Enhancing Generalizable 6D Pose Tracking of an In-Hand Object with Tactile Sensing

Yun Liu*,[1,2], Xiaomeng Xu*,[1], Weihang Chen[3], Haocheng Yuan[4], He Wang[5], Jing Xu[3], Rui Chen[3], and Li Yi[1,2,6]

Abstract*—When manipulating an object to accomplish complex tasks, humans rely on both vision and touch to keep track of the object's 6D pose. However, most existing object pose tracking systems in robotics rely exclusively on visual signals, which hinder a robot's ability to manipulate objects effectively. To address this limitation, we introduce TEG-Track, a tactile-enhanced 6D pose tracking system that can track previously unseen objects held in hand. From consecutive tactile signals, TEG-Track optimizes object velocities from marker flows when slippage does not occur, or regresses velocities using a slippage estimation network when slippage is detected. The estimated object velocities are integrated into a geometric-kinematic optimization scheme to enhance existing visual pose trackers. To evaluate our method and to facilitate future research, we construct a real-world dataset for visual-tactile in-hand object pose tracking. Experimental results demonstrate that TEG-Track consistently enhances state-of-the-art generalizable 6D pose trackers in synthetic and real-world scenarios. *Our code and dataset are available at https://github.com/leolyliu/TEG-Track.

***Index Terms*—Force and Tactile Sensing, Sensor Fusion, Visual Tracking**

I. INTRODUCTION

ACCURATE 6D pose tracking of objects is essential for enabling effective robotic manipulation. Prior research [1]–[3] has demonstrated impressive precision and robustness for tracking known objects using 3D object models. Recent studies have further shifted their focus towards developing generalizable 6D pose tracking methods that can handle novel object instances from known [4]–[6] or even unknown [7]–[9] object categories. In this paper, we contribute to the development of such generalizable 6D pose tracking techniques, specifically addressing the **in-hand** setup shown in Figure 1 that is commonly encountered in robot manipulation tasks. Our goal is to consecutively track the 6D pose of an in-hand object starting from its initial 6D pose. In scenarios where objects are manipulated by robot hands, relying solely on robot proprioception could prove challenging, particularly when external forces from collisions or multi-agent interactions occur. Therefore, it is critical to have an accurate in-hand object tracker that can precisely capture the object's state, especially in contexts involving in-hand manipulation or rich environmental contacts such as peg-hole insertion. Furthermore, this research could significantly benefit human-robot collaboration [10]–[12], where sudden changes in the object's kinematic state caused by interactions are common.

Existing generalizable 6D pose tracking methods face challenges in in-hand manipulation scenarios. Compared with scenes without robot manipulation, visual sensing of the in-hand object become more distorted and less informative due to in-hand occlusions, which could impede existing methods that heavily rely on only visual signals like RGB-D images. As a remedy, tactile sensing could be integrated into the tracking process. By equipping the robot hand with tactile sensors such as GelSight [13], we can capture high-quality geometric and motion signals from contact areas. Such information from tactile sensing can complement the noisy visual sensing caused by occlusions, meanwhile combining with the rapid advancement of tactile sensor technologies [13]–[16], making the integration feasible and promising. Moreover, precise tactile sensing captures accurate motions for object contact regions, providing strong clues for understanding object pose changes.

Therefore, we propose TEG-Track, a general framework for enhancing generalizable 6D pose tracking of an in-hand object with tactile sensing. First, from tactile sensing alone, TEG-Track learns tactile kinematic cues that indicate the kinematic states of the object. Combining with visual sensing, TEG-Track then integrates object kinematic states with existing generalizable visual pose trackers through a geometric-kinematic optimization strategy. TEG-Track can be easily plugged into various generalizable pose trackers, including template-based (introduced in Section V-B), regression-based [5], and keypoint-based [7] approaches.

To evaluate TEG-Track, we curate synthetic and real-world datasets due to the lack of datasets supporting generalizable visual-tactile in-hand object pose tracking research. Since existing datasets [17], [18] only serve for single-frame object pose estimation with a small data scale, we collect a large-scale synthetic object pose tracking dataset with large in-hand motion variations to test TEG-Track widely in various situations. Furthermore, to examine TEG-Track in real scenarios, we contribute a real-world visual-tactile in-hand

Manuscript received: July, 18, 2023; Revised October, 17, 2023; Accepted November, 20, 2023.

This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments.

Project supported by the Young Scientists Fund of the National Natural Science Foundation of China (Grant No. 62203258).

*Yun Liu and Xiaomeng Xu are co-first authors.

Li Yi is the corresponding author.

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China

[2]Shanghai Qizhi Institute, Shanghai, China

[3]Department of Mechanical Engineering, Tsinghua University, Beijing, China

[4]Northwestern Polytechnical University, Xian, China

[5]Center on Frontiers of Computing Studies, Peking University, Beijing, China

[6]Shanghai AI Laboratory, Shanghai, China

Digital Object Identifier (DOI): see top of this page.

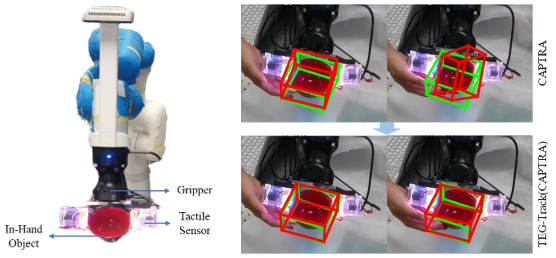

Fig. 1. We propose a general in-hand object pose tracking framework TEG-Track, and evaluate it on our synthetic and real datasets. Our approach enhances generalizable visual trackers such as BundleTrack [7] with tactile sensing. Here we visualize the tracking task as tracking the object's 3D bounding box: green boxes denote ground truth poses whereas red boxes denote estimated poses.

object pose tracking dataset including 200 trajectories covering 17 instances from 5 object categories with careful per-frame object pose annotations. Experiments demonstrate that TEG-Track consistently improves the performances of different generalizable visual pose trackers in both synthetic and real scenarios. Compared to a state-of-the-art generalizable pose tracker BundleTrack [7] on our real evaluation set, TEG-Track achieves 30.9% and 21.4% decreases in the average rotation and translation errors, respectively.

In summary, our main contributions are threefold: 1) To the best of our knowledge, we are among the first to explore generalizable in-hand object pose tracking combining visual and tactile sensing. 2) We present TEG-Track, a visual-tactile framework that learns tactile kinematic cues from tactile sensing and then incorporates them into various visual pose trackers with consistent performance gain. 3) **Dataset**: We construct the first fully-annotated visual-tactile in-hand object pose tracking dataset in real-world scenarios to facilitate future research.

II. Related Work

Generalizable Visual Pose Tracking. Different from instance-level object pose tracking [3], [19] , generalizable object pose tracking methods [4]–[9] aim to track the pose for an unseen object without its 3D model and can be divided into regression-based and keypoint-based methods. Regression-based approaches [5], [8] directly use a neural network to regress 6D object motion from RGB [8] or point cloud [5] sequences, while keypoint-based methods [4], [6], [7], [9] are two-stage that first detect object keypoints and then estimate object pose differences among different frames by keypoint matching. In terms of generalizability, category-level trackers [4]–[6] are limited to objects from known object categories during test time, while category-agnostic ones [7]–[9] can track an arbitrary object without the category information. However, visual signals are the only input for these methods, impeding them to apply to robot manipulation scenarios due to visually heavy occlusions.

Visual-Tactile 3D Perception and Datasets. Tremendous efforts [20]–[28] have been made to combine visual and tactile signals to deal with several 3D perception tasks other than pose tracking. To reconstruct the 3D shape of the object in contact, a line of studies [20], [21], [24], [29] first reconstructs a coarse object mesh by visual sensing and then refines the details with tactile information, and others [22], [23], [27], [28] further design iterative strategies to online search for a local object region with the most informative tactile signals. To estimate the pose of a static object from an active robot movement, a multi-stage method [25] leverages visual and tactile sensing alternatively in different robot states. Various visual-tactile datasets have been collected to facilitate studies on object shape reconstruction [30], [31], in-hand object pose estimation [17], [18], and object grasping [32], [33]. We present the first real-world visual-tactile dataset supporting researches on object pose tracking.

Object Pose Estimation and Tracking via Tactile Feedback. Due to the relatively low quality of visual sensing, previous works have explored object pose estimation and tracking via tactile feedback, but limited to instance-level tracking with a major focus on static grasps. To estimate the object pose in a single frame, a tactile-only method [34] combines multiple tactile images via proprioception. Recent studies [35], [36] in this field combines visual and tactile sensing to achieve object pose estimation. *Wen et.al.* [35] generates pose hypotheses from visual point clouds and then prunes them via hand-object collision check, and *Caddeo et.al.* [36] encodes different modalities to learnable features and fuses them in the feature space. To track the object pose with a known object model, *Álvarez et.al.* [37] separately predicts the object pose using visual and tactile signals alone, then fuses the two predictions by an extended Kalman Filter. Another method [38] fuses visual and tactile point clouds and then align the integral point cloud to the object model.

III. Method

In this section, we introduce TEG-Track in detail. As illustrated in Figure 2, the key idea is leveraging tactile kinematic cues learned from tactile sensing to boost visual pose trackers through a geometric-kinematic optimization strategy. We first revisit generalizable visual pose trackers in Section III-A. We then present tactile kinematic cues that estimate kinematic state of the in-hand object from tactile images and marker flows in Section III-B. Finally, we propose a geometric-kinematic optimization strategy to integrate object kinematic states with various visual pose trackers in Section III-C.

A. Generalizable Visual Pose Trackers

A generalizable pose tracker aims at transferring the learned pose tracking policy to novel objects without their 3D models. For instance, CAPTRA [5] trains one network per object category, and uses it to track an unseen object from the same category during test time. Such a pose tracker can be designed in a template-based, regression-based, or keypoint-based manner. A general object-centric representation (complete object model, NOCS Map [39], object keypoints, etc.) is commonly regarded as an intermediate object feature to build the bridge between visual inputs and 3D object pose. Though generalizable visual pose trackers have achieved impressive results, their heavy reliance on visual sensing makes it difficult to handle heavy occlusion under in-hand situations. TEG-Track leverages a generalizable visual pose tracker to provide

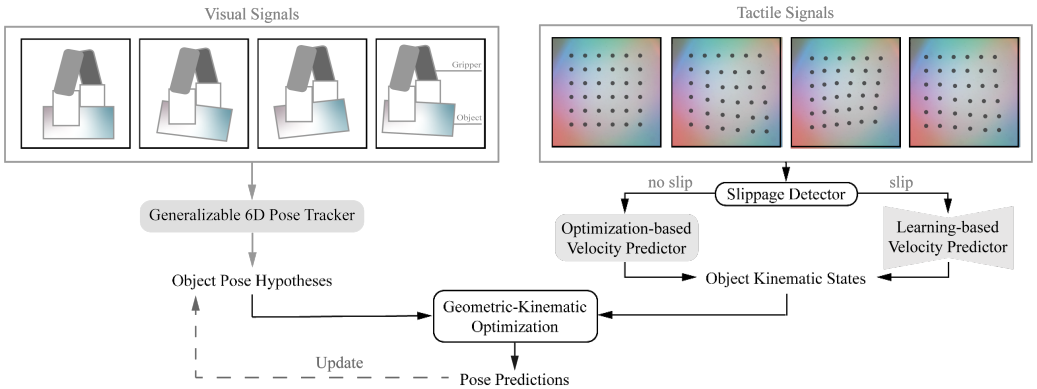

Fig. 2. Overview of TEG-Track. During the online pose tracking process, TEG-Track first estimates object pose hypotheses by a generalizable visual pose tracker, meanwhile predicting the object's kinematic state from tactile kinematic cues learned from tactile signals. Incorporating kinematic states into pose hypotheses, TEG-Track then refines the object poses via a geometric-kinematic optimization strategy and utilizes final results to help subsequent tracking.

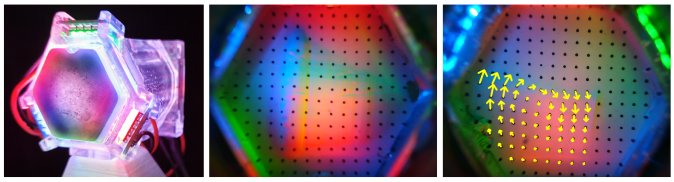

Fig. 3. Tactile sensor, tactile RGB image and detected marker flows.

object pose hypotheses and then incorporates tactile sensing to improve them as final pose results.

B. Tactile Kinematic Cues

As shown in Figure 3, when a robot manipulates an object, tactile RGB images capture high-quality geometry of the object's contact area. Differences between geometries in adjacent frames can indicate kinematic states of the in-hand object consisting of linear and angular velocities, which benefits the understanding of object pose changes. We thus present a learnable mapping from tactile images to object kinematic states, dubbed tactile kinematic cues. In general, such a mapping fulfills in a data-driven manner. Despite complex object motion caused by collisions and interactions, we observe that the mapping is simple when the object approximately sticks to the tactile sensor, thus presenting a two-stage design. First, a slippage detector is used to detect whether the in-hand object is slipping between adjacent frames. Then, object kinematic states are estimated by a velocity predictor via an optimization-based manner if no slippage occurs, otherwise via a learning-based network.

Slippage Detector. Given tactile signals from two adjacent frames, we determine if the in-hand object has slippage. As shown in Figure 3, optical tactile sensors [13], [14], [40] could capture marker positions and further compute marker flows via Nearest-Neighbor matching. We draw inspiration from GelSlim3.0 [40] and detect object slippage by fitting an affine transformation to the marker flows between consecutive frames. Specifically, denoting N_c as the number of marker flows, and $S = \{s_i \in \mathbb{R}^2\}_{i=1}^{N_c}$ and $T = \{t_i \in \mathbb{R}^2\}_{i=1}^{N_c}$ as source and target pixels of marker flows, we compute $E = \min_{A,b} \sum_{i=1}^{N_c} \|As_i + b - t_i\|_2^2$ via the least-square method. A slippage is detected if and only if E is larger than a threshold.

Optimization-based Velocity Predictor. When a visual tactile sensor is in contact with a rigid object, the contact area of the sensor surface sticks to the grasped object and moves simultaneously if no slippage occurs. In this case, the 2D pixel flow f_{c_i} of a contact point p_{c_i} on the sensor, which is approximate to the point on the object surface, can be acquired from the marker flows detected from consecutive tactile images exemplified in Figure 3. f_{c_i} is then transformed to 3D velocity v_{c_i} via tactile depth map estimated by GelSight [13].

As a rigid body, the object's motion can be defined as a linear velocity $v \in \mathbb{R}^3$ and an angular velocity $\omega \in \mathbb{R}^3$ at a pivot point $\hat{p} \in \mathbb{R}^3$ in the world coordinate system. Given N_c contact points detected from the tactile sensor and a point $\hat{p}$, we can optimize v and ω from the detected positions $P_c = \{p_{c_i} \in \mathbb{R}^3\}_{i=1}^{N_c}$ and velocities $V_c = \{v_{c_i} \in \mathbb{R}^3\}_{i=1}^{N_c}$ of these contact points. For each contact point p_{c_i} we have $v_{c_i} = v + \omega \times (p_{c_i} - \hat{p})$. Based on such a constraint from each contact point, we define the energy function:

$$E_{kinematics}(\hat{v}, \hat{\omega}) = \sum_{i=1}^{N_c} \|v_{c_i} - \hat{v} - \hat{\omega} \times (p_{c_i} - \hat{p})\|^2. \quad (1)$$

We set $\hat{p}$ to the estimated object position in the last frame. Then, velocity estimates $\hat{v}$ and $\hat{\omega}$ are computed by minimizing $E_{kinematics}$ via the least-square method. The estimated object kinematic state is formulated as $\{\hat{v}, \hat{\omega}, \hat{p}\}$ and will be used in our geometric-kinematic optimization.

Learning-based Velocity Predictor. When the in-hand object slips on the sensor surface, the contact regions on the object change rapidly, making contact point information of tactile sensors unreliable. We thus design a learning-based approach to directly regress object kinematic states from raw adjacent tactile images without the usage of marker points.

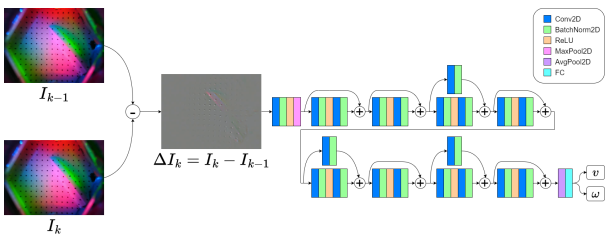

Fig. 4. The network structure of our learning-based velocity predictor.

Figure 4 illustrates our learning-based velocity predictor. Given the difference ΔI_k=I_k-I_{k-1} $\in \mathbb{R}^{384\times288\times3}$ between two adjacent tactile RGB images $\{I_{k-1}, I_k\}$ as input, our network first encodes it to a 512D feature vector using a ResNet [41] structure, then decodes the vector to a 6D velocity representation $[v, \omega]$ via a fully-connected layer. v and ω from the network are defined at the origin of the robot gripper's coordinate system, and finally transformed to velocities at the origin of the world coordinate system by applying known robot gripper pose. The loss function $\mathcal{L} = \lambda_v \|v - \tilde{v}\|_2^2 + \lambda_\omega \|\omega - \tilde{\omega}\|_2^2$ is the weighted sum of mean-square errors on v and ω, where $\tilde{v}$ and $\tilde{\omega}$ are ground-truth velocities. We set $\lambda_v = 100$ and $\lambda_\omega = 0.25$. The network is train by an Adam optimizer with learning rate $1e-4$.

C. Integrating Object Kinematic States with Generalizable Vision-based Pose Trackers

Given the estimated kinematic state of the in-hand object, a straightforward kinematics-only tracking solution is to compute the object pose differences between adjacent frames with object velocities and time differences. However, since velocity information lacks rectification of object pose bias, using such kinematic states alone will yield significant accumulation error. Meanwhile, visual estimate alone has a relatively small accumulation error and maintains a stable error range, while it fluctuates due to heavy sensor noise. We observe that visual and tactile modalities are complementary to achieve a better tracking performance, and thus propose a geometric-kinematic optimization strategy that enhances geometry-based object pose hypotheses from generalizable visual trackers with the estimated kinematic states.

Geometric-Kinematic Optimization. For the i-th frame, the object state is defined as $\{t_i, r_i, v_i, \omega_i, p_i\}$ including position $t_i \in \mathbb{R}^3$, orientation $r_i \in \mathbb{R}^3$, linear velocity $v_i \in \mathbb{R}^3$ and angular velocity $\omega_i \in \mathbb{R}^3$ that are defined at pivot point $p_i \in \mathbb{R}^3$. Object kinematic state prediction $\{\hat{v}_i, \hat{\omega}_i, \hat{p}_i\}$ is provided by velocity predictors in Section III-B. During the online pose tracking process, when handling the k-th frame, we first obtain vision-based object pose hypothesis $\{\bar{t}_k, \bar{r}_k\}$ for frame k from Section III-A, then incorporate object kinematic states $\{\hat{v}_i, \hat{\omega}_i, \hat{p}_i\}$ into pose hypotheses $\{\bar{t}_i, \bar{r}_i\}$ for frames $[k\text{-}N\text{+}1, k]$ to predict current object pose $\{t_k, r_k\}$, where N is a manually-designed parameter. Given a guess $\{\hat{t}_k, \hat{r}_k\}$ for $\{t_k, r_k\}$, the pose predictions $\{\hat{t}_i, \hat{r}_i\}(i \in [k\text{-}N\text{+}1, k\text{-}1])$ can be recursively computed by:

$$
\begin{aligned}
\hat{t}_i &= \hat{t}_{i+1} - (\hat{v}_{i+1} + \hat{\omega}_{i+1} \times \overrightarrow{\hat{t}_{i+1} - \hat{p}_{i+1}}) \cdot \Delta T_{i+1}, \\
\hat{r}_i &= \{I + \sin(\|\hat{\omega}_{i+1}\| \cdot \Delta T_{i+1}) \cdot [\frac{-\hat{\omega}_{i+1}}{\|\hat{\omega}_{i+1}\|}\times] + \\
&\quad [1 - \cos(\|\hat{\omega}_{i+1}\| \cdot \Delta T_{i+1})] \cdot [\frac{-\hat{\omega}_{i+1}}{\|\hat{\omega}_{i+1}\|}\times]^2\} \cdot \hat{r}_{i+1},
\end{aligned}
\tag{2}
$$

where ΔT indicates the time interval between adjacent frames, $[\omega\times]$ denotes the skew-symmetric matrix of ω.

We use $E_t(\hat{t}, \bar{t})$=$\|\hat{t}\text{-}\bar{t}\|_2^2$ to measure the distance between two positions, and $E_r(\hat{r}, \bar{r})$=$\|R(\hat{r})\text{-}R(\bar{r})\|_2^2$ to measure the difference between two orientations, where $R(r)$ indicates the rotation matrix of orientation r. Using pose hypotheses $\{\bar{t}_i, \bar{r}_i\}$ as constraints, the final pose predictions $\{\hat{t}_k, \hat{r}_k\}$ are estimated by minimizing the following energy function via a multi-frame optimization procedure:

$$
E_{track}(\hat{t}_k, \hat{r}_k) = \sum_{i=k-N+1}^{k} E_t(\hat{t}_i, \bar{t}_i) + E_r(\hat{r}_i, \bar{r}_i), \tag{3}
$$

where $\{\hat{t}_i, \hat{r}_i\}(i < k)$ are computed by Equation 2. We use Adam [42] optimizer to find $\{\hat{t}_k^*, \hat{r}_k^*\}$ that minimizes Equation 3, and then obtain $\{\hat{t}_i^*, \hat{r}_i^*\}$ $(i \in [k\text{-}N\text{+}1, k])$ via Equation 2 as our final pose predictions. The final results are used to update object pose hypotheses for the following tracking procedure.

Fig. 5. (a) The data capturing system. The red, green, blue and yellow boxes indicate the Xarm robot arm with Geisight tactile sensors, the Realsense D415 RGB-D sensor, the NOKOV motion capture suite and the in-hand tracked object, respectively. (b) Object categories and instances.

IV. Dataset

The hardware devices of both synthetic and real-world data capturing systems include a Xarm7 robot arm, a Robotiq 2F-140 gripper, a third-view RealSense D415 camera, and two GelSight [13] tactile sensors mounted on the gripper. As illustrated in Figure 5, in our real-world scenario, we additionally set up a NOKOV motion capture system to obtain the ground-truth object poses. The simulation and real-world settings share similar hardware parameters.

A. Synthetic Dataset

We carefully select five object categories (camera, can, bottle, mug, bowl) from NOCS [39] that are easy to hold by robot gripper and incorporate two (earphone, birdhouse) to enrich the variance of object geometry. All objects are chosen from ShapeNet [43] dataset. We establish simulation in the SAPIEN [44] environment, place object collision meshes on the table plane and leverage motion planning to control the robot arm to automatically lift the object, move it, and finally push it to contact the table. For each object category, we collect 10-41 object instances with different geometries and 90-277 successful robot manipulation trajectories. We follow a ratio of 7:3 to randomly separate the objects into training and evaluation sets. The data capturing frequency of robot manipulation is 30 FPS. Each trajectory is approximately 100 frames in total. For each video frame, we capture a visual RGB-D image, tactile RGB images from two tactile sensors mounted on each robot finger, tactile depth images reconstructed via rendering, the position and velocity of each sensor contact point, a 2D object mask, and the object pose directly obtained by the simulation environment. To minimize the gap between simulation and the real world, we model

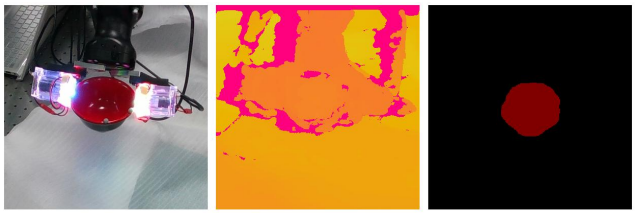

Fig. 6. Visual signals in our real-world dataset: RGB image, depth image, 2D object mask.

TABLE I
AVERAGE MOTION OF THE IN-HAND OBJECT.

Category	Number of Video	AV[1] (°/s)	LV[2] (cm/s)
Camera	44	9.4(±10.4)	1.5(±1.7)
Can	44	8.6(±11.2)	1.5(±1.9)
Bottle	43	7.2(±9.4)	1.5(±2.1)
Mug	36	8.0(±9.4)	1.2(±1.6)
Bowl	33	9.4(±13.4)	0.8(±1.1)

[1]Angular velocity. [2]Linear velocity.

visual and tactile sensor noise following the pipelines in [45], [46], [13], and empirically add multiple Gaussian noises to the positions and linear velocities of contact points based on the sensor noise patterns observed in the real world.

B. Real Dataset

We construct the first real-world visual-tactile in-hand object pose tracking dataset. We believe this is strong support for future relevant research efforts.

We collect 200 videos among 17 object instances (shown in Figure 5(b)) spanning five object categories (camera, can, bottle, mug, and bowl). The integrated system captures the visual-tactile signals and the object pose at approximately 20 FPS, and each video contains 401 RGB-D frames. Besides the object poses captured by the motion capture system, we also provide the 2D masks of the in-hand object via a semi-automatic annotation process using the MIVOS [47] tool. The sorts of visual signals are shown in Figure 6. We follow a ratio of 3:7 to construct training and evaluation sets, respectively. The training set includes three categories (camera, can, bottle) with 6 objects that are not included in the test set, which is only used to train the learning-based velocity predictor in our experiments.

In real scenarios, we manually apply various motions to the in-hand object: 1) We impose collisions between the manipulated object and background objects. The background object held in the human hand touches the manipulated object from multiple positions and directions, which enriches the collision patterns and significantly increases the difficulty of pose tracking. 2) The object could be touched by human hands and hence produce a fast in-hand movement which is smoother than that in 1). 3) We apply zero-force control on the robot arm. The robot arm is moved by manually dragging, hence the robot motion is rich and smooth and could keep the tracking challenging. Table I shows the average object speed in adjacent video frames, indicating the object motion is generally fast with a large variance.

V. Experiments

This section presents a comprehensive evaluation of TEG-Track. We first introduce different baseline approaches including a tactile tracking method (Section V-A) and various vision-based generalizable pose trackers (Section V-B). We then compare TEG-Track with baseline approaches on both synthetic (Section V-C) and real-world (Section V-D) datasets. Additional ablation studies are presented in Section V-E to examine our method design. The tracking speed and robustness of TEG-Track are evaluated in Sections V-F and V-G, respectively. For TEG-Track, parameter N in the geometric-kinematic optimization is set to 5.

A. Kinematics-only Pose Tracking

Leveraging the kinematic cues from tactile sensors, object velocities $\{v, \omega\}$ can be estimated at the speed of 63 FPS on average via kinematic optimization and slippage estimation. Based on $\{v, \omega\}$, we can compute the difference of object pose between adjacent frames by Equation 1. Adding this pose difference to the last pose estimation could directly predict the current pose, hence a continuous pose tracking procedure can be achieved. This method is regarded as a tactile baseline approach in the following experiments.

B. Different Choices of Vision-based Pose Trackers

We incorporate TEG-Track with three generalizable pose trackers: 1) ShapeAlign, a template-based category-level pose tracking method designed by our own, which registers visual point clouds to an estimated object model. ShapeAlign first acquires a category-level complete shape template of the object using PoinTr [48], then predicts object poses by aligning the shape template to visual point clouds via a Chamfer Distance loss. 2) CAPTRA [5], a regression-based category-level pose tracking method. 3) BundleTrack [7], a keypoint-based category-agnostic pose tracking method for novel objects. ShapeAlign and CAPTRA take object depth point clouds alone as visual inputs, while BundleTrack utilizes RGB-D images and object masks.

C. Results on Synthetic Data

Table II summarizes the quantitative results for in-hand object pose tracking on seven object categories from our synthetic dataset, where K denotes the kinematics-only tracking method described in Section V-A, TE denotes our approach TEG-Track, SA denotes ShapeAlign, CA denotes CAPTRA, and BT denotes BundleTrack.

We report the following metrics: 1) $5°5$mm (%), the percentage of pose estimation with translation error ≤ 5mm and rotation error $\leq 5°$. 2) R_e (°), average rotation error. 3) T_e (mm), average translation error.

TEG-Track performs better than kinematics-only and vision-based methods on all three metrics, meanwhile bringing consistent improvements to three different types of generalizable visual pose trackers. For example, TEG-Track respectively brings an increment of 2.3%, 13.3%, and 2.6% to ShapeAlign, CAPTRA, and BundleTrack on the metric $5°5$mm.

TABLE II
QUANTITATIVE RESULTS ON SYNTHETIC DATASET.

Method		K	SA	TE(SA)	CA	TE(CA)	BT	TE(BT)
Visual Signal		N/A	Depth				RGB-D	
Camera	$5°5mm\uparrow$	9.8	42.3	**55.3**	53.5	**60.8**	87.1	**89.8**
	$R_e(°)\downarrow$	4.2	8.1	**6.1**	3.4	**2.6**	3.0	**2.7**
	$T_e(mm)\downarrow$	9.4	4.3	**4.0**	5.2	**4.2**	2.5	**1.9**
Can	$5°5mm\uparrow$	6.9	**88.1**	81.1	84.6	**96.9**	**98.8**	98.7
	$R_e(°)\downarrow$	7.0	**2.3**	2.4	1.3	**0.9**	1.4	**1.2**
	$T_e(mm)\downarrow$	10.5	**1.8**	2.7	3.8	**2.0**	**1.4**	**1.4**
Bottle	$5°5mm\uparrow$	5.0	50.8	**55.2**	61.2	**80.3**	95.5	**96.1**
	$R_e(°)\downarrow$	6.6	5.3	**4.8**	4.8	**2.7**	4.2	**1.3**
	$T_e(mm)\downarrow$	10.6	**5.3**	5.8	5.1	**3.6**	9.6	**4.6**
Earphone	$5°5mm\uparrow$	8.4	41.5	**48.2**	49.0	**56.7**	**67.4**	60.7
	$R_e(°)\downarrow$	3.5	10.9	**6.5**	5.4	**4.2**	4.5	**4.4**
	$T_e(mm)\downarrow$	9.6	**4.1**	4.2	3.3	**2.4**	2.0	**1.2**
Mug	$5°5mm\uparrow$	13.4	32.6	**33.3**	45.0	**58.2**	72.4	**76.9**
	$R_e(°)\downarrow$	3.0	16.0	**7.9**	7.1	**5.5**	5.1	**4.0**
	$T_e(mm)\downarrow$	9.0	4.5	**4.4**	4.0	**3.1**	2.7	**2.2**
Birdhouse	$5°5mm\uparrow$	7.0	29.3	**34.6**	42.9	**50.1**	72.8	**80.3**
	$R_e(°)\downarrow$	3.1	9.4	**6.3**	6.9	**5.3**	6.9	**3.3**
	$T_e(mm)\downarrow$	10.4	**5.4**	5.8	5.8	**5.1**	5.6	**3.4**
Bowl	$5°5mm\uparrow$	2.7	**40.0**	32.9	20.9	**47.3**	48.8	**58.9**
	$R_e(°)\downarrow$	12.5	**8.3**	12.0	19.3	**4.2**	9.3	**6.6**
	$T_e(mm)\downarrow$	11.7	**2.1**	4.2	7.2	**4.8**	7.5	**1.5**
Overall	$5°5mm\uparrow$	7.6	46.4	**48.7**	51.0	**64.3**	77.6	**80.2**
	$R_e(°)\downarrow$	5.7	8.6	**6.6**	6.9	**3.6**	4.9	**3.4**
	$T_e(mm)\downarrow$	10.2	**3.9**	4.4	4.9	**3.6**	4.5	**2.3**

TABLE III
QUANTITATIVE RESULTS ON REAL DATASET.

Method		K	SA	TE(SA)	CA	TE(CA)	BT	TE(BT)
Visual Signal		N/A	Depth				RGB-D	
Camera	$5°5mm\uparrow$	76.9	66.5	**69.8**	0.8	**1.4**	85.2	**88.8**
	$R_e(°)\downarrow$	3.0	4.1	**4.0**	10.1	**6.4**	3.0	**2.8**
	$T_e(mm)\downarrow$	3.3	3.4	**3.2**	14.8	**11.3**	2.4	**2.1**
Can	$5°5mm\uparrow$	82.0	76.4	**78.2**	5.1	**5.6**	85.0	**86.3**
	$R_e(°)\downarrow$	2.4	3.0	**2.8**	3.6	**2.2**	2.8	**2.5**
	$T_e(mm)\downarrow$	3.0	3.0	**2.8**	13.8	**9.1**	2.1	**1.7**
Bottle	$5°5mm\uparrow$	61.2	71.2	**74.2**	2.3	**5.7**	89.2	**91.9**
	$R_e(°)\downarrow$	5.6	3.7	**3.6**	12.0	**5.2**	2.4	**2.1**
	$T_e(mm)\downarrow$	5.0	2.9	**2.7**	16.1	**10.3**	2.0	**1.8**
Mug	$5°5mm\uparrow$	70.4	66.6	**69.5**	1.4	**2.1**	91.4	**94.5**
	$R_e(°)\downarrow$	4.6	4.3	**4.0**	9.6	**6.9**	2.3	**2.1**
	$T_e(mm)\downarrow$	3.8	2.9	**2.6**	13.0	**9.1**	1.9	**1.6**
Bowl	$5°5mm\uparrow$	58.6	58.7	**62.0**	0.2	**0.3**	69.6	**75.8**
	$R_e(°)\downarrow$	6.7	5.4	**5.0**	70.7	**19.8**	8.6	**3.6**
	$T_e(mm)\downarrow$	3.4	4.8	**4.5**	23.4	**15.5**	3.9	**2.5**
Overall	$5°5mm\uparrow$	70.0	67.9	**70.8**	2.0	**3.0**	84.1	**87.5**
	$R_e(°)\downarrow$	4.4	4.1	**3.9**	21.19	**8.1**	3.8	**2.6**
	$T_e(mm)\downarrow$	3.7	3.4	**3.2**	16.23	**11.1**	2.5	**1.9**

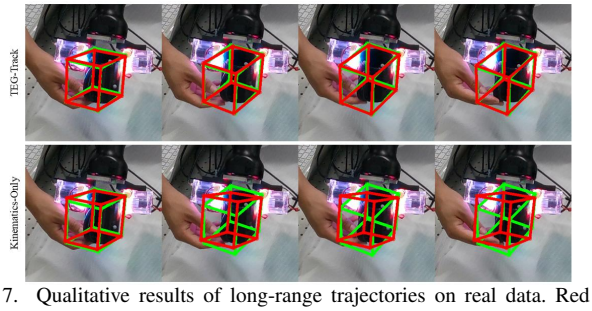

Fig. 7. Qualitative results of long-range trajectories on real data. Red and green bounding boxes indicate the predicted and the ground-truth poses of the in-hand object, respectively.

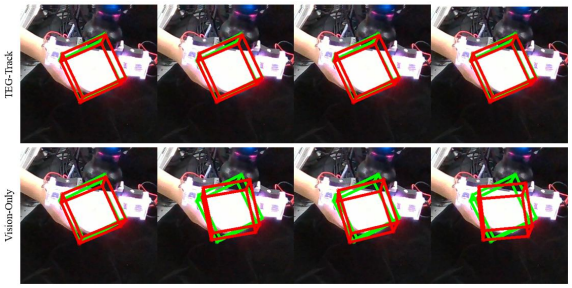

Fig. 8. Qualitative results of consecutive frames on real data.

D. Results on Real Data

Table III shows the quantitative results on our real dataset with the same metrics in Section V-C. Most experimental settings are the same as those on the synthetic dataset, except that CAPTRA is trained on synthetic data with the same object category for the lack of large-scale training data in real world. Consistent with the evaluation results on synthetic data, TEG-Track consistently enhances the performance of different vision-based pose trackers in real-world scenarios. Compared with ShapeAlign, TEG-Track improves learning-based CAPTRA and BundleTrack more significantly. One reason is that the instability of historical pose bias in the learning-based pose predictors could be alleviated by the geometric-kinematic optimization that integrates the information from multiple frames. The cumulative error of object pose severely influence the effect of the kinematics-only method, while it is mitigated by TEG-Track via the fusion with object visual information.

The combination of visual and tactile signals benefits TEG-Track to track accurately and robustly, which helps TEG-Track perform better than the trackers that only leverage a single data modality. Figure 7 shows the object pose predictions of TEG-Track (BundleTrack) at the 50, 150, 250, and 350-th frame of a real-world video, indicating that TEG-Track yields more accurate and stable results compared with the kinematics-only tracking method throughout a long period. Figure 8 shows the tracking results of TEG-Track (BundleTrack) in four adjacent frames, indicating TEG-Track can track more stably and precisely than visual pose trackers with the help of tactile kinematic cues.

E. Ablation Studies

Tactile Kinematic Cues. To examine different tactile kinematic cues, we select real-world video clips containing at least one frame with slippage. Each video clip spans one second. Since two velocity predictors share the same output format, the kinematic states of the object could be fully obtained by either of them. We thus evaluate these two designs on the kinematic-only method and TEG-Track (ShapeAlign) and compare them with our proposed method. Table IV shows the comparison. Compared with the method that ignores the object slippage and uses the optimization-based velocity predictor alone, leveraging a learning-based module to handle slippage cases could consistently achieve performance gain on both kinematics-only and multi-modal tracking approaches, while using such a learning-based module to fully replace optimization would significantly reduce the tracking effect. The reason is that

TABLE IV
ABLATION STUDY ON TACTILE KINEMATIC CUES.

Method	Designs O[1]	Designs L[2]	5°5mm↑	R_e(°)↓	T_e(mm)↓
K		✓	37.69	5.16	4.24
	✓		45.49	3.71	3.16
	✓	✓	**55.56**	**2.95**	**2.76**
TE(SA)		✓	47.05	3.51	2.43
	✓		56.26	2.75	**2.21**
	✓	✓	**59.87**	**2.67**	2.24

[1]Optimization-based VP (Velocity Predictor). [2]Learning-based VP.

TABLE V
GEOMETRIC-KINEMATIC OPTIMIZATION WITH DIFFERENT FRAME NUMBER.

N	1	2	3	5	10	15	20
R_e(°)↓	4.10	4.03	3.99	**3.87**	3.88	3.91	3.96
T_e(mm)↓	3.39	3.29	3.26	**3.19**	3.21	3.27	3.38

contact points could indicate the movement of the sensor surface material with high precision hence its effect varies greatly depending on whether the object is slipping, on the contrary, the neural network could perform better under large object movement while obtaining noisier results in other cases. We thus combine the two designs as our proposed kinematic cues.

Number of frames in Geometric-Kinematic Optimization. While using TEG-Track, frame number N for each geometric-kinematic optimization iteration determines the extent to use tactile signals. We evaluate TEG-Track (ShapeAlign) under different values of N on real-world data and report the mean tracking errors among all categories in Table V. Note that TEG-Track (ShapeAlign) degenerates into ShapeAlign when $N = 1$. The performance of TEG-Track (ShapeAlign) drops with either a small or a large N. A short sequence lacks adequate tactile information to improve vision-based object pose hypotheses, while a long one suffers from a cumulative error of object kinematic states.

F. Tracking Speed

Methods using BundleTrack are tested on a single NVIDIA RTX 2080Ti GPU, while others are tested on a single NVIDIA RTX 3090 GPU. The speed of computing object kinematic states and producing geometric-kinematic optimization is 20 FPS (with N=5), which is faster than 11 FPS for ShapeAlign, 10 FPS for CAPTRA, and 1 FPS for BundleTrack, indicating that TEG-Track can improve the tracking performance for visual pose trackers with low additional time cost.

G. Tracking Robustness

The accuracy of object kinematic states hugely depends on the quality of marker flows detected by tactile sensors. We evaluate TEG-Track under different kinematic noise patterns on synthetic data to test its robustness to various tactile sensing qualities. The kinematic noise contains the possible calibration error of tactile sensors for real scenarios and the detecting error of contact points. We select the *camera* category as the evaluation set and evaluate the kinematics-only method, CAPTRA, and TEG-Track (CAPTRA) under different scales of Gaussian noise added on the position of contact points. As shown in Figure 9, TEG-Track (CAPTRA) is more robust to tactile sensor noise than the kinematics-only tracking method, meanwhile performing better than CAPTRA even under inaccurate tactile sensing.

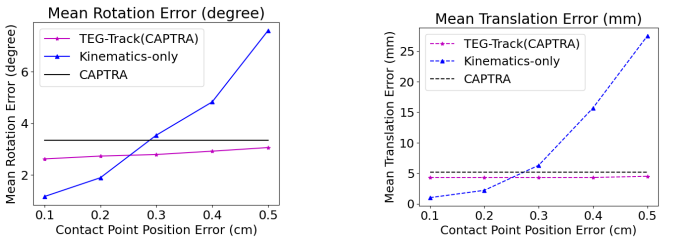

Fig. 9. Mean rotation error (°) and translation error (mm) under different tactile noise patterns. The horizontal axis indicates the standard deviation of the Gaussian noise added to the contact point positions. The solid and dashed lines show rotation and translation errors, respectively.

VI. Limitations and Conclusion

We explore the method leveraging visual and tactile sensing for generalizable in-hand object pose tracking. To this end, we present a novel framework TEG-Track with our core design to model kinematic cues for pose changes and integrate them with visual perception. We incorporate various visual pose trackers into TEG-Track and demonstrate consistent improvements on both synthetic and real-world data. We provide a visual-tactile in-hand object pose tracking dataset supporting relevant studies.

Limitations. One limitation is that our current method deals with only rigid objects but not articulated objects. We may need more complex grippers to support richer contacts with all parts of an articulated object. Another limitation is that we currently do not design domain adaptation techniques for sim-to-real transfer. Multi-modal domain adaptation could also be an interesting future direction.

References

[1] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," 2018.

[2] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] B. Wen, C. Mitash, B. Ren, and K. E. Bekris, "se(3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2020. [Online]. Available: http://dx.doi.org/10.1109/IROS45743.2020.9341314

[4] C. Wang, R. Martín-Martín, D. Xu, J. Lv, C. Lu, L. Fei-Fei, S. Savarese, and Y. Zhu, "6-pack: Category-level 6d pose tracker with anchor-based keypoints," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 059–10 066.

[5] Y. Weng, H. Wang, Q. Zhou, Y. Qin, Y. Duan, Q. Fan, B. Chen, H. Su, and L. J. Guibas, "Captra: Category-level pose tracking for rigid and articulated objects from point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 209–13 218.

[6] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Keypoint-based category-level object pose tracking from an rgb sequence with uncertainty estimation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1258–1264.

[7] B. Wen and K. Bekris, "Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8067–8074.

[8] Y. Du, Y. Xiao, M. Ramamonjisoa, V. Lepetit, *et al.*, "Pizza: A powerful image-only zero-shot zero-cad approach to 6 dof tracking," in *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022, pp. 515–525.

[9] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield, "Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 606–617.

[10] W. Yang, C. Paxton, A. Mousavian, Y.-W. Chao, M. Cakmak, and D. Fox, "Reactive human-to-robot handovers of arbitrary objects," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3118–3124.

[11] J. Laplaza, F. Moreno-Noguer, and A. Sanfeliu, "Context and intention aware 3d human body motion prediction using an attention deep learning model in handover tasks," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4743–4748.

[12] E. Ng, Z. Liu, and M. Kennedy, "It takes two: Learning to plan for human-robot cooperative carrying," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7526–7532.

[13] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[14] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez, "Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1927–1934.

[15] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, *et al.*, "Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3838–3845, 2020.

[16] A. Padmanabha, F. Ebert, S. Tian, R. Calandra, C. Finn, and S. Levine, "Omnitact: A multi-directional high-resolution touch sensor," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 618–624.

[17] S. Dikhale, K. Patel, D. Dhingra, I. Naramura, A. Hayashi, S. Iba, and N. Jamali, "Visuotactile 6d pose estimation of an in-hand object using vision and tactile sensor data," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2148–2155, 2022.

[18] Y. Tu, J. Jiang, S. Li, N. Hendrich, M. Li, and J. Zhang, "Posefusion: Robust object-in-hand pose estimation with selectlstm," 2023.

[19] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1328–1342, 2021.

[20] E. Smith, R. Calandra, A. Romero, G. Gkioxari, D. Meger, J. Malik, and M. Drozdzal, "3d shape reconstruction from vision and touch," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 193–14 206, 2020.

[21] Y. Wang, W. Huang, B. Fang, F. Sun, and C. Li, "Elastic tactile simulation towards tactile-visual perception," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2690–2698.

[22] S. Wang, J. Wu, X. Sun, W. Yuan, W. T. Freeman, J. B. Tenenbaum, and E. H. Adelson, "3d shape perception from monocular vision, touch, and shape priors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1606–1613.

[23] E. Smith, D. Meger, L. Pineda, R. Calandra, J. Malik, A. Romero Soriano, and M. Drozdzal, "Active 3d shape reconstruction from vision and touch," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 064–16 078, 2021.

[24] S. Suresh, Z. Si, J. G. Mangelson, W. Yuan, and M. Kaess, "Shapemap 3-d: Efficient shape mapping through dense touch and vision," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7073–7080.

[25] A. N. Chaudhury, T. Man, W. Yuan, and C. G. Atkeson, "Using collocated vision and tactile sensors for visual servoing and localization," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3427–3434, 2022.

[26] L. Yang, B. Huang, Q. Li, Y.-Y. Tsai, W. W. Lee, C. Song, and J. Pan, "Tacgnn: Learning tactile-based in-hand manipulation with a blind robot using hierarchical graph neural network," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3605–3612, 2023.

[27] L. Rustler, J. Lundell, J. K. Behrens, V. Kyrki, and M. Hoffmann, "Active visuo-haptic object shape completion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5254–5261, 2022.

[28] L. Rustler, J. Matas, and M. Hoffmann, "Efficient visuo-haptic object shape completion for robot manipulation," 2023.

[29] W. Xu, Z. Yu, H. Xue, R. Ye, S. Yao, and C. Lu, "Visual-tactile sensing for in-hand object reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8803–8812.

[30] R. Gao, Y.-Y. Chang, S. Mall, L. Fei-Fei, and J. Wu, "Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations," 2021.

[31] R. Gao, Z. Si, Y.-Y. Chang, S. Clarke, J. Bohg, L. Fei-Fei, W. Yuan, and J. Wu, "Objectfolder 2.0: A multisensory object dataset for sim2real transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 598–10 608.

[32] T. Zhang, Y. Cong, J. Dong, and D. Hou, "Partial visual-tactile fused learning for robotic object recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4349–4361, 2021.

[33] S. Kanitkar, H. Jiang, and W. Yuanl, "Poseit: A visual-tactile dataset of holding poses for grasp stability analysis," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 71–78.

[34] Y. Gao, S. Matsuoka, W. Wan, T. Kiyokawa, K. Koyama, and K. Harada, "In-hand pose estimation using hand-mounted rgb cameras and visuotactile sensors," *IEEE Access*, vol. 11, pp. 17 218–17 232, 2023.

[35] B. Wen, C. Mitash, S. Soorian, A. Kimmel, A. Sintov, and K. E. Bekris, "Robust, occlusion-aware pose estimation for objects grasped by adaptive hands," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6210–6217.

[36] G. M. Caddeo, N. A. Piga, F. Bottarel, and L. Natale, "Collision-aware in-hand 6d object pose estimation using multiple vision-based tactile sensors," 2023.

[37] D. Álvarez, M. A. Roa, and L. Moreno, "Visual and tactile fusion for estimating the pose of a grasped object," in *Iberian Robotics conference*. Springer, 2019, pp. 184–198.

[38] G. Izatt, G. Mirano, E. Adelson, and R. Tedrake, "Tracking objects with point clouds from vision and touch," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4000–4007.

[39] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2642–2651.

[40] I. H. Taylor, S. Dong, and A. Rodriguez, "Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 781–10 787.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[43] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," 2015.

[44] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, *et al.*, "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 097–11 107.

[45] X. Zhang, R. Chen, A. Li, F. Xiang, Y. Qin, J. Gu, Z. Ling, M. Liu, P. Zeng, S. Han, Z. Huang, T. Mu, J. Xu, and H. Su, "Close the optical sensing domain gap by physics-grounded active stereo sensor simulation," 2023.

[46] W. Chen, Y. Xu, Z. Chen, P. Zeng, R. Dang, R. Chen, and J. Xu, "Bidirectional sim-to-real transfer for gelsight tactile sensors with cyclegan," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6187–6194, 2022.

[47] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5559–5568.

[48] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 498–12 507.