

A Synthesizing Tactile Pose Estimation Dataset

We generate a tactile pose estimation dataset under practical conditions, where the objects are securely grasped in hand. As illustrated in Fig. 2 (I), for each object O , we first select several grasping approaches based on the object’s canonical frame, such as $X+$ or $Y-$, which

indicate that the left sensor will face the object along the corresponding axis for grasping. The object is initially positioned at the center of the TCP (Tool Center Point) frame. To introduce variability, randomization is applied in two dimensions: (1) along the xy -plane perpendicular to the grasping approach direction, and (2) in the rotational direction around the grasping approach axis. Specifically, the position in the xy -plane is randomized within $[-b_m, b_m]$, where b_m represents the maximum bounding box of the object in the given xy -plane. The gripper subsequently approaches the object along the selected grasping direction, with an indentation randomized within $[0.2\text{mm}, 1\text{mm}]$. Once the grasp is performed, we render the FEM-simulated left and right RGB tactile images. Finally, we filter out data points where the contact region is smaller than 5% of the sensor’s imaging area to ensure data quality. Notably, we observed that ColorJitter serves as an effective augmentation strategy during training across all methods presented in this paper, as demonstrated by the ablation studies in Sec. 4.2. Specifically, we utilized the PyTorch implementation of ColorJitter with brightness, contrast, saturation, and hue parameters set to 0.3, 0.3, 0.3, and 0.1, respectively.

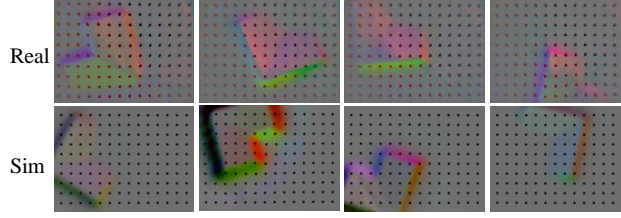


Figure 5: Simulation and real-world tactile images.

B Real-World Data Collection and Hardware Setup

Hardware Setup. We use the GelSlim 3.0 sensor for both simulation and real-world data collection, following the approach of [1]. This sensor provides high-resolution tactile readings in the form of RGB images. It consists of a deformable membrane that responds to contact, and an internal camera that captures the deformation. The sensor transmits tactile observations via ROS as 640×480 compressed images at a frequency of 90Hz. The object CAD model is 3D-printed and securely mounted on a table using a fixture. The sensor is attached to the end-effector of a Franka Panda robot, which randomly samples poses to establish contact with the fixed object model.

Real-world Data Collection. We collected labeled datasets of tactile observations for grasps performed on 30 objects. Each object was mounted in a known position and orientation. For each dataset, we collected pairs of RGB tactile images and their corresponding ground-truth object poses relative to the gripper center. A comparison of tactile imprints obtained from simulation and real-world experiments is shown in Fig. 5. The robotic system used for data collection includes the Franka Panda robot equipped with a GelSlim 3.0 tactile sensor on each finger. The hardware setup is illustrated in Fig. 6. **Data Collection Process.** The data collection procedure consists of the following steps:

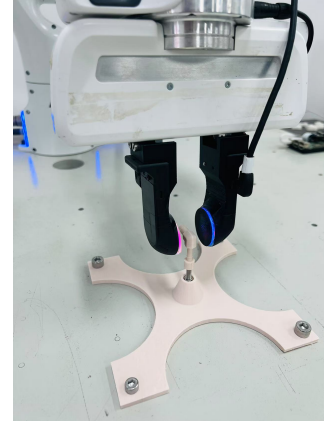


Figure 6: Hardware setup.

- **Mounting the Object:** Each object is attached to a fixed platform using its threaded base. This ensures that the object remains stationary during grasping operations and prevents any slippage.
- **Calibrating Object Pose:** We initially replicate the real-world experimental setup within a simulation environment to estimate the object’s pose relative to the gripper center. The pose is then refined by performing multiple grasps at different orientations and indentation

depths. We iteratively compare tactile images from real-world grasps with those from the simulation and adjust the simulated object pose until both sets of images align closely.

- **Data Collection:** For each object, predefined grasp axes and approach directions are established. We systematically gather tactile data by performing grasps along these directions, varying the gripper’s x , y , and θ positions relative to the object. To simulate real-world uncertainties, we introduce in-plane rotational noise in each grasp. We also conduct grasps at three distinct indentation depths to capture variations in grasp quality and applied force.

C Baseline Implementation

FilterReg utilizes a partial point cloud extracted from the depth image produced by the tactile sensor. It is a probabilistic variant of the traditional ICP algorithm, aligning the tactile point cloud with the CAD model within a Gaussian Mixture Model (GMM) framework. *FilterReg* iteratively minimizes alignment errors to refine the estimated pose. For this method, we provide a rough estimate by introducing noise to the ground truth pose. We implement two variants of *FilterReg*: *FilterReg*(Global) matches the tactile point cloud with the complete point cloud derived from the CAD model. *FilterReg*(Partial) uses the oracle object pose and grasping depth to acquire partial point cloud from the CAD model, serving as an upper bound for *FilterReg*.

Vanilla Regression is a simple baseline where the pose is regressed directly from the tactile RGB image and the point cloud derived from the CAD model. The point cloud is encoded using PointNet, and the tactile image is processed through a combination of convolutional and Vision Transformer blocks, similar to our approach. Features from both modalities are fused and passed through a regression network to predict the pose.

Matching We reimplement Tac2Pose [1], as the original implementation has not been publicly released. For each object, we construct pose grids with a resolution of 2.5 mm and 6 degrees, resulting in approximately 8K to 17K grid points depending on the object geometry. The MoCo module [7] is trained for 30 epochs, ensuring convergence of the training loss for all objects. For fair comparison, the main differences from the original method are: (1) we omit training the image-to-image transfer module, thus making our method purely simulation-based; and (2) during real-world inference, we directly utilize the raw contact masks derived from depth measurements provided by the tactile sensor.

D Extension for Pose Tracking and Uncertainty Estimation

In this section, we provide a detailed description of the extensions to our framework for supporting pose tracking and uncertainty estimation. These additions are crucial for improving the robustness of object manipulation, particularly when dealing with dynamic environments and initial contact uncertainty.

D.1 Pose Tracking

In precise manipulation tasks, objects are often subject to motion, and it is essential to track their poses continuously over time to avoid unintended collisions with the environment. Our pose estimation framework can be naturally extended to pose tracking by modifying the prior distribution during the pre-filtering stage.

Tracking Prior Distribution. In the pose tracking scenario, the primary challenge is to incorporate temporal continuity from previously estimated poses. The tracking prior distribution is designed to reflect this temporal consistency, making it more informative compared to the prior used for the initial pose estimation.

Given the previously estimated in-hand pose \hat{p}_t , we define the tracking prior distribution for the next frame as a Gaussian distribution:

$$\pi_{\text{track}}(\tilde{\mathbf{p}}_{t+1}) = \mathcal{N}(\tilde{\mathbf{p}}_{t+1}; \hat{\mathbf{p}}_t, \sigma_{\text{track}}^2 \mathbf{I}) \quad (9)$$

Here, $\sigma_{\text{track}} = 0.05$ is a hyperparameter that controls the spread of the distribution. This prior encourages the next pose to remain close to the previous estimate while allowing for some flexibility to account for small changes in pose due to motion or noise in the sensing.

Pose Tracking Procedure. Upon receiving a tactile observation T_{t+1} at frame $t+1$, we initialize the particle filter-based ODE solver (PF-ODE) in Eq. 6 with K candidates sampled from the tracking prior π_{track} . This approach leverages the previous pose estimate to more effectively predict the current pose.

Since pose tracking assumes continuity between consecutive frames, the tracking prior is typically closer to the high-density regions of the posterior distribution $p_{\text{data}}(\mathbf{p}|O, T)$, compared to the prior used for initial pose estimation. This allows the pose tracking process to be more efficient.

To further enhance the tracking efficiency, we set a smaller value for the time step parameter t_0 in the PF-ODE, specifically $t_0 = 0.1$, during tracking. This smaller time step allows the model to utilize more informative gradients, which leads to faster convergence during tracking compared to initial pose estimation.

Pose Selection. At the end of the tracking process, we select the pose candidate with the highest energy value as the final estimate. This candidate is the one that best matches the observed tactile feedback, considering both the prior distribution and the sensory information.

Efficiency of Pose Tracking. Compared to frame-by-frame pose estimation, pose tracking is significantly faster, with an inference rate of approximately 10 Hz, which is about 10 times faster than the initial pose estimation pipeline. This speed improvement comes from the reduced number of refinement steps required during tracking, as the previously estimated pose serves as a strong prior.

In practice, it is most beneficial to spend additional time accurately estimating the initial pose at the first contact frame, where the uncertainty is typically higher. Once the initial pose is accurately determined, subsequent frames can be processed efficiently with minimal refinement, allowing for real-time tracking.

D.2 Uncertainty Estimation

Uncertainty plays a crucial role in object manipulation, particularly when the object is grasped at uncertain contact points. Our framework quantifies uncertainty by calculating the variance of the refined pose candidates. The goal is to identify regions with lower uncertainty and guide the robot to re-grasp the object at those locations to improve pose estimation accuracy.

Relative Uncertainty Estimation. The uncertainty of the pose candidates $\{\hat{\mathbf{p}}_i\}_{i=1}^K$ is measured by computing their variance. This is done by first aggregating the candidate poses into a mean pose, denoted as $\hat{\mathbf{p}}_{\text{mean}}$, following the procedure described in GenPose [45].

Once the mean pose is computed, we define the uncertainty as the variance of the pose candidates. Specifically, the uncertainty S^2 is calculated as:

$$S^2 = \frac{1}{K} \sum_{i=1}^K d(\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_{\text{mean}}, O) \quad (10)$$

Here, $d(\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_{\text{mean}}, O)$ is a distance metric that quantifies the difference between the candidate pose $\hat{\mathbf{p}}_i$ and the mean pose $\hat{\mathbf{p}}_{\text{mean}}$ with respect to the object O . For non-symmetric objects, we use the Average Distance Difference (ADD) metric, while for symmetric objects, we use the ADD-S metric. Both of these metrics measure the average Euclidean distance between the model points under two different poses.

577 **Grasp Comparison and Re-grasping.** Given N different grasp candidates $\mathbf{g}_i, i = 1^N$ that result in
578 different tactile images $T_i, i = 1^N$ of the object O , we compare the relative uncertainty of each grasp
579 based on the computed variance S_i^2 of their corresponding pose candidates.

580 The relative uncertainty between two grasps \mathbf{g}_i and \mathbf{g}_j is determined as:

$$\mathbf{g}_i \succ \mathbf{g}_j \iff S_i^2 > S_j^2, \quad \mathbf{g}_{re} = \arg \min_i S_i^2 \quad (11)$$

581 In this context, the grasp with the lowest uncertainty, \mathbf{g}_{re} , is selected for re-grasping. This means
582 that the robot will choose the grasp pose that minimizes uncertainty in the pose estimation, leading
583 to more accurate and reliable manipulation.

584 **Re-grasping for Improved Pose Estimation.** The ability to re-grasp the object at regions of lower
585 uncertainty is critical for improving pose estimation accuracy over time. This approach allows the
586 robot to refine its understanding of the object’s pose by re-grasping at more stable and distinguish-
587 able contact regions, reducing the effect of initial uncertainty and improving the overall robustness
588 of the manipulation process.

589 E Additional Experimental Details

590 In this section, we provide detailed results and discussions for the experiments related to in-hand
591 pose tracking, uncertainty estimation, and the handling of arbitrary contacts.

592 E.1 Pose Tracking and Uncertainty Estimation

593 We conduct extensive studies to evaluate the effectiveness of our method in two key aspects: in-hand
594 pose tracking and relative uncertainty estimation.

595 **In-hand Pose Tracking.** We collect three real-world trajectories on three objects, each consisting
596 of 50 time steps. The initial pose estimate for each tracking trajectory is obtained using our pose
597 estimation method, and subsequent estimations are computed as described in Appendix D. As shown
598 in Tab. 2, our method demonstrates robust performance, maintaining an error within 2mm across all
599 three objects. Additionally, we observe a stable pose tracking frame rate of 10 Hz.

	Bear Housing(mm)	Rail(mm)	Deutsch Connector(mm)
Ours	1.2	1.8	1.5

Table 2: Pose Tracking Results.

600 **Relative Uncertainty Estimation.** We sample 10 sets of data points from the real-world test set of
601 three objects, with each set containing data points from 10 different grasps. For each set, we apply
602 the relative uncertainty estimation method (described in Appendix D) to select the Top-1, Top-3, and
603 Top-5 grasps and compute the average pose error for each selection. As shown in Tab. 3, the pose
604 errors for grasps selected using uncertainty estimation consistently outperform the baseline, where
605 a grasp is randomly chosen from the 10 candidates. Furthermore, as the Top-K selection narrows,
606 the average error decreases, demonstrating the effectiveness of the uncertainty estimation method.

607 E.2 Extending UniTac2Pose to Arbitrary Contacts

608 Our framework is designed to handle an arbitrary number of tactile contacts, thanks to its end-to-end
609 training paradigm. To validate this, we compare three variants of the framework across six objects:

- 610 • *Double*: The original version of the framework, which uses two tactile images as input.
- 611 • *Single*: A variant that ablates the right sensor observation, using only the left tactile sensor
- 612 for render-compare.

	Nut (mm)	Cotter (mm)	Cable Clip (mm)
Random Selection	2.8	2.9	9.9
Top-1 Confidence	1.5	0.6	1.5
Top-3 Confidence	2.1	0.8	4.7
Top-5 Confidence	2.6	1.0	7.7

Table 3: Grasp uncertainty estimation results. We compute the variance of estimated poses of a certain grasp generated by our model. Lower variance indicates lower uncertainty (higher confidence). We compute the mean ADD-S(mm) over top-k confident grasps. The ADD-S error of top-k grasps are lower than the mean error of random selected grasp, demonstrating the effectiveness of our grasp uncertainty estimation.

- *Arbitrary*: A variant where either the left or right tactile image is randomly masked during training and testing.

As shown in Tab. 4, *Double* consistently outperforms both *Single* and *Arbitrary*, as the latter two suffer from higher observation ambiguity. Although *Arbitrary* performs poorly on the *Stud* object, its overall performance is comparable to *Single*, demonstrating that our method can generalize well to scenarios where only a subset of tactile observations is available.

	Single (mm)	Double (mm)	Arbitrary (mm)
Cotter	1.8	1.5	2.5
Hose	2.9	1.2	2.4
Hydraulic	2.3	2.4	2.5
Round Nut	2.9	2.4	3.0
Rail	1.3	1.5	1.2
Stud	1.6	2.2	8.6

Table 4: Evaluation with different contact settings.

F Ablations on three stages.

We conduct ablation studies on the three stages of our method: pre-filtering, refinement, and post-ranking. We also compare refining only the top candidate pose. As shown in Tab. 5, our full method surpasses all variants, demonstrating the effectiveness of each stage.

	Ours	w/o pre-filter	w/o refine	w/o post-rank	refine top-1
Bear Housing	2.4	<u>2.8</u>	13.1	3.1	3.7
Cable Holder	2.4	<u>2.7</u>	12.0	3.3	4.2
Hose	1.5	<u>1.6</u>	16.0	1.7	<u>1.6</u>

Table 5: Ablation on three stages. Ours *w/o pre-filter* randomly samples 16 candidates from the prior, *w/o refine* selects top-1 candidate as the output, *w/o post-rank* reports the average ADD-S of 16 refined poses, and *refine top-1* means refining the top-1 candidate as the output.

G Choices of t .

For pose refinement, we train a model on the Hose object with the time parameter t uniformly sampled from the range $[0, 1]$. During inference, we initialize the RK45 solver with t values ranging from 0.4 to 1.0 in increments of 0.1 and evaluate the corresponding performance. As shown in Fig. 7, initialization with $t \geq 0.6$ results in consistent performance, whereas $t \leq 0.5$ leads to a notable drop

in accuracy. Based on this observation and inference efficiency, we choose $t = 0.6$ as the default setting. For pose selection, we vary t from 0.1 to 0.6 and evaluate performance across these values. As illustrated in Fig. 8, our pose selection method demonstrates robustness to the choice of t .

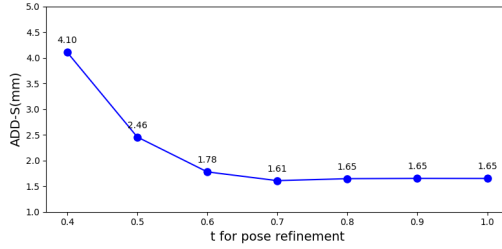


Figure 7: Evaluation of different initial t for pose refinement.

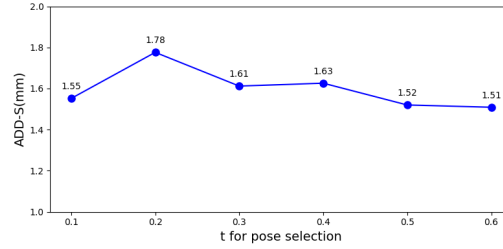


Figure 8: Evaluation of different t for pose selection.

H Training UniTac2Pose on objects from multiple categories.

We train a single model using data from three objects: Round Nut, Hose, and Rail. These objects were selected for their distinct shapes and differing symmetry properties—Round Nut exhibits half-turn symmetry, Rail has quarter-turn symmetry, and Hose is asymmetric. As shown in Tab. 6, the multi-object model successfully fits all three objects, achieving performance comparable to individual models trained separately for each object. This result suggests that our method is capable of learning across multiple object categories within a single model and holds promise for improved out-of-distribution (OOD) generalization as more diverse training data becomes available.

	Round Nut(mm)	Hose(mm)	Rail(mm)
Multi-object Model	2.3	1.5	1.7
Single-object Model	2.4	1.5	1.5

Table 6: Comparison of a model trained on three objects versus models trained on individual objects.

I The sim-to-real performance gap.

We report the evaluation performance in both simulation and the real world in Tab. 7. For most objects, the models achieve comparable performance in simulation. However, the Hook object is an exception, as its uneven surface results in severe partial observations and ambiguities in the contact images, ultimately degrading performance. The sim-to-real performance gap varies across objects, largely due to differences in contact surface geometry, object size, and symmetry. Overall, our method demonstrates a relatively small sim-to-real gap, owing to the use of extensive data augmentation and randomization during training.

	Bear Housing	Cable Holder	Cable Clip	Round Nut	Cotter	Hook	Hose	Hydraulic	Stud	Rail
Sim	1.4	0.8	1.0	1.1	1.3	3.5	0.9	1.3	1.3	1.4
Real	2.5	2.7	1.6	2.4	1.5	3.0	1.5	2.4	2.2	1.5
Δ	1.1	1.9	0.6	1.3	0.2	-0.5	0.6	1.1	0.9	0.1

Table 7: The sim-to-real performance gap. We report ADD-S (mm) and ADD (mm) errors for symmetric and non-symmetric objects respectively. Lower ADD/ADD-S error implies better performance.