arXiv:2209.13864v2 [cs.RO] 17 Feb 2023

# USEEK: Unsupervised SE(3)-Equivariant 3D Keypoints for Generalizable Manipulation

Zhengrong Xue[1,3], Zhecheng Yuan[2,1], Jiashun Wang[4], Xueqian Wang[2], Yang Gao[2,5,1], Huazhe Xu[2,5,1]

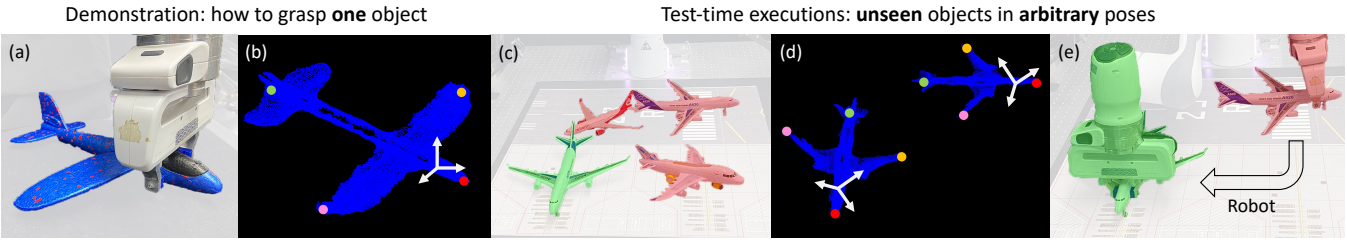


**Fig. 1:** (a) Given an object point cloud and the mere demonstration of a functional grasping pose, (b) USEEK infers a set of keypoints and the task-relevant local coordinate frame. (c) When tested, unseen objects within the category in initial poses unobserved at the training time are shown to the robot. The robot is required to pick and then place it to an arbitrary target pose. (d) The properties of intra-category alignment and SE(3)-equivariance make USEEK generalizable to novel shapes and poses. (e) With the help of keypoints and local coordinate frames, the robot manages to transfer the functional knowledge and execute the manipulation tasks.

***Abstract*— Can a robot manipulate intra-category unseen objects in arbitrary poses with the help of a mere demonstration of grasping pose on a single object instance? In this paper, we try to address this intriguing challenge by using USEEK, an unsupervised SE(3)-equivariant keypoints method that enjoys alignment across instances in a category, to perform generalizable manipulation. USEEK follows a teacher-student structure to decouple the unsupervised keypoint discovery and SE(3)-equivariant keypoint detection. With USEEK in hand, the robot can infer the category-level task-relevant object frames in an efficient and explainable manner, enabling manipulation of any intra-category objects from and to any poses. Through extensive experiments, we demonstrate that the keypoints produced by USEEK possess rich semantics, thus successfully transferring the functional knowledge from the demonstration object to the novel ones. Compared with other object representations for manipulation, USEEK is more adaptive in the face of large intra-category shape variance, more robust with limited demonstrations, and more efficient at inference time. Project website: `https://sites.google.com/view/useek/`.**

## I. Introduction

When three-year old children think of an object, they recognize it not only as the object itself but also as a symbol for the category [8]. The innate talents of humans to generalize, according to developmental psychology, are known as symbolic functioning [21]. In the context of robotics, the same desire to generalize begs the research question: does there exist a control method that achieves generalizable manipulation across object poses and instances?

With the access to the full 3D geometry of the object, the pipeline for robotic manipulation has long been mature — template matching [10], [1], [26] for perception while trajectory optimization [2], [25] and inverse kinematics [5] for execution. However, these manipulation skills often suffer from intra-category shape variance as common hand-crafted techniques for template matching may fail to generalize.

To enable intra-category any-pose manipulation, an object representation that achieves category-level generalization is crucial. Existing representations can be roughly classified into three kinds: 6-DOF pose estimators [39], [38], [37], [40], [18], 3D keypoints [33], [29], [19], [20], [4], and dense correspondence models [28], [12], [32], [31]. Despite the disparities in form, their ultimate goals are consistent — to determine the local coordinate frame of the object. Thus, we tend to view those representations as different abstraction levels of an object. Among them, 6-DOF pose estimators provide the highest level of abstraction by predicting the object frame directly. However, they are often regarded as not generalizable enough for large shape variance in robotic manipulation tasks [19], [31]. Recently, dense correspondence models implicitly define an object by approximating a continuous function that maps either 2D pixels or 3D points to spatial descriptors. While these spatial descriptors preserve abundant geometric details, the object frames directly guiding manipulation cannot be acquired gratis from the dense correspondence representations.

Compared with the aforementioned representations, 3D keypoints enjoy the benefits of both practicality and simplicity: its semantic correspondences are more informative than 6-DOF poses; its succinct expression is more efficient than dense correspondence models. Despite the advantages of keypoints, for the task of intra-category any-pose robotic manipulation, we may further require the keypoints to possess the following properties:

- (i) *Anti-occlusion.* The keypoints should be repeatable in the face of self-occlusion. Thus, we prefer raw 3D inputs (i.e., point clouds) to multi-view images.

[1]Shanghai Qi Zhi Institute. [2]Tsinghua University. [3]Shanghai Jiao Tong University. [4]Carnegie Mellon University. [5]Shanghai AI Lab.
Contact: `xuezhengrong@sjtu.edu.cn`, `huazhe_xu@mail.tsinghua.edu.cn`.

- (ii) *Unsupervised.* The keypoints should be obtained in an unsupervised or self-supervised manner to avoid the costs and biases from human annotations.
- (iii) *Aligned across instances.* The semantic correspondence of keypoints across instances within a certain category is essential for category-level generalizable manipulation.
- (iv) *SE(3)-equivariant.* The keypoints are further desired to be equivariant w.r.t. the translations and rotations of the objects in the 3D space because the objects in the wild can appear in any poses.

In this paper, we propose a framework that utilizes 3D keypoints for intra-category any-pose robotic manipulation. At the heart of this framework are the discovered 3D keypoints that boast all of the four desired properties. Specifically, we propose a novel teacher-student architecture for unsupervised SE(3)-equivariant keypoint (USEEK) discovery. We then first evaluate USEEK against state-of-the-art keypoint discovery baselines through visual metrics. Next, we leverage USEEK to enable a robot to pick up an intra-category object from a randomly initialized pose and then place it in a specified pose via one-shot imitation learning. Despite the difficulty, rich semantics of the keypoints given by USEEK enables the robot to execute pick-and-place by transferring the functional knowledge from limited demonstration to unseen instances in any poses. Quantitative and qualitative results in the simulator as well as on the real robot indicate that USEEK is competent to serve as an object representation for generalizable manipulation tasks.

## II. Related Work

### A. Object Representations for Manipulation

**Explicit 6-DOF pose estimation.** Pose estimation techniques start from the early works such as RANSAC [10] or Iterative Closest Point (ICP) [26]. Though very efficient, these works usually struggle with the shape variance of unknown objects. Learning-based 6-DOF pose estimators [27], [39], [37], [18] manage to represent an object on a category level. But when applied to robotic manipulation, they are often viewed as either ambiguous under large intra-category shape variance [19], or incapable to provide enough geometric information for control [31].

**Dense correspondence.** In contrast to the explicit pose prediction, dense correspondence methods [28], [12], [11], [32], [31] define an object in a continuous and implicit way. One example is the recently proposed Neural Descriptor Fields (NDF) [31], which encodes the spatial relations of external rigid bodies and the demonstrated object. Effective as it is for few-shot imitation learning, NDF is inefficient because it has to regress the descriptor fields of hundreds of query points via iterative optimization [14].

**3D keypoints.** The use of 3D keypoints for control is extensively studied in computer vision [33], [17], [41], [29], robotics [20], [19], [13], and reinforcement learning [36], [4]. However, we find that none of the existing methods shown in Table I meets all the requirements we have listed that are beneficial to the task of generalizable robotic manipulation.

| Property | (i) | (ii) | (iii) | (iv) |
|---|---|---|---|---|
| KeypointNet [33] | ✗ | ✓ | ✓ | ✗ |
| USIP [17] | ✓ | ✓ | ✗ | ✗ |
| UKPGAN [41] | ✓ | ✓ | ✗ | ✓ |
| Skeleton Merger [29] | ✓ | ✓ | ✓ | ✗ |
| kPAM [19] and its variants | ✓ | ✗ | ✓ | ✗ |
| Keypoints into the Future [20] | ✗ | ✓ | ✓ | ✗ |
| S3K [36] | ✓ | ✓ | ✓ | ✗ |
| Keypoint3D [4] | ✓ | ✓ | ✗ | ✗ |

**TABLE I:** We compare the features of recently proposed 3D keypoint detectors in the field of computer vision, robotics, and reinforcement learning.

### B. SE(3)-Invariant/Equivariant Neural Networks

The concepts of SE(3)-invariant and SE(3)-equivariant are sometimes intertwined. For the function of keypoint detection, we require it to be *SE(3)-invariant* if the function selects the indices of points in the point cloud; otherwise, we require it to be *SE(3)-equivariant* if the function returns the coordinates of the keypoints. In this paper, we take the second interpretation, requiring the 3D keypoints to be *SE(3)-equivariant*. Noting that we can easily handle translations by normalizing the center of mass of the point cloud to a specified original point, the challenges are mainly entangled with rotations, i.e., SO(3)-invariance/equivariance.

PRIN/SPRIN [43], [44] and Vector Neurons [9] are recently proposed SO(3)-invariant networks that directly takes point clouds as inputs. PRIN extracts rotation invariant features by absorbing the advantages of both Spherical CNN [6] and PointNet [23]-like networks. SPRIN improves PRIN in sparsity and achieves state-of-the-art performance. Concurrently, Vector Neurons enjoy SO(3)-equivariance by extending neurons from 1D scalars to 3D vectors and providing the corresponding SO(3)-equivariant neural operations.

## III. Methods

We present a framework that utilizes the unsupervised, SE(3)-equivariant keypoints (USEEK) for intra-category any-pose object manipulation. The keypoints from USEEK are first used to specify the task-relevant local coordinate frame of a demonstration object. Then, they generalize to the corresponding points from objects of unseen shapes within the same category and in unobserved poses at training time. Leveraging such keypoints, we manage to transfer the task-relevant frames, and finally perform motion planning algorithms to manipulate the objects.

### A. Preliminaries on Keypoints

We first define a keypoint detector as $f(\cdot)$ that maps an object point cloud $\mathbf{P}$ to an ordered set of keypoints $\mathbf{p}$:

$$f(\mathbf{P}) : \mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^{K\times 3}, \tag{1}$$

where $N$ is the number of points and $K$ is the number of keypoints. The function is SE(3)-equivariant if for any point cloud $\mathbf{P}$ and any rigid body transformation $(\mathbf{R}, \mathbf{t}) \in \mathrm{SE}(3)$, the following equation holds:

$$f(\mathbf{RP} + \mathbf{t}) \equiv \mathbf{R} f(\mathbf{P}) + \mathbf{t}. \tag{2}$$

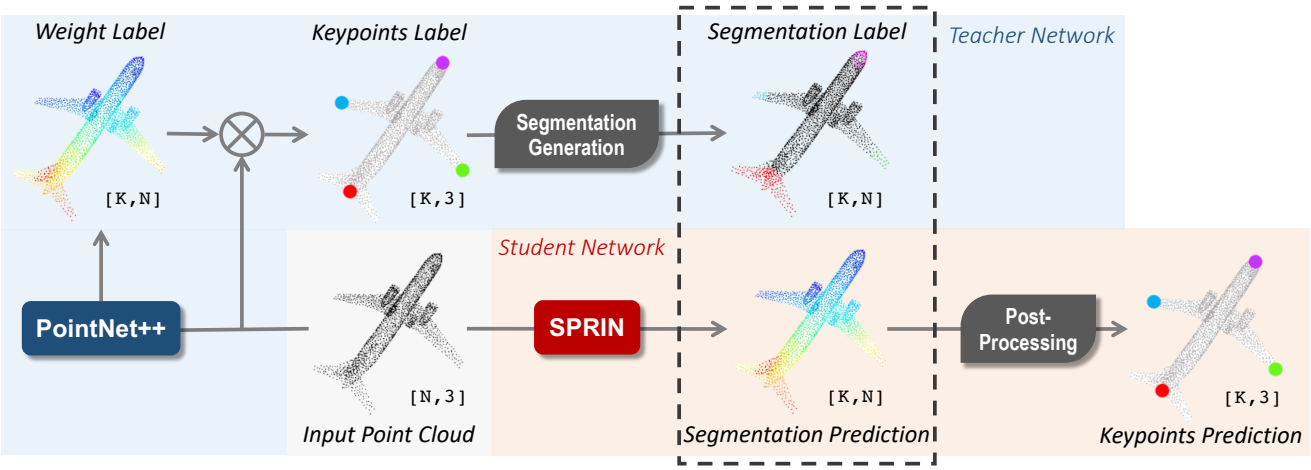


**Fig. 2:** The pipelines of USEEK, which follows a teacher-student architecture. All the "labels" are pseudo ground-truth labels generated by the teacher network, free from any additional human annotations. The PointNet++ [24] module is with fixed parameters, extracted from a pre-trained Skeleton Merger [29]. The SPRIN [44] network is to be optimized in the training process. Binary Cross Entropy (BCE) loss is used for loss computation.

Furthermore, keypoints detected are considered as category-level if they can best represent the shared geometric features of a category of objects.

### *B. USEEK: a Teacher-Student Framework*

To develop a keypoint detector that is both category-level and SE(3)-equivariant, we propose USEEK that has a teacher-student structure. The teacher network is a category-level keypoint detector that can be pre-trained in a self-supervised manner. The student network consists of an SE(3)-invariant backbone.

Conceptually, the major merit of the teacher-student networks is to decouple the learning process, where each network is only responsible for the property that it is most adept at. Moreover, the SE(3)-invariant networks are usually harder to train. Hence, the teacher-student structure might alleviate the burden of the student network in the process of keypoint discovery. These are the central reasons why the teacher-student structure is essential and why the simpler approaches (shown in Section IV-A) cannot achieve competitive results. Next, with the blueprint for USEEK established, we instantiate it with concrete details.

**The teacher network.** In the teacher network, every keypoint is considered as the weighted sum of all the point coordinates in the cloud. To produce the desired weight matrix $\mathbf{W} \in \mathbb{R}^{K \times N}$, we extract the PointNet++ [24] encoder from Skeleton Merger [29], a state-of-the-art category-level keypoint detector. The multiplication of the weight matrix and the input cloud directly gives the predicted keypoints

$$\mathbf{p} = \mathbf{W}\mathbf{P}. \tag{3}$$

We follow the same self-supervised training procedure as shown in [29] to pre-train the PointNet++ module.

The predicted keypoints by the teacher network are used to generate pseudo labels for the student network. Since nearby points share the same semantics, all the points within a distance of a certain keypoint are considered as candidates for the corresponding keypoint and therefore marked with positive labels; the rest are the negative ones. With altogether $K$ keypoints detected in an $N$-point cloud, the final pseudo label can be regarded as a $K$-channel binary segmentation mask $\mathbf{M} \in \mathbb{R}^{K \times N}$.

**The student network.** The student network of USEEK utilizes SPRIN [44], a state-of-the-art SE(3)-invariant backbone, to produce SE(3)-equivariant keypoints. The SPRIN module takes as input the canonical object point cloud and predicts the labels generated by the teacher network.

For training, the student network optimizes a Binary Cross Entropy (BCE) loss between the per-point $K$-channel binary predictions and the corresponding pseudo labels. To deal with imbalanced negative labels over positive ones, we perform importance sampling [35]. Besides, we highlight that the whole training procedure does not require any SE(3) data augmentations because the SE(3)-invariant backbone could automatically generalize to unseen poses.

When tested, the predictions are post-processed to produce the final keypoints. We take the `argmax` operation, which means the point that has the highest confidence value is selected as the detected keypoint for each of the $K$ segmentation classes. Moreover, we take Non-Maximum Suppression [22] to encourage sparse locality of the keypoints.

### *C. From Keypoints to Task-Relevant Object Frames*

The essence of generalizable manipulation is arguably to transfer the functional knowledge from known object(s) to the unknowns. In this section, we exhibit an easy-to-execute yet effective procedure that leverages the detected keypoints to determine the task-relevant object frames.

Taking the airplane in Figure 3 (Left) as an illustrative example, the four detected keypoints lie on the nose, the tail, the left wingtip, and the right wingtip, respectively. With clear semantics, we can easily set up simple rules so as to establish the frame: the $x$-axis is parallel to the connected line of the two wingtips; the $z$-axis is parallel to the connected

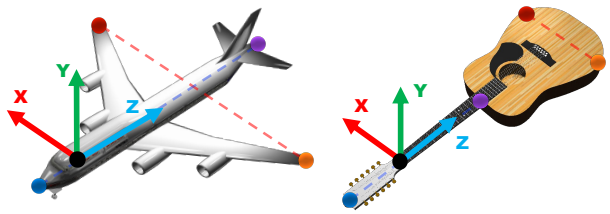


**Fig. 3:** The category-level task-relevant coordinate frames for manipulation can be acquired from the keypoints detected by USEEK together with few human decided priors on only one object.

line of the nose and the tail; the $y$-axis is vertical to both the $x$-axis and the $z$-axis; the origin is the projection of the demonstrated grasping position on the $z$-axis. Thanks to the intra-category alignment property of USEEK, once the rules on *one* specific object are set up, they instantly adapt to *all* the other instances. In this sense, the average labor spent per instance is negligible. In Figure 3 (Right), we provide the category of the guitar as an additional example.

Notably, unlike previous works [19], [31], USEEK avoids any searching or optimization process when inferring object frames, thus dramatically reducing the computational costs.

### D. One-Shot Imitation Learning with USEEK

Equipped with keypoints and task-relevant object frames, we are now ready for category-level manipulation. The task is to pick up an unseen object from a randomly initialized SE(3) pose and place it to another specified pose. To showcase the full potential of USEEK, we follow a more challenging setting of *one*-shot imitation learning rather than the *few*-shot setting commonly seen in prior works [31], since a decreased number of demonstrations calls for increased robustness and consistency of the proposed object representation.

Specifically, the demonstration $\mathscr{D} = (\mathbf{P}_{\text{demo}}, \mathbf{X}^G_{\text{demo}})$ is merely the point cloud of an object $\mathbf{P}_{\text{demo}}$ and a functional grasping pose of the Gripper $\mathbf{X}^G_{\text{demo}}$ in the form of a coordinate frame. Given the demonstration $\mathscr{D}$, USEEK infers the task-relevant coordinate frame of the demonstration Object $\mathbf{X}^O_{\text{demo}}$. Then, we calculate the rigid body transformation from the Object to the Gripper ${}^O\mathbf{T}^G_{\text{demo}}$, s.t.

$$\mathbf{X}^G_{\text{demo}} = \mathbf{X}^O_{\text{demo}}\, {}^O\mathbf{T}^G_{\text{demo}}, \tag{4}$$

where all the poses and transformations are in the homogeneous coordinates. Assuming the object is rigid and the grasp is tight, ${}^O\mathbf{T}^G_{\text{demo}}$ is general for the category. Thus, we rewrite it as ${}^O\mathbf{T}^G$ for simplicity.

At test time, the observation $\mathscr{O} = (\mathbf{P}_{\text{init}}, \mathbf{P}_{\text{targ}})$ consists of the point cloud of an unseen object in an arbitrary initial pose $\mathbf{P}_{\text{init}}$ and another point cloud indicating the target pose $\mathbf{P}_{\text{targ}}$. Note that it is not required that the object in the target pose is the same as the one in the initial pose. USEEK infers the object frame in the initial pose $\mathbf{X}^O_{\text{init}}$ and the frame in the target pose $\mathbf{X}^O_{\text{targ}}$. Then, with ${}^O\mathbf{T}^G$ prepared, we can easily calculate the poses of the gripper for pick and place:

$$\mathbf{X}^G_{\text{pick}} = \mathbf{X}^O_{\text{init}}\, {}^O\mathbf{T}^G, \tag{5}$$

$$\mathbf{X}^G_{\text{place}} = \mathbf{X}^O_{\text{targ}}\, {}^O\mathbf{T}^G. \tag{6}$$

| | Airplanes | Chairs | Guitars | Knives |
|---|---|---|---|---|
| ISS [45] | 31.9 | 9.9 | 34.6 | 35.8 |
| Skeleton Merger [29] | 19.9 | 10.7 | 24.2 | 18.8 |
| w/ data augmentation | 14.0 | 7.3 | 10.9 | 21.2 |
| w/ ICP [26] | 22.2 | 10.0 | 25.8 | 17.2 |
| w/ SPRIN [44] encoder | 74.7 | 18.4 | 41.1 | 45.6 |
| USEEK w/ KL divergence | 55.7 | 40.1 | 64.9 | 17.0 |
| USEEK (ours) | **85.5** | **53.8** | **70.2** | **60.3** |
| Teacher Network | 87.0 | 65.3 | 71.1 | 40.3 |

**TABLE II:** mIoU scores of keypoints detected by USEEK and the baselines on the SE(3) KeypointNet [42] dataset. The best results are shown in **bold** type. In addition, *Teacher Network* tested on the canonical dataset is also included but marked in gray for distinction.

Finally, we leverage off-the-shelf motion planning [15] and inverse kinematics [30] tools to execute the predicted poses.

## IV. Experiments: Semantics of Keypoints

In this section, we evaluate whether the keypoints detected by USEEK are with proper and accurate semantics when the input point clouds are under SE(3) transformations.

### A. Setup and Baselines

The experiments are conducted on the KeypointNet [42] dataset, where keypoints with category-level semantic labels are annotated by experts. We use the mean Intersection over Unions (mIoU) [34] score to measure the alignment between the predictions and the human annotations. To evaluate the property of SE(3)-equivariance, the inputs and their annotations are under the same random SE(3) transformations. We compare USEEK with the following methods:

- *Intrinsic Shape Signatures (ISS).* ISS [45] is a classic hand-crafted 3D keypoint detector.
- *Skeleton Merger.* The Skeleton Merger is trained on ShapeNet [3] with canonical point clouds.
- *Skeleton Merger w/ data augmentation.* We apply SE(3) data augmentations to the training dataset.
- *Skeleton Merger w/ ICP.* During test time, we randomly take one instance in canonical pose on the training dataset as the template, and adopt ICP [26] initialized with RANSAC [10] for point cloud registration.
- *Skeleton Merger w/ SPRIN encoder.* The encoder of Skeleton Merger is replaced with an SE(3)-invariant SPRIN encoder.
- *USEEK w/ KL divergence.* The SE(3)-invariant backbone in USEEK is slightly revised to predict weight matrices. It is optimized via the Kullback–Leibler (KL) divergence [16] between the predicted weights and the pseudo weight labels.

Additionally, we evaluate the *Teacher Network* of USEEK on the *canonical* KeypointNet dataset w/o SE(3) transformations. This auxiliary configuration reflects the quality of the semantics that USEEK could learn from.

### B. Results and Discussion

The qualitative results of keypoints detected by USEEK are shown in Figure 4. Under SE(3) transformations and

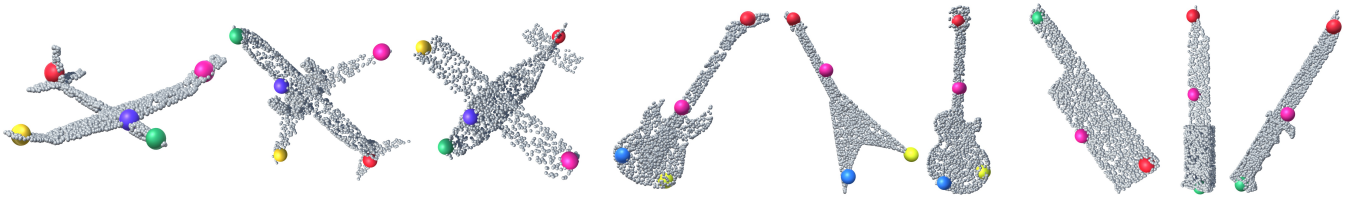

**Fig. 4:** Qualitative results of the keypoints detected by USEEK. The input point clouds are under random SE(3) transformations. The color of the keypoints stands for the predicted category-level semantic correspondence (i.e., keypoints of a category are color-aligned).

with large shape variance, the keypoints are well aligned across the category and identify semantic parts that are akin to human intuition. The quantitative results given in Table II reveal that USEEK substantially outperforms all the other baselines by a large margin. Surprisingly, USEEK approaches Teacher Network performance on the categories of airplanes and guitars, and even surpasses it on knives. In fact, it is reasonable that the student in USEEK may excel its teacher, given the elaborately designed label generation mechanism where surrounding points of a predicted keypoint are all considered as the keypoint candidates.

For the other baselines, it is as expected that baselines such as Skeleton Merger w/ data augmentation are unable to produce semantically meaningful keypoints because it is extremely hard, if not impossible, for the naive PointNet++ encoder to capture invariant patterns from arbitrary SE(3) transformations. Further, we also notice that stronger baseline methods such as Skeleton Merger w/ SPRIN encoder is not as capable as USEEK. We attribute its failure to the fact that the SE(3)-invariant guarantee of the encoder often results in more parameters in the network as well as much more difficulty in training. Compared with the baselines, USEEK enables the training process by following a teacher-student structure, which transforms unsupervised causal discovery of latent keypoints to a simpler supervised learning task.

## V. Experiments: Keypoints for Manipulation

In this section, we evaluate the power of USEEK as an object representation for generalizable manipulation. We conduct experiments in both simulated and real-world environments where USEEK is utilized to perform category-level pick-and-place via one-shot imitation learning.

### A. Setups

We build simulated environment mimicking the physical setup in PyBullet [7]. For the real-world environment, we use a Franka Panda robot arm for manipulation and four Intel RealSense D435i depth cameras at each corner of the table for the capture of point clouds. We use wooden wedges, metal holders, or plasticine-kneaded holders to support the objects so that they can be placed in arbitrary poses. The real-world experiment setup is visualized in Figure 5.

The robot needs to pick up unseen objects in randomly initialized SE(3) poses and place them in a specified target pose. For achieving this task, only one demonstration of a functional grasping pose on a single object is provided for each category. For greater diversity, we design the tasks on three categories: 1) for airplanes, we demonstrate two distinct functional poses (i.e., in the front and in the rear) to inspect whether the robot can acts accordingly with the demonstration; 2) for guitars, the robot is asked to transfer the guitar from one metal stand to the other by holding its neck. This task is more challenging because the neck of a guitar is delicate and thus asks for very precise control; 3) for knives, the robot needs to pick up a knife on the table and use it to chop tofu. The chopping direction is inferred from a given target pose.

### B. Baselines and Evaluation Metrics

To show the effectiveness of USEEK for manipulation, we perform comparison in the simulated environment with ICP [26] initialized with RANSAC [10] coarse registration and Neural Descriptor Fields (NDF) [31]. ICP estimates the frame of an unseen object by trying to align it to the given demonstration object. NDF is a recently proposed state-of-the-art dense correspondence representation for category-level manipulation. NDF encodes the relation of every SE(3) pose of external rigid bodies and a demonstration object through a set of query points. The task-relevant coordinate frame of an unseen object is determined by regressing its descriptor field to the demonstrated one.

We measure the success rates for both grasping itself and the whole process of pick-and-place. Specifically, the grasping is successful if the gripper stably takes the object off the table with a vertical clearance of at least 10 centimeters. For pick-and-place, the transnational and rotational tolerance of the final pose is 5 centimeters and 0.2 radian respectively for each degree of freedom. The experiments are performed on 100 different object instances randomly taken from ShapeNet [3] without cherry-picking for each category. Besides, we also present the average inference time that each method takes to perform one execution.

### C. Training Details

We train both USEEK and the NDF baseline on the ShapeNet [3] dataset and directly deploy the models to the manipulation tasks. We use Adam [14] for optimization with a default learning rate of 0.001. While NDF could handle multiple categories with one unified model, we find the performance of the given pre-trained model drops significantly because of the large shape variance in the chosen categories. Therefore, we train independent models of each category for both USEEK and NDF for fair comparison.

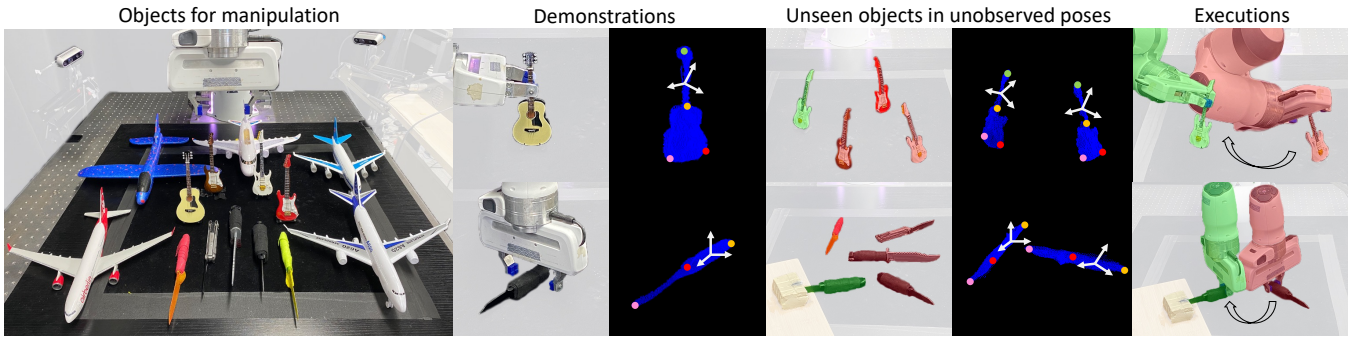


**Fig. 5:** In real-world experiments, objects of large intra-category shape variance are manipulated. The alignment and SE(3)-equivariance properties enable USEEK to transfer the functional knowledge from one demonstration object to various unseen objects in arbitrary poses.

| | Airplanes | | Guitars | | Knives | |
|---|---|---|---|---|---|---|
| | Grasp | Overall | Grasp | Overall | Grasp | Overall |
| ICP [26] | 0.08 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 |
| NDF [31] | 0.25 | 0.04 | 0.22 | 0.01 | 0.71 | 0.61 |
| USEEK | **0.90** | **0.81** | **0.89** | **0.69** | **0.93** | **0.85** |

**TABLE III:** The success rates for grasping and the overall process of pick-and-place with the setting of one-shot imitation learning.

### D. Simulation Experiments

The success rates of ICP [26], NDF [31], and USEEK are shown in Table III. USEEK significantly outperforms the two baselines in the challenging one-shot imitation learning setting, showcasing its strengths as an object representation for generalizable manipulation. In comparison, ICP as a template matching method is virtually unable to deal with large intra-category shape variance and cannot provide accurate information for manipulation. Meanwhile, NDF manages to offer relatively decent results on the category of knives. But it is overall far less competitive than USEEK. We observe a failure mode where NDF cannot find the correct grasping orientation. We attribute this to the difficulty in regressing the neural descriptors of a large number of ($\sim$500 in default) query points with only one demonstration available. Furthermore, for NDF, it takes 3.65 seconds on average to infer for one execution on an NVIDIA RTX 3070 GPU while for USEEK, it only takes 0.11 seconds, indicating a more than 30$\times$ efficiency boost.

### E. Real World Execution

Finally, we validate USEEK can be successfully deployed for generalizable manipulation on the real robot without sim-to-real finetuning. We apply the RRT-Connect [15] algorithm for motion planning and the default methods from the `Panda Robot` library [30] for executing inverse kinematics.

Quantitatively, a total of 50 executions are conducted on a series of novel airplanes of various appearances and materials, and in arbitrary initial poses. Among them, 37 trials end up with success, resulting in a success rate of 74%. We find that USEEK is robust to the quality degradation of raw point clouds taken from depth cameras. The results also show that USEEK can well adapt to the color changing and shape variance in real-world scenes. A common failure mode is that USEEK is unable to handle the severe artifact in the point clouds (e.g., half of the wing is missing) largely due to high reflective rates of the oil paint on the surface of some airplanes. We believe this can be alleviated in the future with the access to more advanced depth cameras. Qualitatively, we present example execution trajectories for airplanes, guitars, and knives in Figure 6. More results can be found in the supplementary video.

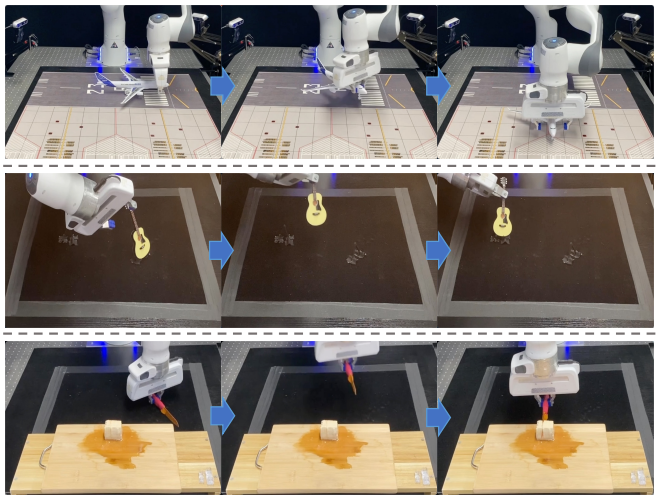

**Fig. 6:** Leveraging USEEK, we present the example execution trajectories of pick-of-place by manipulating the front of an airplane, the neck of a guitar, and the handle of a knife.

## VI. Conclusion

We present USEEK, an unsupervised SE(3)-equivariant keypoints detector to empower generalizable pick-and-place via one-shot imitation learning. USEEK uses teacher-student structure so that keypoints with the desired properties can be acquired in an unsupervised manner. Extensive experiments show that the keypoints detected by USEEK enjoy rich semantics, which enables the transfer of functional knowledge from one demonstration object to unseen objects with large intra-category shape variance in arbitrary poses. Further, USEEK is found to be robust with limited demonstrations and efficient at inference time. Given all these advantages, we believe USEEK is a capable object representation and has the potential to empower many manipulation tasks.

## REFERENCES


[1] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[2] R. A. Brooks. Planning collision-free motions for pick-and-place operations. *The International Journal of Robotics Research*, 2(4):19–44, 1983.

[3] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[4] B. Chen, P. Abbeel, and D. Pathak. Unsupervised learning of visual 3d keypoints for control. In *International Conference on Machine Learning*, pages 1539–1549. PMLR, 2021.

[5] S. Chiaverini, B. Siciliano, and O. Egeland. Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on control systems technology*, 2(2):123–134, 1994.

[6] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling. Spherical cnns. In *International Conference on Learning Representations*, 2018.

[7] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2019.

[8] J. S. DeLoache. Rapid change in the symbolic functioning of very young children. *Science*, 238(4833):1556–1557, 1987.

[9] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. J. Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.

[10] 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*, 24(6):381–395, 1981.

[11] P. Florence, L. Manuelli, and R. Tedrake. Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters*, 5(2):492–499, 2019.

[12] P. R. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018.

[13] W. Gao and R. Tedrake. kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6527–6533. IEEE, 2021.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.

[16] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[17] J. Li and G. H. Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 361–370, 2019.

[18] X. Li, Y. Weng, L. Yi, L. J. Guibas, A. Abbott, S. Song, and H. Wang. Leveraging se (3) equivariance for self-supervised category-level object pose estimation from point clouds. *Advances in Neural Information Processing Systems*, 34:15370–15381, 2021.

[19] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *The International Symposium of Robotics Research*, pages 132–157. Springer, 2019.

[20] L. Manuelli, Y. Li, P. Florence, and R. Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning. In *Conference on Robot Learning*, pages 693–710. PMLR, 2021.

[21] L. McCune-Nicolich. Toward symbolic functioning: Structure of early pretend games and potential parallels with language. *Child development*, pages 785–797, 1981.

[22] A. Neubeck and L. Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[23] 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*, pages 652–660, 2017.

[24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[25] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494. IEEE, 2009.

[26] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001.

[27] C. Sahin and T.-K. Kim. Category-level 6d object pose recovery in depth images. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[28] T. Schmidt, R. Newcombe, and D. Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2016.

[29] R. Shi, Z. Xue, Y. You, and C. Lu. Skeleton merger: an unsupervised aligned keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 43–52, 2021.

[30] S. Sidhik. Franka ROS Interface: A ROS/Python API for controlling and managing the Franka Emika Panda robot (real and simulated)., Dec. 2020.

[31] 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. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[32] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg. Learning rope manipulation policies using dense object descriptors trained on synthetic depth data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9411–9418. IEEE, 2020.

[33] S. Suwajanakorn, N. Snavely, J. J. Tompson, and M. Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. *Advances in neural information processing systems*, 31, 2018.

[34] L. Teran and P. Mordohai. 3d interest point detection via discriminative learning. In *European conference on computer vision*, pages 159–173. Springer, 2014.

[35] S. T. Tokdar and R. E. Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.

[36] M. Vecerik, J.-B. Regli, O. Sushkov, D. Barker, R. Pevceviciute, T. Rothörl, R. Hadsell, L. Agapito, and J. Scholz. S3k: Self-supervised semantic keypoints for robotic manipulation via multi-view consistency. In *Conference on Robot Learning*, pages 449–460. PMLR, 2021.

[37] 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)*, pages 10059–10066. IEEE, 2020.

[38] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019.

[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*, pages 2642–2651, 2019.

[40] B. Wen, W. Lian, K. Bekris, and S. Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. *arXiv preprint arXiv:2201.12716*, 2022.

[41] Y. You, W. Liu, Y. Ze, Y.-L. Li, W. Wang, and C. Lu. Ukpgan: A general self-supervised keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17042–17051, 2022.

[42] Y. You, Y. Lou, C. Li, Z. Cheng, L. Li, L. Ma, C. Lu, and W. Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13647–13656, 2020.

[43] Y. You, Y. Lou, Q. Liu, Y.-W. Tai, L. Ma, C. Lu, and W. Wang. Pointwise rotation-invariant network with adaptive sampling and 3d spherical voxel convolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12717–12724, 2020.

[44] Y. You, Y. Lou, R. Shi, Q. Liu, Y.-W. Tai, L. Ma, W. Wang, and C. Lu. Prin/sprin: On extracting point-wise rotation invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[45] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops*, pages 689–696. IEEE, 2009.