2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

SwitchOpt: Trajectory Optimization with Adaptive Grasp Target Switching

Elisabeth Menendez[1], Santiago Martínez[1], and Carlos Balaguer[1]

Abstract— **We address the problem of trajectory optimization in scenarios where multiple grasp targets are available for the same object. We introduce SwitchOpt, an adaptive optimization strategy that dynamically switches between grasp targets during trajectory optimization. Instead of committing to a single candidate, SwitchOpt monitors progress step by step using a merit function that captures trajectory quality and constraint satisfaction. A prediction horizon is used to assess whether the current trajectory is likely to improve further, while a minimum-stay mechanism ensures sufficient refinement before considering a switch. Whenever a switch is considered, SwitchOpt reconstructs candidate trajectories by combining the current head with interpolated tails toward alternative grasp targets, and evaluates each of these full trajectories with the same merit function over the prediction horizon. If the best candidate is predicted to outperform continuing toward the current target, that target is selected as the new goal and its reconstructed trajectory is used as the new initialization, allowing the solver to continue from a promising adapted trajectory. This principled selection strategy balances local exploitation of the current target with structured exploration of alternative grasp poses, maintaining optimization continuity between switches. Experiments in simulation demonstrate that SwitchOpt improves final trajectory quality, accelerates convergence, and increases feasibility in multi-target trajectory optimization.**

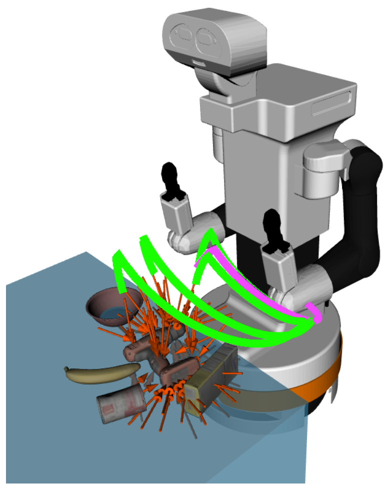

Fig. 1. While SwitchOpt optimizes the current target (pink), it conceptually evaluates alternative grasp targets. Green curves show the predicted motion after H steps if each candidate were pursued independently. These are for visualization only, SwitchOpt actually evaluates targets via deterministic tail reconstruction and short-horizon merit prediction.

I. INTRODUCTION

In robot manipulation, planning a robot trajectory to grasp an object is a fundamental research problem. The challenge lies in jointly addressing motion planning, which aims to find a collision-free arm trajectory, and grasp planning, which decides how to grasp the target object. Traditionally, these problems have been tackled separately. Motion planning approaches focus on reaching a given end-effector goal while avoiding obstacles, using either sampling-based planners such as Rapidly-exploring Random Trees (RRT) [1], Fast Marching Trees (FMT) [2], or optimization-based methods such as CHOMP [3] and Sequential Convex Optimization (TrajOpt) [4]. Grasp planning methods, such as GraspIt! [5], 6D GraspNet [6] or SE(3)-DiffusionFields [7], synthesize feasible grasp poses for a robotic gripper given object models or point clouds, but do not consider the arm motion required to reach them.

Combining grasp planning and motion planning directly addresses the robotic grasping problem. A common pipeline first generates candidate grasps of the target object, then plans a trajectory to reach one of them. A naive approach tests these grasps sequentially until it finds a collision-free trajectory and a sufficiently high-quality solution, which is complete but becomes inefficient when the grasp set is large. More advanced approaches couple the two stages. Several works have explored planning methods that account for multiple possible grasp poses. Examples include extending CHOMP to support goal sets as end-point constraints [8], combining sampling and gradient-based optimization to reach sampled grasps [9], or formulating grasp and manipulation planning jointly as an optimal control problem [10]. Other approaches integrate grasp quality predictors such as Dex-Net [11] with SQP-based trajectory optimization [12], evaluate alternative grasps using quality metrics [13], construct occupancy maps from point clouds for planning in clutter [14], or define goal regions in the manipulator workspace [15]. A related approach [16] combines random grasp pose sampling with repeated full TrajOpt runs, incorporating a waypoint commitment strategy to incrementally refine the final trajectory. However, their algorithm treats TrajOpt as a black box and restarts the optimization from scratch for each new grasp. While some planners, such as OMG [17], support online goal selection, they do so within CHOMP-based frameworks or using specialized goal-set projections. To our knowledge, no existing method performs grasp-target switching inside a TrajOpt optimization run by updating only the target pose and continuing the ongoing

[1]Robotics Lab, Department of Systems Engineering and Automation, Universidad Carlos III de Madrid (UC3M), {emenende, scasa, balaguer}@ing.uc3m.es.

optimization instead of restarting TrajOpt for each grasp.

In this paper, we propose SwitchOpt, an adaptive target switching strategy for grasp trajectory planning. The key idea is to refine the current trajectory step by step toward its present target, while simultaneously evaluating short-horizon continuations predicted toward alternative grasp poses. As illustrated in Figure 1, the optimizer progresses along one candidate (pink), while the green trajectories conceptually depict how motion would evolve if other grasp targets were pursued. In practice, SwitchOpt does not optimize these alternatives; instead, it evaluates each candidate through deterministic tail reconstructions and short-horizon merit prediction. These evaluations provide a lookahead on whether continuing toward the current goal remains cost-effective, or whether switching would yield better progress. Apart from a basic feasibility check for inverse kinematics, we do not pre-rank or externally score grasp candidates. Instead, the method relies entirely on these evaluations: if a different target shows a more promising cost trend, the optimizer switches while reusing the feasible trajectory prefix built so far, using it as initialization while reconstructing only the remaining tail toward the new goal. This enables efficient exploration of the grasp set while still exploiting already promising solutions.

We evaluate our approach in simulated experiments with the dual-arm TIAGo++ robot on five tabletop scenes from the SceneReplica benchmark [18], using precomputed grasp sets for 16 YCB objects [19]. All methods are assessed under identical cost terms and constraints. Results demonstrate improvements in merit value, convergence speed, and feasibility. The contributions of this paper are: 1) the SwitchOpt strategy for adaptive target switching during trajectory optimization, 2) a horizon-based evaluation of candidate tails that guides switching without pre-ranking grasps, and 3) validation in simulation with the dual-arm TIAGo++ robot across multiple scenes and objects.

II. METHODOLOGY

A. Problem Formulation

We formulate trajectory optimization for grasping as the problem of finding a collision-free and feasible motion toward one of several candidate grasp poses. Let $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_K\}$ denote the set of feasible end-effector configurations for grasping the object. To this end, we optimize the joint-space trajectory $\mathbf{x} = [\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_N]$, where $\mathbf{q}_i \in \mathbb{R}^d$ is the robot configuration at the waypoint i, d is the number of joints, and N is the number of waypoints. Unlike standard formulations with a fixed grasp target, our method may switch among candidates $\mathbf{T}_k$ during optimization (see Sec. II-B).

The optimization problem is written as:

$$\begin{aligned}\min_{\mathbf{x}} \quad & w_{\text{vel}} J_{\text{vel}}(\mathbf{x}) + w_{\text{cart}} J_{\text{cart}}(\mathbf{x}) \\ & + w_{\text{goal}} J_{\text{goal}}(\mathbf{x}) + w_{\text{stand}} J_{\text{standoff}}(\mathbf{x})\end{aligned} \tag{1}$$

$$\begin{aligned}\text{subject to} \quad & \mathbf{q}_0 = \mathbf{q}_{\text{start}} \\ & \mathbf{q}_{\min} \leq \mathbf{q}_i \leq \mathbf{q}_{\max} \quad \forall i \\ & \text{Collision}(\mathbf{q}_i) \leq 0 \quad \forall i\end{aligned} \tag{2}$$

Cost Terms: The objective in Eq. 1 is composed of multiple cost terms, each encouraging desirable trajectory properties. These terms act together to ensure smooth and task-specific motions. We detail them below.

a) Joint Velocity Cost: Penalizes joint-space motion between consecutive waypoints:

$$J_{\text{vel}} = \sum_{i=1}^{N} \|\mathbf{q}_i - \mathbf{q}_{i-1}\|^2 \tag{3}$$

b) Cartesian Smoothness Cost: Penalizes changes in the end-effector pose (via forward kinematics) between consecutive waypoints:

$$J_{\text{cart}} = \sum_{i=1}^{N-1} \left\|\text{FK}(\mathbf{q}_i) - \text{FK}(\mathbf{q}_{i-1})\right\|^2 \tag{4}$$

c) Final Goal Pose Cost: Encourages the final waypoint to match the current target grasp pose:

$$J_{\text{goal}} = \|\text{FK}(\mathbf{q}_N) - \mathbf{T}_k\|^2, \quad \mathbf{T}_k \in \mathcal{T} \tag{5}$$

d) Standoff Pose Cost: To ensure a consistent approach to the object, we introduce a standoff pose $\mathbf{T}_{\text{standoff}}$, defined as having the same orientation as the selected grasp pose but translated slightly along the approach direction. The trajectory is encouraged to pass through this pose at a dedicated waypoint $\mathbf{q}_{\text{standoff}}$ (*standoff_wp*), where the gripper begins to open. This avoids unstable behavior where the optimizer may unintentionally open and close the gripper between waypoints to bypass collisions. The corresponding cost is

$$J_{\text{standoff}} = \left\|\text{FK}(\mathbf{q}_{\text{standoff}}) - \mathbf{T}_{\text{standoff}}\right\|^2 \tag{6}$$

Constraints: In addition to the cost terms, the optimization problem is subject to a set of constraints that guarantee feasibility. These constraints enforce consistency with the robot's physical limits, the initial state, and collision safety. TrajOpt handles costs and constraints together through a merit cost, which combines the objective value and the constraint violation into a single scalar used by the sequential quadratic programming (SQP) solver. This allows soft trade-offs when constraints are temporarily violated during intermediate iterations, while ensuring that the final solution satisfies feasibility.

- *Start configuration:* The first waypoint equals the start state.
- *Joint limits:* Each joint satisfies $\mathbf{q}_{\min} \leq \mathbf{q}_i \leq \mathbf{q}_{\max}$ for all i. During approach ($i <$ *standoff_wp*), gripper joints are held near closed; afterward they interpolate toward the grasp configuration.
- *Collision avoidance:* Discrete collision constraints are enforced at each waypoint.

B. SwitchOpt Strategy

In Sec. II-A, we formulated grasping as trajectory optimization toward one of several feasible grasp targets $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_K\}$, and note that, unlike standard formulations, our method may switch between candidates during optimization. However, the challenge lies in deciding when to remain committed to the current target and when to adapt to a new one.

In many grasping problems, the robot has access to multiple feasible grasp poses for the same object. Optimizing toward only one target from the start can be risky: if that target lies in a poor region of the search space (e.g., near a collision, a singularity, or a bad local minimum), the trajectory optimizer may converge slowly or even fail altogether. On the other hand, allowing frequent target changes destabilizes convergence, since the solver is forced to constantly re-adapt the entire trajectory.

SwitchOpt addresses this dilemma by introducing a strategy that (i) predicts the expected improvement of staying with the current target, (ii) evaluates alternative targets through a controlled tail reconstruction, (iii) decides adaptively when a switch is beneficial, and (iv) restores and refines the best trajectory at the end of the iteration budget. The central aim is to balance the exploitation of the current trajectory with the exploration of alternative goals, ensuring stable convergence without missing better opportunities.

As a first step, we quantify the progress of the current optimization through the merit cost and predict its expected decrease if we continue to stay with the same target.

a) Merit cost and stay prediction: Let $\mathbf{x}_t$ denote the current trajectory at iteration t. Its merit cost combines the weighted sum of trajectory costs $J(\mathbf{x}_t)$ and the ℓ_2 norm of constraint violations $\mathbf{g}(\mathbf{x}_t)$:

$$m_t = J(\mathbf{x}_t) + \|\mathbf{g}(\mathbf{x}_t)\|_2 \tag{7}$$

From a sliding window of the past w_{slope} iterations, we estimate a log-relative slope b_{rel} representing the rate of merit decrease. If we were to *stay* with the current target for H for more iterations, we predict:

$$M_{\text{stay}} = m_t \, e^{b_{\text{rel}} H} \tag{8}$$

b) Shrinking tail and editable horizon: To evaluate alternative targets without destabilizing the ongoing optimization, SwitchOpt restricts the portion of the trajectory that can be modified. Early iterations expose a large editable tail that can be rebuilt if a switch occurs, while later iterations progressively shrink the editable region to only the final segment. This mechanism ensures that exploration is possible at the beginning, but convergence is preserved near the end.

Let N be the number of waypoints in the trajectory and N_{it} the maximum iteration budget. At iteration t, the number of tail waypoints allowed to change is:

$$s_t = \max\Big(s_{\min},\ \big\lfloor N - \tfrac{t}{\max(1, N_{\text{it}})}(N - s_{\min})\big\rceil\Big) \tag{9}$$

where $s_{\min}$ is the minimum tail length (equal to the number of waypoints from standoff to grasp). The head is fixed for waypoints $0:(N - s_t - 1)$, and the editable tail is $(N - s_t):(N-1)$.

c) Deterministic tail reconstruction for candidates: Given the current head of the trajectory, candidate targets must be evaluated consistently and fairly. SwitchOpt achieves this by deterministically reconstructing the editable tail toward each candidate grasp, ensuring that all alternatives are compared under the same initialization and constraints. For a candidate $g \in \mathcal{T}$, the trajectory $\mathbf{x}_t$ is rebuilt into $\tilde{\mathbf{x}}^{(g)}$ by (i) preserving the head up to the head–tail boundary, (ii) interpolating from the last head waypoint to the candidate standoff IK pose $\mathbf{q}^{(g)}_{\text{standoff}}$, and (iii) interpolating further from standoff to the candidate grasp IK pose $\mathbf{q}^{(g)}_{\text{grasp}}$.

The immediate merit of the reconstructed trajectory is

$$m^{(g)}_{\text{tail}} = J(\tilde{\mathbf{x}}^{(g)}) + \|\mathbf{g}(\tilde{\mathbf{x}}^{(g)})\|_2, \tag{10}$$

and its predicted horizon merit is estimated using a learned cold-start slope b_{cold}:

$$M^{(g)}_{\text{sw,core}} = m^{(g)}_{\text{tail}} \, e^{b_{\text{cold}} H} + \lambda_{\text{sw}} \frac{s_t}{N}. \tag{11}$$

d) Switching decision and adaptation: Having predicted the merit of staying (M_{stay}) and of candidate reconstructions ($M^{(g)}_{\text{sw,core}}$), SwitchOpt augments the evaluation with two penalties to discourage disruptive changes:

1) a penalty on the ℓ_2 deviation between the rebuilt and current tail, and
2) a penalty on the IK jump between current and candidate targets.

The penalized merit is

$$\begin{aligned} \widehat{M}^{(g)}_{\text{sw}} = M^{(g)}_{\text{sw,core}} &+ \rho_{\text{tail}} \left\|\tilde{\mathbf{x}}^{(g)}_{\text{tail}} - \mathbf{x}_{t,\text{tail}}\right\|_2 \\ &+ \rho_{\text{ik}} \left\|\mathbf{q}^{(g)}_{\text{ik}} - \mathbf{q}^{(\text{cur})}_{\text{ik}}\right\|_2 \end{aligned} \tag{12}$$

To ensure stability, we define a hysteresis threshold $M_{\text{thresh}} = \max\big(\tau_{\text{abs}},\ \tau_{\text{rel}} M_{\text{stay}}\big)$, and select the best alternative $g^\star = \arg\min_{g \neq \text{cur}} \widehat{M}^{(g)}_{\text{sw}}$. A switch is triggered if $\widehat{M}^{(g^\star)}_{\text{sw}} + M_{\text{thresh}} < M_{\text{stay}}$, or with a small probability $\varepsilon_{\text{explore}}$, we relax this condition and still accept $g^\star$ as the new target. Figure 2 illustrates the evolution of the merit cost and the resulting switching decisions during optimization.

When a switch occurs, SwitchOpt updates the goal and standoff constraints to $g^\star$, reinitializes the SQP solver at $\tilde{\mathbf{x}}^{(g^\star)}$, and enforces a cooldown period of $\max(H, w_{\text{slope}})$ iterations before re-evaluating switches. The cold-start slope b_{cold} is then re-estimated from post-switch progress to maintain consistent prediction accuracy.

e) Final refinement: When the iteration budget N_{it} is reached, SwitchOpt restores the trajectory snapshot associated with the best target encountered during the optimization. A short refinement phase is then executed on this trajectory, with early stopping triggered by stagnation of the merit decrease. This guarantees that the final solution is locally optimized around the most promising candidate observed within the budget.

SwitchOpt integrates adaptive target switching with a shrinking tail mechanism to balance exploitation and exploration during trajectory optimization. The overall procedure is summarized in Algorithm 1.

Algorithm 1 SwitchOpt algorithm

Initialize trajectory $\mathbf{x} \leftarrow$ initial seed to first target
Compute initial merit m_0 and cold-start slope b_{cold}
cooldown $\leftarrow 0$
for $t = 1$ **to** N_{it} **do**
    Perform one SQP step on $\mathbf{x}$ toward current target
    Update merit history and slope b_{rel}
    if cooldown > 0 **then**
        cooldown $\leftarrow$ cooldown -1
        continue
    $H \leftarrow$ horizon from s_t
    $s_t \leftarrow$ shrinking tail length (Eq. 9)
    Predict stay merit: $M_{\text{stay}} \leftarrow m_t e^{b_{\text{rel}} H}$ (Eq. 8)
    for all candidates $g \neq$ current **do**
        Build $\tilde{\mathbf{x}}^{(g)}$
        Compute $\widehat{M}_{\text{sw}}^{(g)}$ (Eq. 12)
    $g^\star \leftarrow \arg\min_g \widehat{M}_{\text{sw}}^{(g)}$
    if $\widehat{M}_{\text{sw}}^{(g^\star)} + M_{\text{thresh}} < M_{\text{stay}}$ **or** rand() $< \varepsilon_{\text{explore}}$ **then**
        Switch target $\leftarrow g^\star$
        $\mathbf{x} \leftarrow \tilde{\mathbf{x}}^{(g^\star)}$
        Set cooldown, learn new b_{cold}
Restore best trajectory and run final refinement

III. EXPERIMENTS AND RESULTS

A. Experimental Setup

All experiments were conducted in simulation using the dual-arm TIAGo++ robot[1] equipped with parallel grippers. The experiments were implemented in the Tesseract motion planning framework[2], which integrates TrajOpt as a trajectory optimizer. Unlike the standard black-box usage of TrajOpt, our method leverages a step-by-step optimization interface, allowing goal adaptation and partial trajectory reconstruction during the optimization process. This fine-grained control is essential for the adaptive target switching strategy proposed in this work. All experiments were run on a laptop with a 12th-Gen Intel® Core™ i7-12700H CPU; TrajOpt is CPU-bound and does not rely on GPU acceleration.

We evaluate our method in simulation across five tabletop scenes derived from the SceneReplica benchmark [18]. Each scene contains a subset of YCB objects [19] arranged according to the benchmark layouts. For each object, grasp candidates were obtained from the MultiGripperGrasp dataset [20], using the subset generated for the Fetch parallel gripper. These grasps were adapted to the TIAGo gripper frame and downsampled to 300 grasp candidates per object to control computational cost. These grasp candidates are not guaranteed to be kinematically reachable. IK feasibility is checked on demand: when a target is first considered during a trial, its IK solution is computed and cached for the remainder of that trial. This cache is local to each trial and is not shared across methods or across trials. In each trial, the robot is assigned a single target object from the scene and is tasked with planning a grasp using only the grasps available in this precomputed set.

[1]https://pal-robotics.com/robot/tiago/
[2]https://github.com/tesseract-robotics/tesseract

An open-source implementation of SwitchOpt, including all experiment scripts and configuration files, is available at https://github.com/elisabeth-ms/switchOpt.

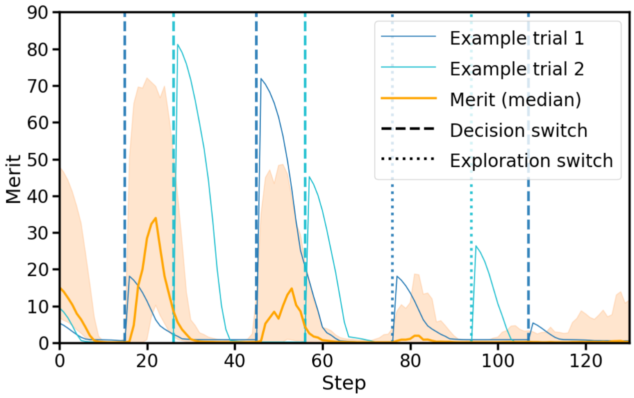

Fig. 2. Merit evolution during SwitchOpt optimization for SceneReplica scene 10 (power drill). The shaded band shows the distribution of instantaneous merit across 50 trials, and two example trials are overlaid. Vertical dashed lines indicate decision-driven switches, while dotted lines correspond to exploration-driven switches.

B. Experimental Protocol

For each scene, each object, and each method, we ran 50 independent trials with different random seeds to account for stochastic effects in initialization and IK solution selection. The robot's starting configuration was fixed across all runs to ensure comparability. To guarantee fair comparison, all methods used the exact same candidate grasp set $\mathcal{T}$, cost terms, constraints, collision settings, and joint limits. Additionally, for every trial, the initial grasp target was identical across all methods.

SwitchOpt experiments used the following parameter values: $\lambda_{\text{sw}}{=}0.05$, $w_{\text{slope}}{=}12$, $\varepsilon_{\text{explore}}{=}0.1$, $N_{\text{it}}{=}200$, $\tau_{\text{abs}}{=}10^{-3}$, $\tau_{\text{rel}}{=}8{\times}10^{-3}$, $H \in [6, 30]$, $\rho_{\text{tail}}{=}10^{-3}$, $\rho_{\text{ik}}{=}10^{-2}$. These parameters control switching detection, exploration probability, and the regularization terms for tail interpolation and IK feasibility.

C. Comparison Methods

We compare SwitchOpt against two standard TrajOpt baselines and two ablations of our method. All methods use the same costs, constraints, and solver settings; differences lie only in how grasp targets are handled.

- **Single-Goal TrajOpt**: A single grasp target is sampled at the start of the trial. A standard TrajOpt optimization is run until stagnation, with no target changes or reinitializations. This corresponds to the classical fixed-goal TrajOpt setting.
- **Multi-Goal TrajOpt (Random Sequential)**: At stagnation, the optimizer discards the current trajectory and reinitializes TrajOpt from scratch toward a new randomly chosen grasp. This continues until all grasp targets have been attempted. No information or partial solutions are reused across targets.

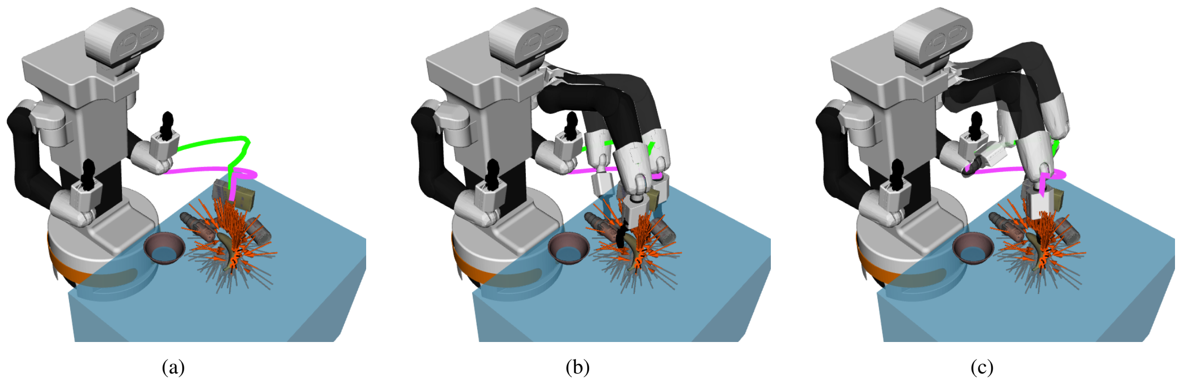

Fig. 3. Illustration of adaptive goal switching in trajectory optimization. (a) Planning stage: the robot switches from an initial (pink) trajectory to an adapted (green) trajectory, with the complete grasp set shown in orange. (b) Executing the initial (pink) plan leads to collision with the environment. (c) Executing the adapted (green) plan successfully reaches a feasible grasp without collision.

- **SwitchOpt without Exploration**: Identical to SwitchOpt (Sec. II-B) but with the exploration probability $\varepsilon_{\text{explore}}$ set to zero. A switch is performed only when the predicted merit strictly outperforms staying. This ablation isolates the contribution of stochastic exploration.
- **SwitchOpt with Full-Seed Reconstruction**: Instead of reusing the optimized trajectory head, each switch rebuilds a full trajectory toward every candidate using only start–to–standoff and standoff–to–grasp interpolation. This removes the head–tail reuse mechanism and tests the importance of trajectory continuity during switching.

D. Evaluation Metrics

We evaluate all comparison methods using the following metrics:

- **Best merit** (↓): The lowest merit value obtained during optimization. The merit combines the weighted trajectory costs and the ℓ_2 norm of constraint violations.
- **Time to best** (↓): Wall-clock time required to reach the best merit value.
- **Time to finish** (↓): Total wall-clock time taken by the method to complete its run under its own stopping rules.
- **Success rate** (↑): Percentage of trials whose final solution achieves a merit below a fixed threshold ($m \leq 0.03$) and is collision-free.

E. Results

Table I summarizes performance across all scenes, objects, and methods. Figure 3 provides a qualitative example in which SwitchOpt switches from a colliding trajectory to an adapted, collision-free grasp plan. The comparison is designed to assess four aspects of SwitchOpt: (i) whether adaptive switching improves over fixed-target optimization, (ii) the importance of the exploration mechanism for escaping stagnation, (iii) whether structured tail reuse is necessary for efficient switching, and (iv) how far the method is from an exhaustive multi-goal TrajOpt.

Single-Goal TrajOpt exhibits extremely high merit variance (std. 1.53) and low success, confirming that committing to an arbitrary grasp frequently leads to poor local minima. Although this baseline is naturally fast because it optimizes only one target, its reliability is too low for practical use.

The first ablation evaluates the role of exploration. Without exploration, switching occurs only when the predicted merit clearly improves, causing the optimizer to remain on suboptimal targets in borderline cases. This leads to reduced robustness (success drops to 65%), whereas full SwitchOpt benefits from occasional exploratory switches to escape flat regions.

The second ablation isolates the effect of structured tail reuse. When the trajectory head is discarded and a full seed is rebuilt at each switch, the method still benefits from adaptive target changes and achieves better reliability than the no-exploration variant (success 72.2% vs. 64.7%), but its best-merit values remain higher and its outcomes more dispersed than full SwitchOpt. The lack of continuity across switches forces the solver to repeatedly recover feasibility from scratch, which makes progress less stable and degrades overall optimization efficiency. In contrast, reusing the refined trajectory head in full SwitchOpt leads to lower costs and more consistent performance for a comparable runtime, highlighting trajectory continuity as a key factor in making target switching effective rather than wasteful.

The multi-goal baseline represents the upper bound of exhaustive switching. It achieves the lowest merits (0.0015 ± 0.0003) and perfect success (100%) by reoptimizing a full trajectory for every grasp target. However, this comes at the cost of extremely high runtimes (time to best 33.9 s, finish 70.4 s), since TrajOpt is restarted after each target change. As a result, the method is computationally impractical for large grasp sets and serves mainly as a best-case reference rather than a realistic planning strategy.

Overall, the comparisons show that neither fixed-target nor exhaustive multi-goal optimization offers a practical solution. The ablations confirm that exploration and tail reuse are both important for stable progress, and full SwitchOpt provides the best balance of cost, reliability, and runtime. These results

TABLE I

COMPARISON OF ALL BASELINE METHODS, ABLATIONS, AND SWITCHOPT EVALUATED ACROSS ALL TESTED SCENES AND OBJECTS. REPORTED VALUES ARE MEAN $\pm$ STD OVER ALL TRIALS.

Method	**Best merit** $\downarrow$	**Time to best [ms]** $\downarrow$	**Time to finish [ms]** $\downarrow$	**Success rate** $\uparrow$
Single-Goal TrajOpt	0.3664 ± 1.5327	982.1 ± 533.6	1198.5 ± 543.9	46.1%
Multi-Goal TrajOpt	0.0015 ± 0.0003	33941.0 ± 21170.3	70373.3 ± 5678.3	100.0%
SwitchOpt without Exploration	0.1230 ± 0.4180	2155.4 ± 1762.6	9599.0 ± 2618.4	64.7%
SwitchOpt with Full-Seed Reconstruction	0.0795 ± 0.2347	3759.5 ± 2871.5	9313.7 ± 2448.0	72.2%
SwitchOpt (ours)	0.0505 ± 0.1581	3365.2 ± 2292.4	9468.6 ± 2525.5	80.0%

indicate that structured, in-place switching is an effective strategy for multi-target grasp optimization.

IV. CONCLUSIONS

We presented SwitchOpt, an adaptive target switching strategy for grasp planning that integrates dynamic goal adaptation into a step-by-step TrajOpt optimization process. By monitoring optimization costs in real time, SwitchOpt predicts when alternative grasp targets are likely to yield better results and switches accordingly. This targeted switching allows partial reuse of previously optimized trajectory segments, preserving optimization progress while avoiding the full trajectory restarts required by naive multi-goal strategies.

Experimental results on the SceneReplica benchmark with the TIAGo++ robot in simulation show that SwitchOpt achieves lower costs, faster convergence, and higher-quality solutions overall. The method effectively balances exploration and exploitation by selectively switching to promising targets while refining already feasible solutions.

Several extensions could further enhance SwitchOpt. Learned motion primitives (e.g., DMPs) could replace direct trajectory interpolation to generate smoother motion during switches. The fixed grasp set could be adapted during the optimization process, adding or refining candidates as the optimization evolves. Incorporating grasp quality metrics into the switching criterion could improve overall performance, complementing the current feasibility-based filtering. SwitchOpt could be extended to dual-arm systems by running optimization threads in parallel and retaining the best arm-object solution. Finally, real-robot experiments on structured datasets are needed to validate the method's robustness beyond simulation.

ACKNOWLEDGMENT

This research has been supported by project iRoboCity2030-CM, Robótica inteligente para ciudades sostenibles (TEC-2024/TEC-62), funded by Programas de Actividades I+D en tecnologías de la Comunidad de Madrid.

REFERENCES

[1] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.

[2] L. Janson, E. Schmerling, A. Clark *et al.*, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International journal of robotics research*, vol. 34, no. 7, pp. 883–921, 2015.

[3] N. Ratliff, M. Zucker, J. A. Bagnell *et al.*, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 489–494.

[4] J. Schulman, Y. Duan, J. Ho *et al.*, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[5] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[6] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2901–2910.

[7] J. Urain, N. Funk, J. Peters *et al.*, "Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," *arXiv preprint arXiv:2209.03855*, 2022.

[8] A. D. Dragan, N. D. Ratliff, and S. S. Srinivasa, "Manipulation planning with goal sets using constrained trajectory optimization," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 4582–4588.

[9] T. Murooka, A. I. Karoly, F. von Drigalski *et al.*, "Simultaneous planning of grasp and motion using sample regions and gradient-based optimization," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 1102–1109.

[10] M. B. Horowitz and J. W. Burdick, "Combined grasp and manipulation planning as a trajectory optimization problem," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 584–591.

[11] J. Mahler, M. Matl, V. Satish *et al.*, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.

[12] J. Ichnowski, M. Danielczuk, J. Xu *et al.*, "Gomp: Grasp-optimized motion planning for bin picking," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 5270–5277.

[13] N. Vahrenkamp, M. Do, T. Asfour *et al.*, "Integrated grasp and motion planning," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2883–2888.

[14] N. Jetchev and M. Toussaint, "Trajectory prediction in cluttered voxel environments," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2523–2528.

[15] D. Berenson, S. S. Srinivasa, D. Ferguson *et al.*, "Manipulation planning with workspace goal regions," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 618–624.

[16] J. Pan, K. He, J. M. Ong *et al.*, "Variable grasp pose and commitment for trajectory optimization," in *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 2023, pp. 1–6.

[17] L. Wang, Y. Xiang, and D. Fox, "Manipulation trajectory optimization with online grasp synthesis and selection," *arXiv preprint arXiv:1911.10280*, 2019.

[18] N. Khargonkar, S. H. Allu, Y. Lu *et al.*, "Scenereplica: Benchmarking real-world robot manipulation by creating reproducible scenes," *CoRR*, 2023.

[19] B. Calli, A. Walsman, A. Singh *et al.*, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143*, 2015.

[20] L. F. Casas, N. Khargonkar, B. Prabhakaran *et al.*, "Multigrippergrasp: A dataset for robotic grasping from parallel jaw grippers to dexterous hands," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 2978–2984.