APPENDIX A
ANALYTICAL GRADIENTS OF THE OPTIMIZATION

In this section, we introduce the key analytical gradients of the trajectory optimization problem.

A. Preliminaries

According to the theory of Lie groups and Lie algebra for robotics [27], we denote the conversion from the axis-angle rotation vector $\boldsymbol{r} \in \mathfrak{so}(3)$ to the rotation matrix $\boldsymbol{R} \in \mathrm{SO}(3)$ as $\boldsymbol{R} = \exp(\boldsymbol{r}^\wedge)$, and the inverse conversion is denoted as $\boldsymbol{r} = \ln(\boldsymbol{R})^\vee$.

According to the linear approximation of the Baker-Campbell-Hausdorff (BCH) formula [29], we have

$$\ln(\exp(\boldsymbol{r}_1^\wedge)\exp(\boldsymbol{r}_2^\wedge))^\vee \approx \begin{cases} \boldsymbol{J}_l(\boldsymbol{r}_2)^{-1}\boldsymbol{r}_1 + \boldsymbol{r}_2, & \text{when } \boldsymbol{r}_1 \to \boldsymbol{0}, \\ \boldsymbol{J}_r(\boldsymbol{r}_1)^{-1}\boldsymbol{r}_2 + \boldsymbol{r}_1, & \text{when } \boldsymbol{r}_2 \to \boldsymbol{0}. \end{cases} \tag{8}$$

Here, $\boldsymbol{J}_r(\boldsymbol{r}) = \boldsymbol{J}_l(-\boldsymbol{r})$, and $\boldsymbol{J}_l(\boldsymbol{r})$ can be calculated as

$$\boldsymbol{J}_l(\boldsymbol{r}) = \frac{\sin\theta}{\theta}\boldsymbol{I} + \left(1 - \frac{\sin\theta}{\theta}\right)\boldsymbol{a}\boldsymbol{a}^\mathsf{T} + \frac{1-\cos\theta}{\theta}\boldsymbol{a}^\wedge, \tag{9}$$

$$\boldsymbol{J}_l(\boldsymbol{r})^{-1} = \frac{\theta}{2}\cot\frac{\theta}{2}\boldsymbol{I} + \left(1 - \frac{\theta}{2}\cot\frac{\theta}{2}\right)\boldsymbol{a}\boldsymbol{a}^\mathsf{T} - \frac{\theta}{2}\boldsymbol{a}^\wedge, \tag{10}$$

where θ and $\boldsymbol{a}$ are the angle and axis of $\boldsymbol{r}$, respectively.

B. Gradients of Orientation Distances

Here we generally introduce the gradient regarding the orientation distance. We define the weighted scalar distance of orientations $\boldsymbol{R}$ and $\boldsymbol{R}_\mathrm{d}$ as

$$d_\mathrm{r}(\boldsymbol{R}, \boldsymbol{R}_\mathrm{d}, \boldsymbol{W}_\mathrm{r}) = \frac{1}{2}\boldsymbol{r}_\mathrm{e}^\mathsf{T}\boldsymbol{W}_\mathrm{r}\boldsymbol{r}_\mathrm{e}, \tag{11}$$

where $\boldsymbol{r}_\mathrm{e}$ is defined as $\ln\left(\boldsymbol{R}\boldsymbol{R}_\mathrm{d}^{-1}\right)^\vee$. Note that $\boldsymbol{r}_\mathrm{e}$ is defined in the world frame $\mathcal{W}$. In addition, $\boldsymbol{W}_\mathrm{r}$ is a semi-positive definite matrix for weighting. Here, $\boldsymbol{R}$ is determined by a variable $\boldsymbol{x}$, and $\boldsymbol{R}_\mathrm{d}$ is a constant desired orientation. Note that $\boldsymbol{r}_\mathrm{e}$ is defined in the same frame as $\boldsymbol{R}$ and $\boldsymbol{R}_\mathrm{d}$.

In the following derivation, for convenience, we use both $\boldsymbol{R} \in \mathrm{SO}(3)$ and $\boldsymbol{r} \in \mathfrak{so}(3)$ to represent the same rotation; e.g., $\boldsymbol{R} = \exp(\boldsymbol{r}^\wedge)$ and $\boldsymbol{R}_\mathrm{d} = \exp(\boldsymbol{r}_\mathrm{d}^\wedge)$.

The gradient of d_r w.r.t. the variable $\boldsymbol{x}$ is derived as

$$\frac{\partial d_\mathrm{r}}{\partial \boldsymbol{x}} = \frac{\partial d_\mathrm{r}}{\partial \boldsymbol{r}_\mathrm{e}}\frac{\partial \boldsymbol{r}_\mathrm{e}}{\partial \boldsymbol{x}} = \frac{\partial d_\mathrm{r}}{\partial \boldsymbol{r}_\mathrm{e}}\frac{\partial \boldsymbol{r}_\mathrm{e}}{\partial \boldsymbol{\varphi}}\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{x}} \tag{12}$$

For convenience of the following calculation, here we introduce a perturbation variable $\boldsymbol{\varphi} \in \mathfrak{so}(3)$, which means that we left perturb $\boldsymbol{R}$ by $\Delta\boldsymbol{R}$ (i.e., $(\Delta\boldsymbol{R})\boldsymbol{R}$), where $\Delta\boldsymbol{R} = \exp(\boldsymbol{\varphi}^\wedge)$.

First, it is easy to obtain

$$\frac{\partial d_\mathrm{r}}{\partial \boldsymbol{r}_\mathrm{e}} = \boldsymbol{r}_\mathrm{e}^\mathsf{T}\boldsymbol{W}_\mathrm{r} \tag{13}$$

Second, regarding $\frac{\partial \boldsymbol{r}_\mathrm{e}}{\partial \boldsymbol{\varphi}}$, we have

$$\begin{aligned}\frac{\partial \boldsymbol{r}_\mathrm{e}}{\partial \boldsymbol{\varphi}} &= \lim_{\boldsymbol{\varphi}\to\boldsymbol{0}} \frac{\ln\left(\exp(\boldsymbol{\varphi}^\wedge)\exp(\boldsymbol{r}^\wedge)\left(\exp(\boldsymbol{r}_\mathrm{d}^\wedge)\right)^{-1}\right)^\vee}{\boldsymbol{\varphi}} \\ &\quad - \frac{\ln\left(\exp(\boldsymbol{r}^\wedge)\left(\exp(\boldsymbol{r}_\mathrm{d}^\wedge)\right)^{-1}\right)^\vee}{\boldsymbol{\varphi}} \\ &= \lim_{\boldsymbol{\varphi}\to\boldsymbol{0}} \frac{\ln\left(\exp(\boldsymbol{\varphi}^\wedge)\exp(\boldsymbol{r}_\mathrm{e}^\wedge)\right)^\vee - \ln\left(\exp(\boldsymbol{r}_\mathrm{e}^\wedge)\right)^\vee}{\boldsymbol{\varphi}}\end{aligned} \tag{14}$$

It follows from the BCH formula (8) that

$$\frac{\partial \boldsymbol{r}_\mathrm{e}}{\partial \boldsymbol{\varphi}} = \lim_{\boldsymbol{\varphi}\to\boldsymbol{0}} \frac{\boldsymbol{J}_l(\boldsymbol{r}_\mathrm{e})^{-1}\boldsymbol{\varphi} + \boldsymbol{r}_\mathrm{e} - \boldsymbol{r}_\mathrm{e}}{\boldsymbol{\varphi}} = \boldsymbol{J}_l(\boldsymbol{r}_\mathrm{e})^{-1} \tag{15}$$

We thus obtain $\frac{\partial d_\mathrm{r}}{\partial \boldsymbol{\varphi}} = \boldsymbol{r}_\mathrm{e}^\mathsf{T}\boldsymbol{W}_\mathrm{r}\boldsymbol{J}_l(\boldsymbol{r}_\mathrm{e})^{-1}$. Moreover, using (10), we can obtain $\boldsymbol{r}_\mathrm{e}^\mathsf{T}\boldsymbol{J}_l(\boldsymbol{r}_\mathrm{e})^{-1} = \boldsymbol{r}_\mathrm{e}^\mathsf{T}$. Thus, in special cases where the weights for each orientation dimension are the same (i.e., $\boldsymbol{W}_\mathrm{r} = w\boldsymbol{I}$), we further have $\frac{\partial d_\mathrm{r}}{\partial \boldsymbol{\varphi}} = w\boldsymbol{r}_\mathrm{e}^\mathsf{T}\boldsymbol{J}_l(\boldsymbol{r}_\mathrm{e})^{-1} = w\boldsymbol{r}_\mathrm{e}^\mathsf{T}$.

Third, as the perturbation variable $\boldsymbol{\varphi} \to \boldsymbol{0}$, we have $\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{x}} = \boldsymbol{J}_\mathrm{a}(\boldsymbol{x})$, where $\boldsymbol{J}_\mathrm{a}(\boldsymbol{x})$ is the space Jacobian that relates the spatial angular velocity to $\dot{\boldsymbol{x}}$.

C. Gradients of Pose Distances

Similar to the derivation of the gradients of orientation distances, the general formula of the gradients of pose distances is derived as follows.

The weighted scalar distance between poses $\boldsymbol{T}$ and $\boldsymbol{T}_\mathrm{d}$ is defined as

$$d(\boldsymbol{T}, \boldsymbol{T}_\mathrm{d}, \boldsymbol{W}) = \frac{1}{2}\boldsymbol{e}^\mathsf{T}\boldsymbol{W}\boldsymbol{e} \tag{16}$$

where $\boldsymbol{e} = [\boldsymbol{p}_\mathrm{e}; \boldsymbol{r}_\mathrm{e}]$, in which $\boldsymbol{p}_\mathrm{e} = \boldsymbol{p} - \boldsymbol{p}_\mathrm{d}$ and $\boldsymbol{r}_\mathrm{e} = \ln\left(\exp(\boldsymbol{r}^\wedge)\left(\exp(\boldsymbol{r}_\mathrm{d}^\wedge)\right)^{-1}\right)^\vee$. Here, $\boldsymbol{T}$ is determined by a variable $\boldsymbol{x}$, and $\boldsymbol{T}_\mathrm{d}$ is a constant desired pose. Note that $\boldsymbol{e}$ is defined in the same frame as $\boldsymbol{T}$ and $\boldsymbol{T}_\mathrm{d}$.

Similar to (12), we introduce a perturbation variable $\boldsymbol{\phi} \in \mathfrak{se}(3)$. Then, the gradient of d w.r.t. $\boldsymbol{x}$ is derived as

$$\frac{\partial d}{\partial \boldsymbol{x}} = \frac{\partial d}{\partial \boldsymbol{e}}\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{\phi}}\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} \tag{17}$$

where $\frac{\partial d}{\partial \boldsymbol{e}} = \boldsymbol{e}^\mathsf{T}\boldsymbol{W}$ and

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_l(\boldsymbol{r}_\mathrm{e})^{-1} \end{bmatrix} \tag{18}$$

Additionally, we have $\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{x})$, where $\boldsymbol{J}(\boldsymbol{x})$ is the space Jacobian that relates the spatial twist to $\dot{\boldsymbol{x}}$.

D. Gradients of $\mathcal{J}_\mathrm{object}$

It is easy to see that $\mathcal{J}_\mathrm{object}$ is only relevant to the object pose at time T. The gradient between the position distance cost and the object position variable is easy to derive. Here, we introduce the gradient regarding the orientation distance (i.e., $\frac{\partial d_\mathrm{r}}{\partial \boldsymbol{r}_{\mathrm{o},T}}$). For brevity, we omit the subscripts o and T.

Similar to (14) and (15), we derive that

$$\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{r}} = \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\varphi}}\right)^{-1} = \boldsymbol{J}_l(\boldsymbol{r}) \tag{19}$$

We then have

$$\frac{\partial d_\mathrm{r}}{\partial \boldsymbol{r}} = \boldsymbol{r}_\mathrm{e}^\mathsf{T}\boldsymbol{W}_\mathrm{r}\boldsymbol{J}_l(\boldsymbol{r}_\mathrm{e})^{-1}\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{r}} = \boldsymbol{r}_\mathrm{e}^\mathsf{T}\boldsymbol{W}_\mathrm{r}\boldsymbol{J}_l(\boldsymbol{r}_\mathrm{e})^{-1}\boldsymbol{J}_l(\boldsymbol{r}) \tag{20}$$

E. Gradients of $\mathcal{J}_{\text{finger}}$

We denote the Lie algebra corresponding to the object pose $\boldsymbol{T}_{\text{o},t} \in SE(3)$ as $\boldsymbol{\xi}_{\text{o},t} = [\boldsymbol{p}_{\text{o},t}; \boldsymbol{r}_{\text{o},t}] \in \mathfrak{se}(3)$, which is defined in $\mathcal{W}$. The optimization variable related to $d({}^{\mathcal{O}}\boldsymbol{T}_{i,t}, {}^{\mathcal{O}}\boldsymbol{T}_{i,0}, \boldsymbol{W}_{\text{f}})$ contains the object pose $\boldsymbol{\xi}_{\text{o},t}$ and finger joint angle $\boldsymbol{q}_{i,t}$. For brevity, we omit the subscripts o, i, and t. We further denote $\boldsymbol{\vartheta} = [\boldsymbol{\xi}; \boldsymbol{q}]$.

We can use (17) to calculate the gradient, but we still need to know the space Jacobian that relates the fingertip twist in $\mathcal{O}$ to $\dot{\boldsymbol{\vartheta}}$. As the object frame $\mathcal{O}$ is moving, this Jacobian is a relative Jacobian between the finger and object. We can calculate this relative Jacobian using individual manipulator Jacobians defined in $\mathcal{W}$ [30], in which we regard the object as a virtual manipulator. According to (2) in [30], the relative Jacobian between the fingertip twist in $\mathcal{O}$ and $\dot{\boldsymbol{\vartheta}}$ can be expressed as

$$ {}^{\mathcal{O}}\boldsymbol{J}_{\text{f}}(\boldsymbol{\vartheta}) = \left[\begin{array}{cc} -{}^{\mathcal{O}}\boldsymbol{\Psi}_{\text{f}} {}^{\mathcal{O}}\boldsymbol{\Omega}_{\text{w}} \boldsymbol{J}_{\text{o}}(\boldsymbol{\xi}) & {}^{\mathcal{O}}\boldsymbol{\Omega}_{\text{w}} \boldsymbol{J}_{\text{f}}(\boldsymbol{q}) \end{array} \right], \tag{21} $$

where $\boldsymbol{J}_{\text{o}}(\boldsymbol{\xi})$ is the space Jacobian that relates the object's twist in $\mathcal{W}$ to $\dot{\boldsymbol{\xi}}$, and $\boldsymbol{J}_{\text{f}}(\boldsymbol{q})$ is the space Jacobian that relates the fingertip's twist in $\mathcal{W}$ to $\dot{\boldsymbol{q}}$. Similar to (19), it can be obtained that

$$ \boldsymbol{J}_{\text{o}}(\boldsymbol{\xi}) = \left[\begin{array}{cc} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_l(\boldsymbol{r}) \end{array} \right], \tag{22} $$

where $\boldsymbol{r}$ refers to the object orientation in $\mathcal{W}$. The finger Jacobian $\boldsymbol{J}_{\text{f}}(\cdot)$ can be obtained from the finger's kinematics. The transformation matrices $\boldsymbol{\Psi}$ and $\boldsymbol{\Omega}$ are defined as

$$ {}^{a}\boldsymbol{\Psi}_b = \left[\begin{array}{cc} \boldsymbol{I} & -\boldsymbol{S}({}^{a}\boldsymbol{p}_b) \\ \boldsymbol{0} & \boldsymbol{I} \end{array} \right], \quad {}^{a}\boldsymbol{\Omega}_b = \left[\begin{array}{cc} {}^{a}\boldsymbol{R}_b & \boldsymbol{0} \\ \boldsymbol{0} & {}^{a}\boldsymbol{R}_b \end{array} \right], \tag{23} $$

where $\boldsymbol{S}(\boldsymbol{p})$ refers to the skew-symmetric matrix of vector $\boldsymbol{p}$.

F. Other Gradients

Other gradients, including the gradients of $\mathcal{J}_{\text{joint}}$ and those of the constraints, can be easily derived. The details are omitted here for brevity.

Appendix B
Additional Details of the Competition

A. Hyper-Parameters

The hyper-parameters we used in the competition were set as $\boldsymbol{W}_{\text{o}} = \text{diag}(10, 10, 10, 0.01, 0.01, 0.0)$, and $\boldsymbol{W}_{\text{f}} = \text{diag}(10, 10, 10, 0.001, 0.001, 0.001)$; for the first trajectory optimization for each waypoint, we set $T = 3$ and $\lambda = 4e-4$; for the re-planning, we set $T = 1$ and $\lambda = 5e-3$. These parameters are also used for the experimental evaluation in Section V.

As analyzed in Section V-B, for the competition, we set $N_{\text{replan}} = 4$ in the first run to ensure more conservative results and $N_{\text{replan}} = 8$ in the second run to aim for higher precision.

B. Performance Results

The performance details of our approach in the RGMC are provided by the competition organizers, which are summarized in the following tables. Specifically, the positions of the ten goal waypoints in the competition are listed in Table I, ranging from (-2.5, -2.5, -2.5) to (2.5, 2.5, 2.5) (cm). The average task error of the ten waypoints in each run is shown in Table II. Note that the task error of each individual waypoint is not provided separately, as the organizer did not record them in detail. From the results, it can be seen that the precision of the second run is higher than that of the first run. This improvement is attributed to the different choices of N_{replan}.

TABLE I
TEN GOAL WAYPOINTS IN THE COMPETITION.

Waypoint index	Position (x, y, z) (cm)
1	(2.5, 2.5, 0)
2	(2.5, 2.5, 2.5)
3	(-2.5, -2.5, -2.5)
4	(-1.3, -2.0, 0.6)
5	(-1.2, 0.7, 0.6)
6	(0.6, 0.4, 0.2)
7	(0.9, -1.2, -1.3)
8	(-2.0, 2.0, 2.0)
9	(0.0, 0.0, 2.0)
10	(0.0, 0.0, 0.0)

TABLE II
FINAL RESULTS IN THE COMPETITION.

		Average task error (cm)
Cylinder Object	Run 1	0.080
	Run 2	0.054
Novel Object[a]	Run 1	0.125
	Run 2	0.063

[a] A mustard bottle from the YCB Dataset, as shown in Fig. 2)

Appendix C
Additional Results and Analysis

A. Analysis of Task Error Variance

The diamond points within Figs. 6, 7, 8 and 10 represent the task error of each waypoint along with the summary statistic. Overall, the variance in task errors can be attributed to factors such as different cubic corners, different iterations, and slight differences between initial grasps. We further investigate the variance through extensive experiments. Specifically, we apply the proposed approach to manipulate the object to the corners of the $5 \times 5 \times 5$ (cm) cubic space over five iterations without human intervention. This process was repeated five times, with the initial grasp reset by a human before each trial. We provide a video of four initial grasps at url[1].

Fig. 11(a) and (b) summarize the average task errors for different runs and different cubic corners. The results indicate that 1) the average task errors of different runs are relatively consistent, with the gap between the largest and smallest closed-loop error being less than 0.08 cm, demonstrating that the slight differences in initial grasps have little impact on the overall statistic results; and 2) certain corners (e.g., a, c, h) exhibit averagely larger errors, as discussed in Section V-D. Then, we further investigate the variance among different runs for each corner, as shown in Fig. 11(c). It can be seen that the planned errors for the same corner remains relatively

[1] https://rgmc-xl-team.github.io/ingrasp_manipulation/initial_grasp.mp4

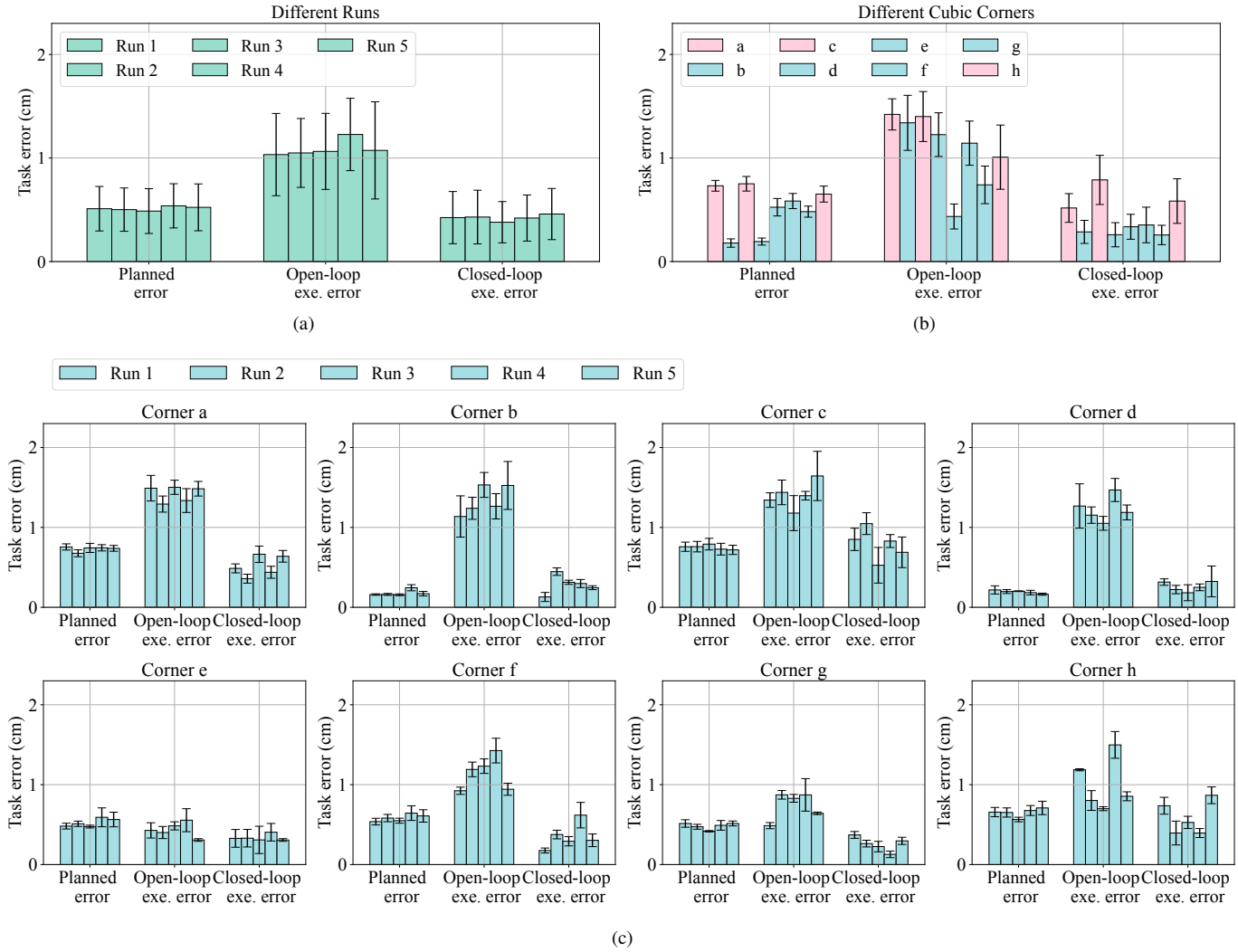

Fig. 11. Variance of task errors. (a) Variance w.r.t. different runs, where each bar represents the average error over 40 waypoints (five iterations of eight corners) in each run. (b) Variance w.r.t. different cubic corners, where each bar represents the average error over 25 attempts (five iterations in five runs) at each corner. (c) Variance w.r.t. different runs for each corner, where each bar represents the average error over five iterations for each corner in each run. The error bars represent the standard deviation.

consistent; in contrast, the execution errors for the same corner vary across different runs and iterations. This variability arises from minor differences in the manually established initial grasps and slight changes in the grasp (e.g., slippage) during continuous manipulation.

B. Effect of Moving Back to Initial State

As described in Section IV-D, we employ a strategy where the fingers return to the initial state (following the forward trajectory) after reaching each waypoint. This approach is adopted because the initial state typically provides a more favorable starting point for trajectory optimization toward the next goal. To experimentally evaluate the effectiveness of this strategy, we task the hand with manipulating the object to the corners of the $5 \times 5 \times 5$ (cm) cubic space over five iterations. We compare the performance of the approach with and without this strategy, conducting five trials for each condition. An example of the manipulation process without returning to the initial state is shown in a video at url[2].

When using this strategy, the average task error of the first iteration (8 corners) is 0.40 cm, and the average task error of all five iteration (40 waypoints) is 0.42 cm. When not using this strategy, the average task error of the first iteration is 0.47 cm; however, the object falls in the second iteration in all five tests due to low contact quality. These results indicate that employing this strategy can enhance the robustness in continuous waypoint reaching.

C. Impact of Excessive Re-planning

In Section V-B, we point out that excessive re-planning may degrade contact quality and lead to larger task errors. This problem is primarily due to the "initial state" of the trajectory optimization. We assume the initial grasp is a stable and manipulable grasp, which holds true for human-designed

[2]https://rgmc-xl-team.github.io/ingrasp_manipulation/not_move_back.mp4

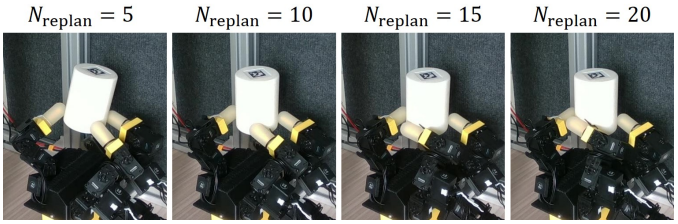

Fig. 12. Example of applying excessive re-planning.

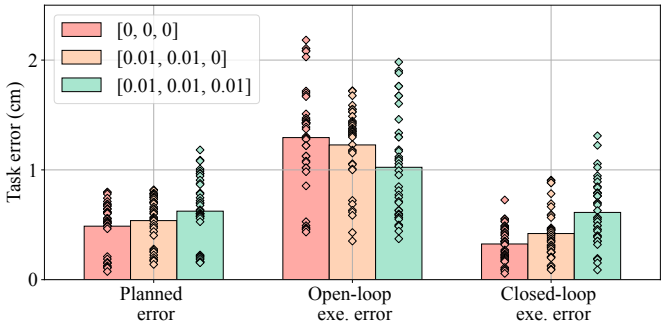

Fig. 13. Effect of different weights for the object orientation cost. Each bar shows the average error over 40 waypoints on the corners of the $5 \times 5 \times 5$ (cm) space, and the values of each waypoint are also plotted by the scattered diamond-shaped points.

TABLE III
EVALUATION OF OUR APPROACH WITH GOALS OF OBJECT POSES.

Index	Goal translation (cm)	Goal rotation (degree)	Position error (cm)	Orientation error (degree)
1	(0, -1, -1)	(0, 20, 0)	0.22	1.7
2	(-2, 0, 0)	(30, 0, 0)	0.74	4.5
3	(0, 0, 2)	(0, 0, 40)	0.16	1.2

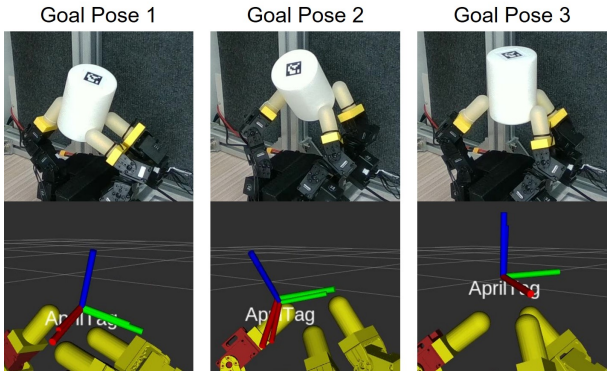

Fig. 14. Snapshots of reaching goal object poses. The figures in the second row visualize the poses, where the larger axes represent the goal poses and the smaller axes represent the poses of the AprilTag. The manipulation process is also shown in the supplementary video.

initial grasps. However, during replanning, the "initial state" of the new optimization problem is set as the terminal state from the previous execution. Note that our simple trajectory optimization problem formulation does not explicitly constrain the quality of the finger-object contacts. Instead, it tries to implicitly maintain the contact quality by softly penalizing deviations between the terminal and initial configuration (through the cost term $\mathcal{J}_{\text{finger}}$ and $\mathcal{J}_{\text{joint}}$). Consequently, the terminal state from the previous execution may not be as good a grasp as the initial state, leading to a gradual decline in contact quality as the number of replanning iterations increases.

We provide an example to illustrate this problem in Fig. 12. It is shown that when applying too many replanning times to reach the goal waypoint, unmodeled contact occurs between the object and non-spherical parts of the fingers, which leads to significant slippage.

D. Effect of Weights for Object Pose Cost

In the competition, we empirically used $\boldsymbol{W}_{\text{o}} = \text{diag}(10, 10, 10, 0.01, 0.01, 0.0)$ for the object pose cost $\mathcal{J}_{\text{object}}$. We set the goal object orientation as the current (initial) orientation, and applied the non-zero weights to slightly regulate the rotation along the X and Y axes, which improved the manipulation robustness in our experience.

We further quantitatively evaluate the effect of different choices of the weights for the orientation, including $\text{diag}(0.0, 0.0, 0.0)$, $\text{diag}(0.01, 0.01, 0.0)$, and $\text{diag}(0.01, 0.01, 0.01)$. The results of one run (40 waypoints) are shown in Fig. 13. It can be seen that 1) increasing the regulation on object orientation leads to larger planned and closed-loop errors, as it restricts the object's reachable space; however, it results in smaller open-loop errors, as smaller rotation generally reduces the risk of unexpected slippage during manipulation; and 2) regulating the Z-axis rotation further increases the closed-loop task errors, compared with regulating only the X and Y axes. Although the results of this run suggest that no regulation might yield higher precision, we find that significant slippage sometimes occur during continuous manipulation without any regulation. Consequently, we chose $\text{diag}(0.01, 0.01, 0.0)$ for the competition as a trade-off between accuracy and robustness.

E. Goals of Object Poses

We further demonstrate our approach's capability to handle goals of object poses, which include both positions and orientations. The task involves continuous reaching three goal poses, for which we set $\boldsymbol{W}_{\text{o}} = \text{diag}(10, 10, 10, 1, 1, 1)$. The details of the goals and closed-loop manipulation results are listed in Table III. Snapshots of the manipulation process are shown in Fig. 14. The results demonstrate that our approach can be easily adapted to address goal orientations.

F. Implementation of Baseline

In Section V-C, we implement a baseline similar to [5] for comparison. Here we provide the details regarding the re-implementation. We adopt the same framework as our approach, whereas the optimization variables include only joint angles, and the object pose is derived from the thumb-tip pose under the rigid contact assumption. Our implementation closely follows that in [5], with the following differences: 1) we do not include the cost of in-trajectory object poses,

whose references are obtained by linear interpolation between the start and goal object poses in theirs; 2) we treat the joint velocity/movement limits as a soft constraint (penalty) instead of a hard constraint in theirs; and 3) we use the same hyper-parameters as those in our approach.