pot}} = \lambda_{\text{pot}} \phi_{\theta, 3D}(x^*, q^*)^2, \quad (9)$$

i.e. the three gripper points are at their target locations.

Gradient Magnitude Loss: Near the target, the field must have a strong, non-vanishing slope. We sample N SE(3) poses (x_i, q_i) around the target, convert each query quaternion to the relative rotation vector r_i , and penalize weak combined gradients:

$$\mathcal{L}_{\text{mag}} = \lambda_{\text{mag}} \mathbb{E}_{(x_i, q_i) \sim \mathcal{B}} \left[\max \left(0, \gamma_{\min} - \sqrt{\|\nabla_x \phi\|^2 + (s \|\nabla_r \phi\|)^2} \right) \right], \quad (10)$$

where $\gamma_{\min} = 0.1$, $s_p = 0.5$ m, $s_q = \pi$ rad, and $\mathcal{B}$ is the sampler defined above, which mixes 20% local orientation perturbations with 80% samples up to the full geodesic radius.

Gradient Alignment Loss: The negative gradient should point toward the target in the coupled position-rotation-vector space. Let $g = [\nabla_x \phi^\top, s \nabla_r \phi^\top]^\top$ and $t = [(x^* - x)^\top, -s r^\top]^\top$, where $-r$ is the geodesic direction back to the target orientation. Then:

$$\mathcal{L}_{\text{align}} = \lambda_{\text{align}} \mathbb{E} \left[1 - \frac{g^\top t}{\|g\| \|t\|} \right]. \quad (11)$$

Supervising the *projected gradient* rather than the raw potential value makes the target invariant to the unknown per-view offset of ϕ (only differences of ϕ are observable in a projection); the loss nevertheless compares the full gradient vectors, so both the direction and the magnitude of the learned field are constrained, and the magnitude carries the distance-to-obstacle information. Default weights are $\lambda_{\text{pot}} = 50$, $\lambda_{\text{mag}} = 10$, and $\lambda_{\text{align}} = 50$.

Orientation Bowl Loss: Because $R(q) = R(-q)$, the three-point potential is double-cover invariant and needs no term against a minimum at $-q^*$; we nevertheless keep a hinge regularizer $\mathcal{L}_{\text{bowl}}$ that requires $\phi_{\theta, 3D}(x, q) \geq \|x - x^*\| + d_q(q, q^*)$ on poses drawn over the full orientation

ball, so that ϕ grows with distance from the target and large initial orientation errors receive a usable descent direction. A multi-view projection-consistency term on the goal center is retained from the position-only model but disabled when the grasp buffer is used.

E. Network Training

The network outputs σ_θ , $\mathbf{c}_\theta$ and $\phi_{\theta,3D}$ and is trained on

$$\begin{aligned} \mathcal{L}_{\text{total}} = & \lambda_c \mathcal{L}_c + \lambda_g \mathcal{L}_g + \mathcal{L}_{\text{pot}} + \mathcal{L}_{\text{mag}} \\ & + \mathcal{L}_{\text{align}} + \lambda_{\text{bowl}} \mathcal{L}_{\text{bowl}} + \lambda_{\text{mv}} \mathcal{L}_{\text{mv}}, \end{aligned} \quad (12)$$

with the SE(3) terms updating only the potential network; gradients are clipped to unit norm and accumulated over four mini-batches. The complete architecture and training configuration (4-layer, width-512 point MLP, Adam 5×10^{-4} , 1024 rays per step, 8 epochs, loss weights $\lambda_c=1$, $\lambda_g=2$, $\lambda_{\text{pot}}=50$, $\lambda_{\text{mag}}=10$, $\lambda_{\text{align}}=50$, $\lambda_{\text{bowl}}=20$) is given in the supplementary material; nothing is tuned per scene except the target-object class.

F. Navigation-Function Supervision: Removing Stalls and Spurious Minima

A potential field assembled from a goal attractor and obstacle repulsors, the construction behind the image-space targets above, inherits the classical weakness of artificial potential fields: wherever the attractive and repulsive gradients cancel, the field has a *spurious* minimum and gradient descent stalls short of the goal, typically in the shadow of an obstacle or in a corridor between two of them. In our executed trials, this appears as the Stall column of Table II, and the same cancellation is what lets the descent graze surfaces on the approach: the field only pushes *away*, it never says *around*.

A navigation function [13] removes this failure by construction: it is a scalar field on the free space whose only minimum is the goal, so following its negative gradient from any free configuration reaches the goal without entering an obstacle. We supervise the learned field with such a function. Obstacle geometry is recovered from the same images used for training, as the visual hull of the detector masks over the posed views (no depth, no reconstruction mesh); the hull is inflated by the gripper radius (7 cm) and the table becomes a floor, which yields the configuration-space free set $\mathcal{F}$. The navigation function is the geodesic distance to the grasp pose inside $\mathcal{F}$, $D(x) = \min_{\gamma \subset \mathcal{F}} \text{len}(\gamma(x \rightarrow x^*))$, computed once by Dijkstra on a 1 cm voxel grid. This geometry is a *training-time* construct only: the hull is a coarse silhouette-carved occupancy envelope, built from the masks already used by the 2D supervision and never refined into a surface. Neither the hull nor D exists at run time; planning remains gradient descent on the learned field alone, with no geometric model in the loop. What the method avoids is a reconstruction on which the *planner* depends, not the use of multi-view consistency during training. D has no spurious minima: for every $x \neq x^*$ the first step of the shortest path lowers D , so a descent direction exists everywhere in $\mathcal{F}$, and level sets of D bend around obstacles instead of terminating on them. Clearance is not a tuned constant but a property of $\mathcal{F}$: every

path of D keeps at least the inflation radius from the observed surfaces.

The field distills D through point-sampled losses that need no volume rendering, so the supervision is placed exactly where the obstacles are. At random poses (x, q) in $\mathcal{F}$ we require (i) *direction*: $1 - \cos(\nabla_x \phi_{\theta,3D}(x, q), \nabla D(x))$; (ii) *descent progress*: $\max(0, 1 - \nabla_x \phi_{\theta,3D} \cdot \hat{\nabla} D)^2$, a minimum slope along the navigation direction, which is the term that forbids flat regions and hence stalls, and is sampled preferentially where the current slope is smallest; (iii) a scale term that keeps $\|\nabla_x \phi_{\theta,3D}\|$ at the value of the image-trained field; and (iv) a goal-basin anchor that preserves the learned grasp basin within 10 cm of x^* . The repulsive head is trained to the bounded distance potential $\rho(\text{sdf}(x))$, which makes it a clearance predictor at run time.

G. Trajectory Planning Using the SE(3) Neural Potential Field

Planning is damped gradient descent on $\phi_{\theta,3D}$: position and quaternion gradients are obtained by automatic differentiation through the three-point mapping, smoothed by an exponential moving average and momentum, clipped, and applied with a step that shrinks within 0.2 m and 0.1 m of the target, the quaternion being re-normalized after every update. The rollout stops when $d_{SE(3)} < \epsilon$. At run time only this field is evaluated; the learned repulsive head of Sec. IV-F is not needed for steering.

V. EXPERIMENTAL EVALUATION

A. Datasets

Experiments run in Gazebo with a 6-DoF UR10 and a Robotiq three-point gripper. Two earlier position-only scenes (**Balls, 3 Obj.**) are retained in Table I for reference. For the 6-DoF evaluation we collected two RGB-D scenes in which the end effector must reach a *grasp pose*: **3 Obj. (book)**, where the book is the target and a cup and a crate are obstacles, and **8 Obj. (orange)**, a cluttered tabletop of eight everyday objects with an orange as the target (67 and 83 training views, camera orbit centered on the target). The free corridor around the target is wider than the gripper (nearest obstacle 20.0 cm and 20.8 cm from the goal, 9 cm gripper). Real-robot rollouts use the setup of Fig. 3.

B. Evaluation Metrics

We evaluate trajectory-planning performance using the following metrics [27], [18]: success rate (SR), collision-free rate, trajectory length, planning time, execution time, final orientation error, and obstacle clearance. A trial is successful if the final end-effector pose is within 3 cm and 15° of the target grasp pose. The collision-free rate against ground-truth geometry (CF_{GT}) is the fraction of executed TCP trajectories that do not intersect any ground-truth obstacle mesh, excluding the target object and support table. Clearance is the minimum TCP-to-obstacle distance along the trajectory. To additionally evaluate whole-arm safety, Link coll. reports the fraction of executed arm configurations in which any robot link intersects a ground-truth obstacle. This is computed by

forward-kinematics replay of the recorded joint trajectories, modeling each link as a capsule between consecutive joint origins and the gripper as a capsule along the tool axis. Trajectory length is the accumulated Euclidean distance between consecutive TCP waypoints; planning time measures the time required to generate a trajectory, and execution time measures the corresponding robot motion time. Orientation error (OE) is the geodesic quaternion distance $d_q(q_{\text{final}}, q^*)$ between the final and target orientations.

C. Baseline and Comparison Methods

We compare our NPF (SE(3) field learned from multi-view images, trajectories by damped gradient descent, Sec. IV) with the reconstruct-then-plan pipeline: a scene mesh from a vanilla NeRF trained on the same RGB-D views (any reconstruction method could be substituted [28], [29]), the target from YOLO [30] and AnyGrasp, and RRT* [31] with a length-plus-clearance cost and a 2 cm collision margin on that mesh (target geometry excluded). Unlike our earlier evaluation, which gave the baseline the ground-truth models, the baseline here plans on the mesh the pipeline actually produced, and the ground-truth-geometry variant is reported separately as an upper bound; both planners therefore consume the same perception, which is the only comparison that isolates the planner.

VI. RESULTS AND ANALYSIS

A. Quantitative Results

TABLE I: Executed trajectory-planning performance in Gazebo. Comparison of the image-supervised NPF, navigation-function NPF, and RRT* using reconstructed or ground-truth geometry over 10 SE(3) start poses per scene. Success requires a final pose within 3 cm and 15° of the target. N_{exec} : trials in which the arm executed a motion; in the remainder no motion took place (the Cartesian executor aborted the plan, or, for the RRT* rows, the planner found no path); CF_{GT} denotes collision-free executed TCP paths; Link coll. denotes the percentage of executed arm configurations in collision; OE is the final orientation error; † RRT* interpolates to the goal quaternion, so its final orientation error is zero by construction rather than by search. Values are mean $\pm$ std across successful starts.

Scene	Method	SR (%)	N_{exec}	CF_{GT} (%)	Plan. T. (s)	Exec. T. (s)	Traj. (m)	Link coll.	OE ($^\circ$)
3 Obj. (book)	Recon. mesh + RRT*	80.0	8/10	100.0	99.5 $\pm$ 2.6	3.8 $\pm$ 1.7	2.83 $\pm$ 4.14	1.8	0.0 †
3 Obj. (book)	GT geometry + RRT*	80.0	8/10	87.5	124.4 $\pm$ 3.5	4.5 $\pm$ 2.0	2.79 $\pm$ 3.40	1.5	0.0 †
3 Obj. (book)	NPF SE(3), image supervision only (prior)	50.0	6/10	0.0	1.2 $\pm$ 0.3	1.9 $\pm$ 0.3	0.79 $\pm$ 0.07	50.4	3.5 $\pm$ 2.5
3 Obj. (book)	NPF SE(3) + nav. function (ours)	40.0	6/10	100.0	2.2 $\pm$ 0.3	3.3 $\pm$ 0.9	0.61 $\pm$ 0.09	2.7	2.8 $\pm$ 2.1
8 Obj. (orange)	Recon. mesh + RRT*	50.0	5/10	100.0	67.4 $\pm$ 1.8	3.2 $\pm$ 0.3	0.86 $\pm$ 0.11	20.6	0.0 †
8 Obj. (orange)	GT geometry + RRT*	80.0	9/10	100.0	132.9 $\pm$ 13.8	2.8 $\pm$ 0.7	0.73 $\pm$ 0.23	12.0	0.0 †
8 Obj. (orange)	NPF SE(3), image supervision only (prior)	80.0	8/10	25.0	0.8	4.3 $\pm$ 0.5	0.67 $\pm$ 0.22	20.6	3.8
8 Obj. (orange)	NPF SE(3) + nav. function (ours)	90.0	9/10	100.0	1.9 $\pm$ 0.8	5.4 $\pm$ 1.2	0.92 $\pm$ 0.30	5.5	1.0 $\pm$ 1.5

Tables I and II compare, on the same start poses and the same perception, the image-supervised NPF, the NPF with navigation-function supervision, and RRT* planning on the reconstructed mesh (with the ground-truth-geometry variant as an upper bound). Table II repeats the comparison offline, without the simulator, with RRT* over five seeds per start. Image supervision alone, which extends the supervision of the earlier position-only formulation to SE(3), already

reaches the grasp reliably and plans two orders of magnitude faster than RRT* (0.8-1.2 s against 67-133 s), but it does not produce a detour: on both scenes the descent drives toward the goal and passes within half a centimeter of the obstacle blocking the start, and the object it touches is precisely that blocker in 14 of 14 book and 14 of 20 eight-object rollouts, leaving its executed paths collision-free in only 25 % (8 objects) and 0 % (book) of trials. The same supervision is collision-free on 95 % of *unblocked* starts but only 25 % of blocked ones in the position-only setting, so the grazing is a property of image-space repulsion a thin band at the projected mask boundary rather than of the SE(3) extension or the three-point gripper model; a fixed-start protocol hides it because the straight-line approach is already clear. The navigation function changes the behavior rather than the margin, and without changing the run-time procedure: every executed path in both scenes is collision-free against the ground-truth geometry, mean clearance rises to 8.6-8.8 cm (against 2-3 cm for RRT*, which plans with a 2 cm margin), arm-body contacts fall from 20.6 % to 5.5 % of configurations, and planning stays at about 2 s. The planning-only protocol of Table II isolates this safety gain from the run-time: there the separation is entirely in collision avoidance.6-2.6 s), and the separation is entirely in collision avoidance, the navigation-function field reaches every start of both scenes without penetrating an object, at 8.2-9.7 cm clearance, against 1.8-2.1 cm for RRT* (which plans to a 2 cm margin) and under 0.5 cm for image supervision alone. The deployed speed-up reflects the cost of querying a dense reconstruction with gripper-body checks at every waypoint; the safety gain holds in both settings. Executed success is 90 % in the 8-object scene and is executor-bounded in the book scene (Sec. VI-D). Execution is slower than RRT* (5 s against 3 s) because of conservative gradient steps, and paths are of comparable length.

B. Executed Trials in Gazebo

We execute every planned SE(3) trajectory on a UR10 with a Robotiq three-point gripper in Gazebo under MoveIt Cartesian control. Each method is evaluated from the same 10 arbitrary start poses per scene, with varying initial positions and orientations. Fig. 2 shows representative executed trajectories from one obstacle-blocked start in each scene. Both methods consume the same perception: NPF plans directly on the learned field, while RRT* plans on the NeRF-reconstructed mesh (2 cm margin, KD-tree over the surface samples) or, as an upper bound, on the ground-truth object meshes. We record the executed TCP path and the full joint trajectory of every trial so that both end-effector and full-arm clearance can be audited offline against the ground-truth geometry. In all executed trials the learned repulsive head is left off at inference: with the navigation-function field it adds no clearance and only blocks the grasp approach (Table III, last rows).

TABLE II: **Planning-only comparison under a common offline protocol.** All methods are evaluated on the same start sets and collision geometry using a single-CPU harness, with RRT* averaged over five random seeds per start. CF-SR_{mesh} and CF-SR_{GT} denote collision-free success against reconstructed and ground-truth geometry, respectively; Clear_{GT} is the minimum TCP distance to ground-truth obstacles. Values are mean $\pm$ std across starts.

Scene	Method	N	SR (%)	CF-SR _{mesh} (%)	CF-SR _{GT} (%)	Stall (%)	Traj. (m)	Clear _{GT} (m)
3 Obj. (book)	Recon. mesh + RRT* (5 seeds)	14	100.0	92.9	100.0	0.0	0.52 $\pm$ 0.16	0.021 $\pm$ 0.013
3 Obj. (book)	NPF SE(3), image supervision only (prior)	14	100.0	14.3	0.0	0.0	0.43 $\pm$ 0.10	0.004 $\pm$ 0.003
3 Obj. (book)	NPF SE(3) + nav. function (ours)	14	100.0	100.0	100.0	0.0	0.81 $\pm$ 0.23	0.082 $\pm$ 0.012
8 Obj. (orange)	Recon. mesh + RRT* (5 seeds)	20	100.0	60.0	100.0	0.0	0.89 $\pm$ 0.32	0.018 $\pm$ 0.013
8 Obj. (orange)	NPF SE(3), image supervision only (prior)	20	100.0	40.0	25.0	0.0	0.82 $\pm$ 0.20	0.005 $\pm$ 0.003
8 Obj. (orange)	NPF SE(3) + nav. function (ours)	20	100.0	45.0	100.0	0.0	0.91 $\pm$ 0.26	0.097 $\pm$ 0.020

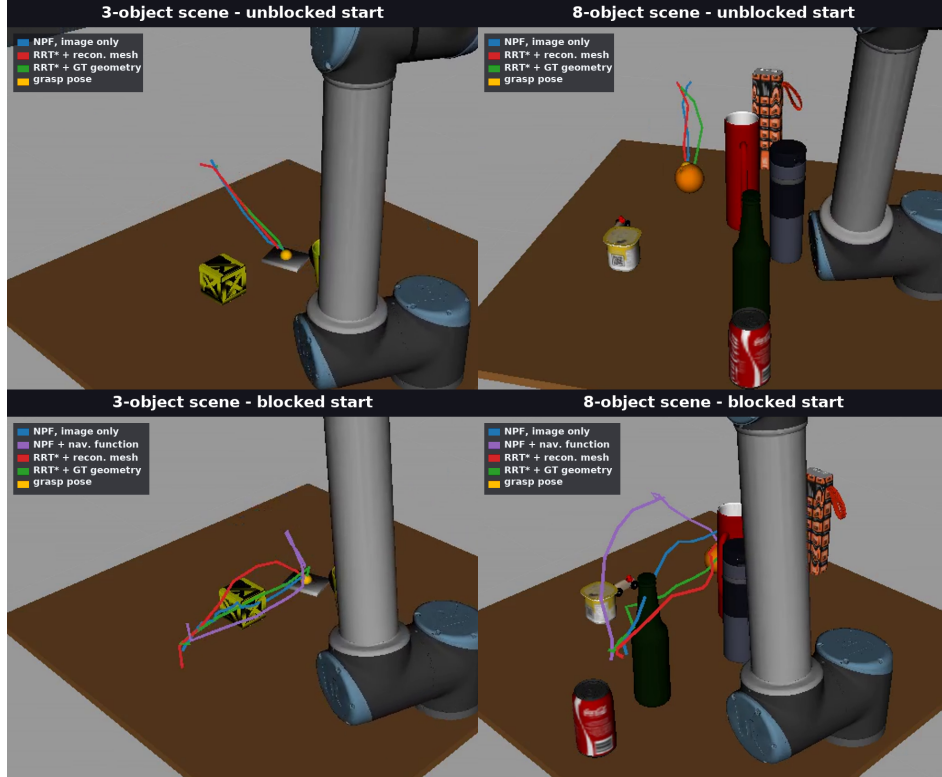


Fig. 2: Executed trajectories in both scenes, one start per case, rendered in the robot’s frame. Top row: unblocked starts, where a straight line from the start to the grasp is already clear and every planner reaches it. Bottom row: obstacle-blocked starts, where the image-supervised field drives at the goal and through the blocking object, while the navigation-function field detours around it; the two RRT* baselines are shown for reference. Blue: NPF with image supervision only. Purple: NPF with navigation-function supervision. Red: RRT* on the reconstructed mesh. Green: RRT* on the ground-truth geometry. Gold: the grasp pose. The four panels share a common field of view.

C. Training Time and Amortized Cost

The one-off cost of each pipeline is NeRF reconstruction of 10.8 h and 8.53 h against NPF training of 7.4 h and 6.0 h on one RTX 3090, from the same images (totals 13.1 h vs. 9.7 h and 9.9 h vs. 8.0 h including pre-processing). Both are paid once per scene; the fair comparison is the amortized time per query, $t_{\text{offline}}/N + t_{\text{plan}}$, which NPF wins after a handful of grasps because a plan is a single forward pass (1.9 ± 0.8 s against 67.4 ± 1.8 s).

D. Failure Modes and Local Minima

With image supervision alone the NPF reaches the grasp pose reliably but grazes obstacles: the summed attractive and

repulsive gradients cancel rather than steer around (CF_{GT} and Link coll. in Table I). The other classical failure of potential fields descent halting at a spurious minimum does not appear in these two scenes (Stall is zero for both variants in Table II); the wide near-goal sampling radius of the SE(3) losses makes the attractor effectively global, so the field always reaches the goal, it simply reaches it through the obstacle rather than around it. The navigation-function supervision of Sec. IV-F addresses exactly this: its target has no minimum other than the goal; the distilled network inherits no such guarantee, so the descent-progress term enforces it empirically by forbidding flat regions in

the learned approximation. On these starts it removes the penetrations entirely (100.0% CF_{GT} , mean clearance 8.8 cm in the 8-object scene) and drops the arm-body contacts from 20.6% to 5.5% of executed configurations. Neither variant stalls on this start setboth fields converge to within 3 cm of the grasp on every rollout, and the residual failures are refusals of the Cartesian executor rather than field behaviour. Scored over all ten starts of each scene rather than only the executed ones, the field reaches the grasp 10/10 times and penetrates no object: zero collisions, with clearance beyond 7 cm on 10/10 paths (8-object). The book scene, whose starts are laterally blocked by the cup and the crate, separates the field from the executor: the image-supervised NPF reaches only half of the starts and every one of its executed paths hits the cup or the crate, whereas the navigation-function NPF converges within 3 cm in all ten rollouts and every executed path is collision-free with $13\times$ the clearance. Its lower executed success rate on this set (40%) is entirely an execution-layer effect the Cartesian executor aborted four of the converged rollouts before the arm had moved, and two others executed the full path but finished 2.6 and 7.0 cm from the grasprather than a field failure; the image-supervised row is bounded the same way: four of its ten rollouts were aborted before motion. The asymmetry with RRT* is in what each planner hands the executor: 7–8 widely spaced waypoints against 26–42 densely specified poses. The refusals persist with collision checking disabled, so they are kinematic, and are not a reorientation effect: the baseline executes even with 129° of interpolated rotation.

E. Ablation of the Navigation-Function Supervision

Table III varies the ingredients of Sec. IV-F on the 8-object scene, each variant starting from the same image-trained checkpoint and audited offline from the 10 Gazebo starts (collide: TCP within the 4.5 cm gripper radius of an obstacle; clear: beyond 7 cm). The geometry source dominates: targets from the NeRF reconstruction wall off the goal with reconstruction artifacts and no start reaches it, whereas the visual hull recovers every start. The inflation radius trades margin against reach (3.5 cm never collides but leaves half the paths in the graze band, 10 cm pinches the approach lanes; 7 cm is the working point); further distillation rounds convert collisions into clearance, the descent-progress term removes the residual stalls (7/10 to 10/10 reach).

F. Loss-Term Ablations

Table IV ablates each loss term in the position-only setting (each variant retrained from scratch; 20 fixed and 20 obstacle-blocked starts). The multi-view goal and gradient-alignment terms are structurally necessary without them the learned goal drifts by 62.8 cm or the goal is not a basin while the photometric term is dispensable for planning, supporting the no-reconstruction thesis. The projected-gradient, potential and magnitude terms carry the obstacle information: removing any of them roughly halves the collision-free rate on blocked starts. The anisotropic weight w_{aniso} is not justified empirically (removing it *raises* the blocked-start

TABLE III: Ablation of the navigation-function supervision (8-object scene, offline ground-truth audit, 10 Gazebo start poses). Collide/graze/clear: minimum TCP-obstacle distance < 4.5 cm, $4.5\text{--}7$ cm, ≥ 7 cm.

Factor	Setting	Reach	Collide	Graze	Clear
Inflation radius (3 rounds)	3.5 cm	10/10	0	5	5
	7 cm	8/10	0	4	6
	10 cm	10/10	1	1	8
Distillation rounds (7 cm)	1	10/10	2	4	4
	3	8/10	0	4	6
	8 (model)	10/10	0	1	9
Geometry source (7 cm, 3 rounds)	visual hull	8/10	0	4	6
	NeRF recon. mesh	0/10	6	2	2
Descent-progress term	without	7/10	0	2	8
	with (model)	10/10	0	1	9

TABLE IV: Loss-term ablation (position-only field, shared terms; each row a single training seed). SR and collision-free rates in %, on 20 fixed / 20 obstacle-blocked (blk) starts vs. ground-truth geometry; Grad.q. is the validation gradient-alignment score.

Variant (3 Obj., position-only)	SR	CF_{GT}	CF_{GT} blk	Grad. q.
Full model	100.0	95.0	25.0	0.859
w/o $\mathcal{L}_c$ (photometric)	100.0	95.0	20.0	0.863
w/o $\mathcal{L}_g$ (gradient field)	100.0	95.0	10.0	0.694
w/o $\mathcal{L}_{\text{pot}}$	100.0	90.0	10.0	0.852
w/o $\mathcal{L}_{\text{mag}}$	100.0	95.0	5.0	0.853
w/o $\mathcal{L}_{\text{align}}$	0.0	0.0	0.0	0.845
w/o $\mathcal{L}_{\text{mv}}$	0.0	0.0	0.0	0.000
w/o w_{aniso}	100.0	100.0	60.0	0.760

collision-free rate). With matched gradient steps, 8 training views already match 64, and the supervision is robust to segmentation noise: eroding or dilating the masks by 10 px, or dropping the goal mask in 20% of views, leaves every metric within noise of the clean model. The remaining variants (view counts, segmentation noise, repulsive decay, near-goal radius, and the 8-object scene) are tabulated in the supplementary material.

G. Qualitative Analysis

Fig. 3 shows rollouts from start to grasp on the real robot, using the position-only field; because the field is followed continuously rather than sampled, the executed paths are spatially smoother than the RRT* paths of Fig. 2. These hardware runs are demonstrations of the earlier position-only formulation: the SE(3) formulation of this paper is evaluated in the executed simulation trials reported above, and quantitative hardware trials remain future work.

VII. DISCUSSION AND LIMITATIONS

Reconstruct-then-plan pipelines commit early to a hard geometric representation and pay for it in reconstruction time (10.8 h and 8.53 h of NeRF training here) and in planner queries that must check a dense model. Our field is trained once from the same images, poorly observed regions contribute weak, smooth gradients rather than a possibly wrong surface, and a plan is a sequence of network evaluations; the navigation-function supervision keeps this architecture,

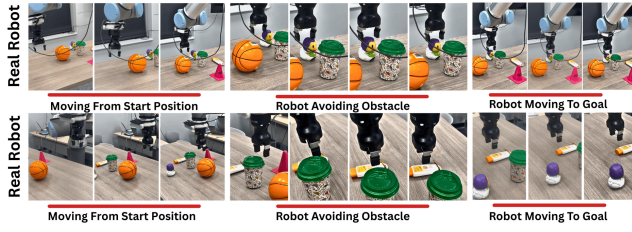


Fig. 3: Real-robot rollouts of the *position-only* field, to two different targets: the arm leaves the start pose, avoids the intervening objects, and reaches the goal. Hardware runs use the position-only formulation; the SE(3) formulation of this paper is evaluated in executed simulation trials (Tables I–II). These rollouts are demonstrations and are reported as such (Sec. VI-D).

running the planner once offline on image-derived geometry and evaluating only the distilled field at run time. Limitations: executed success is bounded by the Cartesian executor rather than by the field (Sec. VI-D); execution is slower than RRT* because of conservative gradient steps; the target pose and the visual hull inherit the errors of AnyGrasp and of the instance detector; and performance depends on a scene’s training distribution, as for reconstruction systems. Adaptive step sizing, learned velocity control and dynamic scenes are natural next steps.

VIII. CONCLUSION

We presented an SE(3) neural potential field learned directly from images, whose gradient plans 6-DoF grasp reaches without explicit 3D reconstruction, and showed how supervising it with a navigation function computed on image-derived geometry removes the spurious minima and grazing of classical potential fields. On executed Gazebo trials the navigation-function NPF is collision-free against the ground-truth geometry in every executed path of both scenes (from 25 % and 0 % with image supervision alone), keeps 8.6–8.8 cm of clearance, and plans in about 2 s against 67–133 s for RRT* on the reconstruction, a margin set by dense collision checking rather than planner complexity (Table II).

REFERENCES

- [1] Y. Xiang, S. H. Allu, R. Peddi, T. Summers, and V. Gogate, “Grasping trajectory optimization with point clouds,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 9885–9892.
- [2] B. H. Leebron, K. Ren, Y. Chen, and K. Hang, “B4p: Simultaneous grasp and motion planning for object placement via parallelized bidirectional forests and path repair,” *arXiv preprint arXiv:2504.04598*, 2025.
- [3] M. K. Ali, A. Rajput, M. Shahzad, F. Khan, F. Akhtar, and A. Börner, “Multi-sensor depth fusion framework for real-time 3d reconstruction,” *Ieee Access*, vol. 7, pp. 136 471–136 480, 2019.
- [4] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, “Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera,” in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.
- [5] 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,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

- [6] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.
- [7] Z. Xie, X. Yu, X. Gao, K. Li, and S. Shen, “Recent advances in conventional and deep learning-based depth completion: A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 3, pp. 3395–3415, 2024.
- [8] C. Bi, M. Teng, T. Xie, K. Li, and J. Yang, “Jasrnet: Learning joint adaptive sampling and reconstruction for depth sensing,” *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 2, pp. 1–23, 2025.
- [9] T.-H. Wang, W. Xiao, T. Seyde, R. Hasani, and D. Rus, “Measuring interpretability of neural policies of robots with disentangled representation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 602–641.
- [10] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.
- [11] L. Wang, Z. Sun, Y. Wang, J. Wang, Z. Zhao, C. Yang, and C. Yan, “A pre-grasping motion planning method based on improved artificial potential field for continuum robots,” *Sensors*, vol. 23, no. 22, p. 9105, 2023.
- [12] L. Hao *et al.*, “Improved potential field path planning via primal-dual neural optimization for surgical robots,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2343–2354, 2022.
- [13] E. Rimon and D. E. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
- [14] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, “Vision-only robot navigation in a neural radiance world,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.
- [15] T. Chen, P. Culbertson, and M. Schwager, “CATNIPS: Collision avoidance through neural implicit probabilistic scenes,” *IEEE Transactions on Robotics*, vol. 40, pp. 2712–2728, 2024.
- [16] M. Thapliyal *et al.*, “Safe navigation using neural radiance fields via reachable sets,” *arXiv preprint*, 2026.
- [17] L. Wang, X. Meng, Y. Xiang, and D. Fox, “Hierarchical policies for cluttered-scene grasping with latent plans,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2883–2890, 2022.
- [18] P. Huang, Y. Shaoul, and J. Li, “Benchmarking shortcutting techniques for multi-robot-arm motion planning,” *arXiv preprint arXiv:2508.05027*, 2025.
- [19] M. Makarova, Q. Liu, and D. Tsetserukou, “Diffusionrl: Efficient training of diffusion policies for robotic grasping using rl-adapted large-scale datasets,” *arXiv preprint arXiv:2505.18876*, 2025.
- [20] Y. Zheng, X. Chen, Y. Zheng, S. Gu, R. Yang, B. Jin, P. Li, C. Zhong, Z. Wang, L. Liu *et al.*, “Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping,” *IEEE Robotics and Automation Letters*, 2024.
- [21] M. N. Vu, F. Grandeur, and A. Nguyen, “Online trajectory replanner for dynamically grasping irregular objects,” *arXiv preprint arXiv:2501.17968*, 2025.
- [22] Y. Huang, T. Davies, J. Yan, J. Sun, X. Chen, and L. Hu, “Spatial robograsp: Generalized robotic grasping control policy,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.20814>
- [23] Q. Liang, B. Cai, R. He, H. Li, T. Teng, H. Duan, C. Huang, and R. Zeng, “Whole-body coordination for dynamic object grasping with legged manipulators,” *arXiv preprint arXiv:2508.08328*, 2025.
- [24] B. Zhao, L. Chang, C. Wu, and Z. Liu, “Robot multi-target high performance grasping detection based on random sub-path fusion,” *Scientific Reports*, vol. 15, no. 1, p. 8709, 2025.
- [25] M. Asad, R. Dorent, and T. Vercauteren, “Fastgeodis: Fast generalised geodesic distance transform,” *Journal of Open Source Software*, vol. 7, no. 79, p. 4532, 2022. [Online]. Available: <https://doi.org/10.21105/joss.04532>
- [26] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, “Anygrasp: Robust and efficient grasp perception in spatial and temporal domains,” *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3929–3945, 2023.
- [27] S. Yang, P. Liu, and N. Pears, “Benchmarking of robot arm motion planning in cluttered environments,” in *2023 28th International Conference on Automation and Computing (ICAC)*. IEEE, 2023, pp. 1–6.
- [28] V. Leroy, Y. Cabon, and J. Revaud, “Grounding image matching in 3d

- with mast3r,” in *European Conference on Computer Vision*. Springer, 2024, pp. 71–91.
- [29] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, “Vggt: Visual geometry grounded transformer,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5294–5306.
- [30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [31] I. Noreen, A. Khan, Z. Habib *et al.*, “Optimal path planning using rrt* based approaches: a survey and future directions,” *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, pp. 97–107, 2016.