APPENDIX I
IMPLEMENTATION DETAILS

A. 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.

B. 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.

C. Real-world Tactile Sensing

We use Tac3D, a vision-based tactile sensor, to measure contact states in the real world, as shown in Fig. 9. 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 $\times$ 2 cm plane, which limits the application of our approach in current real-world experiments.

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

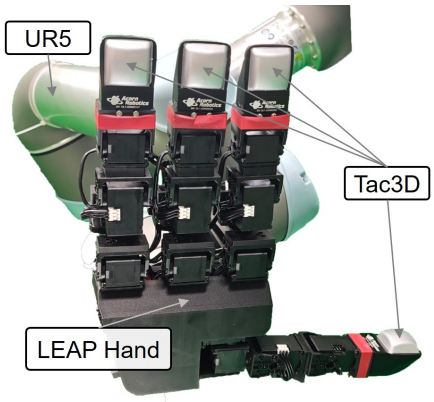


Fig. 9: Hardware setup for real-world experiments. We use Tac3D, a vision-based tactile sensor, to measure the contact states.

APPENDIX II
ACCURACY OF MOTION-CONTACT PREDICTION

The analytical motion-contact model described in Section IV-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

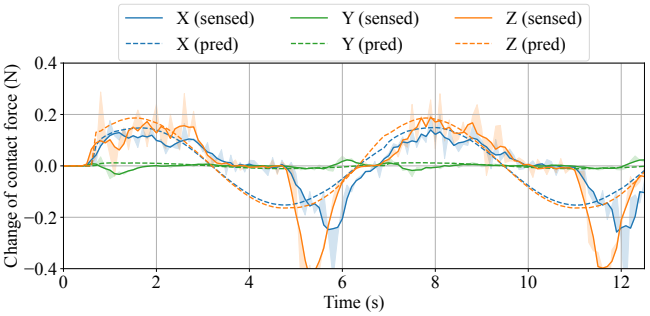


Fig. 10: Real-world test of the motion-contact model. The X-axis represents the normal force, while the Y- and Z-axes represent the friction forces.

(i.e., $\boldsymbol{f}_t^i - \boldsymbol{f}_{t-1}^i$), as illustrated in Fig. 10. 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 III
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. 10, 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 IV
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 V-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 IV, where the reported normalized wrench is computed using the ground-truth contact 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.

TABLE IV: Effect of gaussian force sensing noises in simulation evaluation.

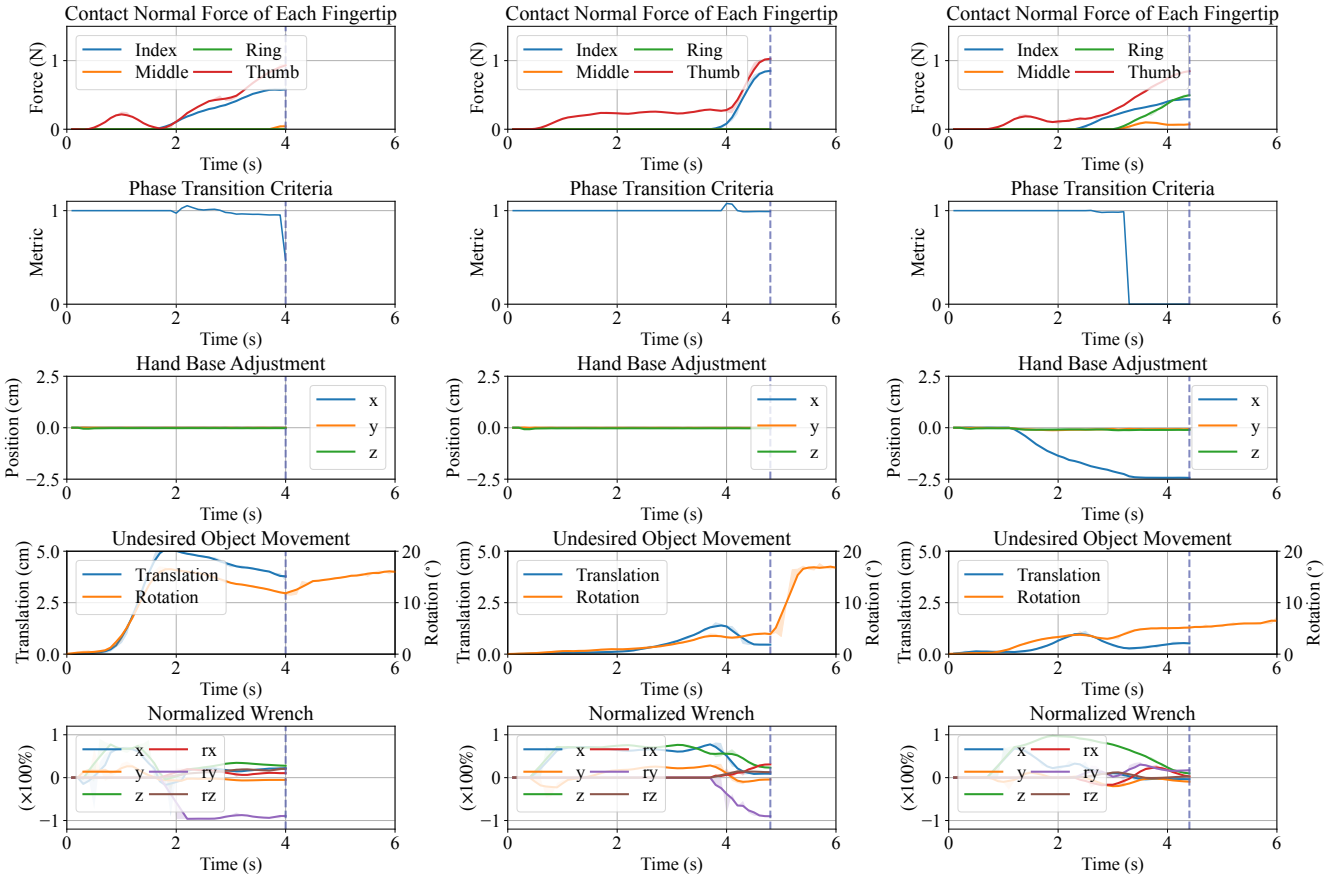
Noise scale	SR ↑	Pos. ↓	Rot. ↓	Wrench ↓
0	93.1	2.5±4.6	2.4±4.0	0.17±0.16
0.1	92.9	2.6±4.6	2.5±4.3	0.21±0.16
0.2	92.7	2.9±4.7	2.9±4.4	0.25±0.16
0.5	92.2	4.8±5.6	4.4±4.9	0.35±0.17

B. 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. 8. 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 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



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

Fig. 11: Manipulation processes for grasping the position-perturbed mosquito repellent bottle (Fig. 8), 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.

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/`.