2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Local Neural Descriptor Fields: Locally Conditioned Object Representations for Manipulation

Ethan Chun[1], Yilun Du[1], Anthony Simeonov[1], Tomas Lozano-Perez[1], Leslie Kaelbling[1]
[1]Computer Science and Artificial Intelligence Laboratory, MIT, USA

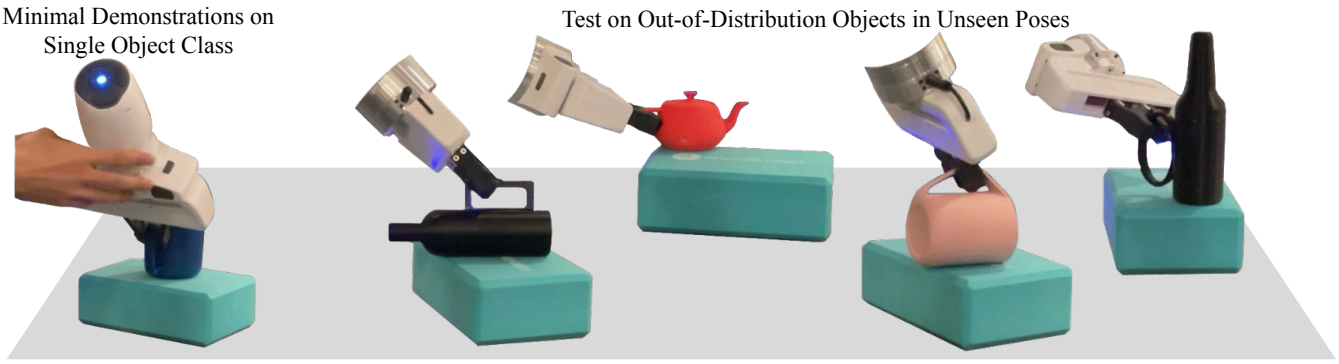


Fig. 1: Given minimal (5-10) real world demonstrations of grasping and picking up two different upright mugs, Local Neural Descriptor Field can successfully grasp and pick up a set of geometrically distinct objects at arbitrary SE(3) poses.

***Abstract*— A robot operating in a household environment will see a wide range of unique and unfamiliar objects. While a system could train on many of these, it is infeasible to predict all the objects a robot will see. In this paper, we present a method to generalize object manipulation skills acquired from a limited number of demonstrations, to novel objects from unseen shape categories. Our approach, Local Neural Descriptor Fields (L-NDF), utilizes neural descriptors defined on the local geometry of the object to effectively transfer manipulation demonstrations to novel objects at test time. In doing so, we leverage the local geometry shared between objects to produce a more general manipulation framework. We illustrate the efficacy of our approach in manipulating novel objects in novel poses – both in simulation and in the real world. Project website, videos, and code: https://elchun.github.io/lndf/.**

I. Introduction

A robot operating autonomously in an household environment will encounter a wide variety of unseen objects. While individual objects may be novel in shape, many can be decomposed into a set of previously seen constituent parts. Consider the novel objects illustrated in Fig. 1 – while a bottle with a handle may be unseen, both bottles and mugs are individually known. Therefore, one may propose that a robot manipulate the novel object via skills learned on both bottles and mugs. In this paper, we investigate enabling such generalization using an imitation learning paradigm. In particular, we wish to construct a system which, when given a small set (5 - 10) of manipulation demonstrations on a single category of objects, can successfully execute this skill on novel objects types in arbitrary SE(3) orientations.

To enable efficient learning, we build on the Neural Descriptor Fields (NDF) system [21]. NDF assigns a dense descriptor to each point in a shape, with similar points across different objects in a given category assigned similar descriptors. Object manipulation may be generalized to novel objects in the same category by finding a corresponding set of dense descriptors in the novel object. A limitation of NDF, however, is that it relies on a single global latent to encode all geometric aspects of a shape in a given category. When given an object of a new category, this representation cannot capture the resultant geometry, preventing NDF from transferring object manipulation to new categories of objects.

We circumvent this problem by using a voxel grid of latents to locally capture the geometry and descriptors of a shape (see Fig. 2); where each latent encodes the geometry and descriptors of a local spatial region. With this encoding scheme, descriptors of shapes in new categories can be more accurately encoded, as individual patches of the new shape correspond to patches from various categories of training object. We illustrate how this encoding enables generalization of object manipulation to new categories, referring to our approach as Local Neural Descriptor Fields (L-NDF).

An issue that arises when encoding descriptors locally is that descriptors of a portion of an object may change as the object is transformed. For example, the handle of a mug is represented with different voxel latents when it is translated and rotated. To ensure that descriptors are consistent across rigid object transformations, we propose a contrastive loss which explicitly enforces descriptor consistency when objects are transformed.

To transfer object manipulation demonstrations from one object to another, we must find corresponding sets of descriptors between the objects. In NDFs, a global gradient optimization procedure is used to minimize descriptor distance. With L-NDF, a similar global optimization procedure

Correspondence to: yilundu@mit.edu

is difficult to run, as descriptors of an object are only locally encoded – lacking a consistent global direction in which descriptors are changing in a shape. To overcome this difficulty, we propose to initialize optimization across a diverse set of positions in a shape – running local optimization to choose the descriptor with a minimal descriptor distance as our final, matching, descriptor.

We demonstrate that L-NDFs can be reliably used to generalize object manipulation to both novel objects and objects at unseen SE(3) poses. Given only (5-10) demonstration, our framework is able to manipulate novel objects (such as a tea cup or a bowl with a handle attached) in both simulation as well as on a real robot.

II. Related Work

A. Generalizable Manipulation

Our work follows a long line of work on using imitation learning for manipulation. When object models are known, pose estimation may be used for manipulation [20, 28, 29]. When the precise geometry of objects is unknown, template matching with coarse 3D primitives [8, 14, 25] or nonrigid registration [20] can be used; but such methods still suffer when objects deviate substantially from templates. Recent work has explored more flexible representations for imitation learning, such as keypoint [6, 7, 12] or dense descriptors [4, 21, 24]. Most similar to our work – DON [4] and NDF explore 2D and 3D dense descriptors for object manipulation – but both only demonstrate generalization within the same category of objects. In contrast, our approach enables object manipulation for novel categories of shapes at test time.

B. Neural Implicit Representations for Robotics

Neural implicit representations [13, 17] have emerged as a promising representation of 3D geometry in robotics. Different works have explored how implicit representations may be used in navigation [1], localization [5, 15, 26], SLAM [16, 23, 30], and manipulation [10, 11, 19, 21, 22, 27]. In the context of manipulation, [10, 27] utilize NeRF as an approach to extract the underlying 3D geometry of a scene. In contrast, [19, 21, 22] build on the Neural Descriptor Field framework for learning manipulation skills, where underlying high-dimensional neural descriptors are used to transfer and generalize demonstrations. Our work extends NDFs to work with locally conditioned implicit representations.

III. Background: Manipulation with Neural Descriptor Fields

A Neural Descriptor Field (NDF) [21] encodes the shape of an object using a function f that maps a 3D point $\mathbf{x} \in \mathbb{R}^3$ and an partial object point cloud $\mathbf{P} \in \mathbb{R}^{3\times N}$ to a spatial descriptor in $\mathbb{R}^d$:

$$f(\mathbf{x}|\mathbf{P}) : \mathbb{R}^3 \times \mathbb{R}^{3\times N} \rightarrow \mathbb{R}^d. \tag{1}$$

NDFs are also trained to learn correspondence over objects in the same category, so that points near similar geometric features of different instances (e.g., a point near the neck of two different bottles) are mapped to similar descriptor values.

NDFs can be generalized to assign descriptors to full SE(3) poses, rather than individual points. This is achieved by concatenating the descriptors of the individual points in a *rigid set* of query points $\mathcal{X} \in \mathbb{R}^{3\times N_q}$, i.e., a set of three or more non-collinear points $\mathbf{x}_i,\ i = 1...N_q$, that are constrained to transform together rigidly. This construction allows NDFs to represent an SE(3) pose $\mathbf{T}$ via its action on $\mathcal{X}$, i.e., via the points of the *transformed query point cloud* $\mathbf{T}\mathcal{X}$:

$$\mathcal{Z} = F(\mathbf{T}|\mathbf{P}) = \bigoplus_{\mathbf{x}_i \in \mathcal{X}} f(\mathbf{T}\mathbf{x}_i|\mathbf{P}) \tag{2}$$

Thus, F maps a point cloud $\mathbf{P}$ and an SE(3) pose $\mathbf{T}$ to a category-level pose descriptor $\mathcal{Z} \in \mathbb{R}^{d\times N_q}$.

Few-Shot Manipulation Learning with NDFs. Next, we discuss how to leverage NDF for few-shot learning of object manipulation skills. Consider a set of K demonstrations, $\{\mathcal{D}_i\}_{i=1}^K$, where each demonstration, $\mathcal{D}_i = (\mathbf{P}^i, \mathbf{T}^i_{pick}, \mathbf{T}^i_{place})$ consists of a object $\mathbf{P}^i$, and two poses: the end-effector pose before grasping, $\mathbf{T}^i_{pick}$, and the relative pose of the placement surface $\mathbf{T}^i_{place}$. We define a set of query points $\mathcal{X}_{pick}$ and $\mathcal{X}_{place}$ to represent the gripper and placement surface, respectively. We then utilize (2) to encode each pose $\mathbf{T}^i_*$ into its vector of descriptors $\mathcal{Z}^i_*$, conditional on the respective object point cloud $\mathbf{P}^i$, obtaining a set of spatial descriptor tuples $\{(\mathcal{Z}^i_{pick}, \mathcal{Z}^i_{rel})\}_{i=1}^K$. The set of descriptors is averaged over the K demonstrations to obtain *single* pick and place descriptors $\bar{\mathcal{Z}}_{pick}$ and $\bar{\mathcal{Z}}_{rel}$

When a new object is placed in the scene at test time, we obtain a point cloud $\mathbf{P}^{test}$ and leverage (3) to recover $\mathbf{T}^{test}_{pick}$ and $\mathbf{T}^{test}_{rel}$ by minimizing the distance to spatial descriptors $\bar{\mathcal{Z}}_{pick}$ and $\bar{\mathcal{Z}}_{rel}$.

$$\bar{\mathbf{T}} = \underset{\mathbf{T}}{\operatorname{argmin}} \|F(\mathbf{T}|\mathbf{P}) - F(\hat{\mathbf{T}}|\hat{\mathbf{P}})\| \tag{3}$$

We rely on off-the-shelf inverse kinematics and motion planning algorithms to execute the final predicted poses.

IV. Local Neural Descriptor Fields

Given a set of K, single object class, pick and place demonstrations, $\{\mathcal{D}_i\}_{i=1}^K$, where each demonstration, $\mathcal{D}_i = (\mathbf{P}^i, \mathbf{T}^i_{pick}, \mathbf{T}^i_{place})$, consists of a partial object point cloud $\mathbf{P}^i$, end-effector pick pose $\mathbf{T}^i_{pick}$ and place pose $\mathbf{T}^i_{place}$, we are interested in generalizing the tasks to a set of new objects $\mathbf{P}'$ from unseen object classes. To solve this problem, we develop an approach using locally defined descriptors and propose suitable modifications of the NDF pipeline (Section III) to utilize such descriptors.

In particular, in Section IV-A, we introduce Local Neural Descriptor Fields and illustrate how they may be used to locally encode the geometry of objects. In Section IV-B, we discuss how we may build SE(3) equivariance into the underlying descriptor of L-NDF. Finally, in Section IV-C, we discuss how to modify the underlying optimization procedure to allow us to search for an ideal pose within the local descriptor field landscape.

A. Local Descriptor Fields

A global NDF model cannot generalize effectively to new categories of objects. To solve this problem, we use local descriptor fields for objects: each element of a voxel grid contains a latent vector representation of the object's local shape near that voxel.

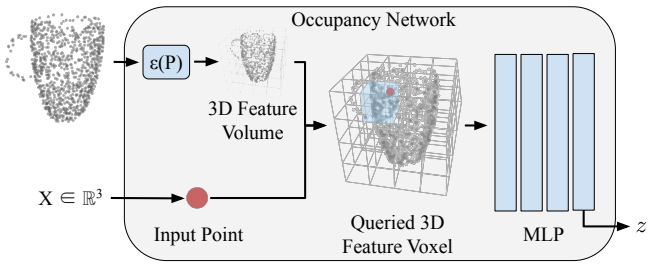


Fig. 2: **Local Neural Descriptor Field Architecture** – A L-NDF takes any coordinate in 3D space, $\mathbf{x}$, and a conditioning point cloud $\mathbf{P}$. It then uses an encoder $\epsilon(\mathbf{P})$ to encode $\mathbf{P}$ into a 3D feature volume from which the voxel containing $\mathbf{x}$ is queried. These feature are passed into an MLP decoder where the activations of the decoder's final layer are extracted to create the spatial descriptor, z.

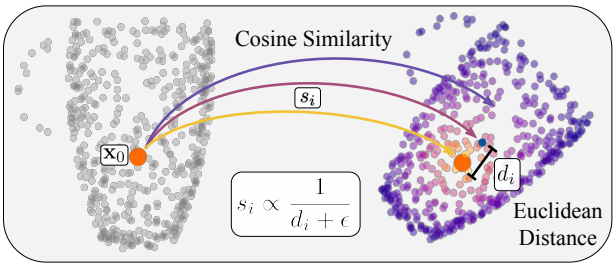


Fig. 3: **Contrastive Loss Term for L-NDF** – The spatial descriptor of a 3D coordinate, $\mathbf{x}$, with respect to an observed point cloud, $\mathbf{P}$, is similar across any transform, $\mathbf{T} \in \mathrm{SE}(3)$. Additionally, geometrically farther points have decreasingly similar descriptors.

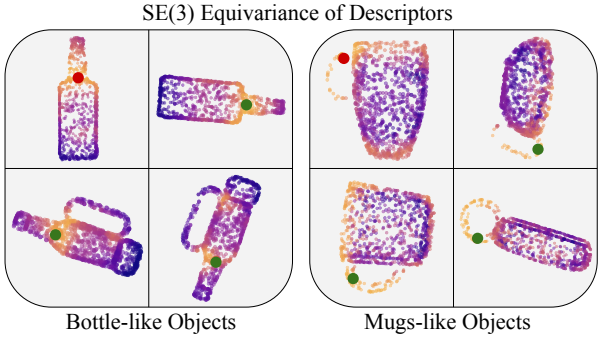


Fig. 4: **SE(3) Equivariance of Object Encoding** – Heat map of cosine descriptor difference from selected point (in red). The descriptor field remains consistent across different objects in arbitrary SE(3) transformations.

In L-NDF, we use a convolutional occupancy network encoder [18], $\epsilon(\mathbf{P})$, to encode a partial point cloud $\mathbf{P}$ into a voxel grid of latents (illustrated in Fig. 2). When querying a particular point, $\mathbf{x}$, the corresponding voxel from the latent feature, $\epsilon(\mathbf{P})$, is retrieved and processed through MLP layers. The final set of MLP activations are then concatenated to produce a latent code z. Formally, this encoder is defined as (4).

$$z = f(\mathbf{x}|\mathbf{P}) = \Phi(\mathbf{x}|\epsilon(\mathbf{P})_{\lfloor \mathbf{x} \rfloor}). \tag{4}$$

Following [21], we utilize occupancy reconstruction to train and learn features for NDFs.

B. Training and Learning SE(3) Equivariance

To ensure that our models generalize to rigid transformations of the target object, we design a training regime that enforces the descriptors at the same point of an object (in its local frame) to remain invariant under SE(3)transformations of the object.

Enforcing SE(3) Equivariance. In contrast to [21], our system is not inherently SE(3) equivariant. Instead, we utilize a contrastive loss term to shape the network activations such that they exhibit SE(3) equivariance. Formally, an encoder, $f(\mathbf{x}|\mathbf{P})$, is SE(3) equivariance if, for any rigid body transform $\mathbf{T} \in \mathrm{SE}(3)$,

$$f(\mathbf{x}|\mathbf{P}) \equiv f(\mathbf{T}\mathbf{x}|\mathbf{T}\mathbf{P}) \tag{5}$$

A simple approach to enforce equivariance is to directly enforce that the encoding of corresponding points should be preserved across SE(3) transformations. However, directly enforcing this constraint was problematic as we found f to map all inputs to the same encoding. Therefore, we considered directly enforcing an additional constraint, that different input points produce different encodings, but found the resultant descriptors were no longer semantically consistent between shapes.

We found that a robust alternative to construct descriptors that are both SE(3) equivariant and semantically consistent was to enforce (6), that descriptor similarity between two points is roughly proportional to their inverse distance across different rigid transformations $\mathbf{T}$ (illustrated in Fig. 3).

$$\mathrm{sim}(f(\mathbf{x}_1|P), f(\mathbf{T}\mathbf{x}_2|\mathbf{T}P)) \propto \frac{1}{\|\mathbf{x}_1 - \mathbf{x}_2\| + \epsilon}, \tag{6}$$

This constraint enforces that descriptors are both equivariant across rigid transformations of a shape, but also that they vary smoothly with respect to small Euclidean perturbations of the point.

To enforce this loss, we sample k points within the bounding box of the object. We designate the first point, $\mathbf{x}_0$, as the point we compute descriptor similarity with respect to in the remaining $k-1$ points. For each point, we compute the cosine similarity, s_i, between $f(\mathbf{x}_0|\mathbf{P})$ and $f(\mathbf{T}\mathbf{x}_i|\mathbf{T}\mathbf{P})$,

$$s_i = \frac{f(\mathbf{x}_0|\mathbf{P}) \cdot f(\mathbf{T}\mathbf{x}_i|\mathbf{T}\mathbf{P})}{\max(||f(\mathbf{x}_0|\mathbf{P})|| \cdot ||f(\mathbf{T}\mathbf{x}_i|\mathbf{T}\mathbf{P})||, \epsilon)} \tag{7}$$

We compute corresponding target similarity values for each $\mathbf{x}_i$ with respected to the first point $\mathbf{x}_0$

$$t_i = \frac{1}{d(\mathbf{x}_0, \mathbf{x}_i) + \beta}, \tag{8}$$

and enforce that similarities are roughly proportional to the inverse distance. As illustrated in Fig. 4, this loss successfully enables SE(3)equivariance across objects.

C. Pose optimization

When using L-NDFs for few shot task learning, we must optimize a pose, $\mathbf{T}$, on a new point cloud $\mathbf{P}$, to match a desired reference pose, $\mathbf{T}^*$ on a reference point cloud $\mathbf{P}$. This optimization procedure is described in (3). Conventional NDFs run global optimization on a set of query points to obtain the optimal pose $\mathbf{T}$, where optimization is initialized at a random orientation centered at the origin of the object.

However, this method fails when using Local NDFs. Since L-NDFs only aggregate information across local geometry, there is little information relating distant geometric features.

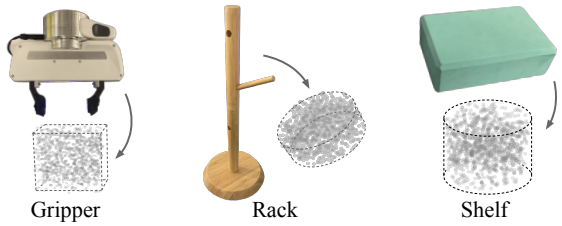


Fig. 5: **Selecting Query Points** – Relative size of query points for each executed task. For grasp and rack placement tasks, we use query points similar in size to contact geometry of the known object (gripper and peg size). For placement surfaces, we find larger query point selections performs well.

To mitigate these challenges, we introduce two techniques: initial translation and query point selection.

Initial translation. In contrast to conventional NDFs, we initialize query points at random rotations and translations within the bounding box of the observed point cloud. When using a sufficient number of query points instances (We found 20 to be adequate), we find that at least one of the translated query point sets will initialize close to our target geometric feature. Subsequent pose optimization tunes the query point cloud to the correct target location.

Query Point Tuning. We find that query point selection is critical to the performance of L-NDFs. If a query point cloud is too large, it encodes confounding geometry and empty space. If a query point cloud is too small, it does not capture enough local geometry. We find that for precise manipulation, query points can be sampled near the expected contact geometry of the known object. For more general poses (such as placing on a surface), a query point cloud which maximizes the expected volume of observed objects contained within the point cloud while minimizing the volume of empty space contained produces robust results. See Fig. 5 for additional details.

V. Experiments: Design and Setup

We design our experiments to test the following: (1) How well do L-NDF's generalize to unseen objects classes? (2) Can L-NDF's be used on a real robot to achieve generalization from a small number of single object class demonstrations?

A. Robot Environment Setup

Our environment consists of a Franka Panda arm mounted on a table. Depth cameras are placed at each corner of the table, all calibrated to obtain fused point clouds of objects within the robot's reach. We use four depth cameras in simulation, and two depth cameras in real life. Our simulation cameras produce a complete point cloud, while the real life cameras produce a partial point cloud. Depending on the task, a rack or shelf is placed on the table. For quantitative data, this setup is simulated in Pybullet [3]. For our simulation setup, refer to Fig. 6. For our real world setup, refer to Fig. 7.

B. Task Setup

We test four tasks: (1) Grasping a mug-like object by its rim and hanging it on a rack by its handle. (2) Grasping a bowl-like object by its rim and placing it upright on a shelf. (3) Grasping a bottle-like object by its neck and placing it upright on a shelf. (4) Grasping a handle placed on an object from an arbitrary object class. Tasks 1, 2, and 3 use demos containing normal mugs, bowls, and bottles, respectively. Task 4 uses demos of normal mugs.

We define mug-like objects as standard mugs and bowls with handles attached to them; bowl-like objects as standard bowls, standard mugs, and bowls with handles attached to them; and bottle-like objects as standard bottles and bottles with handles attached to them.

We provide 10 demonstrations per task and test on 200 unseen objects at randomly generated poses, orientations, and uniform scalings. We assume the environment remains static between demonstrations and test and that (potentially partial) point clouds of the object can be obtained. In simulation, we use Shapenet [2] objects for each in-distribution class, filtering objects that are incompatible with our tasks. For out-of-distribution objects, we modified Shapenet objects as required. Refer to Fig. 6 for examples.

C. Training Details

We pre-train NDFs and L-NDF's by using each system's occupancy network to reconstruct objects from partial depth images. We train each system for 300,000 iterations on a joint dataset containing objects from all three object categories at random rotations and translations. For each object, point cloud data is gathered by placing the object in a PyBullet simulation and taking depth images.

At test time, we gather a small number (10 in simulation and 4-6 in real life) of task specific demonstrations using a single object class. These demonstrations are then used by the systems to execute the desired tasks on the demonstration object class, as well as on unseen object classes.

D. Evaluation Metrics

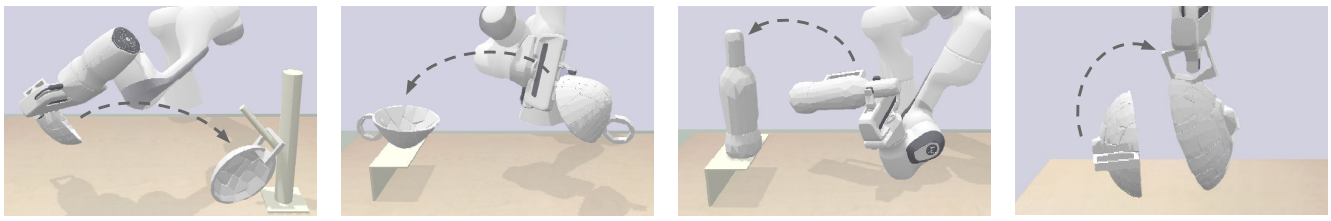
In simulation, we evaluate each method by measuring grasp success (stable contact between object and end effector), place success (stable contact with placement surface in the correct orientation), and overall task success, for which both grasp success and place success must have occurred. On the physical robot, human evaluators assert whether the object has been grasped and placed in the correct location.

E. Baselines

We compare L-NDF performance to conventional NDFs on each of four tasks. The L-NDF query points were selected using the heuristics described above. NDF query points are extracted from the codebase provided by [21].

VI. Experiments: Results

We conduct experiments in simulation to compare the performance of NDFs and L-NDF's on each of the four tasks (illustrated in Fig. 6) with relevant in and out of distribution objects. We then perform ablation studies to examine the effect of different loss functions and different 3D feature volumes on L-NDF performance. Finally, we apply L-NDFs on a physical robot and validate that the proposed method generalizes to out-of-distribution poses and objects classes in the real world.



"Mug" on Rack "Bowl" on Shelf "Bottle" on Shelf "Mug" Handle Grasp

Fig. 6: **Experimental Setup** – We provide ten simulated demonstrations of each task, then execute each on a set of 200 unseen objects. We measure grasp success, place success, and overall success. Grasp and place success check that the simulated object is in a stable configuration. Overall success checks if both grasp and place success occurred.

Upright Pose		Mug Demo			Bowl Demo			Bottle Demo		Mug Handle Demo			
		Mug	Bowl*	Bottle*	Bowl	Bowl*	Mug	Bottle	Bottle*	Mug	Bowl*	Bottle*	Bowl
NDF	Grasp	1.000	0.615	0.010	0.925	0.725	0.265	0.805	0.695	0.805	0.305	0.235	0.000
	Place	0.925	0.620	0.225	0.910	0.730	0.145	0.935	0.870	-	-	-	-
	Overall	0.925	0.450	0.000	0.885	0.670	0.125	0.805	**0.665**	0.805	0.305	0.235	0.000
L-NDF	Grasp	1.000	0.950	0.160	0.990	0.985	0.970	0.875	0.760	0.980	0.730	0.915	0.190
	Place	0.995	0.830	0.900	0.990	0.990	0.865	0.975	0.670	-	-	-	-
	Overall	**0.995**	**0.800**	**0.135**	**0.985**	**0.975**	**0.845**	**0.850**	0.590	**0.980**	**0.730**	**0.915**	**0.190**
Arbitrary Pose		Mug Demo			Bowl Demo			Bottle Demo		Mug Handle Demo			
		Mug	Bowl*	Bottle*	Bowl	Bowl*	Mug	Bottle	Bottle*	Mug	Bowl*	Bottle*	Bowl
NDF	Grasp	0.900	0.460	0.045	0.675	0.575	0.150	0.575	0.385	0.555	0.105	0.190	0.070
	Place	0.735	0.370	0.235	0.840	0.800	0.565	0.955	0.955	-	-	-	-
	Overall	0.655	0.250	0.010	0.655	0.565	0.120	0.570	0.365	0.555	0.105	0.190	0.070
L-NDF	Grasp	0.770	0.755	0.110	0.910	0.960	0.880	0.790	0.720	0.930	0.540	0.815	0.130
	Place	0.960	0.635	0.850	0.985	0.940	0.885	0.970	0.820	-	-	-	-
	Overall	**0.735**	**0.470**	**0.095**	**0.905**	**0.820**	**0.795**	**0.775**	**0.635**	**0.930**	**0.540**	**0.815**	**0.130**

TABLE I: **Unseen instance pick-and-place success rates in simulation.** Given demonstrations using a single object class, we test performance on a variety of other object classes. NDF performs well on unseen objects from the demonstration object class but struggles with new object classes. L-NDF performs well with unseen objects from both the demonstration and analogous object classes at upright and arbitrary rotations. Green indicates that the test object is the same class as the demonstrations; blue indicates that the test object is from an analogous class to the demonstrations; red indicates that the test object is from a substantially different class. *Objects are modified to include a handle. See illustrations of each task in Fig. 6.

A. Simulation Experiments

In-distribution objects. We first consider how skills are transferred to unseen objects from the demonstration class in novel upright or arbitrarily rotated poses. Referring to the green columns of Table I, we find that in all pick and place tasks, L-NDFs outperform conventional NDFs – sometimes in excess of a 0.25 increase in success rate. Furthermore, we find that L-NDFs dramatically outperform NDFs on handle grasping, achieving a 0.38 improvement over NDFs in task success on arbitrarily rotated mug handles (Table I, last green column). We observe that NDF's handle grasp failures occur when a grasp is found near the desired location, but at a slight offset or rotation from the expected location. Given the fully connected nature of NDFs, we hypothesize that the descriptor fields near an observed object's salient features may be confounded by the irrelevant geometry of the object itself, an issue which local fields address.

Analogous Out-of-distribution objects. We next consider a more difficult task. We still wish to transfer skills from demonstrations to test objects at novel upright or arbitrarily rotated poses. However, now the test objects have analogous geometry to the demonstration objects, but in different arrangements or with confounding features. Referring to the first and last blue columns of Table I, we find that on tasks where the rearranged geometry is integral to the task success, NDF performance drops substantially. In contrast, L-NDF performance does fall, but significantly less than NDF does. In many cases, we observe that tuning NDF query points to more closely match L-NDF query points can recoup some of this performance loss. However, we still find that NDF performance lags behind L-NDF success.

In the middle three blue columns of Table I, we find that, in tasks where the additional feature acts as a confounding feature, NDF and L-NDF overall task success drops by similar amounts. We note that in NDFs, this drop in performance is attributed to both drops in both grasp and placement success. However, with L-NDFs, this drop is mostly attributed to a decrease in placement success. We hypothesize that, while grasping is a highly local task – only concerned with the location of the manipulator fingers; placement reflects a global task where the orientation of an object is largely defined by its aggregate geometry. Thus, the advantages of using a local field are diminished in placement and confounding features still affect performance.

Substantially Different Out-of-distribution objects. Finally, we test the limits of L-NDF's generalization capabilities by testing on objects that are substantially different from the demonstration object class. Of particular interest is placing a bottle with handle on a rack, given mug demos, and grasping

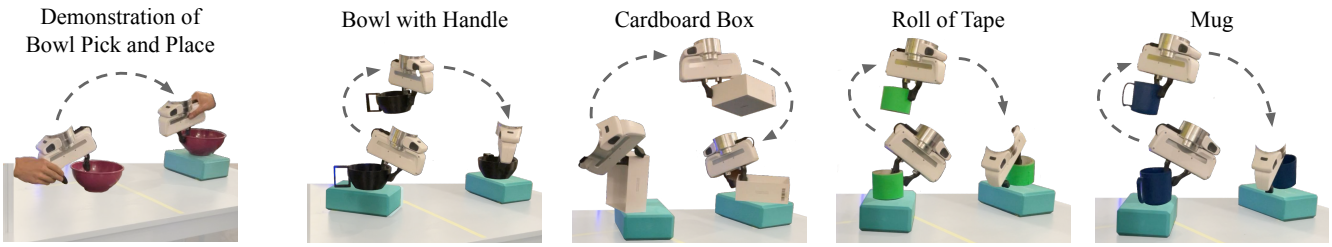


Fig. 7: **Real world Execution** – We provide four real world demonstrations of grasping and placing two different bowls. We then successfully grasp and place a variety of unseen objects using a Franka Panda arm. Refer to our supplementary video for additional results.

Random L-NDF			Occupancy Only			Hard Contrast			Distance Contrast		
G	P	O	G	P	O	G	P	O	G	P	O
0.02	0.73	0.02	0.70	0.66	0.47	0.64	0.63	0.39	0.79	0.97	0.78

TABLE II: **Effect of Loss Function.** We test a randomly initialized system and systems trained with pure 3d reconstruction, simple contrastive loss, and our distance based contrastive loss.

32^3			64^3			128^3		
Grasp	Place	Overall	Grasp	Place	Overall	Grasp	Place	Overall
0.63	0.90	0.56	0.77	0.96	0.75	0.79	0.97	0.78

TABLE III: **Effect of 3D Feature Volume Size.** We examine the effect of 3D feature volume size (in voxels) on L-NDF performance. All systems are trained using our distance based contrastive loss

the "handle" of a bowl with no handle (shown in the red columns of Table I). In these extreme cases, we find that NDFs fail completely, achieving negligible success. L-NDFs fare slightly better, achieving between 10% and 20% success. Interestingly, L-NDFs achieved above 80% place success on bottles with handles. As expected, these overall success rates are unsuitable for general robotic manipulation, but suggest that local fields may be a promising direction to explore for more general robotic manipulation.

B. Ablation Analysis

Next, we run an ablation study on L-NDF using the arbitrary rotation bottle placement task.

Loss Function. First, we analyze the impact of the loss function on L-NDF performance. In Table II, we find that a random network achieves negligible grasp success and subpar place success. This indicates that pretraining L-NDF is important. A simple contrastive loss function where similar points have ground truth similarity of 1 and different points have ground truth similarity of 0 performs poorly as well. We hypothesize that enforcing this sort of loss incorrectly describes our objectives for the network, as different example points should, intuitively, have different costs. Solely training on reconstructive tasks performs better than simple contrastive loss, but yields poor performance at arbitrary rotations. However, our distance based contrastive loss dramatically improves on both methods, enforcing SE(3) equivariance while preserving reconstruction quality.

3D Feature Volume Size. We next analyze the impact of voxel size on L-NDF performance. Referring to Table III, we find that task success monotonically increases with 3D feature volume size. Increasing the feature volume from 32^3 voxels to 64^3 voxels produces a dramatic improvement, while increasing from 64^3 voxels to 128^3 voxels produces increased success, but at a diminishing rate. We elect to use the 128^3 voxel system as it ran in similar time to the 64^3 voxel while providing slightly higher success rates.

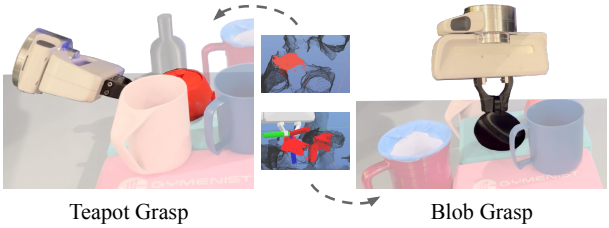


Fig. 8: **Operating in Clutter** – We provide four real world demos of grasping a mug in an uncluttered scene. We then grasp a variety of objects from a cluttered environment using partial point clouds. We used Mask R-CNN [9] for scene segmentation. Please see our supplementary video for additional results.

C. Real world

Finally, we evaluate our system in a real world environment. We collect 5-10 task demonstrations for handle grasping and bowl pick and place using upright objects, then evaluate our system on a variety of unseen objects in arbitrary poses. Additionally, we evaluate our system in a cluttered environment, using Mask R-CNN [9] for scene segmentation and L-NDF for pose estimation. As can be seen in Fig. 8, the resultant point clouds from scene segmentation are often incomplete and noisy, yet LNDF successfully deduces object pose. Please see Fig. 1 and Fig. 7 for our single object trials, Fig. 8 for our evaluation in cluttered environments, and our supplementary video for additional details and qualitative results.

VII. Conclusion

We introduce Local Neural Descriptor Fields, an object representation that allow few-shot imitation learning of manipulation tasks on potentially novel categories of shapes at test time. We illustrate the capability of our work to exhibit strong generalization – given only examples of grasping the handle of a mug, we can generalize to shapes such as teacups or bottles in both simulation and the real world.

VIII. Acknowledgement

We gratefully acknowledge support from NSF grant 2214177; from AFOSR grant FA9550-22-1-0249; from ONR MURI grant N00014-22-1-2740; from ARO grant W911NF-23-1-0034; from the MIT-IBM Watson Lab; and from the MIT Quest for Intelligence.

REFERENCES

[1] Michal Adamkiewicz et al. "Vision-only robot navigation in a neural radiance world". In: *RA-L*. 2022.

[2] Angel X Chang et al. "Shapenet: An information-rich 3d model repository". In: *arXiv preprint arXiv:1512.03012* (2015).

[3] Erwin Coumans and Yunfei Bai. "Pybullet, a python module for physics simulation for games, robotics and machine learning". In: *GitHub repository* (2016).

[4] Peter R Florence, Lucas Manuelli, and Russ Tedrake. "Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation". In: *Conference on Robot Learning*. PMLR. 2018, pp. 373–385.

[5] Jiahui Fu et al. "Robust Change Detection Based on Neural Descriptor Fields". In: *arXiv preprint arXiv:2208.01014* (2022).

[6] Wei Gao and Russ Tedrake. "kPAM 2.0: Feedback Control for Category-Level Robotic Manipulation". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2962–2969.

[7] Wei Gao and Russ Tedrake. "kPAM-SC: Generalizable Manipulation Planning using KeyPoint Affordance and Shape Completion". In: *arXiv preprint arXiv:1909.06980* (2019).

[8] Kensuke Harada et al. "Probabilistic approach for object bin picking approximated by cylinders". In: *2013 IEEE International Conference on Robotics and Automation*. IEEE. 2013, pp. 3742–3747.

[9] Kaiming He et al. "Mask R-CNN". In: *CoRR* abs/1703.06870 (2017). arXiv: 1703.06870.

[10] Jeffrey Ichnowski* et al. "Dex-NeRF: Using a Neural Radiance field to Grasp Transparent Objects". In: *CoRL*. 2020.

[11] Zhenyu Jiang et al. "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations". In: *RSS*. 2021.

[12] Lucas Manuelli et al. "kpam: Keypoint affordances for category-level robotic manipulation". In: *arXiv preprint arXiv:1903.06684* (2019).

[13] Lars Mescheder et al. "Occupancy Networks: Learning 3D Reconstruction in Function Space". In: *Proc. CVPR*. 2019.

[14] Andrew T Miller et al. "Automatic grasp planning using shape primitives". In: *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. Vol. 2. IEEE. 2003, pp. 1824–1829.

[15] Arthur Moreau et al. "LENS: Localization enhanced by NeRF synthesis". In: *Conference on Robot Learning*. 2022.

[16] Joseph Ortiz et al. "iSDF: Real-Time Neural Signed Distance Fields for Robot Perception". In: *RSS*. 2022.

[17] Jeong Joon Park et al. "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation". In: *Proc. CVPR*. 2019.

[18] Songyou Peng et al. "Convolutional occupancy networks". In: *Proc. ECCV*. 2020.

[19] Hyunwoo Ryu et al. "Equivariant Descriptor Fields: SE (3)-Equivariant Energy-Based Models for End-to-End Visual Robotic Manipulation Learning". In: *arXiv preprint arXiv:2206.08321* (2022).

[20] John Schulman et al. "Learning from Demonstrations Through the Use of Non-rigid Registration". In: *Robotics Research: The 16th International Symposium ISRR*. Ed. by Masayuki Inaba and Peter Corke. Cham: Springer International Publishing, 2016, pp. 339–354.

[21] Anthony Simeonov et al. "Neural Descriptor Fields: SE (3)-Equivariant Object Representations for Manipulation". In: *arXiv preprint arXiv:2112.05124* (2021).

[22] Anthony Simeonov et al. "SE(3)-Equivariant Relational Rearrangement with Neural Descriptor Fields". In: *Conference on Robot Learning (CoRL)* (2022).

[23] Edgar Sucar et al. "iMAP: Implicit Mapping and Positioning in Real-Time". In: *ICCV*. 2021.

[24] Priya Sundaresan et al. "Learning Rope Manipulation Policies Using Dense Object Descriptors Trained on Synthetic Depth Data". In: *arXiv preprint arXiv:2003.01835* (2020).

[25] Skye Thompson, Leslie Pack Kaelbling, and Tomas Lozano-Perez. "Shape-Based Transfer of Generic Skills". In: *Proc. of The International Conference in Robotics and Automation (ICRA)*. 2021.

[26] Lin Yen-Chen et al. "iNeRF: Inverting Neural Radiance Fields for Pose Estimation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021.

[27] Lin Yen-Chen et al. "NeRF-Supervision: Learning Dense Object Descriptors from Neural Radiance Fields". In: *ICRA*. 2022.

[28] Youngrock Yoon, Guilherme N DeSouza, and Avinash C Kak. "Real-time tracking and pose estimation for industrial objects using geometric features". In: *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. Vol. 3. IEEE. 2003, pp. 3473–3478.

[29] Menglong Zhu et al. "Single image 3D object detection and pose estimation for grasping". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 3936–3943.

[30] Zihan Zhu et al. "Nice-slam: Neural implicit scalable encoding for slam". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12786–12796.