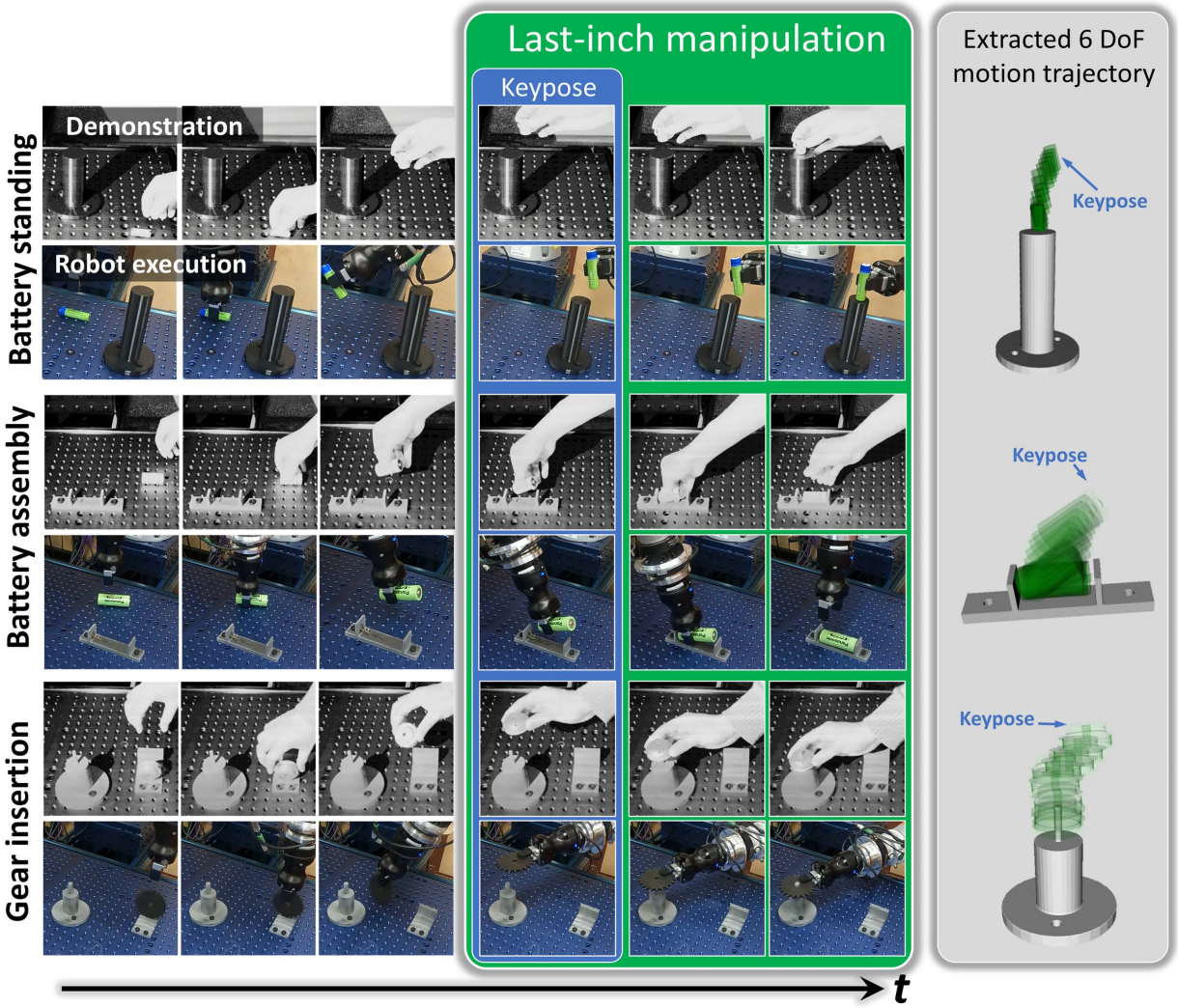
Accepted in Robotics: Science and Systems (RSS) 2022

You Only Demonstrate Once: Category-Level Manipulation from Single Visual Demonstration

Bowen Wen[§†], Wenzhao Lian[†], Kostas Bekris[§] and Stefan Schaal[†]
[†]Intrinsic Innovation LLC, CA, USA.
Email: {wenzhaol, sschaal}@intrinsic.ai
[§]Department of Computer Science, Rutgers University, NJ, USA
Email: {bw344, kostas.bekris}@cs.rutgers.edu



***Abstract*—Promising results have been achieved recently in category-level manipulation that generalizes across object instances. Nevertheless, it often requires expensive real-world data collection and manual specification of semantic keypoints for each object category and task. Additionally, coarse keypoint predictions and ignoring intermediate action sequences hinder adoption in complex manipulation tasks beyond pick-and-place. This work proposes a novel, category-level manipulation framework that leverages an object-centric, category-level representation and model-free 6 DoF motion tracking. The canonical object representation is learned solely in simulation and then used to parse a category-level, task trajectory from a single demonstration video. The demonstration is reprojected to a target trajectory tailored to a novel object via the canonical representation. During execution, the manipulation horizon is decomposed into long-range, collision-free motion and last-inch manipulation. For the latter part, a category-level behavior cloning (CatBC) method leverages motion tracking to perform closed-loop control. CatBC follows the target trajectory, projected from the demonstration and anchored to a dynamically selected category-level coordinate frame. The frame is automatically selected along the manipulation horizon by a local attention mechanism. This**

framework allows to teach different manipulation strategies by solely providing a single demonstration, without complicated manual programming. Extensive experiments demonstrate its efficacy in a range of challenging industrial tasks in high-precision assembly, which involve learning complex, long-horizon policies. The process exhibits robustness against uncertainty due to dynamics as well as generalization across object instances and scene configurations. The supplementary video is available at https://www.youtube.com/watch?v=WAr8ZY3mYyw

I. INTRODUCTION

Significant progress has been achieved in robotic manipulation for known objects [1]–[3], such as methods for acquiring and encoding task-relevant object knowledge. These methods range from training 6D pose estimators with CAD models to end-to-end reinforcement learning from repeated robot interaction with the task object. While manipulation skills for the exact object instance can be acquired with either strategy, it requires significant time and effort to transfer these skills to similar but novel instances. Additionally, the exact object instance is often unavailable until task execution, particularly in less structured environments.

This has motivated recent, promising results in improving the generalizability of robotic manipulation by learning category-level representations, such as semantic keypoints or dense correspondence [4], [5], where transferring manipulation skills across instances is formulated as aligning the semantic keypoints between intra-class object instances [5]. This direction has a few limitations, however:

Manipulation Task Complexity: Trajectory optimization with manually specified goals and constraints has been demonstrated on simple tasks, such as pick-and-place and board-wiping [5], [6]. This process, however, becomes non-intuitive and tedious for more complex and long-horizon tasks. For instance, placing the battery into a spring-loaded charge device requires a sequence of actions, such as pressing towards one end along an angle, then aligning the battery with both ends and pressing down. In such cases, segmenting the manipulation action sequence, and manually specifying the goal and constraints for each segment are challenging.

Robustness: Often in existing work, the manipulated object is assumed to remain static relative to the gripper at the time of the grasp and during manipulation. This assumption enables open-loop execution given only the initially detected keypoints [5] and forward kinematics (FK) [5], [6]. Although force sensing is incorporated in prior work [6], it is not always available and typically handles limited local disturbances. When visual sensing is the primary modality, as in the setting of the current work, open-loop manipulation becomes less reliable, especially in long-horizon manipulation scenarios, as will be shown in the experiments (Sec. V).

Time and cost: To ensure that the training distribution covers the category for learning correspondences, a large number of diverse object instances need to be collected and manually configured for scanning. Multi-view data collection, even when performed by a robot, is time-consuming and cumbersome. The same is true for human annotation of semantic keypoints.

To address these limitations, this work proposes closed-loop, category-level manipulation framework based exclusively on visual feedback, which can be applied to novel objects fast and inexpensively. Leveraging state-of-the-art solutions for model-free 6 DoF object motion tracking, manipulation trajectories are automatically extracted from a single demonstration video. The extracted demonstration trajectory is then represented in a category-level canonical space, which learns solely over synthetic data and enables transferring the manipulation skill across intra-class objects. The manipulation skill is also readily transferable across different task configurations by relying on self-adaptive local correspondences, which are regularized via an attention mechanism. During online robot execution, model-free 6 DoF object tracking is again used for visual feedback to aid the category-level behavior cloning process that guides the robot to follow the canonical trajectory for the target object. Overall, the contributions of this work can be summarized as follows:

- A novel, category-level manipulation framework leveraging object-centric representations trained solely in simulation and model-free 6 DoF object tracking. It achieves robustness and high-precision using only visual feedback. The manipulation skills are transferred across category instances via a Category-level Behavior Cloning (CatBC) process.
- The framework is enabled by one-shot imitation learning where only a single third-person-view video demonstration is used. By virtue of the framework's modular design, the acquired skill is also generalizable to different environments and task configurations. Additionally, it allows quickly teaching the robot with different manipulation strategies without otherwise complicated manual programming.
- An attention mechanism is proposed for dynamic category-level coordinate frame selection. It automatically and dynamically identifies the task-relevant local part of the object for the manipulation task and anchors the category-level, canonical frame for more fine-grained cross-instance alignment when performing behavior cloning.
- This work focuses on challenging manipulation tasks that require high precision and long-horizon actions. Extensive real-world experiments demonstrate significantly superior performance of the proposed framework compared to alternatives for category-level manipulation, in reliability, robustness, and training cost.

II. RELATED WORK

Category-Level Manipulation aims to learn manipulation skills that generalize across instances in the same category. During testing, it is expected to be readily applicable to novel instances, without the need for CAD models or additional robot-object interactions. To achieve this, representative work learns correspondences shared among similar object instances via dense pixel-wise representation [4], [7], [8] or semantic keypoints [5], [9]. In particular, sparse semantic keypoint representations are often assigned task-relevant semantic priors via human annotation [5], [9], [10]. It is cumbersome, however, to manually specify semantic keypoints for each task and

object category. To circumvent this annotation effort, a dense correspondence model was recently proposed, together with self-supervised training over 2D image pairs acquired with a camera-mounted robot [4]. This work aims to avoid manual or time-consuming processes. Instead of reasoning on 2D image pairs that are constrained to specific views, the proposed category-level, object-centric representation allows to directly reason in 3D space. It also imposes an explicit mapping among object instances as training supervision. Therefore, more reliable dense correspondence can be established so as to achieve higher precision than those based on contrastive learning [4], [7], [8], as shown in the experiments (Sec. V).

Behavior Cloning (BC) collects expert demonstrations, and learns a policy taking as input observations and output actions [11]. BC methods can be categorized into model-based [12]–[14] and model-free methods. The latter, which do not estimate dynamic models, can be further grouped into two types: policy learning [15]–[17] and trajectory learning [18]–[20]. More related to this work are vision-based BC methods that aim to learn manipulation skills from demonstration videos [21]–[26]. In addition to video streams, they often rely on robot action labels acquired via extensive robot-object interaction. In contrast, the proposed approach is object-centric and eliminates this requirement. Related work aims to learn from a single demonstration video, but was only applied to simple tasks, such as pushing and stacking, and constrained to a 4D space (3D translation plus in-plane rotation) [27]. In contrast, this work learns object manipulation in the full $SE(3)$ space and considers more complex and high-precision tasks, such as gear insertion and battery assembly.

Visual Feedback Closed-Loop Manipulation involves monitoring task state during execution and provides feedback for reactive planning to compensate execution error and scene updates. Recent work has developed closed-loop manipulation policies by integrating a 6 DoF object motion tracker and a reactive motion planner [3], [28]. Nevertheless, the dependency on an object CAD model for tracking prevents generalization beyond a specific object. Given recent advances in deep reinforcement learning (RL), a number of efforts learn visuo-motor controllers by directly predicting optimal control commands from image observations [29]–[32], or design model predictive controllers with learned visual dynamic models [33]–[35] In contrast to these methods, the proposed closed-loop manipulation framework based on visual feedback does not require robot-object interaction for training, and can be applied to novel objects and environments without time-consuming data collection or re-training.

III. PROBLEM SETUP

The input to the framework per object category $\mathcal{C}$ and associated task $\mathcal{T}_\mathcal{C}$, e.g., `Gear` as an object category and inserting a gear into a shaft as the task, is the following:

- Offline: A collection $\mathbb{O}_{\text{train}}$ of 3D CAD object models in $\mathcal{C}$ for training, which do not include the testing objects.
- Demonstration: A single visual demonstration $\mathcal{D}^\mathcal{C}_\mathcal{T}(\mathcal{O}_\mathcal{D})$ of task $\mathcal{T}_\mathcal{C}$, i.e., an RGBD video (a gray scale and depth video are used in the accompanying experiments) recording the task execution trajectory using one of the training objects $\mathcal{O}_\mathcal{D} \in \mathbb{O}_{\text{train}}$. $\mathcal{D}^\mathcal{C}_\mathcal{T}$ can be a third-person view of a tele-operated robot, or of a human performing the task.
- Online: RGBD images $\mathcal{I}_t$ streamed from a camera during the execution stage when manipulating a new object instance within $\mathcal{C}$.

The objective is to master the manipulation skill from the single demonstration so it can be readily applied to unseen objects in the same category $\mathcal{T}_\mathcal{C}$ without additional fine-tuning or robot-object interactions. In addition, the manipulation skills considered here may require a sequence of actions to be executed, for which solely specifying the target configuration is not sufficient.

IV. APPROACH

Fig. 1 provides an overview of the proposed framework. For each demonstration video frame, the object state is extracted via a model-free 6 DoF motion tracker [37]. This allows to represent the task demonstration with an extracted trajectory $\mathcal{J}^\mathcal{C}_\mathcal{T} := \{\xi_0, \xi_1, ..., \xi_t\}$, where $\xi \in SE(3)$ denotes the object pose at a given timestamp. Object poses are expressed in the receptacle's coordinate frame (e.g., the gear's pose relative to the shaft in the gear insertion task), which allows generalization to new scene configurations regardless of absolute poses.

Given the object pose trajectory parsed from the single visual demonstration, the goal is to "reproject" this trajectory to other object instances in the same category. To this end, this work proposes **category-level behavior cloning (CatBC)**, which follows a virtual target pose trajectory tailored for a novel instance $\mathcal{O}$, reprojected from the demonstrated trajectory $\mathcal{J}^\mathcal{C}_\mathcal{T}$. Specifically, dense correspondence between $\mathcal{O}_\mathcal{D}$ and $\mathcal{O}$ can be established via a category-level canonical space representation, and consequently their relative transformation can be computed. Once the virtual target trajectory for object $\mathcal{O}$ is obtained, behavior cloning reduces to path following by comparing the tracked pose with the reference pose.

The original demonstration video starts before $\mathcal{O}_\mathcal{D}$ is grasped. The initial image frame is used to estimate the category-level pose to initialize the 6 DoF motion tracker. The "last-inch" action sequence is the crucial part of the manipulation process for task success. Consider a concrete example where a gear is grasped without its opening being obstructed. In order to insert the gear into the shaft, it is the final part of the demonstration when the gear is close to the shaft that encodes the relevant spatial relation sequence between the gear and the shaft. Loosely speaking, this spatial relation sequence defines an effective manipulation policy leading to task success. Inspired by this observation, this work identifies first a *keypose* as the pose that corresponds to the start of the "last-inch" demonstration trajectory, and marks the beginning of the category-level behavior cloning process. During the testing stage, a robot path planner is adopted to find a collision-free path that brings the manipulated object to the *keypose*. This step is followed by the category-level behavior cloning for last-inch manipulation until task accomplishment.

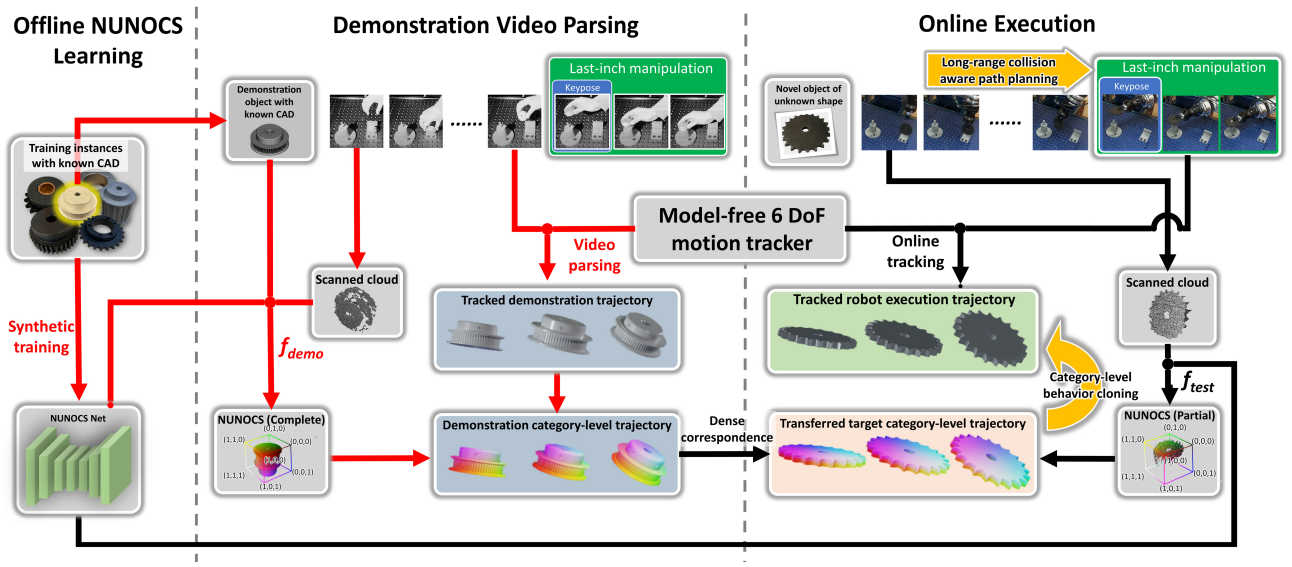


Fig. 1. **During offline NUNOCS learning**, the NUNOCS Net is trained using synthetic data generated using the training CAD models $\mathbb{O}_{\text{train}}$. The purpose of NUNOCS Net is to map an input point cloud to the Non-Uniform Normalized Object Coordinate Space (NUNOCS) for the object category, from which a 9D pose (translation, rotation and 3D scaling) of the observed instance in the category canonical frame can be solved in closed-form [36]. **Upon demonstration**, a model-free 6 DoF motion tracker parses the video and tracks the trajectory of the demonstrated object $\mathcal{O}_\mathcal{D}$. This tracked trajectory is then lifted to a category-level demonstration trajectory by using the NUNOCS representation. In particular, the NUNOCS Net predicts the category-level object pose of the demonstrated object $\mathcal{O}_\mathcal{D}$ in the first video frame. Given the 3D model of $\mathcal{O}_\mathcal{D}$ and the category-level pose, a mapping f_{demo} between the NUNOCS shape and the scanned cloud of $\mathcal{O}_\mathcal{D}$ in the first video frame is obtained. **During testing** on a novel object O, the NUNOCS Net takes the scanned cloud and predicts the mapping f_{test} to its category-level NUNOCS representation. It then establishes a dense correspondence between the NUNOCS representation of O and the NUNOCS shape of $\mathcal{O}_\mathcal{D}$ by finding nearest neighbors, which enables to transfer the demonstration category-level trajectory to a new trajectory tailored for the target novel object O. The manipulation process is split into long-range, collision-free motion and last-inch manipulation. For the latter part, category-level behavior cloning is employed, which aims to clone the target category-level trajectory. Visual feedback for this process is provided by a 6 DoF motion tracker and allows behavioral cloning to adapt the manipulation of the object so that it closely follows the target trajectory until task completion. Red arrows and text denote data flow that occurs exclusively offline.

This work assumes the robot grasps the object in a way that doesn't obstruct the downstream manipulation task.

A. Offline Learning of a Category-Level Representation

Given the single visual demonstration for object $\mathcal{O}_\mathcal{D} \in \mathbb{O}_{\text{train}}$ and in order to "project" the trajectory so it works for a novel object $\mathcal{O}$ during online execution, category-level data association between $\mathcal{O}_\mathcal{D}$ and $\mathcal{O}$ is required. To do so, this work establishes dense correspondence in a 9-dim. space, which refers to a 6D pose and 3D scaling, to relate $\mathcal{O}_\mathcal{D}$ to an arbitrary object instance $\mathcal{O}$ in the same category. The 9-dim. space is an extension of the Normalized Object Coordinate Space (NOCS) [38] developed for category-level 6D pose and 1D uniform scale estimation. The 9-dim. extension, referred to as "Non-Uniform Normalized Object Coordinate Space" (NUNOCS), allows for 3D scaling and has been used before for category-level task-relevant grasp planning [39]. This work adopts the NUNOCS representation for complex, longer horizon tasks and category-level behavior cloning.

Concretely, given the training object models, the category-level NUNOCS representation is obtained by normalizing the corresponding point clouds along each dimension to reside within a unit cube space:

$$\mathcal{O}_\mathbb{C} = (p - p_{min})/(p_{max} - p_{min}), \forall p \in P_\mathcal{O}, \mathcal{O} \in \mathbb{O}_{\text{train}},$$

where p is a 3D point from the object point cloud, $\mathbb{C}$ denotes the canonical unit cube space shared among all objects within the same category. For an arbitrary, unknown instance $\mathcal{O}$, if its NUNOCS representation is available, its relationship with the known object set $\mathbb{O}_{\text{train}}$ can be established.

During online execution, however, only a scanned partial point cloud of the object $\mathcal{P}_\mathcal{O} \in R^{N\times 3}$ is available, preventing the above operation from being applied directly. To address this issue, this work constructs a neural network, referred to here as the NUNOCS Net, to learn a mapping from a scanned partial point cloud of an instance to its configuration in the canonical unit cube space of NUNOCS, i.e. $\Phi(\mathcal{P}_\mathcal{O}) = \mathcal{P}_\mathbb{C} \in R^{N\times 3}$. The mapping Φ is built with a PointNet-like architecture [40]. Different from [39], this work uses a separate branch to predict 3D non-uniform scales simultaneously with the input cloud's point-wise coordinates in the NUNOCS, i.e., $\Phi(\mathcal{P}_\mathcal{O}) = (\mathcal{P}_\mathbb{C}, s)$ where $s = (1, \alpha, \beta)^T \in R^3$. The 3D scales s are normalized w.r.t. the first dimension for compactness. During online execution, the predicted non-uniform scaling is first applied to the predicted NUNOCS coordinates as $s \circ \mathcal{P}_\mathbb{C}$. Subsequently, the 7D uniform scaled transformation between $s \circ \mathcal{P}_\mathbb{C}$ and $\mathcal{P}_\mathcal{O}$ is solved in closed form using least-squares [36], which circumvents exhaustive RANSAC iterations for solving 9D transformation in [39]. Hence, the training loss is the weighted sum of the NUNOCS loss and the scaling loss:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{NUNOCS}} + \lambda_2 \mathcal{L}_s, \tag{1}$$

$$\mathcal{L}_{\text{NUNOCS}} = \min_{Q\in\mathbb{Q}} \sum_{p=1}^{N} \sum_{b=1}^{B} -\overline{\mathcal{P}}_\mathbb{C}^{(p,b)} \log(Q\mathcal{P}_\mathbb{C}^{(p,b)}), \tag{2}$$

$$\mathcal{L}_s = \|s - \overline{s}\|_2, \tag{3}$$

where $\overline{s}$ and $\overline{\mathcal{P}}_{\mathbb{C}}$ are the ground-truth labels. The terms λ_1 and λ_2 are the balancing weights and are empirically set to 1 in all experiments. The NUNOCS representation learning with $\mathcal{L}_{\text{NUNOCS}}$ is formulated as a classification problem by discretizing each coordinate dimension into B bins ($B = 100$ in all experiments) for one-hot vector encoding. This classification formulation with a cross-entropy loss is more effective than regression as it reduces the continuous solution space to a finite number of bins B [38]. To handle symmetrical objects, $Q \in \mathbb{Q}$ are the equivalent symmetric transformations [38], which are pre-defined for each category. For learning the non-uniform scale mapping with $\mathcal{L}_s$, the L_2 loss is adopted.

The NUNOCS Net is trained solely with simulated data and then directly applied to the real world without any retraining or fine-tuning. To achieve this, a synthetic training data generation process is developed using Blender [41] and the details are introduced in the appendix. In order to bridge the sim-to-real domain gap, domain randomization [42] is employed by extensively randomizing the object instance types, physical parameters, object's initial poses, and the table height. In addition, the bidirectional alignment technique over depth modality [43] is employed to reduce the discrepancy between the simulated and real world depth data. Compared to alternatives [4]–[6], the NUNOCS learning process dramatically reduces human effort by avoiding real-world data collection and additional manual annotation of keypoints [5], [6]. Dense point-wise correspondence inherited from NUNOCS also circumvents the trouble of defining the number of semantic keypoints and their locations for each category or task. While there is prior work that builds upon dense correspondence [4], matching points over 2D image pairs tends to suffer from view ambiguity and occlusions, as validated in the experiments.

B. Model-free 6 DoF Object Motion Tracking

This work utilizes 6 DoF motion tracking for 2 purposes. During the demonstration phase, it parses the recorded video to extract the 6 DoF motion trajectory of the manipulated object in the receptacle's coordinate frame. Compared to learning directly in the image pixel space [44], this approach disentangles the object of interest from the background and represents the extracted trajectory independent of any specific scene configuration. This enables the representation to generalize to novel environments, where the initial object and receptacle placement might differ from the demonstration.

During online execution, motion tracking provides visual feedback for closed-loop control when manipulating a testing object. Uncertainty due to dynamics is unavoidable in manipulation, such as unsynchronized finger touching during grasping, in-hand object slipping and object motion caused by contacts with the receptacle during last inch manipulation. In the context of high-precision manipulation tasks, as in this work, the uncertainty introduces non-negligible errors and complicates the process of following the nominal trajectory.

For the above purposes, this work leverages an RGBD-based 6 DoF object motion tracker `BundleTrack` [37], which provides near real-time feedback to guide the execution by comparing the estimated object pose at each timestamp against the demonstrated nominal trajectory. Alternative 6 DoF object motion trackers that rely on object CAD models [43], [45], [46] would impede instant deployment to novel objects.

For initialization, the tracker takes as input the binary segmentation mask indicating the foreground region of interest. In particular, it reuses the inferred object mask also used by the NUNOCS Net (Sec. IV-A). The segmentation is computed by background point cloud subtraction and plane removal, followed by DBSCAN clustering [47]. Alternative learning-based segmentation methods on 2D image [48], [49] or 3D point cloud [50] can also be used. At each timestamp $\tau \in \{1, 2, ..., t\}$, the process tracks the object motion relative to the initial timestamp in the camera's frame, $\xi_{0\to\tau} \in SE(3)$. To obtain the absolute category-level pose in the camera frame at τ, a transformation is applied to the initial category-level pose ξ_0 inferred by the NUNOCS Net, i.e. $\xi_\tau = \xi_0[(\xi_0)^{-1}\xi_{0\to\tau}\xi_0] = \xi_{0\to\tau}\xi_0 \in SE(3)$. As the statically-mounted camera has been calibrated relative to the robot base, the poses ξ_τ can be further expressed in the robot frame so as to be used by other modules. During the demonstration, the receptacle pose is estimated using a model-based pose estimation approach [51]. In this way, the demonstrated object pose trajectory can be represented in the receptacle's coordinate frame, enabling behavior cloning in different environment configurations.

Once initialized, the method is able to track the object at the RGBD camera's acquisition rate (10 Hz) without any re-initialization. The neural network weights in the tracker are adopted from the available, open-sourced implementation[1] and remain fixed in all experiments, eliminating the requirement of training data collection. Since the tracker does not require CAD models for the objects, it can be instantly applied to arbitrary objects in both scenarios, i.e., demonstration video parsing and online visual feedback control.

C. Category-Level Behavior Cloning as Last-Inch Policy

Algorithm 1: Category-Level Behavior Cloning

Input: tracker, **robot**, $\mathcal{J}$ `// virtual target trajectory`
`// starting from the keypose`

for $\overline{\xi}_i$ *in* $\mathcal{J}$ **do**
  $\xi \leftarrow$ **tracker**.$get\_pose()$ `// object SE(3) pose`
  $\Delta\xi \leftarrow \overline{\xi}_i \ominus \xi$ `// relative pose`
  $q \leftarrow$ **robot**.$get\_joints()$ `// joint configuration`
  $J \leftarrow$ **robot**.$get\_jacobian(q)$ `// Jacobian matrix`
  $\Delta q \leftarrow J^{\dagger}\Delta\xi$ `// Jacobian steering`
  $q' \leftarrow q + \Delta q$
  robot.$reach(q')$ `// move joints`
end

Once the object pose trajectory is obtained by parsing the demonstration video (Sec. IV-B), a canonicalization in the NUNOCS space is performed (Sec. IV-A) to establish dense correspondence between the demonstrated object $\mathcal{O}_{\mathcal{D}}$ and the testing object $\mathcal{O}$, as shown in Fig. 1. This effectively

[1] https://github.com/wenbowen123/BundleTrack

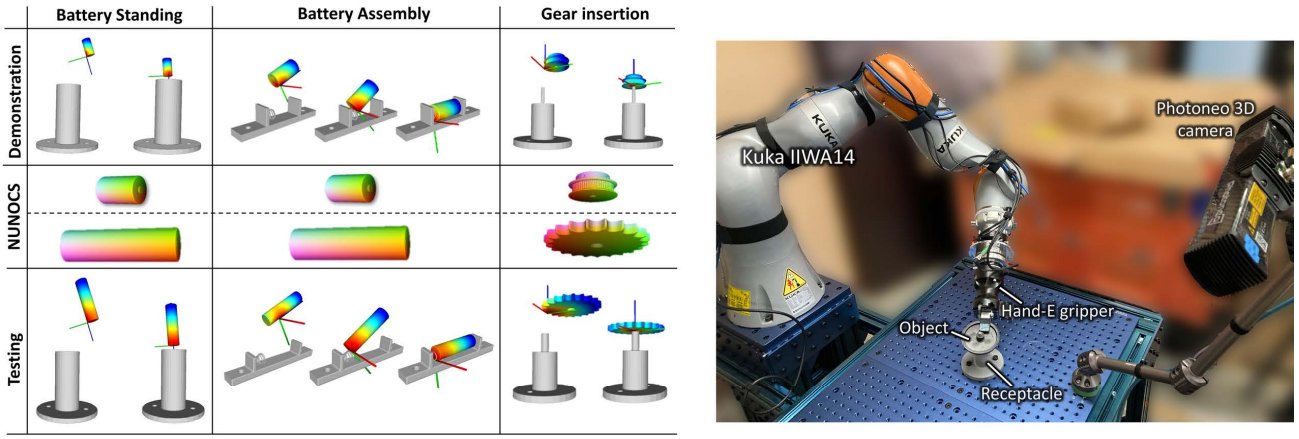


Fig. 2. **Left:** Heatmap visualizations of the local attention mechanism are shown in the top and bottom rows for the training and testing objects respectively. During demonstration, given the 3D model of the demonstration object and its paired receptacle, an attention heatmap is precomputed. During online execution, the attention heatmap can be transferred to a novel object given the dense correspondence established through their NUNOCS representations shown in the middle row. The attention mechanism allows to dynamically anchor the coordinate system to the local attended point (located at the warmest color), capturing the variation between demonstration and testing objects in scaling and local typology. The testing objects' 3D models are shown for visualization only and are unknown during execution. **Right:** Experimental hardware setup.

"reprojects" the demonstration trajectory to a virtual target trajectory tailored for $\mathcal{O}$. This allow to replay the actions performed in the demonstration so as to accomplish the task with the novel instance $\mathcal{O}$. Specifically, even without prior knowledge about $\mathcal{O}$, following the target trajectory allows task success. This is because by following the virtual target trajectory, the novel object $\mathcal{O}$ traverses the critical task-relevant configurations relative to the receptacle in a desired sequence. This process is referred to here as **Category-level Behavior Cloning (CatBC)**. CatBC realizes a manipulation policy by replicating the demonstration trajectory, which is defined in an object-centric manner and is agnostic to how the object is grasped by the robot. To increase the robustness of CatBC against uncertainties due to manipulation dynamics, constantly updated object state is needed to ensure the object follows the target path. In contrast, previous treatments of the grasped object as an extended kinematic frame [5], [6], [52] tend to be brittle, as shown in the accompanying experiments (Sec. V).

Alg. 1 outlines the CatBC process. A model-free 6 DoF object motion tracker (Sec. IV-B) provides online visual feedback for closed-loop control. During the last-inch manipulation, dynamics uncertainty arising from contacts and robot-object-receptacle interaction causes the behavior cloning process to deviate from the desired trajectory. It is thus necessary to discretize the trajectory densely (the neighboring poses' distance is around $2mm$ or $2°$ in our implementation) so that the visual feedback ensures the target trajectory is followed to the highest degree when moving to the next immediate goal along the trajectory.

D. Dynamic Category-Level Frame via Local Attention

Typically, 6D object poses are used to represent a manipulated object's state in a predefined local coordinate system and define the transformation relative to a task-relevant target frame. Given a 6D pose and the 3D model of an object instance, any point on the rigid object is always uniquely defined. This allows to implicitly define the object's parts and orientations relevant to a specific task. Nevertheless, it is challenging to adopt one constant, category-level, canonical coordinate frame for different tasks while capturing the geometric variation across all instances.

Consider the *Batteries* class as an example. If the commonly selected center-of-mass is used as the canonical coordinate frame origin, when aligning a novel battery instance to the demonstrated one, it may collide with or float away from the receptacle, depending on its particular larger or smaller diameter. Instead, the surface center of one of the terminal ends (e.g., the negative pole of the battery) is more appropriate as the frame origin for the battery standing task. In contrast, the negative pole's lowest edge center is more appropriate as the frame origin for the battery assembly task, which comes in contact with both the receptacle and the spring. Nevertheless, it is cumbersome to manually specify a suitable local frame for each task. Moreover, the task-relevant local frame may not stay constant throughout a complex task.

This work proposes a local attention mechanism to automatically and dynamically select an anchor point p_τ^* that defines the origin of the category-level canonical coordinate system, as in Fig. 2. Concretely, during the demonstration, a signed distance function (positive external to the object's surface) of the receptacle is computed, noted as $\Omega(\cdot)$ [53]. Then, an attention heatmap and the anchor point at any timestamp along the manipulation horizon $\tau \in \{0, 1, ..., t\}$ are computed as:

$$Attn_\tau(p_i) = 1 - \frac{exp(\Omega(\xi_\tau p_i))}{\sum_j exp(\Omega(\xi_\tau p_j))}, \tag{4}$$

$$p_\tau^* = \underset{p_i}{\text{argmax}}\, Attn_\tau(p_i), \tag{5}$$

where p_i are the points on the 3D model of $\mathcal{O}_\mathcal{D}$. ξ_τ denotes the demonstration object's pose relative to the receptacle along the trajectory $\mathcal{J}_\mathcal{T}^\mathcal{C} := \{\xi_0, \xi_1, ..., \xi_t\}$, which is parsed from the demonstration video. Intuitively, the local object part that is closer to the receptacle should be assigned higher attention. During online execution, however, the novel object's shape is not available to directly compute the attention heatmap.

By virtue of the established dense correspondence using NUNOCS (Fig. 2 middle row), the attention heatmap can be transferred from $\mathcal{O}_{\mathcal{D}}$ to novel objects (Fig. 2 last row). The attention mechanism allows to dynamically anchor the coordinate system to the local attended point, capturing the variation between demonstration and testing objects in scaling and local typology. The coordinate system is only translated to attend to the task-relevant local region, while the orientation remains the same as the original learned category-level canonical frame.

Compared to using a small number of pre-specified keypoints as in previous work [5], [6], the proposed dynamic attention mechanism allows for versatile implicit keypoint generation augmented with orientation information, which self-adjusts along the manipulation horizon. This improves expressiveness, reduces human effort, and enables high precision category-level behavior cloning.

E. Grasping the Object and Transferring it to the Keypose

The proposed framework is not constrained to a particular grasp planning approach and in general, any CAD model-free grasping planning method [54], [55] can be adopted. As long as the grasp complies with the downstream manipulation task, as in the considered setup, task-relevant grasp planning can be adopted. The core idea is to utilize the category-level affordance priors unified by NUNOCS representations to propose and select task-relevant grasps [39]. Grasp planning is not the focus of this work.

The proposed framework is robust to uncertainty due to robot-object interactions, such as the object moving during finger closing. This is achieved due to the online object state update from the 6 DoF tracker. Once the object is grasped, the tracker provides the latest in-hand object pose, which serves as the start pose for a path planner (RRT* [56] in the implementation) to find a collision-free path that transports the object to the *keypose*. The decomposition into long-range, collision-free motion to bring the object to the *keypose*, and then, last-inch manipulation, provides an implicit attention mechanism that ignores the unimportant background information in the demonstration. It focuses on the critical actions that define task accomplishment. The long-range collision-aware path planning also ensures safety when applying the framework to new scenes with novel obstacles (Sec. V-F). The choice of *keypose* is insensitive and empirically set as the pose $5cm$ away from the receptacle along the demonstrated trajectory in all our experiments.

V. EXPERIMENTS

A. Experimental Setup

The evaluation is performed exclusively with real-world experiments. The hardware is composed of a Kuka IIWA14 arm, a Robotiq Hand-E gripper, a Photoneo 3D camera providing gray scale and depth images at 10 Hz, as well as a spring-damper device mounted between the gripper and the robot flange that provides passive compliance (see Fig. 2). For better accessibility, the robot is controlled in position mode with the joint position commands computed by manipulation policies. Computations are conducted on a standard desktop with an Intel Core i9-10900X CPU processor and a single NVIDIA RTX 2080 Ti GPU for both training and testing.

Fig. 3. Experimental objects in categories *Gears* and *Batteries*. In each category, the testing object set are labeled with IDs. The rest are the training object instances with known CAD models. Note that the real world training objects are only used for data collection to train baseline methods. The proposed approach learns solely in simulation using their CAD models. The testing objects are selected to be manipulable with the gripper but otherwise diverse in shape and appearance cross instances.

There are 2 object categories considered: *Gears* and *Batteries*. The training and testing splits are depicted in Fig. 3. The real world training objects are used for data collection to train baseline methods while the proposed approach only requires virtual 3D models to learn in simulation. The test set, 7 instances for *Gears* and 9 for *Batteries*, are real industrial or commercial objects purchased from popular retailers. They are chosen to vary in shape and appearance to evaluate the cross-instance generalization of the methods. They are different from the training objects.

Three manipulation tasks (Sec. V-C, V-D, and V-E) are defined with different complexity and tolerance levels. For each task, a single video demonstration of a human manipulating a randomly selected training object is recorded offline. This single demonstration is utilized to perform manipulation of novel objects with no additional data collection. The experiment in each object-task-tolerance setting consists of 5 trials unless otherwise specified, with different initial configurations for both the object and the receptacle. The receptacles' shapes are designed from geometric primitives and their configurations during testing are detected using RANSAC [57]. A total number of 1560 task trials were executed with the proposed method and 7 baselines (Sec. V-B). The following questions are explored: 1) How well are the manipulation skills learned from a single video demonstration? 2) How well do they generalize to novel instances? 3) How robust are the skills to different scenes and uncertainty due to dynamics? 4) What level of contact-rich interaction can the learned policy achieve?

B. Baseline Methods

Comparison points correspond to state-of-the-art, vision-based, category-level manipulation methods. The baselines have been tuned for improved performance. Details are provided in the Appendix.

DON [4], **KPAM** [5], and **KPAM 2.0** [6]: These 3 methods are based on open-source implementations. The original training data collection pipeline is applied with the training objects in this work. The tasks are defined by manually specifying the task-relevant keypoints. The Appendix provides example training data and annotated keypoints.

DON BC, **KPAM BC**, and **KPAM 2.0 BC**: These 3 methods are Behavior Cloning **(BC)**-augmented versions, where the manual goal specification is replaced by visual demonstration. The proposed video parsing and CatBC framework are integrated with the prior methods to capture the intermediate action sequence and realize a closed-loop policy.

Ours-no-tracking: In order to study the effectiveness of visual feedback control for handling uncertainty due to manipulation dynamics, this variation performs open-loop control by disabling the visual motion tracker.

C. Battery Standing Task

Setup: This task requires the robot to grasp the battery from the table-top and place it vertically on a small cylindrical platform, such that the battery stands stably after being released from the gripper. The demonstration video and an example successful robot execution are shown in the first page's figure (top). This task represents commonly considered pick-and-place tasks in prior efforts on visual imitation learning [5], [25]–[27], [44], [58]–[60]. This task is the simplest among the ones considered here but still requires accurate orientation reasoning for stable placement due to the batteries' shape, which is usually long and thin.

Instance	KPAM [5]	KPAM BC [5]	KPAM 2.0 [6]	KPAM 2.0 BC [6]	DON [4]	DON BC [4]	Ours-no tracking	Ours
battery1	3/5	3/5	5/5	5/5	1/5	2/5	5/5	5/5
battery2	5/5	5/5	4/5	4/5	2/5	1/5	5/5	5/5
battery3	1/5	1/5	1/5	2/5	3/5	2/5	5/5	5/5
battery4	1/5	0/5	0/5	0/5	0/5	0/5	2/5	5/5
battery5	5/5	5/5	5/5	5/5	0/5	0/5	5/5	5/5
battery6	1/5	1/5	2/5	2/5	0/5	0/5	3/5	5/5
battery7	3/5	3/5	3/5	5/5	2/5	1/5	5/5	5/5
battery8	4/5	5/5	4/5	3/5	0/5	0/5	2/5	5/5
battery9	0/5	0/5	2/5	1/5	0/5	0/5	4/5	5/5
Total	51.1%	51.1%	57.8%	60.0%	17.8%	13.3%	80.0%	**100.0%**

TABLE I: Results of battery standing task. The testing instances are shown in Fig. 3.

Results: The quantitative comparison is presented in Table I. For **KPAM**, **KPAM 2.0**, and **DON**, common failure cases arise due to the detected keypoints or dense correspondence not being able to provide reliable constraints for goal specification, i.e., the battery standing vertically. The performance gain by behavior cloning **(BC)** is insignificant. The reason is the exact last-inch action sequence is not crucial in this pick-and-place task; namely, the task succeeds as long as collision can be avoided while moving the battery towards the correct goal configuration. **Ours** consistently accomplishes the task regardless of the battery instance. Nevertheless, when motion tracking is disabled, the performance drops to 80.0% as the gripper-battery interaction during the gripper's closing perturbs the battery to a different orientation from the initial estimate, thus leading to inclined and unstable battery placement. When compared to **KPAM BC**, **KPAM 2.0 BC** and **DON BC**, **Ours-no-tracking** yields a higher success rate, indicating that the NUNOCS representation achieves more accurate characterization of the objects' poses and improves cross-instance generalizability.

D. Battery Assembly Task

Setup: This task requires the robot to pick the battery and insert it into a receptacle where a spring is mounted on the internal side of a vertical wall. The receptacles are designed with their lengths and wall heights matching their paired battery's dimension. The spring must be pressed to at least 1/2 of its original length (16 mm) to reserve enough space for the battery. To accomplish the task, the battery has to first press the spring with its negative terminal end to reserve enough space for the positive end to be pressed down. Finally, the gripper releasing the battery allows the spring to stretch, pushing the battery tightly towards the other wall to stay stably inside the receptacle. The experiments evaluate the steady state and mark the cases as failed when the spring is squeezed sideways due to brute force from the gripper. The demonstration video and an example successful robot execution are illustrated in the first page's figure (middle). This task requires robustness against uncertainty due to rich contact and external forces from the environment. It highlights the challenges in learning long-horizon manipulation policies where the last-inch action sequence is critical to task success.

Instance	KPAM [5]	KPAM BC [5]	KPAM 2.0 [6]	KPAM 2.0 BC [6]	DON [4]	DON BC [4]	Ours-no tracking	Ours
battery1	0/5	2/5	0/5	2/5	0/5	0/5	2/5	5/5
battery2	0/5	1/5	0/5	1/5	0/5	2/5	1/5	3/5
battery3	0/5	2/5	0/5	2/5	0/5	2/5	3/5	5/5
battery4	0/5	0/5	0/5	0/5	0/5	0/5	0/5	2/5
battery5	0/5	1/5	0/5	1/5	0/5	0/5	2/5	5/5
battery6	0/5	0/5	0/5	0/5	0/5	0/5	2/5	5/5
battery7	0/5	2/5	0/5	2/5	0/5	1/5	2/5	5/5
battery8	0/5	2/5	0/5	2/5	0/5	1/5	3/5	3/5
battery9	0/5	0/5	0/5	1/5	0/5	0/5	2/5	4/5
Total	0.00%	22.22%	0.00%	24.44%	0.00%	13.33%	37.78%	**82.22%**

TABLE II: Results of battery assembly task. The testing instances are shown in Fig. 3.

Results: The quantitative comparison is presented in Table II. When only specifying the goal configuration and directly transporting the battery to the goal, **KPAM**, **KPAM 2.0**, and **DON** are not able to accomplish the task. The spring is often squeezed by excessive force but not along the principal axis. Therefore, upon gripper releasing, the battery usually pops out of the receptacle pushed by the spring. Augmenting the 3 baselines with **BC** enables to reason over the intermediate sequential actions and improves performance. Nevertheless, complicated by the elastic battery-spring interaction and the battery-receptacle friction, dynamics uncertainty frequently results in significant in-hand object motion, causing the open-loop execution to deviate from the desired target trajectory. This is also reflected by the large performance gap between **Ours-no-tracking** and **Ours**. In **Ours**, the motion tracker constantly provides visual feedback about the latest battery state, ensuring the closed-loop policy can guide the battery following the target category-level trajectory for task success.

E. Gear Insertion Task

Setup: This task requires the robot to insert the gear into a tight-tolerance shaft. To investigate the precision boundary of

the category-level manipulation approaches, experiments are conducted on varying levels of gear-shaft tolerances including $0.1mm$, $0.5mm$, $5mm$ (or $3mm$ if limited by the hole diameter of the gear). The shafts are designed to have similar lengths and varying diameters to realize different tolerances. A task trial is marked as success if the gear's hole passes through the shaft. This task highlights the common challenges in contact-rich manipulation: requiring high precision and robustness against uncertainty. The demonstration video and an example successful robot execution are shown in the first page's figure (bottom).

Instance	Tolerance (mm)	KPAM [5]	KPAM BC [5]	KPAM 2.0 [6]	KPAM 2.0 BC [6]	DON [4]	DON BC [4]	Ours-no tracking	Ours
gear1	0.1	0/5	0/5	0/5	0/5	0/5	0/5	0/5	2/5
	0.5	0/5	0/5	0/5	0/5	0/5	0/5	1/5	5/5
	5	2/5	2/5	2/5	2/5	1/5	2/5	3/5	5/5
gear2	0.1	0/5	0/5	0/5	0/5	0/5	0/5	0/5	2/5
	0.5	0/5	0/5	0/5	0/5	0/5	0/5	0/5	5/5
	5	2/5	2/5	2/5	2/5	0/5	0/5	3/5	5/5
gear3	0.1	0/5	0/5	0/5	0/5	0/5	0/5	0/5	2/5
	0.5	0/5	0/5	0/5	0/5	0/5	0/5	1/5	5/5
	5	2/5	2/5	2/5	2/5	2/5	1/5	3/5	5/5
gear4	0.1	0/5	0/5	0/5	0/5	0/5	0/5	0/5	1/5
	0.5	0/5	0/5	0/5	0/5	0/5	0/5	0/5	4/5
	5	1/5	1/5	1/5	1/5	0/5	0/5	2/5	5/5
gear5	0.1	0/5	0/5	0/5	0/5	0/5	0/5	0/5	1/5
	0.5	0/5	0/5	0/5	0/5	0/5	0/5	0/5	4/5
	3	0/5	0/5	1/5	1/5	0/5	0/5	1/5	5/5
gear6	0.1	0/5	0/5	0/5	0/5	0/5	0/5	0/5	1/5
	0.5	0/5	0/5	0/5	0/5	0/5	0/5	0/5	2/5
	5	0/5	0/5	2/5	2/5	0/5	0/5	2/5	5/5
gear7	0.1	0/5	0/5	0/5	0/5	0/5	0/5	0/5	1/5
	0.5	0/5	0/5	0/5	0/5	0/5	0/5	0/5	4/5
	5	0/5	1/5	1/5	1/5	0/5	0/5	1/5	5/5
Total	0.1	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	**28.6%**
	0.5	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	5.7%	**82.9%**
	5	20.0%	22.9%	31.4%	31.4%	8.6%	8.6%	42.9%	**100.0%**

TABLE III: RESULTS OF GEAR INSERTION TASK. THE TESTING INSTANCES ARE SHOWN IN FIG. 3.

Results: The quantitative comparison is presented in Table III. Even with the most relaxed tolerance ($5mm$), **KPAM**, **KPAM 2.0**, and **DON** struggle as the predicted semantic keypoints and cross-image 2D dense correspondences are not sufficiently reliable to achieve the required precision in the 3D space. Augmenting with **BC** delivers small improvement due to the unreliably estimated gear state. Additionally, *Gears*' being textureless and reflective, poses notable challenges to the training data collection pipeline shared by the above baselines [4]–[6]. In particular, a pre-determined scanning trajectory with a fixed number of view points is not able to cover the potentially novel views with different reflections. In contrast, even without visual feedback, **Ours-no-tracking** yields superior performance, validating the effectiveness of the robust and reliable NUNOCS representation as well as the domain-randomized and bidirectional aligned synthetic training pipeline. Comparing **Ours** against **Ours-no-tracking**, when motion tracking is utilized to provide visual feedback, the performance is dramatically boosted from 42.9% to 100%, indicating the benefit of continuously tracking the object state in high precision contact-rich tasks. With a tighter tolerance $0.5mm$, the performance gap between our approach and baseline methods becomes more significant, demonstrating superior precision and robustness. Finally, **Ours** remains feasible in solving the task with $0.1mm$ tolerance, but the success rate decreases to 28.6%. We expect a further boosted performance by adding force feedback to the control policy [6] or using advanced compliant control methods [3].

F. Framework Analysis

Generalizability to scene configurations: In addition to generalization across a category, it's also interesting to explore whether the skills learned from single visual demonstration generalizes to novel scene configurations not seen in the video. Fig. 4 (a) illustrates the initial gear poses relative to the receptacle in the demonstrated configuration along with the testing cases in the "gear insertion" experiments (Sec. V-E). Fig. 4 (d) provides an example testing case for the battery assembly task where an obstacle impedes the direct transport towards the *keypose*, which is unseen in the demonstration video (middle in first page's figure). In this case, a collision-free path is planned and executed. As observed, the approach generalizes to different scenes, including unstructured environments with novel obstacles. This is attributed to the separation of the long-range motion and last-inch manipulation, leveraging video parsing with object motion tracking (Sec. IV-B). Thus, the method disentangles the object of interest from the background and represents the task-relevant trajectory independent of specific scenes.

Robustness against external disturbance: Fig. 4(e) shows an example of gear insertion, which is successful despite external disturbances due to the human operator, who drags the object away from the gripper, causing a change of the gear's pose. With visual motion tracking constantly feeding the latest object state to the controller (tracking visualizations shown in the bottom-left corners), the desired trajectory is closely followed, thus providing robustness to CatBC against disturbances.

Endowing different manipulation strategies: A key advantage of the proposed approach is the ability to conveniently endow the robot with various manipulation strategies, which is complicated to program otherwise. In the above $0.5mm$ tolerance "gear insertion" task (Sec. V-E), a plain top-down insertion strategy is illustrated in the demonstration video (first page's figure). Then, a more reliable strategy is re-demonstrated using a different action sequence. Specifically, the gear's internal edge is first anchored against the shaft top, and the gear pivots around the anchored point. As shown in Fig. 4 (f), the demonstration encoding this strategy, named "human pivot", notably improves the success rate compared with the simple top-down strategy.

Sensitivity to the demonstration format: The demonstration video parsing formulation (Sec. IV-B) not only disentangles the object of interest from the background, but also from the manipulator. Therefore, the framework smoothly works with other demonstration formats, such as kinesthetic teaching or tele-operation. Fig. 4 (b) includes evaluations in the "gear insertion" task with teleop-collected demonstration using the plain top-down strategy (teleop naive). Compared to human arm demonstration, tele-operation uses the same robotic arm and thus does not suffer from the kinematics or compliance gap. Additionally, the robot arm motion is practically more stable than the human motion, providing a less noisy target trajectory. These lead to slightly improved performance. However,

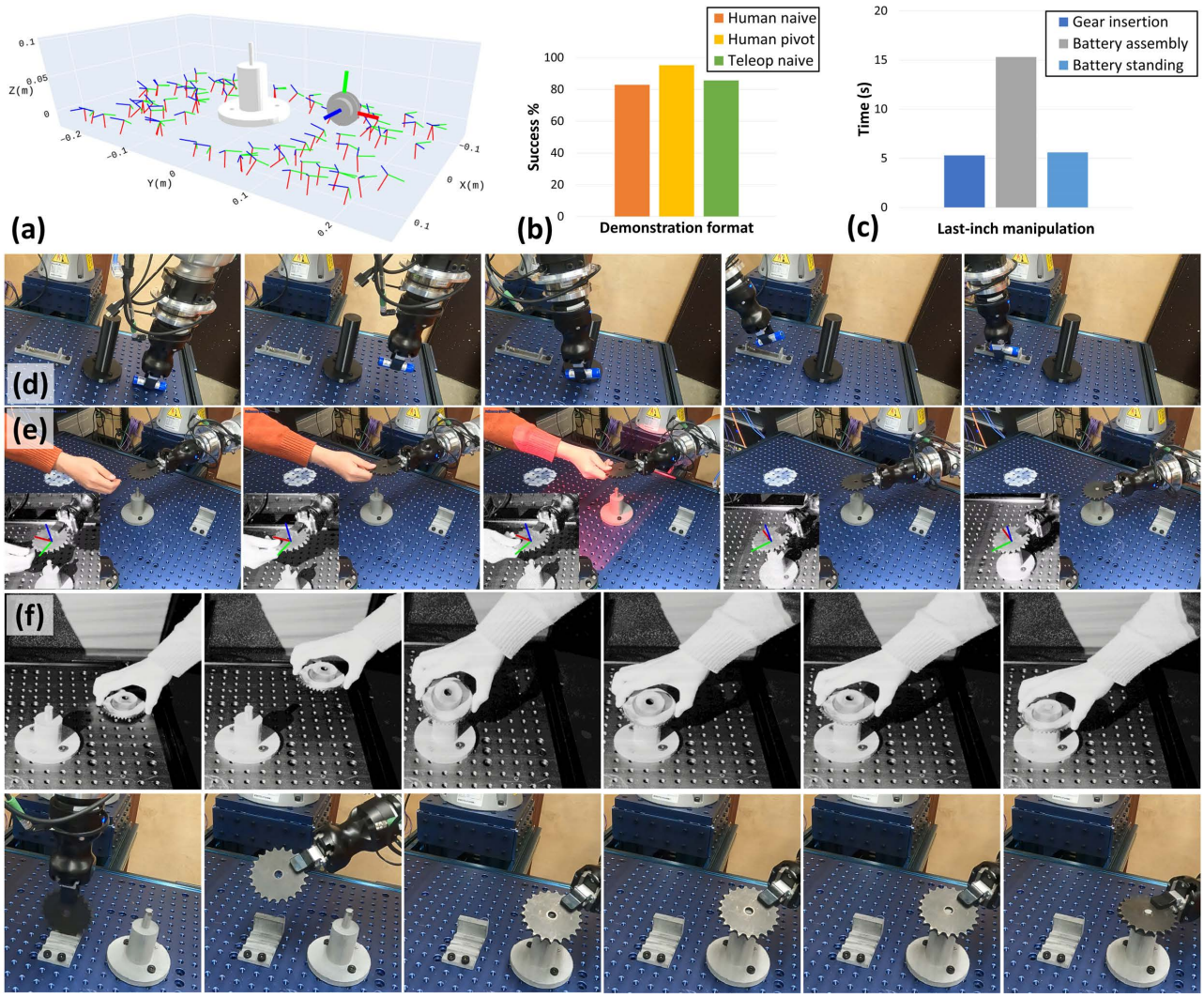


Fig. 4. (a) Distribution of gears' initial poses relative to the receptacle in the "gear insertion" experiments (Sec. V-E). The gray gear mesh represents the demonstration object $\mathcal{O}_{\mathcal{D}}$ in its initial configuration. During testing, the framework generalizes to unseen configurations. (b) Overall success rates of the 3 policies learned from different demonstrations in the "gear insertion" task. The success rates are averaged across object instances and the same number of runs for the $0.5mm$ tolerance as in Table III. The method **Ours** in Table III is based on "human naive". (c) Running time of last-inch manipulation in different tasks. (d) An example testing case of the "battery assembly" task, where the proposed approach generalizes to unstructured environments with obstacles unseen in the demonstration video. (e) Visual motion tracker constantly updates the object pose for robust CatBC against external disturbances, such as human dragging. Pose visualization thumbnails are in bottom-left corners. (f) For the "gear insertion" task, an anchor-and-pivot manipulation strategy is provided instead (first row), and the robot executes the learned policy on a testing object (second row). Complete videos are available in supplementary material.

for tasks involving complex long-horizon sequential actions, tele-operation might become cumbersome. In principal, the approach is not constrained to specific demonstration formats, and one can choose the format most suitable to the task.

Running time: The average running time of the last-inch manipulation is reported in Fig. 4 (c). The 6 DoF motion tracker provides visual feedback to CatBC in real time at the camera's frequency (10 Hz) and runs in a separate thread in parallel, adding nearly no delay other than the communication cost. The running time difference among the tasks is mainly due to the length of each target trajectory. In particular, for the "battery assembly" task, intricate long-horizon sequential actions are required and the CatBC process takes longer. For the long-range collision-aware motion, the running time primarily depends on the distance between the grasping pose and the *keypose*, along with the complexity of the obstacles in the environment, thus omitted in the summary.

VI. DISCUSSIONS AND FUTURE WORK

This work presents a closed-loop category-level manipulation framework that uses visual feedback. The framework can be applied to novel objects given a single visual demonstration. Extensive experiments demonstrate its efficacy in a range of high-precision assembly tasks that require learning complex, long-horizon sequential policies. The approach provides robustness against uncertainty due to manipulation dynamics, and generalization across object instances and scenes. It also allows teaching a robot different manipulation strategies by solely providing a single demonstration, without the need for manual programming.

There are a few limitations that open up future work directions. First, the current framework utilizes vision as the

single sensing modality. Integrating additional sensor modalities, such as force or tactile sensing, can further improve the accuracy of behavioral cloning for high precision manipulation tasks. In addition, only rigid objects are considered here. Many manipulation tasks involve articulated or deformable objects, such as cables, and suitable category-level representations are needed for such object categories.

Acknowledgments

Bowen Wen and Kostas Bekris were partially supported by the US NSF Grant IIS-1734492. The opinions expressed here are of the authors and do not reflect the views of the sponsor.

References

[1] W. Lian, T. Kelch, D. Holz, A. Norton, and S. Schaal, "Benchmarking Off-The-Shelf Solutions to Robotic Assembly Tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021*, pp. 1046–1053.

[2] M. Kyrarini, M. A. Haseeb, D. Ristić-Durrant, and A. Gräser, "Robot learning of industrial assembly task via human demonstrations," *Autonomous Robots*, vol. 43, no. 1, pp. 239–257, 2019.

[3] A. S. Morgan, B. Wen, J. Liang, A. Boularias, A. M. Dollar, and K. Bekris, "Vision-driven Compliant Manipulation for Reliable, High-Precision Assembly Tasks," *RSS*, 2021.

[4] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," *CoRL*, 2018.

[5] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "KPAM: Keypoint affordances for category-level robotic manipulation," *ISRR*, 2019.

[6] W. Gao and R. Tedrake, "kPAM 2.0: Feedback Control for Category-Level Robotic Manipulation," *IEEE Robotics and Automation Letter (RA-L)*, 2020.

[7] S. Yang, W. Zhang, R. Song, J. Cheng, and Y. Li, "Learning Multi-Object Dense Descriptor for Autonomous Goal-Conditioned Grasping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4109–4116, 2021.

[8] C.-Y. Chai, K.-F. Hsu, and S.-L. Tsao, "Multi-step pick-and-place tasks using object-centric dense correspondences," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4004–4011.

[9] Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei, and S. Savarese, "KETO: Learning keypoint representations for tool manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7278–7285.

[10] M. Vecerik, J.-B. Regli, O. Sushkov, D. Barker, R. Pevceviciute, T. Rothörl, C. Schuster, R. Hadsell, L. Agapito, and J. Scholz, "S3K: Self-Supervised Semantic Keypoints for Robotic Manipulation via Multi-View Consistency," in *Conference on Robot Learning (CoRL)*, 2020.

[11] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *Foundations and Trends in Robotics*, 2018.

[12] D. B. Grimes, R. Chalodhorn, and R. P. Rao, "Dynamic Imitation in a Humanoid Robot through Nonparametric Probabilistic Inference." in *Robotics: science and systems*. Cambridge, MA, 2006, pp. 199–206.

[13] P. Englert, A. Paraschos, M. P. Deisenroth, and J. Peters, "Probabilistic model-based imitation learning," *Adaptive Behavior*, vol. 21, no. 5, pp. 388–403, 2013.

[14] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2146–2153.

[15] Y. Duan, M. Andrychowicz, B. C. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-shot imitation learning," *NIPS*, 2017.

[16] "Third-person imitation learning, author=Stadie, Bradly C and Abbeel, Pieter and Sutskever, Ilya," *ICLR*, 2017.

[17] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *Conference on Robot Learning*. PMLR, 2017, pp. 357–368.

[18] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Robotics research. the eleventh international symposium*. Springer, 2005, pp. 561–572.

[19] S. Calinon and A. Billard, "Statistical learning by imitation of competing constraints in joint space and task space," *Advanced Robotics*, vol. 23, no. 15, pp. 2059–2076, 2009.

[20] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration," in *Robotics Research*. Springer, 2016, pp. 339–354.

[21] P. Florence, L. Manuelli, and R. Tedrake, "Self-supervised correspondence in visuomotor policy learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 492–499, 2019.

[22] S. Paradis, M. Hwang, B. Thananjeyan, J. Ichnowski, D. Seita, D. Fer, T. Low, J. E. Gonzalez, and K. Goldberg, "Intermittent visual servoing: Efficiently learning policies robust to instrument changes for high-precision surgical manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7166–7173.

[23] E. Johns, "Coarse-to-Fine Imitation Learning: Robot Manipulation from a Single Demonstration," *ICRA*, 2021.

[24] B. Wu, F. Xu, Z. He, A. Gupta, and P. K. Allen, "SQUIRL: Robust and Efficient Learning from Video Demonstration of Long-Horizon Robotic Manipulation Tasks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9720–9727.

[25] D.-A. Huang, S. Nair, D. Xu, Y. Zhu, A. Garg, L. Fei-Fei, S. Savarese, and J. C. Niebles, "Neural task graphs: Generalizing to unseen tasks from a single video demonstration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8565–8574.

[26] J. Liang, B. Wen, K. Bekris, and A. Boularias, "Learning Sensorimotor Primitives of Sequential Manipulation Tasks from Visual Demonstrations," *ICRA*, 2022.

[27] M. Sieb, Z. Xian, A. Huang, O. Kroemer, and K. Fragkiadaki, "Graph-structured visual imitation," in *Conference on Robot Learning*. PMLR, 2020, pp. 979–989.

[28] D. Kappler, F. Meier, J. Issac, J. Mainprice, C. G. Cifuentes, M. Wüthrich, V. Berenz, S. Schaal, N. Ratliff, and J. Bohg, "Real-time perception meets reactive motion generation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1864–1871, 2018.

[29] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[30] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[31] U. Viereck, A. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," in *Conference on Robot Learning*. PMLR, 2017, pp. 291–300.

[32] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[33] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[34] L. Manuelli, Y. Li, P. Florence, and R. Tedrake, "Keypoints into the Future: Self-Supervised Correspondence in Model-Based Reinforcement Learning," *CoRL*, 2020.

[35] A. Byravan, F. Leeb, F. Meier, and D. Fox, "Se3-pose-nets: Structured deep dynamics models for visuomotor control," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3339–3346.

[36] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Computer Architecture Letters*, vol. 13, no. 04, pp. 376–380, 1991.

[37] B. Wen and K. E. Bekris, "BundleTrack: 6D Pose Tracking for Novel Objects without Instance or Category-Level 3D Models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[38] 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.

[39] B. Wen, W. Lian, K. Bekris, and S. Schaal, "CaTGrasp: Learning

Category-Level Task-Relevant Grasping in Clutter from Simulation," *arXiv preprint arXiv:2109.09163*, 2021.
[40] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.
[41] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.
[42] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IROS 2017*.
[43] 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," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 10 367–10 373.
[44] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, "Learning to generalize across long-horizon tasks from human demonstrations," *RSS*, 2020.
[45] 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*, 2021.
[46] 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," 2020.
[47] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
[48] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
[49] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
[50] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 108–11 117.
[51] C. Mitash, B. Wen, K. Bekris, and A. Boularias, "Scene-level pose estimation for multiple instances of densely packed objects," in *Conference on Robot Learning*. PMLR, 2020, pp. 1133–1145.
[52] M. Gualtieri and R. Platt, "Robotic Pick-and-Place With Uncertain Object Instance Segmentation and Shape Completion," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1753–1760, 2021.
[53] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 2, pp. 158–175, 1995.
[54] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *RSS*, 2017.
[55] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.
[56] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
[57] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[58] J. Jin, L. Petrich, Z. Zhang, M. Dehghan, and M. Jagersand, "Visual geometric skill inference by watching human demonstration," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8985–8991.
[59] H. Xiong, Q. Li, Y.-C. Chen, H. Bharadhwaj, S. Sinha, and A. Garg, "Learning by Watching: Physical Imitation of Manipulation Skills from Human Videos," *IROS*, 2021.
[60] J. Tremblay, T. To, A. Molchanov, S. Tyree, J. Kautz, and S. Birchfield, "Synthetically trained neural networks for learning human-readable plans from real-world demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5659–5666.
[61] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.
[62] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural Descriptor Fields: SE (3)-Equivariant Object Representations for Manipulation," *arXiv preprint arXiv:2112.05124*, 2021.

APPENDIX

A. Implementation details

The NUNOCS learning process is solely conducted over synthetic data. Example training data is exhibited in Fig. 6 and Fig. 7. The data generation pipeline is developed with Blender [41]. For each data point generation, an object is randomly selected from the training set and dropped onto a table surface. After the object stabilizes, the partially scanned point cloud is computed from the rendered 2D depth image. Additionally, the object's ground-truth pose is retrieved along with its CAD model to compute the training labels $\overline{\mathcal{P}}_{\mathbb{C}}$ and $\bar{s}$. The object type, initial pose relative to the camera, table's height, gravity, friction and bouncing parameters are all randomized in each scene. Additionally, random non-uniform 3D scaling is applied to the object model to attain a novel object with different dimensions, which allows to create greatly enriched kinds of object instances beyond the ones provided in the training set. Each data point includes the scanned depth image, instance segmentation mask and the ground-truth labels of NUNOCS. The NUNOCS Net is implemented in Pytorch [61] and is trained with Adam optimizer for 100 epochs with a batch size of 50. Learning rate starts from 0.01 and is scaled by 0.1 at epochs 50 and 80. Depth-missing corruption is applied to the 2D depth image at a missing percentage between 0 to 0.4 before being converted into the point cloud. Additional data augmentations include random translation and rotation applied to the point cloud.

For the trajectory extracted from the single demonstration video, timestamps are discretized such that the interval distance between any neighboring sub-goals are at least $2mm$ or $2°$. This discretizes the continuous trajectory while ensures to timely correct from deviations when performing the CatBC (Sec. IV-C). One thing noteworthy is symmetry handling when following the target trajectory during online CatBC. For both the categories of *Gears* and *Batteries*, the equivalent symmetric transformations are the rotations around the Z axis from 0 to 360° discretized by 5°. In such cases, for each of the intermediate goals, each equivalent transformed pose, sorted by their distances to the current object pose in ascending order, is sequentially checked for a collision-free IK solution. The first feasible equivalent pose is then selected as the next sub-goal pose.

B. Limitations and failure modes

Example failure modes are presented in Fig. 5. In the "battery assembly" task (top), as the battery squeezes the spring, the elastic force gradually increases and eventually pushes the battery out of the gripper. In this case, it would be beneficial to add tactile sensing together with the visual tracker in the feedback loop to predict slippery early. In the "gear insertion" task (bottom), when using "anchor-and-pivot" strategy, rich contact leads to significant change of the gear's in hand orientation, causing no collision-free IK solutions found to continue. In this scenario, the object is still accurately tracked, so regrasping or adjusting the object pose via in-hand manipulation with a dexterous hand can recover from this failure mode.

C. Baseline methods

DON - This is based on the open-sourced implementation[2] of [4]. The same training data collection pipeline in [4] is adopted, where dense correspondence across object instances is learned through contrastive learning, without requiring annotated labels. Some example training data collected on our task objects are displayed in Appendix. We closely follow the semantic grasping learning workflow in [4] and specify semantic keypoints that are required to define the manipulation task target. For fair comparison, the semantic keypoints are specified on images corresponding to the same $\mathcal{O}_{\mathcal{D}}$ in our used demonstration video. The keypoints are then transferred to novel object instances during testing based on the learned cross-image dense correspondence.

DON BC - This is a modified version of **DON** [4] by replacing the manual goal specification with the single visual demonstration, similar to ours. The same video parsing and CatBC formulation proposed in our approach are integrated, except that the cross-image dense correspondence learned from real world data is inherited from **DON** instead of our category-level representation for the CacBC process. When directly using the predicted dense correspondence for parsing the in-hand object motion in the demonstration video, where significant occlusions involve, the trajectory quality is poor. Therefore, the same extracted trajectory by our approach is utilized. During testing, the predicted dense correspondence initializes the object's state and object motion is thereafter tracked for CatBC by treating them as additional virtual links in the kinematic tree, same as [5].

KPAM - This is based on the open-sourced implementation[3] of [5]. The training data collection is similar to that of **DON** [4]. Following the original work [5], human demonstration is provided via manual goal specification during the testing stage for trajectory optimization.

KPAM BC - This is a modified version of **KPAM** [5] augmented with behavior cloning by replacing the manual goal specification with the single visual demonstration, similar to ours. The same video parsing and CatBC formulation proposed in our approach are integrated similar to the treatment to **DON BC**, except that during testing the cross-instance correspondence is established using the **KPAM** predicted semantic keypoints for the CatBC process. Similar to the reason in **DON BC**, the same extracted trajectory by our approach is utilized. Strictly following **KPAM**, the keypoints are tracked by treating them as additional virtual links in the kinematic tree [5].

KPAM 2.0 - This is based on the related work [6] that extends **KPAM** [5] by augmenting semantic keypoints with orientation information for improved expressiveness. The force sensing is disabled to make vision as the primary sensing

[2]https://github.com/RobotLocomotion/pytorch-dense-correspondence
[3]https://github.com/weigao95/kPAM

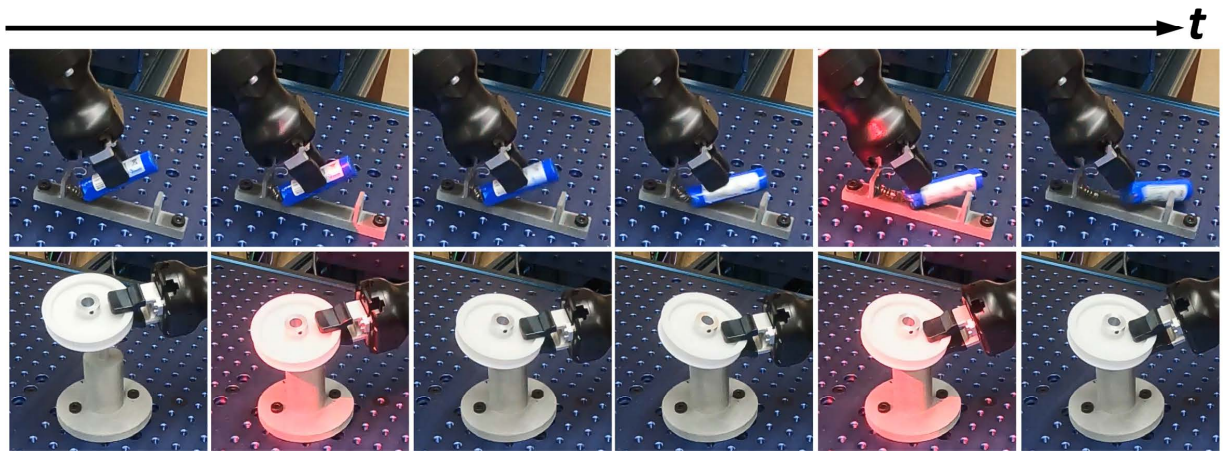


Fig. 5. Example failure modes. **Top:** In the "battery assembly" task, as the battery squeezes the spring, the elastic force gradually increases and eventually pushes the battery out of the gripper. In this case, it would be beneficial to add tactile sensing together with the visual tracker in the feedback loop to predict slippery early. **Bottom:** In the "gear insertion" task, when using "anchor-and-pivot" strategy, rich contact leads to significant change of the gear's in hand orientation, causing no collision-free IK solutions found to continue. In this scenario, the object is still accurately tracked, so regrasping or adjusting the object pose via in-hand manipulation with a dexterous hand can recover from this failure mode.

modality, sharing the same setup as other evaluated approaches.

KPAM 2.0 BC - This is a modified version of **KPAM 2.0** [6] by replacing the manual goal specification with the single visual demonstration, similar to ours. The same video parsing and CatBC formulation proposed in our approach are integrated similar to the treatment to **DON BC**, except that during testing the cross-instance correspondence is established from **KPAM 2.0**'s predicted semantic keypoints for the CatBC process. Similar to the reason in **DON BC**, the same extracted trajectory by our approach is utilized. The keypoints are tracked kinematically in the same fashion as in **KPAM BC**.

Besides the above category-level manipulation approaches, another concurrent work [62] requires 4 RGBD cameras mounted at each corner on a tabletop, which allows to fuse the point cloud to reconstruct the 3D shape of the novel objects. This is different from the single camera setting considered in all the evaluated approaches and thus not included for comparison.

D. Training data collection for baseline methods

Data collection process described in [4]–[6] was closely followed. For **KPAM** [5], some example training data with keypoint annotations are shown in Fig. 8 and 9. For **KPAM 2.0** [6], some example training data with keypoint annotations augmented with orientations are shown in Fig. 10 and 11. For **DON** [4], some example training data with sampled inlier correspondences obtained by re-projecting points based on their relative transformations is shown in Fig. 12 and 13.

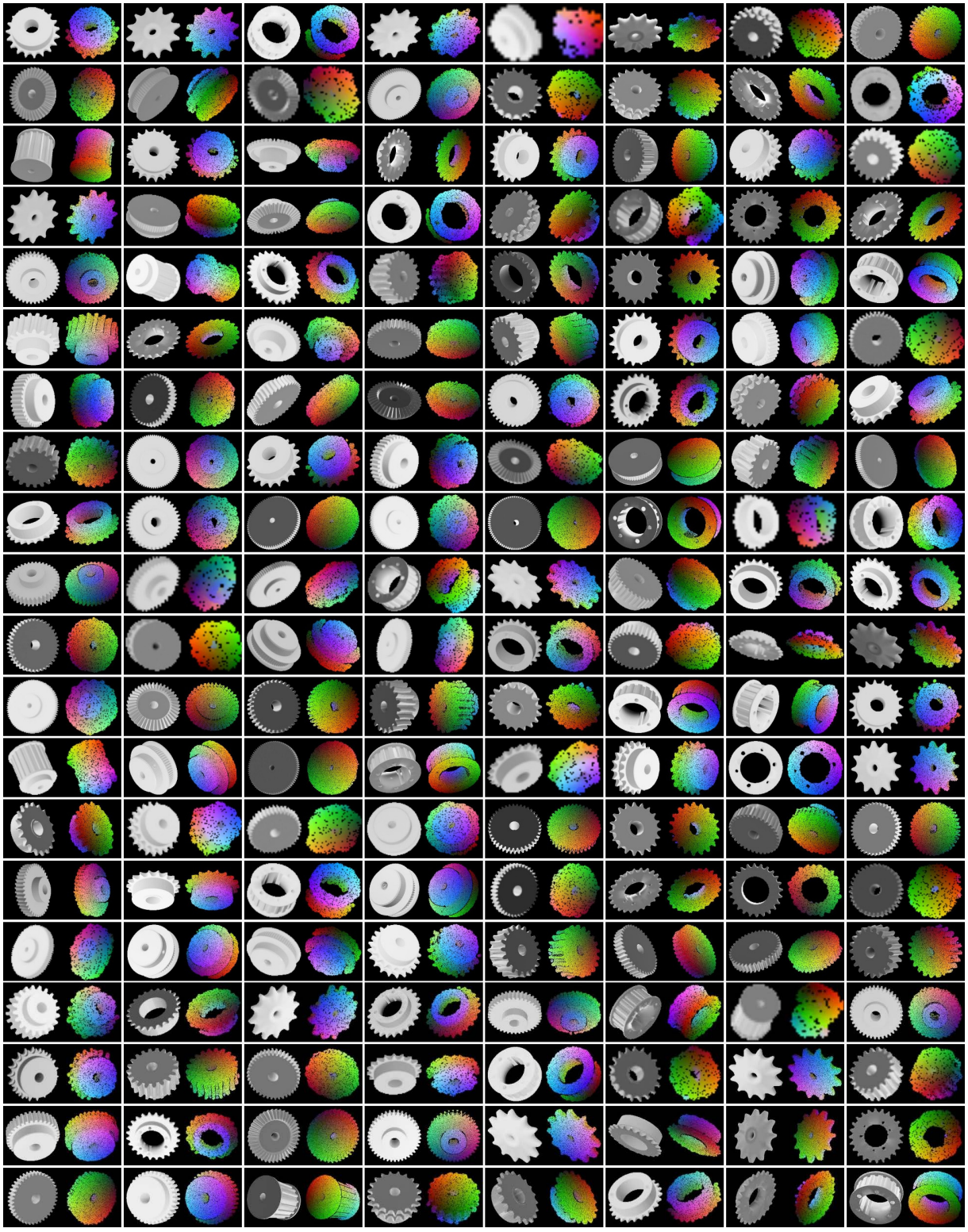


Fig. 6. Example synthetic training data in *Gears* category to train the NUNOCS Net in our approach, where each pair consists of a color image and the ground-truth NUNOCS label corresponding to the corrupted noisy depth image. In total, 100000 data points have been generated in simulation, where each data point includes color and depth images, an instance segmentation mask, and the ground-truth NUNOCS label.

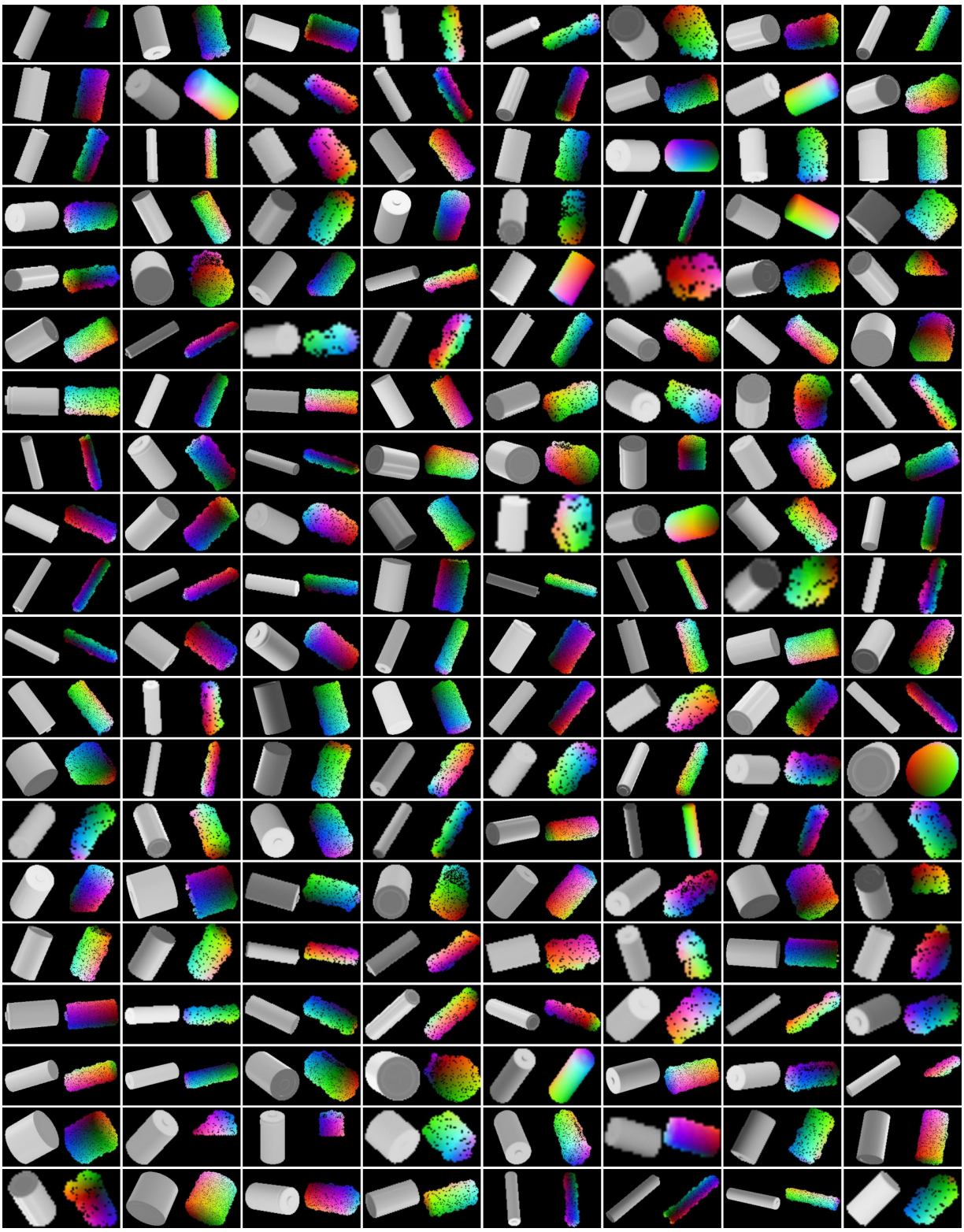


Fig. 7. Example synthetic training data in *Battery* category to train the NUNOCS Net in our approach, where each pair consists of a color image and the ground-truth NUNOCS label corresponding to the corrupted noisy depth image. In total, 100000 data points have been generated in simulation, where each data point includes color and depth images, an instance segmentation mask, and the ground-truth NUNOCS label.

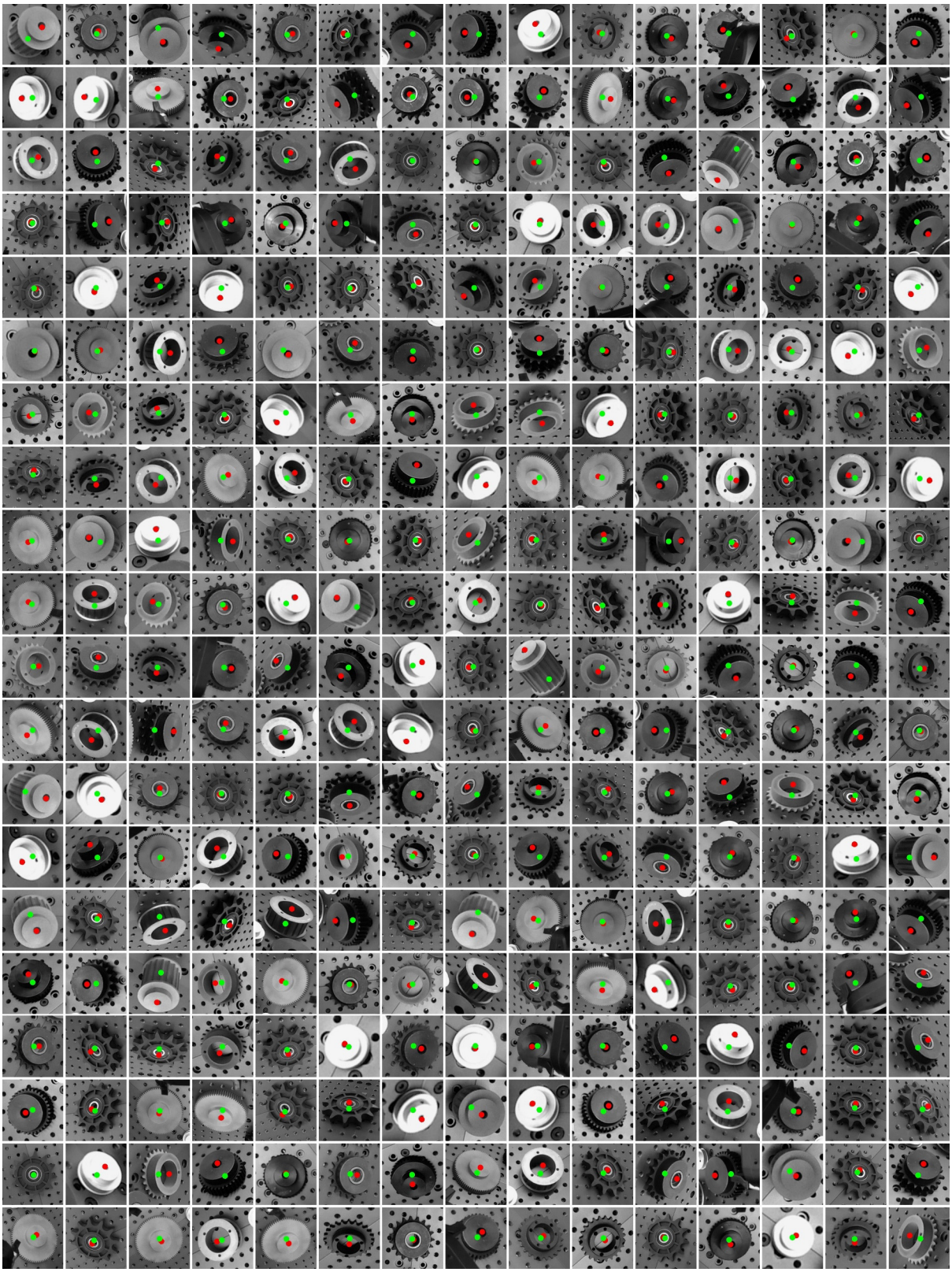


Fig. 8. Example real-world training data in *Gears* category collected in this work to train KPAM [5] for comparison. In total, 98340 data points have been generated, where each data point includes color and depth images, an object bounding box, an instance segmentation mask, and ground-truth semantic keypoints annotated in red and green.

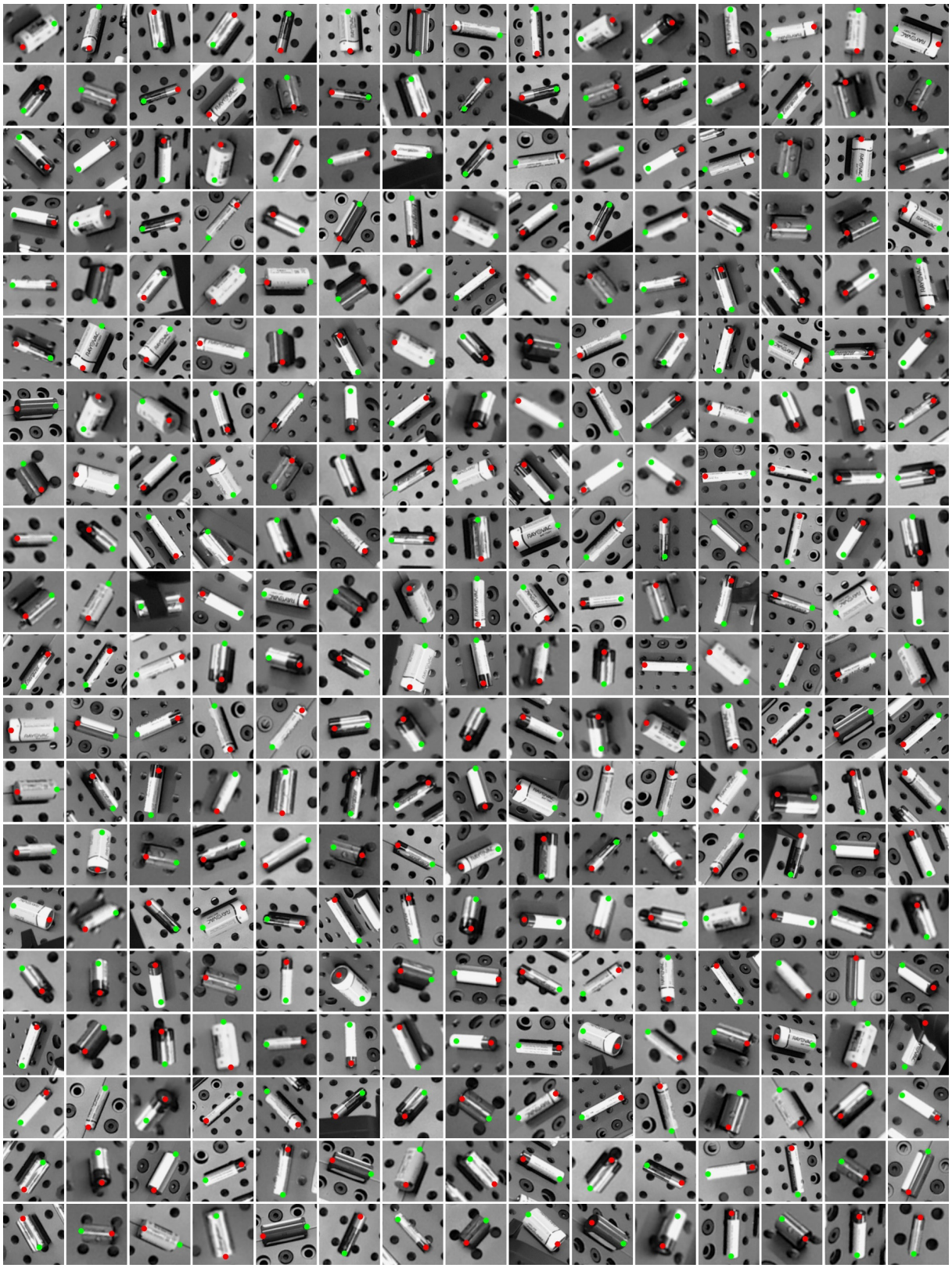


Fig. 9. Example real-world training data in *Batteries* category collected in this work to train KPAM [5] for comparison. In total, 90000 data points have been generated, where each data point includes color and depth images, an object bounding box, an instance segmentation mask, and ground-truth semantic keypoints annotated in red and green.

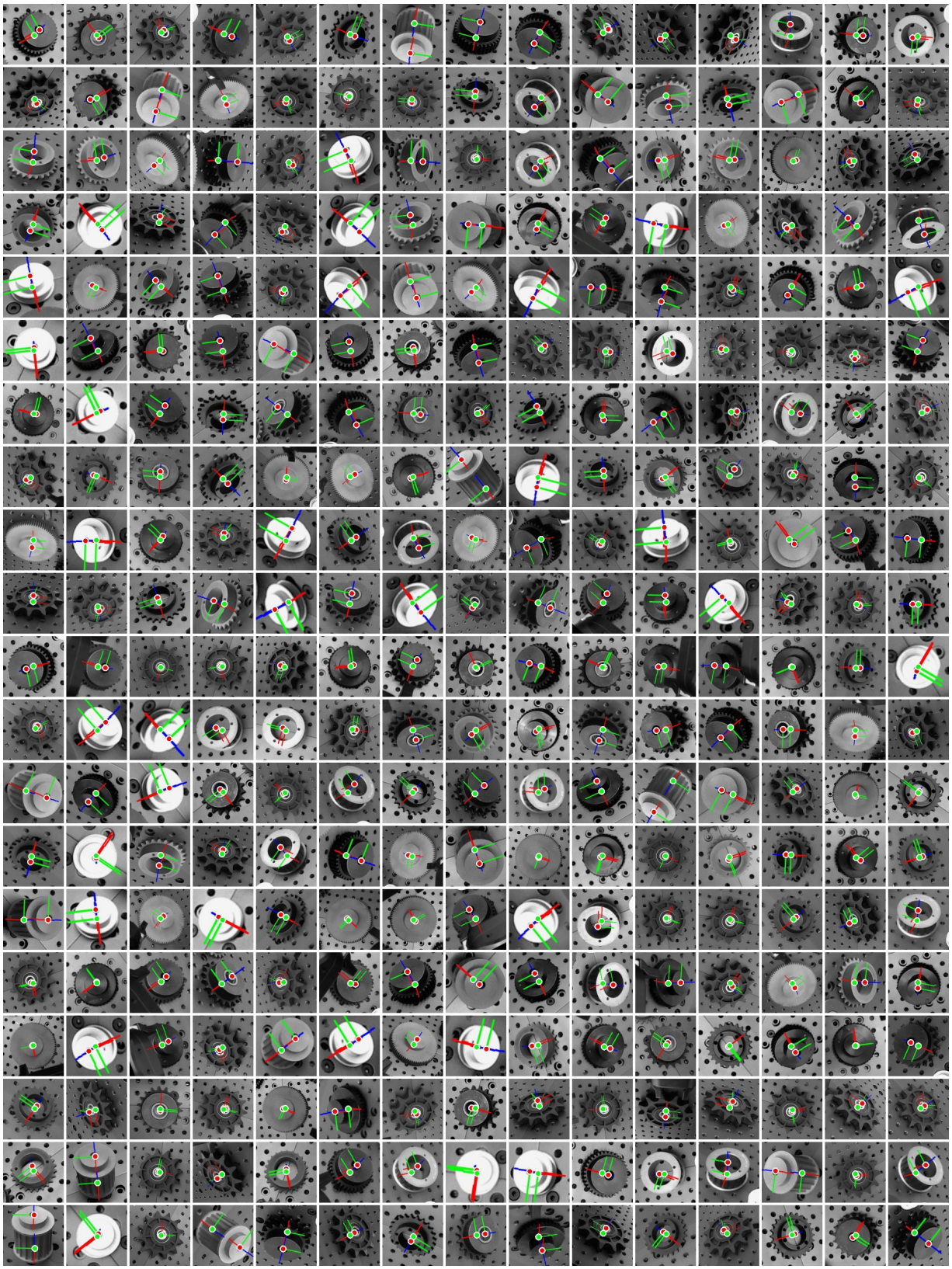


Fig. 10. Example real-world training data in *Gears* category collected in this work to train KPAM 2.0 [6] for comparison. In total, 90000 data points have been generated, where each data point includes color and depth images, an object bounding box, an instance segmentation mask, ground-truth semantic keypoints annotated in red and green, and their augmented orientations.

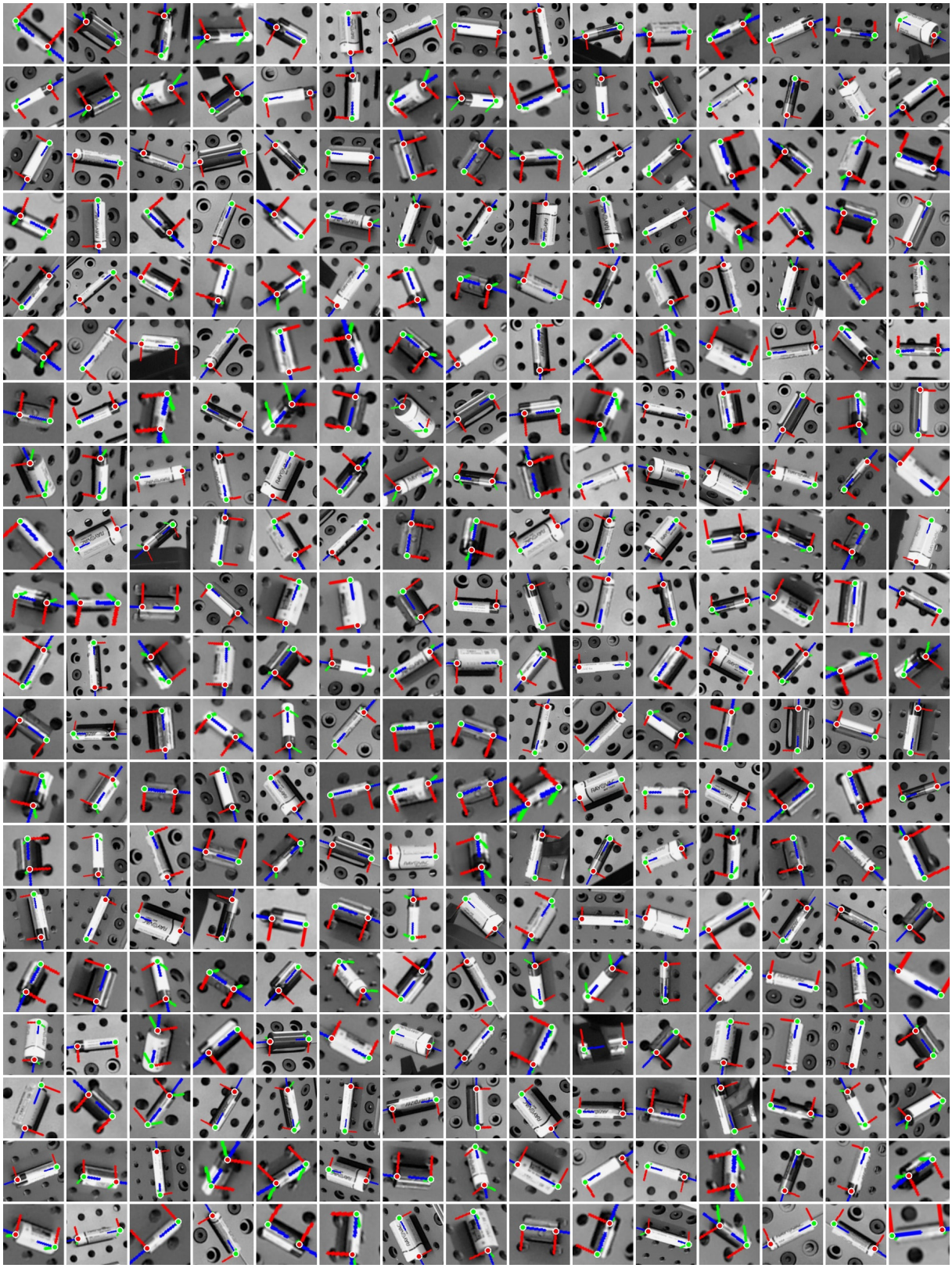


Fig. 11. Example real-world training data in *Batteries* category collected in this work to train KPAM 2.0 [6] for comparison. In total, 90000 data points have been generated, where each data point includes color and depth images, an object bounding box, an instance segmentation mask, ground-truth semantic keypoints annotated in red and green, and their augmented orientations.

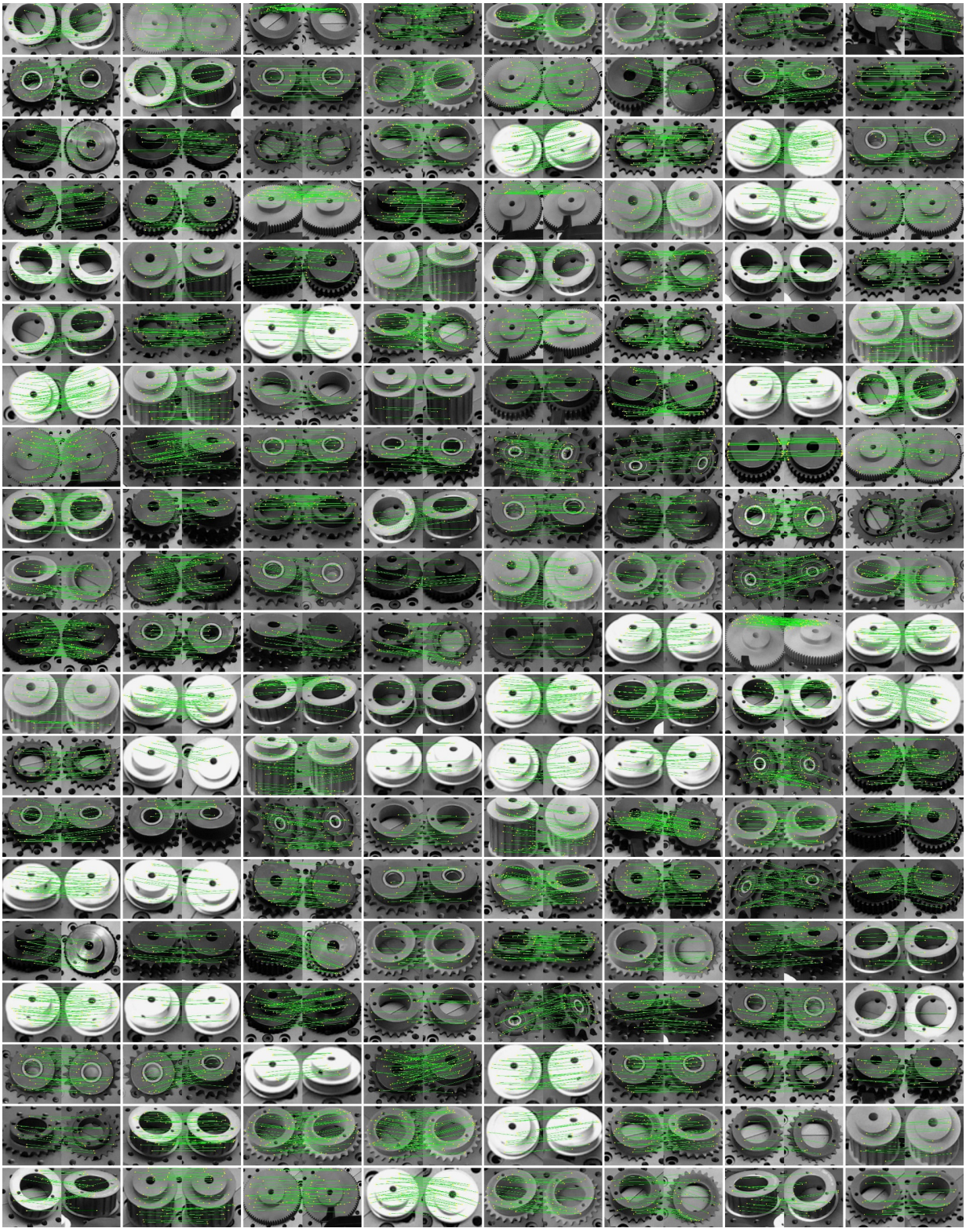


Fig. 12. Example real-world training data illustrating sampled inlier correspondences in *Gears* category collected in this work to train the comparison approach DON [4]. During training, DON randomly samples inlier and outlier correspondences for contrastive learning.

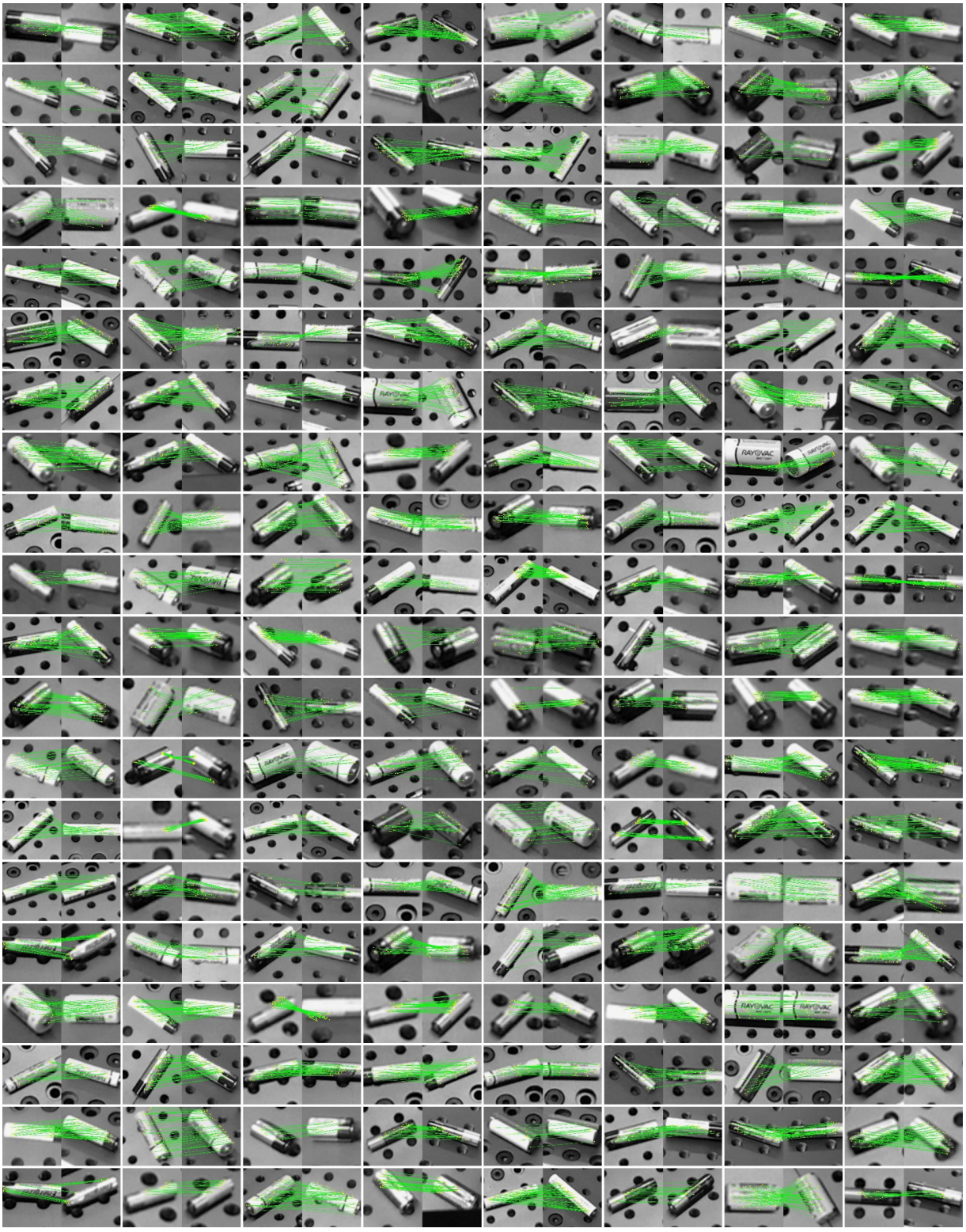


Fig. 13. Example real-world training data illustrating sampled inlier correspondences in *Batteries* category collected in this work to train the comparison approach DON [4]. During training, DON randomly samples inlier and outlier correspondences for contrastive learning.