CoorGrasp: Coordinated Contact Control for Adaptive Dexterous Grasping Under Uncertainty

Anonymous Author(s)

***Abstract*— While recent research has focused heavily on dexterous grasp pose generation, less attention has been devoted to the execution of planned grasps. Under shape and position uncertainty, open-loop execution often yields uncoordinated contacts, causing undesired in-hand object motion and even grasp failures. To address this, this paper proposes a tactile-driven model predictive controller for adaptive and delicate execution of diverse dexterous grasps. Our approach emphasizes multi-contact coordination across both approaching and grasping phases, with three key novelties: (i) coordination-aware phase separation, (ii) arm–hand coordination to compensate for position errors, and (iii) adaptive force coordination to increase contact forces in a balanced manner. An analytical model is employed to relate contact forces to robot joint motions for predictive control. Our formulation imposes no restrictions on grasp types or contact configurations and integrates seamlessly with state-of-the-art grasp pose generation methods. Supplementary materials are available at `https://ada-grasp-ctrl.github.io/`.**

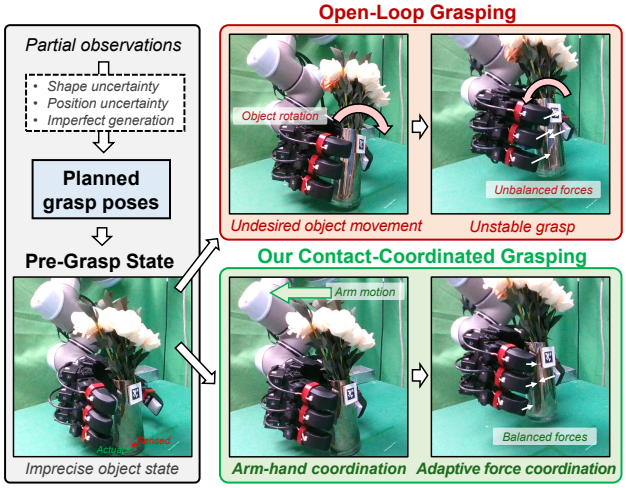


Fig. 1: **Execution of planned dexterous grasps.** Due to uncertainty, open-loop execution of imperfect planned grasp poses can cause unintended in-hand object movements or grasp failures.

I. Introduction

While substantial progress has been made in dexterous grasp pose synthesis and generation, comparatively less attention has been devoted to the actual execution of planned grasps, which demands delicate control of the physical hand-object interactions. We define a preferable *delicate grasping process* as one where the robotic hand approaches an object on a horizontal tabletop and grasps the object without altering its original pose. Given a grasp pose planned by grasp pose generation approaches, the most straightforward and commonly used execution strategy is to move the robot to the planned joint configuration in an open-loop manner. Owing to perception and planning errors, we usually observe that objects are unintentionally moved in-hand during open-loop grasping, as multiple fingers fail to contact the object simultaneously or apply forces in a coordinated manner, as illustrated in Fig. 1. These *undesired object movements* may lead to incorrect grasp configurations or even grasp failures.

This work aims to enable adaptive execution of planned tabletop grasps under uncertainty while minimizing undesired object movements through tactile-feedback control. The key for achieving this objective is maintaining **coordinated contacts** throughout both the approaching and grasping phases, meaning that contact forces from multiple fingers should remain balanced at all times. Existing tactile-feedback approaches typically treat approaching and grasping separately, primarily using position control during approaching to establish contacts and force control during grasping to apply desired forces [1]–[5]. While this strategy can partly mitigate uncertainty, these methods usually control each finger independently and devote limited attention to the coordination among multiple contacts. See App. VI-B for Related Work.

To address the above objective, we propose a model predictive control (MPC) approach for coordinated contact control during grasping. The key contributions and novelties of our approach beyond existing methods include:

1) Coordination-Aware Phase Separation: Unlike conventional methods that treat each finger independently, our approach separates the approaching and grasping phases based on the collective state of all contacts. The transition occurs once sufficient contacts are established to enable non-zero yet balanced forces. **2) Arm-Hand Coordination during Approaching**: The approaching phase is to establish adequate contacts on the object while avoiding large forces. To adapt to the actual object position without excessively deviating from the planned finger configuration, our method enables coordinated arm motions to adjust the global hand pose, contrasting with conventional methods that rely solely on finger motions. **3) Adaptive Force Coordination during Grasping**: The grasping phase is to coordinately increase contact forces to firmly grasp the object. Unlike conventional methods that prescribe fixed desired forces for each fingertip, our approach adaptively allocates forces of all contacts online based on measured contact locations and forces, guided by wrench balance criteria.

To achieve these functions, our approach employs a contact-driven MPC with an analytical motion-contact model. Unlike methods constrained to fixed grasp types or fingertip-only contacts, our approach imposes no restrictions on the grasp

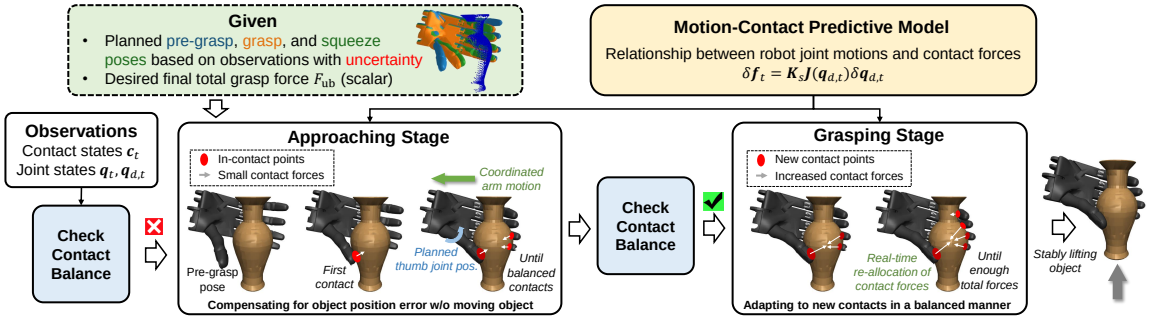


Fig. 2: **Overview of our tactile-driven coordinated contact control method** for adaptive execution of planned grasp poses under uncertainty.

postures, contact numbers, or locations, provided that tactile sensing of these contacts is available. Consequently, it can be seamlessly combined with state-of-the-art (SOTA) grasp pose generation methods and diverse grasp poses.

We perform a large-scale simulation study involving 15k grasps across 478 objects using three robotic hands. In addition, we demonstrate real-world deployment of the proposed method on eight everyday objects. The results show that our method achieves planned grasps with higher success rates and reduced undesired object motion under uncertainty.

II. Method

Planned grasps: The grasp pose generator we use plans three sequential grasp poses based on an point-cloud observation, including a collision-free pre-grasp pose, a grasp pose where the hand is just in contact with the object, and a squeeze pose that can firmly grasp the object [6].

Notations: The joint position of the arm-hand robot at time t is denoted as $\boldsymbol{q}_t \in \mathbb{R}^{N_a+N_h}$, where N_a and N_h are the degrees-of-freedom (DoFs) of the arm and hand, respectively. The contact state is denoted $\boldsymbol{c}_t = \{\boldsymbol{c}_t^1, \cdots, \boldsymbol{c}_t^m\}$, where m is the number of contacts. The state of each contact $\boldsymbol{c}_t^i$ includes a contact force $\boldsymbol{f}_t^i \in \mathbb{R}^3$, a contact position $\boldsymbol{p}_t^i \in \mathbb{R}^3$, and a contact normal $\boldsymbol{n}_t^i \in \mathbb{R}^3$. We denote $\boldsymbol{f}_t = [\boldsymbol{f}_t^1; \cdots; \boldsymbol{f}_t^m] \in \mathbb{R}^{3m}$. Note that we use $[\boldsymbol{a}; \boldsymbol{b}]$ to denote vertical concatenation of column vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

Multi-contact grasp: Defining the contact force $\boldsymbol{f}^i$ in the contact frame, the wrench $\boldsymbol{w}^i \in \mathbb{R}^6$ applied on the object from contact i can be calculated as $\boldsymbol{w}^i = \boldsymbol{G}^i \boldsymbol{f}^i$. The total wrench is obtained as $\boldsymbol{w} = \sum_{i=1}^m \boldsymbol{G}^i \boldsymbol{f}^i = \boldsymbol{G}\boldsymbol{f}$.

A. Overview of Contact-Coordinated Grasping Process

The best open-loop execution strategy leverages the planned three subsequent grasp poses. The path from the pre-grasp pose to grasp pose aims to move from a collision-free pose to a pose just in contact (i.e., approaching phase), and the path from the grasp pose to squeeze pose aims to apply sufficient contact forces (i.e., grasping phase). Our approach leverages the planned grasp poses as a guidance of finger moving directions. We first linearly interpolate between the pre-grasp, grasp, and squeeze pose, obtaining a joint-space *guiding path* denoted as $\bar{\mathcal{P}} = \{\bar{\boldsymbol{q}}_1, \cdots, \bar{\boldsymbol{q}}_T\}$. Due to uncertainty, the separation of the approaching and grasping phase should be adaptively determined based on real-time feedback.

During the first approaching phase, the objective is to move along the guiding path while avoiding applying large forces to

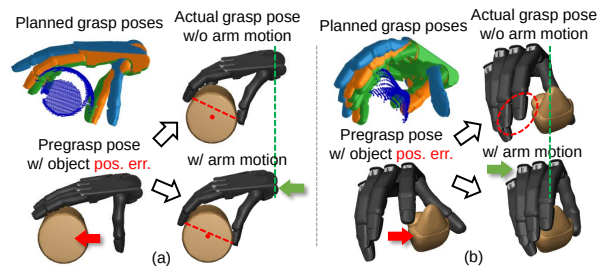


Fig. 3: **Two cases of only using finger motions to compensate for object position errors.** It may lead to issues that (a) the fingers reach an unstable grasp configuration; or (b) the remaining fingers fail to make contact with the object even after reaching the planned final squeeze pose.

the object, until satisfying the phase transition criteria. Owing to position uncertainty, the fingers may not make contact with the object simultaneously (as in Fig. 2). If the first in-contact fingers continue moving forward, they will unintendedly move the object. One strategy is to stop these fingers and actuate the other fingers until they are all in contact [7]. However, since the relative position between the object and hand palm deviates from the planned one, the first-contact fingers will undershoot the desired values while the other fingers will overshoot the desired values, leading to larger deviation from the planned contact configurations and less stable grasps, as illustrated in Fig. 3. Our key insight is that **arm-hand coordination** can be incorporated to simultaneously adjust the palm pose, enabling smaller deviation from the planned finger configurations. We design an MPC with joint optimization of arm and hand motions, which is described in Section II-B.1.

When the phase-transition criteria is satisfied, the robot starts to increase grasp forces. The key question is how to allocate the desired force of each contact. We use the wrench balance criteria to optimize the allocation of the force distribution across all contacts. Note that, due to partial and noisy observations, it is uncertain prior to execution which parts of the hand will make contact with the object. Moreover, new contacts may still emerge after the phase transition (see Fig. 2). Thus, the desired forces are re-allocated in real time to adapt to contact state changes, while also accounting for current contact forces and joint configurations to ensure smoothness and reachability. To achieve this **adaptive force coordination**, we design an MPC that jointly optimizes the desired contact forces and corresponding robot motions. as described in Section II-B.2.

Furthermore, to control contact forces via robot joint movements, we employ a motion-contact model to predict

contact force changes and incorporate it into the MPC for both the approaching and grasping phases, as described in App. I-B. The complete pipeline is illustrated in Fig. 2.

B. Model Predictive Grasping Control

The optimization variables for both phases include control input $\boldsymbol{u}_t = \boldsymbol{q}_{d,t+1} - \boldsymbol{q}_{d,t}$ and desired forces of all contact $\boldsymbol{f}_{t+1}$. Note that this formulation does not take account of the future contact states of current non-contacting points, as they cannot be predicted using tactile feedback alone.

1) Approaching Phase: The MPC is formulated as

$$\min_{\boldsymbol{u}_t, \boldsymbol{f}_{t+1}} \quad \mathcal{J}_{\text{guide}} + \mathcal{J}_{\text{palm}} + \mathcal{J}_{\text{tm}} + \mathcal{J}_{\text{vel}} + \mathcal{J}_{\text{acc}} \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{f}_{i,t+1} = \boldsymbol{f}_{i,t} + \boldsymbol{K}_s \boldsymbol{J}(\boldsymbol{q}_{d,t}) \boldsymbol{u}_t \tag{2}$$

$$\sum_{i=1}^{m} f_{i,1,t+1} \leq F_{\text{appr}} \tag{3}$$

$$\boldsymbol{q}_{\text{lb}} \leq \boldsymbol{q}_{d,t} + \boldsymbol{u}_t \leq \boldsymbol{q}_{\text{ub}} \tag{4}$$

Constraint (3) ensures that the fingers do not apply large forces to the object, where $f_{i,1,t+1}$ represents the normal-axis force and F_{appr} is a small threshold. The cost $\mathcal{J}_{\text{guide}}$ aims to follow the planned guiding path $\bar{\boldsymbol{P}}$ of fingers as $\mathcal{J}_{\text{guide}} = \|(\boldsymbol{q}_{d,t} + \boldsymbol{u}_t) - \bar{\boldsymbol{q}}_{t+1}\|_{\boldsymbol{W}_g}$, where the elements for arm joints in the weight matrix $\boldsymbol{W}_g$ are zeros. The cost $\mathcal{J}_{\text{palm}}$ aims to restrict the vertical-axis movement and rotation of the hand palm, specified as $\mathcal{J}_{\text{palm}} = \|\text{FK}^p(\boldsymbol{q}_{d,t} + \boldsymbol{u}_t) - \text{FK}^p(\bar{\boldsymbol{q}}_{t+1})\|_{\boldsymbol{W}_p}$, where the horizontal axes of the weight matrix $\boldsymbol{W}_p$ are zeros. The cost $\mathcal{J}_{\text{tm}}$ penalizes the tangential movements of the contact points, specified as $\mathcal{J}_{\text{tm}} = \|\boldsymbol{J}(\boldsymbol{q}_{d,t})\boldsymbol{u}_t\|_{\boldsymbol{W}_t}$, where the tangential dimensions of $\boldsymbol{W}_t$ are non-zero. Additionally, the cost $\mathcal{J}_{\text{vel}}$ and $\mathcal{J}_{\text{acc}}$ slightly penalize the velocity and acceleration of $\boldsymbol{q}_d$ to enhance smoothness.

2) Grasping Phase: The MPC is formulated as

$$\min_{\boldsymbol{u}_t, \boldsymbol{f}_{t+1}} \quad \mathcal{J}_{\text{guide}} + \mathcal{J}_{\text{wrench}} + \mathcal{J}_{\text{vel}} + \mathcal{J}_{\text{acc}} \tag{5}$$

$$\text{s.t.} \quad \boldsymbol{f}_{i,t+1} = \boldsymbol{f}_{i,t} + \boldsymbol{K}_s \boldsymbol{J}(\boldsymbol{q}_{d,t}) \boldsymbol{u}_t \tag{6}$$

$$\sum_{i=1}^{m} f_{i,1,t+1} = F_{\text{des},t+1} \tag{7}$$

$$\boldsymbol{f}_{i,t+1} \in \mathcal{F}, \quad i \in [1, m] \tag{8}$$

$$\boldsymbol{q}_{\text{lb}} \leq \boldsymbol{q}_{d,t} + \boldsymbol{u}_t \leq \boldsymbol{q}_{\text{ub}} \tag{9}$$

$$\boldsymbol{W}_a \boldsymbol{u}_t = \boldsymbol{0} \tag{10}$$

Constraint (7) ensures that the sum of normal contact forces equals a scalar desired grasp force $F_{\text{des},t}$. We define it as an increased value over time t:

$$F_{\text{des},t+1} = \min\left(\max\left(\sum_{i=1}^{m} f_{i,1,t}, F_{\text{des},t}\right), F_{\text{ub}}\right) + \delta F \tag{11}$$

where F_{ub} is the pre-defined final grasp force and δF is the step size. Constraint (8) represents the friction cone constraint formulated as $f_{i,1} \geq 0$ and $|f_{i,1}| \geq \hat{\mu}\sqrt{f_{i,2}^2 + f_{i,3}^2}$ for $i \in [1, m]$, where $\hat{\mu}$ is the estimated friction coefficient. The cost $\mathcal{J}_{\text{wrench}}$ is to adaptively coordinate desired forces of multiple contacts by pursuing wrench balance, which is specified as $\mathcal{J}_{\text{wrench}} = \|\boldsymbol{G}\boldsymbol{f}_{t+1}\|_2^2$. The $\mathcal{J}_{\text{guide}}$ is to actuate the non-contact fingers to continue following the guiding path to establish potential additional contacts, and the element of its weight matrix $\boldsymbol{W}_g$ is set to zero if the corresponding finger is already in contact. Additionally, arm motions are not used in this phase and constrained by (10), where the arm-related element of $\boldsymbol{W}_a$ are set to 1.

3) Phase Transition: The criteria of phase transition is defined as whether the multiple in-contact locations can apply a balanced wrench to the object. It is online calculated via the following optimization problem:

$$\min_{\boldsymbol{f}} \quad J = \|\boldsymbol{G}\boldsymbol{f}\|_2^2 \tag{12}$$

$$\text{s.t.} \quad \boldsymbol{f}_i \in \mathcal{F},\ i \in [1, m] \quad \text{and} \quad \sum_{i=1}^{m} f_{i,1} = 1 \tag{13}$$

Considering object scales, we express torques in units of $\text{N}\cdot\text{cm}$ when computing J. The grasping phase is activated if the normalized wrench $\|\bar{\boldsymbol{w}}\|_2$ of the optimal solution is below a threshold ϵ_b.

III. Results

More implementation details, results, and analysis are provided in **Appendix**.

Evaluation metrics: The metrics for evaluation includes 1) **Success rate** (%): a grasp is successful if the object is lifted from the tabletop; 2) **Average object position error** (mm): the distance between the final object position and desired position (i.e., initial object position + lift height), averaged over all successful cases; 3) **Average object rotation error** (°): the angle between the final and initial object orientation, averaged over all successful cases; and 4) **Average normalized wrench**: the normalized wrench during the last 0.5 second before lifting, averaged over all success cases.

Baselines: We choose four baselines for comparison in simulation, including 1) **Open-loop**: The open-loop execution strategy as aforementioned. 2) **Feedback control**: A widely used two-phase tactile-feedback controller concluded from [1]–[5], where each finger are independently controlled. It uses position control for the approaching phase and force control after making contacts. Here the force control law is designed as $\delta \boldsymbol{q}_d^i = \alpha(\boldsymbol{K}_p^i)^{-1}\boldsymbol{J}^i(\boldsymbol{q})^{\mathsf{T}}(\boldsymbol{f}_d^i - \boldsymbol{f}^i)$, where the desired force $\boldsymbol{f}_d^i$ are set to be equal for all contacts. 3) **Without arm motion**: Only adjusting finger motions to adapt to object position errors without leveraging arm motions, like [7]. We implement it by forbidding arm motions in our framework. 4) **Independent forces**: Using independent equal desired forces for each contact in the grasping phase, while other settings are the same as ours.

A. Simulation Studies

We conduct simulation studies to statistically evaluate the methods. We run the grasp generation models on the test set and select top 10 grasps from 100 batched samples for each case based on probability estimation. We then randomly sample 5k grasps across 478 objects for each hand, which are used in the following evaluation. The grasping is simulated in MuJoCo [8]. Some of the grasps are visualized in Fig. 10.

TABLE I: **Large-scale simulation evaluation results under object shape uncertainty.**

Method	Shadow				Allegro				LEAP			
	SR[a]↑	Pos. ↓	Rot. ↓	Wrench ↓	SR ↑	Pos. ↓	Rot. ↓	Wrench ↓	SR ↑	Pos. ↓	Rot. ↓	Wrench ↓
Open-loop	81.6	6.0±5.5	5.2±4.6	0.53±0.24	79.8	5.9±5.5	5.0±4.1	0.44±0.19	79.3	7.3±5.5	5.3±4.3	0.43±0.20
Feedback control	86.7	18±11	13±12	0.43±0.19	86.1	18±11	12±9.8	0.44±0.17	87.8	20±13	11±8.5	0.45±0.20
W/o arm motion	90.8	3.1±5.2	3.4±4.9	0.18±0.19	91.5	2.8±4.7	2.9±4.2	0.19±0.17	93.2	3.2±5.3	3.3±4.5	0.18±0.18
Independ. forces	87.7	12±6.5	7.9±6.4	0.42±0.18	86.2	12±6.6	7.4±5.6	0.43±0.15	89.2	14±9.1	7.0±5.2	0.46±0.19
Ours	**91.6**	**2.4±4.2**	**2.6±4.5**	**0.16±0.17**	**92.4**	**2.2±4.3**	**2.2±3.7**	**0.18±1.15**	**93.6**	**2.5±4.7**	**2.4±3.5**	**0.17±0.17**

[a] SR: success rate (%). Pos. (mm): averaged object position errors. Rot. (°): averaged object rotation error. Wrench: averaged normalized wrench. Results are reported as (mean ±standard deviation).

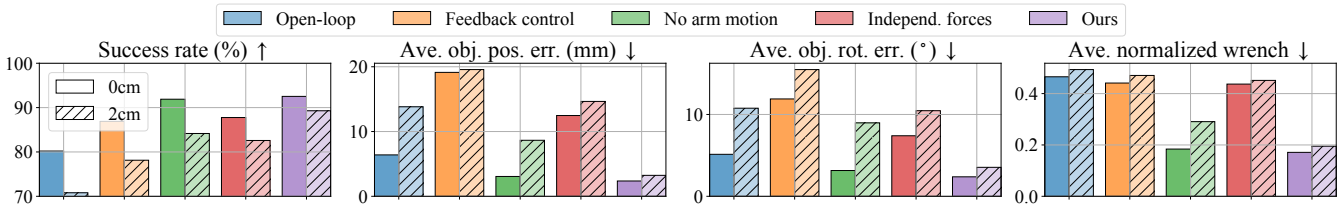


Fig. 4: **Simulation evaluation results under object position uncertainty**, averaged across the three hands. "0cm" refers to no position errors, and "2cm" refers to perturbing the positions by 2 cm along eight uniformly distributed planar directions respectively.

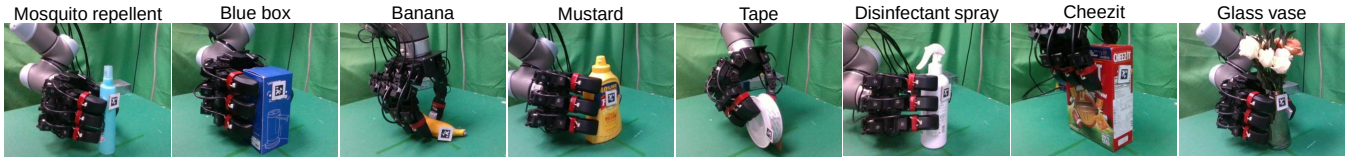


Fig. 5: **Real-world experiments include comparison of our method with baselines on eight different everyday objects and planned grasps.** The AprilTag markers are only used for quantitative evaluation of undesired object movements.

TABLE II: **Quantitative results of the real-world experiments**, averaged across all tests.

Method	Shape uncertainty (8 objects)					Position uncertainty (2 objects, 4 positions)				
	SR[a] ↑	Pos. ↓	Rot. 1 ↓	Rot. 2 ↓	Wrench ↓	SR ↑	Pos. ↓	Rot. 1 ↓	Rot 2 ↓	Wrench ↓
Open-loop	40/40	8.5±4.6	3.5±2.2	6.7±2.5	0.57±0.19	20/20	30±13	12±4.1	12±5.6	0.74±0.09
Feedback control	40/40	11±6.6	3.4±2.1	5.3±3.2	0.54±0.15	20/20	13±7.4	4.9±1.7	8.5±4.9	0.69±0.13
Ours	**40/40**	**3.1±2.4**	**1.7±1.1**	**3.1±1.4**	**0.40±0.12**	**20/20**	**4.1±1.6**	**3.2±1.7**	**5.2±1.9**	**0.42±0.09**

[a] SR: success rate. Pos. (mm): object position error before lifting. Rot. 1 (°): object rotation error before lifting. Rot. 2 (°): object rotation error after lifting. Wrench: normalized wrench before lifting. Results are reported as (mean ±standard deviation).

1) Evaluation with Shape Uncertainty: Since the network-based generation uses partial pointcloud as the observation, the generated grasps reflect the influence of shape uncertainty. The evaluation results on all grasps of the three hands using our method and the baselines are summarized in Table I. The results indicate that 1) our approach performs similarly to the baseline without arm motion, as the visual observations contain no position errors; 2) our approach performs consistently better than other baselines on all three hands, demonstrating the significance of balanced grasping forces; 3) tactile-feedback approaches achieve higher success rates than open-loop execution, as they try to guarantee sufficient forces in every case; and 4) unbalanced forces from multiple contacts lead to large undesired object movements, even worse than open-loop executions that preserves kinematic-level finger coordination.

2) Evaluation with Position Uncertainty: We further evaluate the methods under object position errors. Specifically, the initial object positions are perturbed by 2 cm along eight uniformly distributed planar directions respectively to mimic position uncertainty. Cases with in-collision pre-grasp poses are filtered out. The averaged results across the three hands are shown in Fig. 4. The results indicate that 1) our method achieves the best performance, with the highest success rate as well as the lowest object movements and wrenches; 2) our method exhibits the smallest performance degradation between the settings with and without perturbation, demonstrating its higher adaptability to position errors; and 3) the baseline without arm motions suffers from higher performance drops, highlighting the significance of incorporating coordinated arm motions during approaching.

B. Real-World Experiments

Real-world setup is introduced in App. II-F.

1) Evaluation with Shape Uncertainty: We evaluate our approach against the open-loop strategy and feedback control method on eight planned grasps of different objects, as shown in Fig. 5. Each grasp is executed five times, and the averaged results across all objects are reported in Table II (left). The results show that our approach effectively reduces the undesired object movements.

2) Evaluation with Position Uncertainty: We further conduct experiments under large position errors, using the glass vase and mosquito repellent bottle. The object is displaced by approximately 2 cm from its original position, either towards the thumb or index finger. Each case is executed five times, and the averaged results are reported in Table II (right). Examples are shown in Fig. 1 and 9.

REFERENCES

[1] T. Takahashi, T. Tsuboi, T. Kishida, Y. Kawanami, S. Shimizu, M. Iribe, T. Fukushima, and M. Fujita, "Adaptive grasping by multi-fingered hand with tactile sensor based on robust force and position control," in *IEEE Int. Conf. Robot. Autom.*, 2008, pp. 264–271.

[2] M. Li, K. Hang, D. Kragic, and A. Billard, "Dexterous grasping under shape uncertainty," *Robot. Auton. Syst.*, vol. 75, pp. 352–364, 2016.

[3] Z. Deng, Y. Jonetzko, L. Zhang, and J. Zhang, "Grasping force control of multi-fingered robotic hands through tactile sensing for object stabilization," *Sensors*, vol. 20, no. 4, p. 1050, 2020.

[4] C. J. Ford, H. Li, J. Lloyd, M. G. Catalano, M. Bianchi, E. Psomopoulou, and N. F. Lepora, "Tactile-driven gentle grasping for human-robot collaborative tasks," in *IEEE Int. Conf. Robot. Autom.*, 2023, pp. 10 394–10 400.

[5] C. J. Ford, H. Li, M. G. Catalano, M. Bianchi, E. Psomopoulou, and N. F. Lepora, "Shear-based grasp control for multi-fingered underactuated tactile robotic hands," *IEEE Trans. Robot.*, 2025.

[6] J. Chen, Y. Ke, and H. Wang, "Bodex: Scalable and efficient robotic dexterous grasp synthesis using bilevel optimization," *arXiv preprint arXiv:2412.16490*, 2024.

[7] Z. Chen, T. Wimböck, M. A. Roa, B. Pleintinger, M. Neves, C. Ott, C. Borst, and N. Y. Lii, "An adaptive compliant multi-finger approach-to-grasp strategy for objects with position uncertainties," in *IEEE Int. Conf. Robot. Autom.*, 2015, pp. 4911–4918.

[8] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.

[9] C. Ferrari, J. Canny *et al.*, "Planning optimal grasps," in *IEEE Int. Conf. Robot. Autom.*, vol. 3, 1992, pp. 2290–2295.

[10] T. Gold, A. Völz, and K. Graichen, "Model predictive interaction control for robotic manipulation tasks," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 76–89, 2022.

[11] J. K. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *IEEE Conf. Decis. Control*, 1980, pp. 95–100.

[12] Z. Su, K. Hausman, Y. Chebotar, A. Molchanov, G. E. Loeb, G. S. Sukhatme, and S. Schaal, "Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2015, pp. 297–303.

[13] J. Chen, Y. Ke, L. Peng, and H. Wang, "Dexonomy: Synthesizing all dexterous grasp types in a grasp taxonomy," *arXiv preprint arXiv:2504.18829*, 2025.

[14] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, "Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation," in *IEEE Int. Conf. Robot. Autom.*, 2023, pp. 11 359–11 366.

[15] K. Shaw, A. Agarwal, and D. Pathak, "Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning," *Robotics: Science and Systems (RSS)*, 2023.

[16] L. Zhang, Y. Wang, and Y. Jiang, "Tac3d: A novel vision-based tactile sensor for measuring force distribution and estimating friction coefficient distribution," *arXiv preprint arXiv:2202.06211*, 2022.

[17] H. Liu, B. Huang, Q. Li, Y. Zheng, Y. Ling, W. Lee, Y. Liu, Y.-Y. Tsai, and C. Yang, "Multi-fingered tactile servoing for grasping adjustment under partial observation," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 7781–7788.

[18] H. Liang, L. Cong, N. Hendrich, S. Li, F. Sun, and J. Zhang, "Multifingered grasping based on multimodal reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1174–1181, 2021.

[19] S. Wang, W. Hu, L. Sun, X. Wang, and Z. Li, "Learning adaptive grasping from human demonstrations," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 5, pp. 3865–3873, 2022.

[20] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu, "Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 470–477, 2021.

[21] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, and S. Huang, "Gendexgrasp: Generalizable dexterous grasping," in *IEEE Int. Conf. Robot. Autom.*, 2023, pp. 8068–8074.

[22] Y. Xu, W. Wan, J. Zhang, H. Liu, Z. Shan, H. Shen, R. Wang, H. Geng, Y. Weng, J. Chen *et al.*, "Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4737–4746.

[23] W. Wei, P. Wang, S. Wang, Y. Luo, W. Li, D. Li, Y. Huang, and H. Duan, "Learning human-like functional grasping for multi-finger hands from few demonstrations," *IEEE Trans. Robot.*, 2024.

[24] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao, "D (r, o) grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping," in *IEEE Int. Conf. Robot. Autom.*, 2025.

[25] S. Chen, J. Bohg, and C. K. Liu, "Springgrasp: Synthesizing compliant, dexterous grasps under shape uncertainty," *arXiv preprint arXiv:2404.13532*, 2024.

[26] W. Wan, H. Geng, Y. Liu, Z. Shan, Y. Yang, L. Yi, and H. Wang, "Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 3891–3902.

[27] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, "Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation," in *Conf. Robot. Learn.*, 2023, pp. 594–605.

[28] Q. Liu, Y. Cui, Q. Ye, Z. Sun, H. Li, G. Li, L. Shao, and J. Chen, "Dexrepnet: Learning dexterous robotic grasping network with geometric and spatial hand-object representations," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 3153–3160.

[29] H. Dang and P. K. Allen, "Grasp adjustment on novel objects using tactile experience from similar local geometry," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4007–4012.

[30] H. Yan, H.-S. Fang, and C. Lu, "Avita: Adaptive visual-tactile dexterous grasping," *IEEE Robot. Autom. Lett.*, 2024.

[31] M. Li, Y. Bekiroglu, D. Kragic, and A. Billard, "Learning of grasp adaptation through experience and tactile sensing," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3339–3346.

[32] K. Hang, M. Li, J. A. Stork, Y. Bekiroglu, F. T. Pokorny, A. Billard, and D. Kragic, "Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation," *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 960–972, 2016.

[33] F. Khadivar and A. Billard, "Adaptive fingers coordination for robust grasp and in-hand manipulation under disturbances and unknown dynamics," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3350–3367, 2023.

[34] D. Winkelbauer, R. Triebel, and B. Bäuml, "A learning-based controller for multi-contact grasps on unknown objects with a dexterous hand," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 7331–7337.

APPENDIX I
ADDITIONAL DETAILS OF THE METHOD

A. Preliminaries

Assumptions: We make the following assumptions for the mathematical formulation of the manipulation process: 1) All contacts between the hand and object are regarded as *point contacts with friction* [9]; 2) The location, normal, and force of each contact can be measured by the tactile sensors in real time; 3) The fingers are driven by joint-space PD controllers; and 4) the robot is collision-free at the planned pre-grasp pose, and the fingers required for grasp are in contact with the object at the planned squeeze pose.

Multi-contact grasp: Based on the contact normal $\boldsymbol{n}^i$, we can choose two tangent directions $\boldsymbol{d}^i$ and $\boldsymbol{e}^i$ to construct a contact frame $[\boldsymbol{n}^i, \boldsymbol{d}^i, \boldsymbol{e}^i]$. Defining the contact force $\boldsymbol{f}^i$ in the contact frame, the wrench $\boldsymbol{w}^i \in \mathbb{R}^6$ applied on the object from contact i can be calculated as $\boldsymbol{w}^i = \boldsymbol{G}^i \boldsymbol{f}^i$, where

$$\boldsymbol{G}^i = \begin{bmatrix} \boldsymbol{G}_F^i \\ \boldsymbol{G}_M^i \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}^i & \boldsymbol{d}^i & \boldsymbol{e}^i \\ \tilde{\boldsymbol{p}}^i \times \boldsymbol{n}^i & \tilde{\boldsymbol{p}}^i \times \boldsymbol{d}^i & \tilde{\boldsymbol{p}}^i \times \boldsymbol{e}^i \end{bmatrix} \in \mathbb{R}^{6\times 3} \tag{14}$$

We define $\tilde{\boldsymbol{p}}^i$ as the relative position to the centroid of all contact points. The total wrench is obtained as $\boldsymbol{w} = \sum_{i=1}^m \boldsymbol{G}^i \boldsymbol{f}^i = \boldsymbol{G}\boldsymbol{f}$, where $\boldsymbol{G} \in \mathbb{R}^{6\times 3m}$. Additionally, we define a *normalized wrench* as

$$\bar{\boldsymbol{w}} = \left[\frac{\boldsymbol{G}_F \boldsymbol{f}}{\sum_{i=1}^m \|\boldsymbol{G}_F^i \boldsymbol{f}^i\|_2}; \frac{\boldsymbol{G}_M \boldsymbol{f}}{\sum_{i=1}^m \|\boldsymbol{G}_M^i \boldsymbol{f}^i\|_2} \right] \tag{15}$$

B. Motion-Contact Modeling

Inspired by the interaction control of robot arms [10], we employ a motion-contact model to predict the changes of contact forces caused by arm-hand joint movements.

Consider the ith contact as illustrated in Fig. 6. Accounting for compliance of both the environment and robot joints, one contact involves three key positions: $\boldsymbol{p}_o^i, \boldsymbol{p}_e^i$ and $\boldsymbol{p}_d^i$. The $\boldsymbol{p}_o^i$ represents the undeformed position of the contact, and $\boldsymbol{p}_e^i$ represents the actual contact position. For convenience, all positions, forces, and Jacobians are expressed in local contact frames. Under the quasi-static and linear-elastic assumption [11], the contact force $\boldsymbol{f}^i$ is determined by

$$\boldsymbol{f}^i = \boldsymbol{K}_o \, \boldsymbol{p}_{o,e}^i = \boldsymbol{K}_o(\boldsymbol{p}_e^i - \boldsymbol{p}_o^i) \tag{16}$$

where $\boldsymbol{K}_o$ is the object stiffness matrix. The actual contact position $\boldsymbol{p}_e$ can be calculated using the robot forward kinematics as $\boldsymbol{p}_e^i = \text{FK}^i(\boldsymbol{q})$. Due to the compliance of the robot joints, the *actual* joint positions $\boldsymbol{q}$ will not be the same as the *desired* joint positions $\boldsymbol{q}_d$ (i.e., the position control command). The aforementioned $\boldsymbol{p}_d^i$ represents the desired Cartesian-space position of $\boldsymbol{q}_d$, as $\boldsymbol{p}_d^i = \text{FK}^i(\boldsymbol{q}_d)$, and $\boldsymbol{p}_{e,d}^i = \boldsymbol{p}_d^i - \boldsymbol{p}_e^i$. Note that multiple contacts may exist on one finger. To consider their coupling, we stack all contact points as $\boldsymbol{p} = [\boldsymbol{p}_{(\cdot)}^1; \cdots; \boldsymbol{p}_{(\cdot)}^m]$ and Jacobians as $\boldsymbol{J} = [\boldsymbol{J}^1(\cdot); \cdots; \boldsymbol{J}^m(\cdot)]$. For quasi-static situations, we have

$$\boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{f} = \boldsymbol{\tau} = \boldsymbol{K}_p(\boldsymbol{q}_d - \boldsymbol{q}) \tag{17}$$

where $\boldsymbol{\tau}$ is joint torques and $\boldsymbol{K}_p$ is the position stiffness of the low-level joint PD controller. Considering that the object stiffness is usually much larger than the robot joint stiffness, the derivative $\mathrm{d}\boldsymbol{q}$ is usually much smaller than $\mathrm{d}\boldsymbol{q}_d$ during in-contact movements. Thus, we approximate that $\mathrm{d}\boldsymbol{p}_{e,d} \approx \boldsymbol{J}(\boldsymbol{q}_d)(\mathrm{d}\boldsymbol{q}_d - \mathrm{d}\boldsymbol{q})$, instead of using $\boldsymbol{J}(\boldsymbol{q})$ in [10]. Substituting it into the derivative of (17) and neglecting the derivative of the Jacobian under quasi-static assumption, we have

$$\mathrm{d}\boldsymbol{p}_{e,d} = \boldsymbol{J}(\boldsymbol{q}_d)\boldsymbol{K}_p^{-1}\boldsymbol{J}(\boldsymbol{q})^\top \, \mathrm{d}\boldsymbol{f} \tag{18}$$

Then, substituting (18) into the derivative of (16), we have

$$\begin{aligned} \mathrm{d}\boldsymbol{f} &= \bar{\boldsymbol{K}}_o \, \mathrm{d}\boldsymbol{p}_{o,e} = \bar{\boldsymbol{K}}_o \mathrm{d}\boldsymbol{p}_{o,d} - \bar{\boldsymbol{K}}_o \mathrm{d}\boldsymbol{p}_{e,d} \\ &= \bar{\boldsymbol{K}}_o \mathrm{d}\boldsymbol{p}_{o,d} - \bar{\boldsymbol{K}}_o \boldsymbol{J}(\boldsymbol{q}_d)\boldsymbol{K}_p^{-1}\boldsymbol{J}(\boldsymbol{q})^\top \, \mathrm{d}\boldsymbol{f} \end{aligned} \tag{19}$$

$$\mathrm{d}\boldsymbol{f} = \boldsymbol{K}_s \mathrm{d}\boldsymbol{p}_{o,d} = \left(\boldsymbol{I} + \bar{\boldsymbol{K}}_o \boldsymbol{J}(\boldsymbol{q}_d)\boldsymbol{K}_p^{-1}\boldsymbol{J}(\boldsymbol{q})^\top\right)^{-1} \bar{\boldsymbol{K}}_o \mathrm{d}\boldsymbol{p}_{o,d} \tag{20}$$

where $\bar{\boldsymbol{K}}_o$ is the diagonal stack of $\boldsymbol{K}_o$. If assuming a temporarily constant object pose (i.e., $\mathrm{d}\boldsymbol{p}_o = \boldsymbol{0}$), we have

$$\mathrm{d}\boldsymbol{f} = \boldsymbol{K}_s \boldsymbol{J}(\boldsymbol{q}_d) \, \mathrm{d}\boldsymbol{q}_d \tag{21}$$

Note that $\boldsymbol{K}_s \to \left(\boldsymbol{J}(\boldsymbol{q}_d)\boldsymbol{K}_p^{-1}\boldsymbol{J}(\boldsymbol{q})^\top\right)^{-1}$ when $\boldsymbol{K}_o \to \infty$. Thus, $\boldsymbol{K}_o$ also acts like a damped regularization term to tackle singularities of the finger Jacobians. Additionally, as assuming temporarily constant object poses, the (21) ignores the potential coupling among multiple contacts caused by slight object movements. The full coupling effects are influenced by frictional and supporting forces/torques from the table, which are difficult to model precisely in practice.

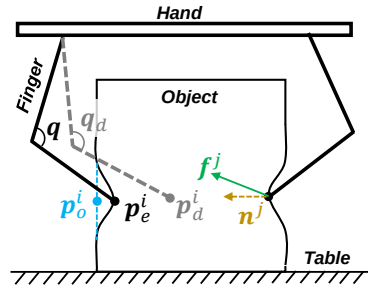


Fig. 6: **Illustration of the mathematical formulation of contacts.** The undeformed, actual, and desired positions of the ith contact as well as the contact normals and forces of the jth contact are shown.

C. Model Predictive Grasping Control

Note that this formulation requires a pre-defined final grasp force F_{ub} which ensures a firm grasp. The work does not tackle the problem of determining F_{ub}, as it depends on the object's mass, friction coefficient, and material softness, which cannot be fully addressed from a control perspective. Possible solutions include visual estimation of object properties via vision-language models or tactile-based slippage detection to increase F_{ub} reactively [3], [12].

Appendix II
Implementation Details

In the experiments, the guiding paths are constructed by linearly interpolating between planned pre-grasp and grasp poses over 2 seconds, and between grasp and squeeze poses over another 2 seconds. The action is computed at 10Hz and then interpolated for high-frequency low-level execution. The optimization is solved using `scipy.optimize.minimize` in Python.

A. Baselines

We choose four baselines for comparison in simulation, including 1) **Open-loop**: The open-loop execution strategy as aforementioned. 2) **Feedback control**: A widely used two-phase tactile-feedback controller concluded from [1]–[5], where each finger are independently controlled. It uses position control for the approaching phase and force control after making contacts. Here the force control law is designed as $\delta\boldsymbol{q}_d^i = \alpha(\boldsymbol{K}_p^i)^{-1}\boldsymbol{J}^i(\boldsymbol{q})^\top(\boldsymbol{f}_d^i - \boldsymbol{f}^i)$, where the desired force $\boldsymbol{f}_d^i$ are set to be equal for all contacts. 3) **Without arm motion**: Only adjusting finger motions to adapt to object position errors without leveraging arm motions, like [7]. We implement it by forbidding arm motions in our framework. 4) **Independent forces**: Using independent equal desired forces for each contact in the grasping phase, while other settings are the same as ours. The first two baselines are well-practiced strategies, and the last two are ablations of ours to respectively show the significance of coordinated arm motions and adaptive contact force coordination. For fair comparison, the desired total force F_{ub} in our method is set to match the averaged final force observed in open-loop execution, and the desired force of each contact in baselines (2) and (4) is set to $F_{\text{ub}}/N_{\text{finger}}$ (the number of fingers).

B. Grasp Pose Planning

As for the grasp pose generation, we first adopt one of the SOTA grasp synthesis approaches [6] to construct a large-scale tabletop grasp pose dataset for the Shadow, Allegro, and LEAP Hand, respectively. To achieve real-time grasp inference from partial observations, we then train flow-based generative models as in [6], [13] using the synthesized grasp poses and single-view point clouds of objects. These models have demonstrated reliable performance and are well applied in the real world. Specifically, 2k objects from the DGN assets [14] are divided into training and testing sets with a 4:1 ratio. Each object is scaled to multiple sizes, placed on the tabletop in different poses, and rendered from various single-view perspectives. The trained models are employed for grasp pose planning.

C. Identification of Robot Joint Stiffness

In the simulation, we use the ground truth values of the robot joint stiffness $\boldsymbol{K}_p$, which are specified in the MuJoCo XML files.

In the real world, we first identify $\boldsymbol{K}_p$ of the fingers. The identification method is as follows. Under quasi-static conditions, we have

$$\boldsymbol{J}(\boldsymbol{q})^{\mathsf{T}}\boldsymbol{f} = \boldsymbol{K}_p(\boldsymbol{q}_d - \boldsymbol{q}) \tag{22}$$

We command one finger to press its fingertip against a fixed object while gradually increasing the force. During this process, we collect the sensed forces $\boldsymbol{f}^i$, joint positions $\boldsymbol{q}$, and commanded positions $\boldsymbol{q}_d$. Then, we apply the linear-squares method to estimate $\boldsymbol{K}_p$ from the collected data. Since the LEAP Hand uses identical motors for all joints, we assume that k_p of all joints are the same. The estimated k_p is approximately 0.8.

It should be noted that the Dynamixel XC-330-M288-T motors on the LEAP Hand actually use a low-level current-based PID controller for high-level position control, and the relationship between joint torques and currents is slightly nonlinear (url [1]). Therefore, the identified $\boldsymbol{K}_p$ should be regarded as an approximation of the actual motor mechanism.

D. Hyper-Parameters

The hyper-parameters are set as $F_{\text{appr}} = 0.2\text{N}$, $\epsilon_b = 0.2$, $\hat{\mu} = 0.3(\text{sim})/0.7(\text{real})$, $\boldsymbol{W}_q = 1.0$ for finger joints and 0 for arm joints, $\boldsymbol{W}_p = \text{diag}(0, 0, 10^2, 10, 10, 10)$, $\boldsymbol{W}_t = \text{diag}(0, 10^3, 10^3)$, and $\boldsymbol{K}_o = 10^5 \times \boldsymbol{I}_3(\text{sim})/10^4 \times \boldsymbol{I}_3(\text{real})$. The δF is set as $\delta F = F_{\text{ub}}/L_g$, in which L_g is the number of waypoints in the interpolated guiding path from the grasp pose to squeeze pose. These values are straightforward to select manually based on reasonable estimates and have not been extensively tuned.

E. Pre-Grasp Poses

To obtain a pre-grasp pose with larger error tolerance to reduce the risk of initial collision under object position uncertainty, we re-compute the pre-grasp pose as the linear extrapolation from the grasp pose towards the original pre-grasp pose generated by the network (three times the original distance).

[1] https://emanual.robotis.com/docs/en/dxl/x/xc330-m288/

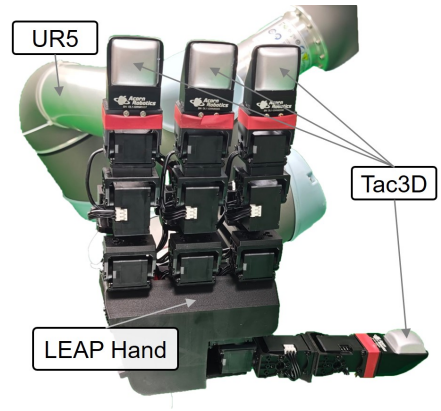


Fig. 7: **Hardware setup for real-world experiments.** We use Tac3D, a vision-based tactile sensor, to measure the contact states.

F. Real-World Experiment Setup

Real-world experiments are conducted on a UR5 arm and a LEAP Hand [15]. Each fingertip is equipped with a vision-based tactile sensor named Tac3D [16], which can robustly estimate the contact surface shape and three-axis contact force distribution. Single-view point clouds of objects are captured by a calibrated Azure Kinect DK camera for grasp generation. To quantitatively evaluate the object movements during grasping, an AprilTag marker is attached to each object and tracked by another RealSense camera. Since the Realsense is not extrinsically calibrated, we report the object position and rotation errors before lifting and the rotation errors after lifting, all computed as relative movements with respect to the initial pose. As tactile sensing is currently available only at the fingertip pads, we restrict our real-world experiments to fingertip grasps. However, our formulation imposes no constraints on contact distribution, as demonstrated in the simulation evaluation. The time cost of MPC solving is around 30 ms on an i9-13900K CPU.

G. Real-World Tactile Sensing

We use Tac3D, a vision-based tactile sensor, to measure contact states in the real world, as shown in Fig. 7. By visually tracking markers on the sensor surface, the sensor estimates local surface deformations and, based on factory calibration, the 3-axis contact forces of each taxel. The total contact force $\boldsymbol{f}^i$ is directly provided by the sensor. To obtain the contact position, we segment the in-contact region using a force threshold and take its centroid. The contact normal $\boldsymbol{n}^i$ is then estimated from the surface normal at the contact position, and the global contact position $\boldsymbol{p}^i$ is obtained through the robot's forward kinematics. We experimentally find that the sensed normal force is relatively accurate, with relative errors below 10%. The directions of the sensed tangential forces are accurate, and their magnitudes appear reasonable, although we have not quantitatively evaluated them. Note that the valid region for reliable tactile sensing is approximately a 2 cm $\times$

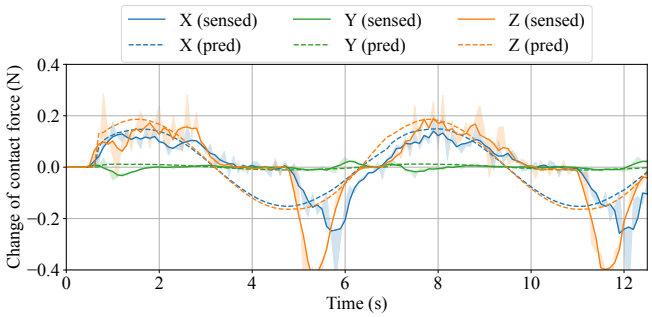


Fig. 8: **Real-world test of the motion-contact model.** The X-axis represents the normal force, while the Y- and Z-axes represent the friction forces.

TABLE III: **Effect of force sensing noises in simulation evaluation.**

Noise scale	SR ↑	Pos. ↓	Rot. ↓	Wrench ↓
0	93.1	2.5±4.6	2.4±4.0	0.17±0.16
0.1	92.9	2.6±4.6	2.5±4.3	0.21±0.16
0.2	92.7	2.9±4.7	2.9±4.4	0.25±0.16
0.5	92.2	4.8±5.6	4.4±4.9	0.35±0.17

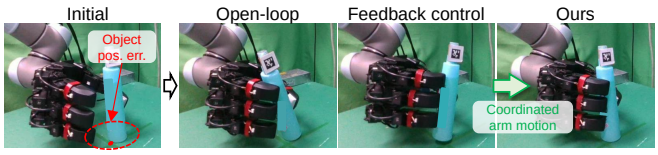


Fig. 9: **Comparison between the methods under position uncertainty.**

2 cm plane, which limits the application of our approach in current real-world experiments.

Appendix III
Accuracy of Motion-Contact Prediction

The analytical motion-contact model described in Section I-B is formulated based on several assumptions:

1) The manipulation process is quasi-static.
2) The object pose is assumed to be temporally constant.
3) Contacts are modeled as point contacts with friction, using a first-order analysis that ignores second-order contact curvatures and frictional moments.
4) No slippage occurs temporally.
5) Robot joint positions are controlled by low-level PD controllers, with joint torques as the control inputs.
6) The model of robot kinematics is accurate.

These assumptions are not fully satisfied in real-world scenarios, making the model a coarse approximation of actual contact dynamics. Nevertheless, we find that its accuracy is acceptable to enable model predictive grasping control and achieve good performance in practice.

We design a relatively ideal experiment to evaluate the accuracy of the motion–contact model in real-world conditions. In this setup, one finger is commanded to follow a sinusoidal trajectory $\boldsymbol{q}_d$ while its fingertip remains in contact with a fixed object. During the motion, we record the sensed forces $\boldsymbol{f}^i$, the actual joint positions $\boldsymbol{q}$, and the commanded positions $\boldsymbol{q}_d$. We then compare the sensed and model-predicted force changes between consecutive steps (i.e., $\boldsymbol{f}_t^i - \boldsymbol{f}_{t-1}^i$), as illustrated in Fig. 8. The results show that the model achieves relatively accurate predictions when the contact forces increase (i.e., when $\boldsymbol{q}_d$ moves the finger further into the object). In contrast, the measured forces behave abnormally when the finger moves back, which will be discussed in the next section.

Appendix IV
Hardware Problems

We find several problems of the LEAP Hand hardware in practice, which may affect the performance of grasping control in the real world.

First, as shown in Fig. 8, when $\boldsymbol{q}_d$ is commanded in the direction of breaking contact, the actual contact force does not decrease immediately but instead remains nearly constant and then drops abruptly. A possible explanation for this phenomenon is related to the motor characteristics of the LEAP Hand. When the commanded change in $\boldsymbol{q}_d$ is small, the motors may not immediately adjust their output torques due to factors such as control deadband, backlash, or static friction in the transmission. As a result, the contact force remains nearly unchanged until the accumulated deviation exceeds a certain threshold, at which point the joint motion occurs and the contact force drops abruptly. This phenomenon significantly affects the contact force tracking performance, particularly when the motors are commanded to move in the reverse direction to adjust frictional forces. As a result, we often observe that the total wrench on the object fails to fully converge before lifting, since the actual frictional forces deviate from the computed desired forces. We plan to investigate this issue further in future work, either by testing alternative hardware or by improving our formulation to mitigate such discrepancies.

Another related phenomenon is that when a human pushes a position-controlled finger, the finger exhibits a much higher resistance (characterized by $\boldsymbol{K}_p$) compared to when it actively pushes other objects. This may not affect the grasping control much as the fingers are actively pressing the object during grasping. However, it affects the calibration of $\boldsymbol{K}_p$, as the calibration data should not be collected by manually pushing the finger.

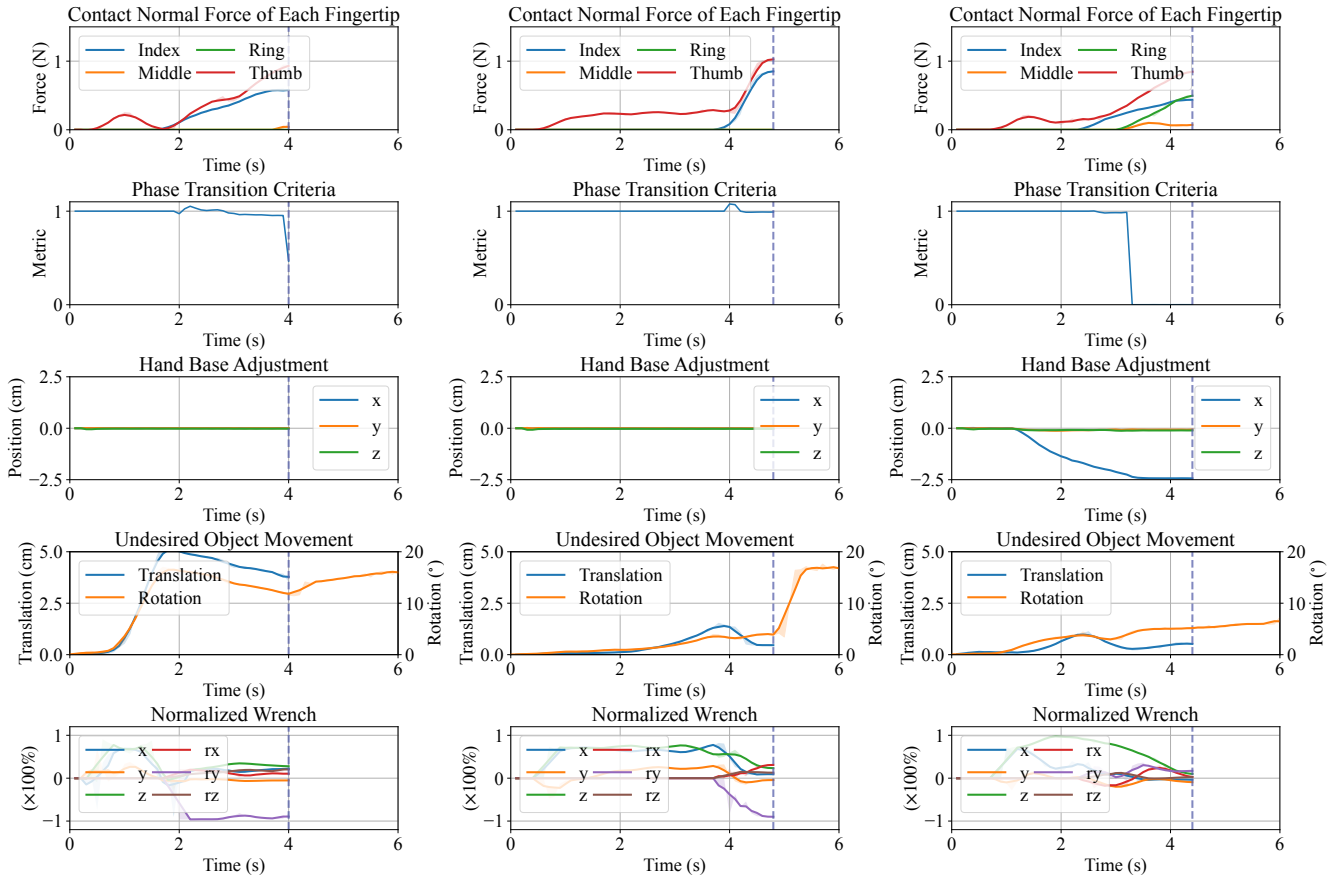
Appendix V
Additional Results

A. Tactile Sensing Noises

As for the Tac3D sensors we used in the real-world experiments, we experimentally find that the sensed normal force is relatively accurate, with relative errors below 10%. The directions of the sensed tangential forces are accurate, and their magnitudes appear reasonable, although we have not quantitatively evaluated them.

We conduct a simulation study to evaluate the effect of tactile sensing noises. We add gaussian noise to the sensed contact forces, and test our approach under the same setting as in Section III-A.1, except that a smaller test set of 3k grasps is used. The standard derivation of the added gaussian noise is defined as (noise scale $\times$ force magnitude) along each axis. The results are shown in Table III, where the reported normalized wrench is computed using the ground-truth contact

Fig. 10: **Visualization of some grasps in our large-scale simulation evaluation.** The evaluation totally involves 15k grasps on 478 objects and three dexterous hands. These figures show the grasp states after lifting from a table, achieved using our approach.



(a) Open-loop (b) Feedback control (c) Ours

Fig. 11: **Manipulation processes for grasping the position-perturbed mosquito repellent bottle** (Fig. 9), comparing (a) open-loop strategy, (b) feedback control, and (c) our method. The time-series plots report contact normal forces, phase transition criteria, hand base adjustments, undesired object movements, and normalized wrenches. The vertical dotted lines indicate the lifting moment.

forces. It can be seen that the grasping performance is not affected much when the noise scale is below 0.2, but undesired object movements are increase significantly at a noise scale of 0.5. It should be noted that while Gaussian noise is used for evaluation, the noise distribution in real-world tactile sensing is more complicated, and further investigation is required.

B. Real-World Case Under Position Uncertainty

We conduct experiments under large position errors in the real world, using the glass vase and mosquito repellent bottle. Examples are shown in Fig. 1 and 9, which shows that our method successfully adapts to the actual object position by leveraging coordinated arm motions, whereas the baselines lead to significant undesired object movements.

C. Analysis of Real-World Manipulation Processes

We provide a detailed analysis of the manipulation process for our method and the two baselines in the case shown in Fig. 9. In this scenario, the mosquito repellent bottle is displaced by approximately 2 cm from its original position toward the thumb, which allows us to evaluate the performance of the methods under significant object position errors. Each method is executed five times. We visualize the recorded variables throughout the manipulation process in Fig. 11, including contact forces from each fingertip, phase transition criteria, arm adjustment motions, undesired object movements, and

TABLE IV: **Distinctive features of our approach compared with existing tactile-driven grasping control methods.**

Work	Method	Coordination-aware phase separation	Arm-hand coordination	Adaptive force coordination	Diverse planned grasps	Evaluation scale
[1]	Feedback control	✗	✗	✗	✗	3 objects (real)
[7]	Feedback control	✓	✗	✗	✗	12 objects (real)
[2]	Feedback control	✗	✗	✗	✓	9 grasps, 3 objects (real)
[3]	Feedback control	✗	✗	✗	✗	8 objects (real)
[17]	Tactile servoing	✗	✓	✗	✗	4 objects (sim) / 2 objects (real)
[18]	Reinforcement learning	-	✗	-	✗	30 objects (sim) / 14 objects (real)
[19]	Imitation learning	-	✗	-	✗	100 objects (real)
[4]	Feedback control	✗	✗	✗	✗	43 objects (real)
[5]	Feedback control	✗	✗	✗	✗	3 objects (real)
Ours	MPC	✓	✓	✓	✓	15k grasps, 478 objects, 3 hands (sim) 8 grasps, 8 objects, 1 hand (real)

normalized wrenches.

Since the object is positioned close to the thumb, the thumb makes the first contact with the bottle in all methods. Afterward, our method activates arm motions to adjust the hand palm's position to better adapt to the actual object position, whereas the baselines keep the palm fixed. During this initial phase (approximately $0 \sim 2$ s), the open-loop strategy moves the thumb along the planned path kinematically, which causes the bottle to be pushed and tilted. The feedback control method regulates the thumb to achieve a desired contact force; since this force is not large enough to move the bottle, the object remains stationary, but the thumb deviates significantly from the planned configuration. In contrast, our method simultaneously moves the thumb along the planned path and regulates the applied contact force on the bottle, by leveraging arm motion adjustments.

In our method, since the palm position is adjusted according to the actual object position, the index, middle, and ring fingers successfully make contact with the object while following their planned paths (approximately $2 \sim 4$ s), resulting in states that satisfy the phase transition criteria. In contrast, in the open-loop execution, the bottle is tilted during the initial process, preventing the middle and ring fingers from making desired contact. The feedback control method encounters the same issue: the middle and ring fingers fail to contact the bottle even at the planned final squeeze pose, because the relative position between the object and the palm deviates from the planned configuration. The undesired grasps formed by only the thumb and index finger cannot apply balanced forces to the object. As a result, during lifting (indicated by the vertical dotted line in Fig. 11), the object rotates significantly when using the open-loop strategy or feedback control method, whereas our method maintains the object close to its initial pose.

Furthermore, in our method, the contact forces of all fingers are increased in a balanced manner during the grasping stage (approximately $3 \sim 4.5$ s), causing the total wrench on the object to gradually converge to zero. In contrast, the baselines generate unbalanced wrenches, which accounts for the large undesired object movements after lifting.

These manipulation processes are also visualized in the supplementary video. The project website is at https://ada-grasp-ctrl.github.io/.

APPENDIX VI
RELATED WORK

A. Dexterous Grasp Planning

Extensive research has been devoted to generating diverse multi-fingered grasp poses. Analytic methods typically compute force-closure grasps from complete object geometries, using either sampling-based search or gradient-based optimization with differentiable metrics, simulators, or bilevel formulations [6], [13], [14], [20], [21]. Building on large-scale synthesized datasets, learning-based approaches employ generative models for real-time grasp generation [22]–[24], possibly with partial observations of objects. Since generated grasps are often imperfect, some open-loop strategies try to introduce intermediate poses (e.g., pre-grasp, grasp, squeeze) to improve robustness, reflecting coordination and phase separation from a purely kinematic perspective [6], [13].

Some works considered object shape uncertainty in planning by placing contacts on object surface with lower uncertainty and applying higher finger compliance for contacts with higher uncertainty [2], [25]. These approaches focus on improving the robustness of planned results with uncertainty modeled from visual observations. In contrast, our method focuses on closed-loop execution, leveraging real-time tactile feedback to adapt to actual positions and shapes.

B. Tactile-Driven Dexterous Grasping Control

Grasping control based solely on vision [26]–[28] is challenging, as vision alone struggles to capture the precise hand-object relationship. To address this, many works incorporated tactile sensing. Some explored continuous re-grasping to adjust contact locations [29], [30] or resistance to disturbances after a stable grasp is achieved [31]–[34]. In contrast, our approach focuses on adaptive and delicate execution of diverse planned grasp, accounting for both the approaching and contact interactions.

A common tactile-driven grasping strategy is to use position control during approaching and force control after making contacts [1]–[5]. Tactile-servoing methods [17] map desired tactile changes to robot motions via tactile Jacobians. Although these methods guarantee firm grasps of objects under shape uncertainty, they require pre-defined desired contacts for each fingertip and do not explicitly consider the multi-contact coordination, often leading to undesired object

movements. An approach-to-grasp strategy was proposed in [7] to handle position uncertainty, which pauses the first contacting finger until others make contacts. However, since only finger motions are used, the final contact configuration may deviate from the target grasp, resulting in unbalanced wrench and object rotation. Tactile-driven reinforcement learning [18] and imitation learning [19] improve grasp success but similarly ignore undesired object movements. In contrast, our approach incorporates arm motions to compensate for object position errors and explicitly coordinates multiple contacts throughout both the approaching and grasping phases. Table IV summarizes existing methods and emphasizes the distinguishing features of our approach.

Appendix VII
Discussion and Conclusion

Discussion: We highlight aspects that are not yet fully explored and potential future improvements: 1) The computing efficiency and control frequency have not yet been fully optimized. The optimization could be solved faster with more efficient solvers. We expect that higher control frequency would further improve force control performance. 2) In the real-world experiments, tracking dynamic desired forces is not very accurate, especially the tangential forces, likely due to limitations in hand's actuation mechanism, tactile sensors, and control delays. Despite these practical challenges, our method still outperforms the baselines. 3) Control of the lifting phase will be studied in future work. For instance, F_{ub} could be adjusted online using slippage detection during lifting, and the direction of frictions could be actively controlled to counteract the gravity.

Conclusion: This work proposes a tactile-driven MPC for coordinated contact control that enables adaptive and delicate execution of planned dexterous grasps under uncertainty. By maintaining coordination across multiple contacts during both the approaching and grasping phases, our approach minimizes undesired in-hand object movements and improves grasp qualities. It is compatible with diverse grasp poses planned by existing well-developed grasp pose generation methods. We evaluate our approach in simulation with 15k grasps across 478 objects on three robotic hands, and in real-world experiments on 8 objects. The results demonstrate that our method achieves planned grasps under uncertainty with higher success rates and reduced undesired object movements compared with the baselines.