

# Neural Descriptor Fields: SE(3)-Equivariant Object Representations for Manipulation

Anthony Simeonov<sup>\*,1</sup>, Yilun Du<sup>\*,1</sup>, Andrea Tagliasacchi<sup>2,3</sup>,  
Joshua B. Tenenbaum<sup>1</sup>, Alberto Rodriguez<sup>1</sup>, Pulkit Agrawal<sup>†,1</sup>, Vincent Sitzmann<sup>†,1</sup>  
<sup>1</sup>Massachusetts Institute of Technology <sup>2</sup>Google Research <sup>3</sup>University of Toronto  
\*Authors contributed equally, order determined by coin flip. <sup>†</sup>Equal Advising.

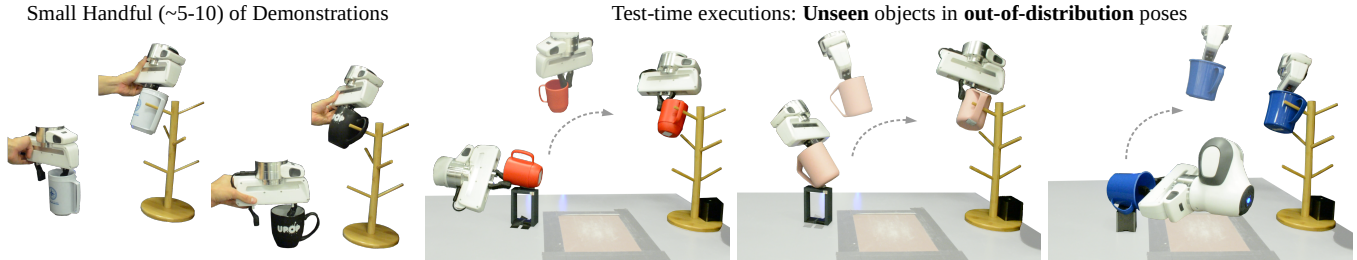


Fig. 1: Given a few ( $\sim 5-10$ ) demonstrations of a manipulation task (left), Neural Descriptor Fields (NDFs) generalize the task to novel object instances in any 6-DoF configuration, *including those unobserved at training time*, such as mugs with arbitrary 3D translation and rotation (right). NDFs are continuous functions that map 3D spatial coordinates to spatial descriptors. We generalize this to functions which encode SE(3) poses, such as those used for grasping and placing. NDFs are trained self-supervised for the surrogate task of 3D reconstruction, do not require labeled keypoints, and are SE(3)-equivariant, guaranteeing generalization to unseen object configurations.

**Abstract**—We present Neural Descriptor Fields (NDFs), an object representation that encodes both points and relative poses between an object and a target (such as a robot gripper or a rack used for hanging) via category-level descriptors. We employ this representation for object manipulation, where given a task demonstration, we want to repeat the same task on a *new* object instance from the *same* category. We propose to achieve this objective by searching (via optimization) for the pose whose descriptor matches that observed in the demonstration. NDFs are conveniently trained in a self-supervised fashion via a 3D auto-encoding task that does not rely on expert-labeled keypoints. Further, NDFs are SE(3)-equivariant, guaranteeing performance that generalizes across all possible 3D object translations *and* rotations. We demonstrate learning of manipulation tasks from few ( $\sim 5-10$ ) demonstrations both in simulation and on a real robot. Our performance generalizes across both object instances and 6-DoF object poses, and significantly outperforms a recent baseline that relies on 2D descriptors. Project website: <https://yilundu.github.io/ndf/>

## I. INTRODUCTION

Task demonstrations are an intuitive and a powerful mechanism for communicating complex tasks to a robot [1, 30, 35]. However, the ability of current methods to learn from demonstrations is severely limited. Consider the task of teaching the robot to pick up a mug and place it on a rack. After learning, if we want the robot to place a novel instance of a mug from any starting location and orientation, state-of-the-art systems would require a large number of demonstrations spanning the space of different initial positions, orientations and mug instances. This requirement makes it extremely tedious to communicate tasks using demonstrations. Moreover, this approach based on data augmentation comes with no algorithmic guarantees to generalization to out-of-distribution

object configurations.

Our goal is to build a robotic system that can learn such pick-and-place tasks for unseen objects in a data-efficient manner. In particular, we desire to construct a system which can manipulate objects from the same category into target configurations, irrespective of the object’s 3D location and orientation (see Figure 1) from just a few training demonstrations ( $\sim 5 - 10$ ).

Consider the task of picking a mug. When task and demonstration objects are identical, the robot can pick up the object by transferring the demonstrated grasp to the new object configuration. For this it suffices to attach a coordinate frame to the demonstration mug, estimate the pose of this frame on the new mug, and move the robot to the relative grasp pose that was recorded in the demonstration with respect to the coordinate frame. Let us now consider mugs that vary in shape and size, wherein grasping requires aligning the gripper to a *local* geometric feature whose location *varies* depending on the shape of the mug. In this case, estimating the coordinate frame on the new mug and moving to the relative grasp pose recorded in the demonstration will fail, *unless* the frame is attached to the specific geometric feature that is used for grasping. However, the choice of which geometric feature to use is *under-specified* unless we consider the task, and different tasks require alignment to *different* features.

For example, to imitate *grasping* along the rim, we may require to define a local frame such that identical gripper poses expressed in this frame all lead to grasping along the rim, irrespective of the height of the mug. On the other hand, imitating a demonstration of object *placing* may require a *new* coordinate frame that can align a placing target (e.g., a shelf)

to the bottom surface of the mugs. These examples elucidate that the relevant geometric structure for alignment is *task-specific*. Having identified the task-relevant geometric feature, we can attach a coordinate frame and measure the pose of the gripper/placing target in task demonstrations relative to this feature on different object instances. Given a new object instance, the task can be performed by first identifying the local coordinate frame on the object and then obtaining a relative grasping/placing pose in this coordinate frame that is consistent with the demonstrations.

The two key questions to be answered in this process are: (1) how to specify the relevant feature and local frame for a given task; and (2) how to solve for the corresponding frame given a new object instance. Prior approaches address these questions by hand-labeling a large dataset of task-specific keypoints and training a neural network to predict their location on new instances [12, 20]. Detected keypoints are then used to recover a local coordinate frame. However, collecting this dataset for each task is expensive, and these methods fail to generalize to new instances in the regime of few demonstrations. To mitigate the generalization issue, other prior work first learns to model *dense* point correspondence in a task-agnostic fashion [10]. At test time, human-annotated keypoints are individually and independently corresponded one-by-one, and the local coordinate frame is established via registration to the keypoints on the demonstration instance. This enables imitation from few demonstrations, but current approaches—which operate in 2D—suffer several key limitations. (i) Keypoints may only lie on the surface of the object, making it difficult to encode important free-space locations (i.e., in the center of a handle). Further, if the object is partially occluded, keypoint locations cannot be inferred. (ii) Small errors in estimating the corresponding location of each keypoint can result in large errors in solving for the transform and consequently the resulting coordinate frame. (iii) Existing methods are not equivariant to SE(3) transformations and thus not guaranteed to provide correct correspondence when instances are in unseen poses. (iv) Human keypoint annotation is required to identify task-specific features.

We propose a novel method to encode dense correspondence across object instances, dubbed Neural Descriptor Fields (NDF), that effectively overcomes the limitations of prior work: (i) We represent an object point cloud  $\mathbf{P}$  as a continuous function  $f(\mathbf{x}|\mathbf{P})$  that maps any 3D coordinate  $\mathbf{x}$  to a spatial descriptor. Descriptors encode the spatial relationship of  $\mathbf{x}$  to the salient geometric features of the object in a way that is consistent across different shapes in a category of objects. Coordinates are not constrained to be on the object and can potentially be occluded.

(ii) We represent a coordinate frames associated with a local geometric structure using a *rigid set* of query points. The configuration of these points is represented as an SE(3) pose with respect to a canonical pose in the world frame. For each object instance in the demonstration, the query point set is converted into a set of feature descriptors by concatenating point descriptors of all the points. The feature representation resulting from evaluating query points at different SE(3)

transformations forms what we call a *pose descriptor field*. To estimate the pose of the local coordinate frame on a new object instance, we optimize for the SE(3) transformation of the query points that minimizes the distance of feature descriptors with those of the demonstration objects. This process solves for feature matching and the coordinate frame’s pose jointly, instead of the two-step process employed by prior work which is more prone to errors. Furthermore, because query points are defined in 3D (as opposed to 2D keypoints) and it is not necessary that location of all query points is observed, the proposed procedure for finding coordinate frame is more robust and has higher accuracy than existing methods.

(iii) To guarantee that we can successfully estimate the local frame for all 6-DoF configurations of the test-time object instance (i.e., generalization), we construct the pose descriptor fields to be equivariant to SE(3) transformations. For this we leverage recent progress in geometric deep learning [7]. (iv) Finally, we devise a procedure for using the demonstrations to obtain the set of query points, such that the pose descriptor - and thus, the recovered coordinate frame - is sensitive to task-relevant local geometric features, overcoming the need for human-annotated keypoints.

Using this novel formulation, we propose a system that can imitate pick-and-place tasks for a category of objects from only a small handful of demonstrations. On three unique pick-and-place tasks, Neural Descriptor Fields enables both pick and place of unseen object instances in out-of-distribution configurations with an overall success rate above 85%, using only 10 expert demonstrations and consistently outperforms baselines that operate in 2D and are not SE(3) equivariant.

## II. METHOD

We present a novel representation that models dense correspondence across object instances at the level of points and local coordinate frames. Our representation enables an intuitive mechanism for specifying a task-relevant local frame using a demonstration task and point cloud  $\hat{\mathbf{P}}$ , along with the efficient and robust computation of a corresponding local frame when presented with a new point cloud  $\mathbf{P}$ .

In Section II-A, we introduce a continuous function  $f(\mathbf{x}|\mathbf{P})$  that maps a 3D coordinate  $\mathbf{x}$  and a point cloud  $\mathbf{P}$  to a spatial descriptor that encodes information about the spatial relationship of  $\mathbf{x}$  to the category-level geometric features of the object. We demonstrate that we can represent this function using a neural network trained in a task-agnostic manner via 3D reconstruction, and that this training objective learns descriptors that encode point-wise correspondence across a category of shapes. We furthermore show how we may equip these point descriptor fields with SE(3)-equivariance, enabling correspondence matching across object instances in arbitrary SE(3) poses. In Section II-B, we leverage these point descriptors to establish correspondence for a rigid *set* of points, whose configuration is used to parameterize a local coordinate frame near the object. This enables us to *directly solve for the SE(3) pose* of the transformed point set whose descriptors best match a reference descriptor set, and recover the corresponding local frame relative to a new object.

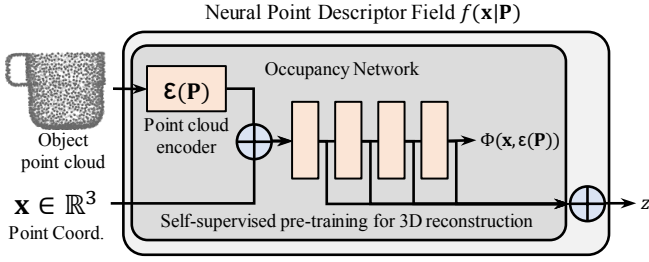


Fig. 2: **Point Descriptor Fields** – We propose to parameterize a Neural Point Descriptor Field  $f$  as the concatenation of the layer-wise activations of an occupancy network  $\Phi(\mathbf{x}, \mathcal{E}(\mathbf{P}))$ . Both the point cloud encoder and the point descriptor function can be pre-trained with a 3D reconstruction task.

We then discuss how to apply this novel representation for transferring grasp and place poses from a set of pick-and-place demonstrations: We first show how contact interactions between the manipulated object and known external rigid bodies (such as a gripper, rack, or shelf) can be used to sample query points near important geometric features, yielding descriptors for *task-relevant* local reference frames directly from demonstrations. Finally, in Section II-C, we show how we use pose descriptor fields and a small handful of demonstrations to reproduce a pick-and-place task on a new object in an arbitrary initial pose.

#### A. Neural Point Descriptor Fields

Our key idea is to represent an object as a function  $f$  that maps a 3D coordinate  $\mathbf{x}$  to a spatial descriptor  $z = f(\mathbf{x})$  of that 3D coordinate:

$$f(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}^n \quad (1)$$

$f$  may further be conditioned on an object point cloud  $\mathbf{P} \in \mathbb{R}^{3 \times N}$  to output *category-level* descriptors  $f(\mathbf{x}|\mathbf{P})$ . We propose to parameterize  $f$  via a neural network. This yields a *differentiable* object representation that continuously maps *every* 3D coordinate to a spatial descriptor. As we will see, this continuous, differentiable formulation enables us to find correspondence across object instances via simple first-order optimization. Finally, it remains to learn the weights of a neural descriptor field. On first glance, this would require setting up a training objective for correspondence matching, and consequently, collection and labeling of a custom dataset. Instead, we propose and demonstrate that we may leverage recently proposed neural implicit shape representations [5, 21, 27] to parameterize  $f$  and learn its weights in a self-supervised manner.

**Background: neural implicit.** Neural implicit representations represent the 3D surface of a shape as the level-set of a neural network. In particular, Mescheder et al. [21] represent a 3D shape as an MLP  $\Phi$  that maps a 3D coordinate  $\mathbf{x}$  to its occupancy value:

$$\Phi(\mathbf{x}) : \mathbb{R}^3 \rightarrow [0, 1] \quad (2)$$

We are interested in learning a low-dimensional latent space of 3D shapes, which can be achieved by parameterizing the latent space with a latent code  $\mathbf{v} \in \mathbb{R}^k$  and concatenating it with  $\mathbf{x}$ , encoding different shapes via different latent codes. These latent codes are obtained as the output of a PointNet [32]-based point cloud encoder  $\mathcal{E}$  that takes as input a point

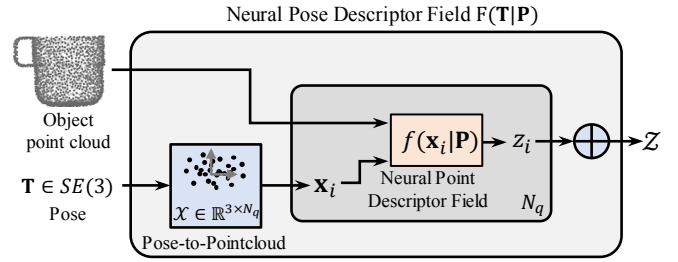


Fig. 3: **Pose Descriptor Fields** – NDFs can extract *pose* descriptors by representing a pose via its action on a *query pointcloud*  $\mathcal{X}$ , and then extracting point-level spatial descriptors  $z_i$  for each point  $\mathbf{x}_i$  with a point-level NDF. Concatenation then yields the final pose descriptor  $Z$ .

cloud  $\mathbf{P}$ , leading to a conditional occupancy function:

$$\Phi(\mathbf{x}, \mathcal{E}(\mathbf{P})) : \mathbb{R}^3 \times \mathbb{R}^k \rightarrow [0, 1] \quad (3)$$

The full model can be trained end-to-end on a dataset of partial point clouds and corresponding occupancy voxelgrids of the objects’ full 3D geometry, thus learning to predict the occupancy of a complete 3D object from a *partial* pointcloud. This is an attractive property, as at test time, we regularly only observe partial point clouds of objects due to occlusions.

**Neural feature extraction – Fig. 2.** To enable category-level object manipulation, a spatial descriptor for a coordinate  $\mathbf{x}$  given a point cloud  $\mathbf{P}$  should encode information about the spatial relationship of  $\mathbf{x}$  to the salient features of the object. That is, for mugs, descriptors should encode information about how far  $\mathbf{x}$  is away from the mug’s handle, rim, etc.

Our key insight is that the category-level 3D reconstruction objective trains  $\Phi(\mathbf{x}, \mathcal{E}(\mathbf{P}))$  to be a hierarchical, coarse-to-fine feature extractor that encodes exactly this information:  $\Phi$  is a classifier whose decision boundary is the surface of the object. Intuitively, each layer of  $\Phi$  is a set of ReLU hyperplanes that are trained to encode *how far a given coordinate  $\mathbf{x}$  is away from this decision boundary*, such that ultimately, the final layer may classify it as *inside* or *outside* the shape, where layers encode increasingly finer surface detail. The output of the pointcloud encoder  $\mathcal{E}(\mathbf{P})$  in turn determines *where* this decision boundary lies in terms of a small set of latent variables. This bottleneck forces the model to use these few latent variables to parameterize the salient features of the object category, which is impressively demonstrated by smooth latent-space interpolations and unconditional shape samples [4, 21, 27]. Prior work has leveraged this property of the activations of  $\Phi$  to classify which semantic part of an object a given coordinate  $\mathbf{x}$  belongs to [17], a task which is closely related to modeling correspondence across a category.

We thus propose to parameterize our neural point descriptor field  $f(\mathbf{x}|\mathbf{P})$  as the function that maps every 3D coordinate  $\mathbf{x}$  to the *vector of concatenated activations* of  $\Phi$ :

$$f(\mathbf{x}|\mathbf{P}) = \bigoplus_{i=1}^L \Phi^i(\mathbf{x}, \mathcal{E}(\mathbf{P})) \quad (4)$$

with the activation of the  $i$ th layer as  $\Phi^i$ , total number of layers  $L$ , and concatenation operator  $\bigoplus$ . We choose to concatenate activations across layers to encourage consideration of features across scales and ablate this effect in Table II.

**Equivariance w.r.t.  $SE(3)$ .** A key requirement of our descrip-

tor field is to ensure descriptors remain constant if the position of  $\mathbf{x}$  relative to  $\mathbf{P}$  remains constant, regardless of their global configuration in the world coordinate system. In other words, we require  $f$  to be *invariant* to joint transformation of  $\mathbf{x}$  and  $\mathbf{P}$ , implying the descriptor field should be *equivariant* to  $SE(3)$  transformations of  $\mathbf{P}$  – we wish that if an object is subject to a rigid body transform  $(\mathbf{R}, \mathbf{t}) \in SE(3)$  its spatial descriptors transform accordingly:

$$f(\mathbf{x}|\mathbf{P}) \equiv f(\mathbf{R}\mathbf{x} + \mathbf{t}|\mathbf{R}\mathbf{P} + \mathbf{t}). \quad (5)$$

Translation equivariance is conveniently implemented by subtracting the center of mass of the point cloud from both the input point cloud and the input coordinate. We thus re-define  $f(\mathbf{x}|\mathbf{P})$  as:

$$f(\mathbf{x}|\mathbf{P}) = f(\mathbf{x} - \mu|\mathbf{P} - \mu); \quad \mu = \frac{1}{N} \sum_{i=1}^N \mathbf{P}_i \quad (6)$$

This results in the input to  $f$  always being zero-centered, irrespective of the absolute position of  $\mathbf{P}$ , making  $f$  invariant to joint translations of  $\mathbf{x}$  and  $\mathbf{P}$ . To achieve *rotation* equivariance, we rely on recently proposed Vector Neurons [7], which propose a network architecture that equips an occupancy network, i.e., the composition of  $\mathcal{E}$  and  $\Phi$  in (3), with full  $SO(3)$  equivariance. By replacing  $\Phi(\mathbf{x}, \mathcal{E}(\mathbf{P}))$  in (4) with this  $SO(3)$ -equivariant architecture,  $f$  immediately inherits this property, such that for  $\mathbf{R} \in SO(3)$ :

$$f(\mathbf{x}|\mathbf{P}) \equiv f(\mathbf{R}\mathbf{x}|\mathbf{R}\mathbf{P}) \quad (7)$$

Combining this with the pointcloud mean-centering scheme yields complete  $SE(3)$  equivariance — i.e.,  $f$  now enjoys a *guarantee* that transforming an input pointcloud by *any*  $SE(3)$  transform will transform the locations of spatial descriptors accordingly, leaving them unchanged otherwise. This guarantees that we can generalize to arbitrary object poses, including those completely unobserved at training time.

**Validation – Fig. 4 and Fig. 5.** To validate the effectiveness of our descriptor fields, let us consider the following energy field:

$$E(\mathbf{x}|\hat{\mathbf{P}}, \mathbf{P}, \hat{\mathbf{x}}) = \|f(\hat{\mathbf{x}}|\hat{\mathbf{P}}) - f(\mathbf{x}|\mathbf{P})\| \quad (8)$$

with its minimizer

$$\bar{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} E(\mathbf{x}|\hat{\mathbf{P}}, \mathbf{P}, \hat{\mathbf{x}}). \quad (9)$$

As shown in Fig. 4, given a reference point cloud  $\hat{\mathbf{P}}$  and a reference point  $\hat{\mathbf{x}}$ , the minimizer  $\bar{\mathbf{x}}$  of Eq. 9 transfers the location of the reference point  $\hat{\mathbf{x}}$  to the test-time object  $\mathbf{P}$ . In Fig. 5, we plot this energy for a reference point on the handle of a reference mug across different mug poses and instances. The colors in the plot reflect that high-energy regions are far from the handle, whereas the energy decreases at positions closer to the handle. We subsequently find that the transferred point  $\bar{\mathbf{x}}$  at the minimum of this energy field correctly corresponds to points on the handles across the different mugs, irrespective of their configuration. This validates that  $f$  may transfer across object instances and generalize across  $SE(3)$  configurations.

### B. Neural Pose Descriptor Fields

The previous section discussed how NDFs induce an energy that can be minimized for transferring points across object instances. However, in manipulation tasks, we need to solve

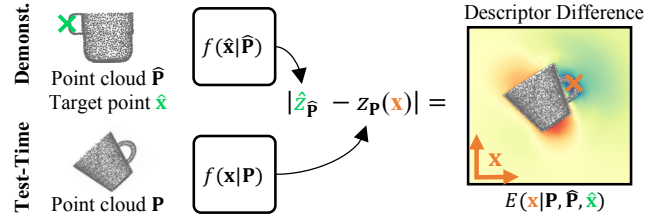


Fig. 4: **Energy landscape induced by NDFs** – Given a demonstration in the form of a pointcloud-point tuple  $(\hat{\mathbf{P}}, \hat{\mathbf{x}})$ , and the pointcloud of an unseen object instance  $\mathbf{P}$ , NDFs induce an energy landscape whose minimizer is the equivalent point for the unseen object. This energy is differentiable w.r.t. the point coordinates.

not only for a position (which may only denote, e.g., a single contact location) but also for the *orientation* of an external rigid body such as the gripper. For example, grasping the rim of a mug requires not only the correct contact position on the rim but also an orientation that enables the fingers to close around the inner and outer surface of the rim. If a grasp were attempted at the rim with an orientation that approached from the side of the mug, it wouldn't work. Similarly, to hang a mug on a rack by its handle, we must not only detect a point in the opening of the handle but also the orientation that allows the rack to pass through this opening.

Generally speaking, our demonstrations regularly consist of a point cloud  $\hat{\mathbf{P}}$  along with a world-frame pose  $\hat{\mathbf{T}} \in SE(3)$  of some rigid body  $\mathbf{S}$  in the vicinity of  $\hat{\mathbf{P}}$  ( $\mathbf{S}$  could be a gripper or a supporting object, like a rack or a shelf). We now wish to transfer both the position and orientation components of this pose when presented with a new point cloud. In this section, we will leverage  $f$  to find an equivalent pose of the rigid body  $\mathbf{S}$  that reproduces the same task for a new object instance defined by its point cloud  $\mathbf{P}$ .

We approach this from the perspective of defining a task-specific local coordinate frame, computing the pose  $\mathbf{T}_{\text{rel}}$  of external object  $\mathbf{S}$  in this local frame, and solving for the corresponding local frame when presented with a new object instance. After finding this corresponding frame, we use the same relative pose  $\mathbf{T}_{\text{rel}}$  in this detected frame to compute a new world-frame pose  $\mathbf{T}$  for object  $\mathbf{S}$ . We leverage our knowledge about the pose  $\hat{\mathbf{T}}$  of object  $\mathbf{S}$  to aid in parameterizing the pose of the local frame by *fixing* the relative pose  $\mathbf{T}_{\text{rel}}$  to be the identity matrix  $\mathbf{I}^4$ , i.e. we constrain the local frame specified in the demonstrations to *exactly align with the body frame defining the pose  $\hat{\mathbf{T}}$  in the world*. The result is that we can directly parameterize the resulting pose  $\mathbf{T}$  by the pose of the detected local frame for the new instance.

With this setup, an initial decision is how to encode local reference frames expressed as  $SE(3)$  poses. Our approach is guided by the observation that we can attach a reference frame to three or more (non-collinear) points which are constrained to move together rigidly, and establish a one-to-one mapping between these points and the configuration of the reference frame. Therefore, by initializing such a set of *query points*  $\mathcal{X} \in \mathbb{R}^{3 \times N}$  in a known canonical configuration, we can represent a local frame represented by an  $SE(3)$  transformation  $\mathbf{T}$  via the action of  $\mathbf{T}$  on  $\mathcal{X}$ .  $\mathbf{T}$  is then represented via the coordinates of the *transformed query*

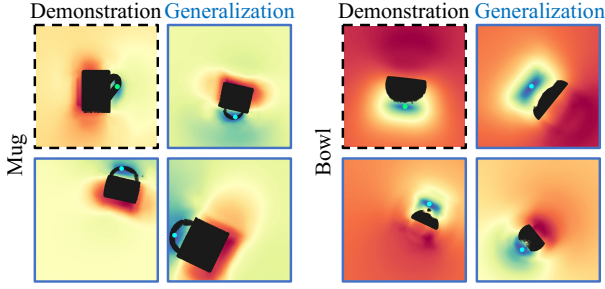


Fig. 5: **Equivariance and generalization of NDFs** – Absolute descriptor differences for a 2D target point  $\hat{\mathbf{x}} \in \mathbb{R}^2$ . The point descriptor field succeeds in transferring the target point to *unseen* SE(3) poses, as well as to *unseen* instances within the same class.

point cloud  $\mathbf{T}\mathcal{X}_h$  (where  $\mathcal{X}_h$  denotes  $\mathcal{X}$  expressed with homogeneous coordinates).

We now define a *Neural Pose Descriptor Field* as the concatenated point descriptors of the individual points in  $\mathbf{T}\mathcal{X}_h$ :

$$\mathcal{Z} = F(\mathbf{T}|\mathbf{P}) = \bigoplus_{\mathbf{x}_i \in \mathcal{X}_h} f(\mathbf{T}\mathbf{x}_i|\mathbf{P}) \quad (10)$$

$F$  maps a point cloud  $\mathbf{P}$  and an SE(3) transformation  $\mathbf{T}$  to a category-level pose descriptor, which we call  $\mathcal{Z}$ . Fig. 3 shows a visualization of the architecture of  $F$ . Note that  $F$  inherits SE(3)-equivariance from  $f$ , and is thus similarly guaranteed to generalize across all 6-DoF object configurations of  $\mathbf{P}$ .

Similar to transferring individual points by minimizing point descriptor distances (Fig. 4), this encoding enables us to transfer a local frame with a reference pose  $\hat{\mathbf{T}}$  when provided with a new point cloud by finding the pose  $\mathbf{T}$  of the query point set  $\mathcal{X}$  that minimizes the distance to the descriptor  $\hat{\mathcal{Z}} = F(\hat{\mathbf{T}}|\hat{\mathbf{P}})$  (our approach for performing this minimization is described at the end of this sub-section). However, an important remaining decision is the choice of points  $\mathbf{x}_i \in \mathcal{X}$ . Any set of three or more points is equally sufficient to represent a reference pose, but the *position of these points relative to  $\hat{\mathbf{P}}$*  has a significant impact on what solutions are obtained when performing pose transfer. In particular, since we represent poses as the concatenation of individual point descriptors, the location of each  $\mathbf{x}_i$  in the demonstration fundamentally determines *which features of the object we are aligning the rigid body to*. For instance, placing  $\mathbf{x}_i$  in the vicinity of the *handle* of a mug would lead to a pose descriptor sensitive to the position of the *handle* across mug instances. Fig. 6 highlights this issue by visualizing the effect of different ways of distributing the points in  $\mathcal{X}$ . To select a set of points that is in the vicinity of the contact that occurs with object  $\mathbf{S}$ , we find that a robust heuristic is to sample points uniformly at random from within the bounding box of the rigid body  $\mathbf{S}$ .

**Pose regression with NDFs.** Similar to how  $f$  induces an energy over *coordinates* across object instances (see Fig. 4 and (9)),  $F$  induces an energy over *poses*. We start with a tuple  $(\hat{\mathbf{T}}, \hat{\mathbf{P}}, \mathbf{S})$  pairing pose  $\hat{\mathbf{T}}$  of rigid body  $\mathbf{S}$  to a point cloud  $\hat{\mathbf{P}}$ . Then, given a novel object instance represented by its point cloud  $\mathbf{P}$ , we can compute a pose  $\mathbf{T}$  such that the relative configuration between  $\mathbf{P}$  and  $\mathbf{S}$  at pose  $\mathbf{T}$  corresponds

to the relative configuration between  $\hat{\mathbf{P}}$  and  $\mathbf{S}$  at pose  $\hat{\mathbf{T}}$ . We initialize  $\mathbf{T} = (\mathbf{R}, \mathbf{t})$  at random and optimize the translation  $\mathbf{t}$  and rotation  $\mathbf{R}$  (parameterized via axis-angle) to minimize the L1 distance between the descriptors of  $\hat{\mathbf{T}}$  and  $\mathbf{T}$ :

$$\hat{\mathbf{T}} = \underset{\mathbf{T}}{\operatorname{argmin}} \|F(\mathbf{T}|\mathbf{P}) - F(\hat{\mathbf{T}}|\hat{\mathbf{P}})\| \quad (11)$$

We solve this directly via iterative optimization (ADAM [16]), minimizing the distance between spatial descriptors of our target pose and our sought-after pose by back-propagating the norm of the differences through  $\mathcal{Z}$ . In Fig. 7 we visualize the optimization steps taken by (11) for optimizing a grasp pose of the end-effector. While we provide an in-depth evaluation in the experiments section, this result is representative in that the end-effector reliably and robustly converges to the correct orientation and location on the object.

### C. Few-shot imitation learning with NDFs

We are now ready to use Neural Descriptor Fields to acquire a pick-and-place skill for a category of objects from only a *handful* of demonstrations. For each category, we are provided with a set of  $K$  demonstrations,  $\{\mathcal{D}_i\}_{i=1}^K$ . Each demonstration  $\mathcal{D}_i = (\mathbf{P}^i, \mathbf{T}_{pick}^i, \mathbf{T}_{rel}^i)$  is a tuple of a (potentially partial) point cloud of the object  $\mathbf{P}^i$ , and two poses: the end-effector pose before grasping,  $\mathbf{T}_{pick}^i$ , and the relative pose  $\mathbf{T}_{rel}^i$  that transforms the grasp pose to the place pose via  $\mathbf{T}_{place}^i = \mathbf{T}_{rel}^i \mathbf{T}_{pick}^i$ . First, we obtain  $\mathcal{X}_{pick}$  and  $\mathcal{X}_{place}$  to represent the gripper and placement surface, respectively. We then leverage (10) 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  $\{(\mathcal{Z}_{pick}^i, \mathcal{Z}_{rel}^i)\}_{i=1}^K$ . Finally, this 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 (11) to recover  $\mathbf{T}_{pick}^{test}$  and  $\mathbf{T}_{rel}^{test}$  by minimizing the distance to spatial descriptors  $\bar{\mathcal{Z}}_{pick}$  and  $\bar{\mathcal{Z}}_{rel}$ . We rely on off-the-shelf inverse kinematics and motion planning algorithms to execute the final predicted pick-and-place task.

## III. EXPERIMENTS: DESIGN AND SETUP

Our experiments are designed to evaluate how effective our method is at generalizing pick-and-place tasks from a small number of demonstrations. In particular, we seek to answer three key questions: (1) How well do NDFs enable manipulation of unseen objects in unseen poses? (2) What impact does the parameterization of NDFs have on its performance? (3) Can NDFs transfer to a real robot?

**Robot Environment Setup.** Our environment includes a Franka Panda arm on a table with a depth camera at each table corner. The depth cameras are extrinsically calibrated to obtain fused point clouds expressed in the robot’s base frame. For our quantitative experiments we simulate the environment in PyBullet [6]. Depending on the task, an additional object such as a rack or a shelf is mounted somewhere on the table to act as a placement/hanging surface; see Fig. 9.

**Task Setup.** We provide 10 demonstrations for each task, and measure execution success rates on unseen object instances with randomly sampled initial poses and a random

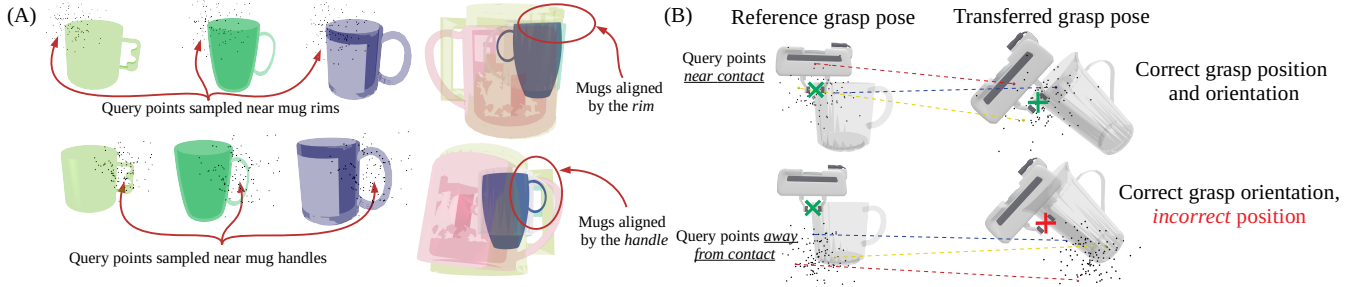


Fig. 6: **Effect of different query points** – (a) (Top) Given a set of reference mugs and query points  $\mathcal{X}$  distributed near the rim of each mug, a set of differently sized test mugs can be aligned by their rim feature by finding a pose whose descriptor matches the average of the reference pose descriptors. (Bottom) Following this procedure with  $\mathcal{X}$  near the mug handles leads the same set of test mugs to be aligned by a different feature (the handle). This highlights the sensitivity to the location of  $\mathcal{X}$  when performing pose transfer. (b) This sensitivity has important implications when transferring gripper poses for grasping: (Top) When the points in  $\mathcal{X}$  are distributed near the rim of the mug and are used to transfer a grasp pose to a taller mug, the gripper position remains near the rim and the grasp can succeed. (Bottom) In contrast, placing query points near the bottom of the mug leads to a transferred pose that is biased toward the bottom of the taller mug, resulting in a grasp that will fail due to collision with the object.

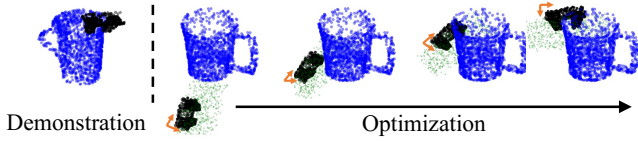


Fig. 7: **Pose regression with NDFs** – Given a demonstration point cloud and gripper pose (left), our method enables solving for the gripper pose (orange) for grasping an unseen object instance (right, blue) by minimizing the difference between demonstration and test pose descriptors, defined via the gripper query point cloud (green).

uniform scaling applied. We assume a segmented object point cloud and a static environment that remains fixed between demonstration-time and test-time. We consider three separate tasks in both the simulated and real environment: 1) grasping a mug by the rim and hanging it on a rack by the handle 2) grasping a bowl by the rim and placing it upright on a shelf 3) grasping the top of a bottle from the side and placing it upright on a shelf. In simulation, we utilize ShapeNet [3] meshes for each object class, where we filter out a subset of meshes that are incompatible with the tasks.

**Baselines.** We run a detailed quantitative comparison with a pick-and-place pipeline utilizing Dense Object Nets (DON) [10]. Our pipeline detects grasp poses following [10] using demonstrated grasp points. To infer object placement, we label a set of semantic keypoints in demonstrations and utilize the DON correspondence model with depth to obtain corresponding 3D keypoints on test objects. We then estimate the relative transformation for placing by optimally registering the detected points to the final configuration of the corresponding points from the demonstrations using SVD.

We also attempted to benchmark with recently proposed TransporterNets [48]. However, the model in [48] is primarily applied to planar tasks that only require top-down pick-and-place. While we were able to reproduce the impressive capabilities of their model in the subset of our tasks in which top-down grasping is sufficient (92% success rate at grasping the rim of a mug), several attempts at implementing a 6-DoF extended version that predicts the remaining rotational and  $z$ -height degrees of freedom (for grasping and placing) failed

|                       | Mug         |             |             | Bowl        |             |             | Bottle      |             |             |
|-----------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                       | Grasp       | Place       | Overall     | Grasp       | Place       | Overall     | Grasp       | Place       | Overall     |
| <b>Upright Pose</b>   |             |             |             |             |             |             |             |             |             |
| DON [10]              | 0.91        | 0.50        | 0.45        | 0.50        | 0.35        | 0.11        | 0.79        | 0.24        | 0.24        |
| NDF                   | <b>0.96</b> | <b>0.92</b> | <b>0.88</b> | <b>0.91</b> | <b>1.00</b> | <b>0.91</b> | <b>0.87</b> | <b>1.00</b> | <b>0.87</b> |
| <b>Arbitrary Pose</b> |             |             |             |             |             |             |             |             |             |
| DON [10]              | 0.35        | 0.45        | 0.17        | 0.08        | 0.20        | 0.00        | 0.05        | 0.02        | 0.01        |
| NDF                   | <b>0.78</b> | <b>0.75</b> | <b>0.58</b> | <b>0.79</b> | <b>0.97</b> | <b>0.78</b> | <b>0.78</b> | <b>0.99</b> | <b>0.77</b> |

TABLE I: **Unseen instance pick-and-place success rates in simulation.** For objects in upright poses (top row), NDFs perform on par with DON on grasp success rate, but outperforms DON on overall pick-and-place success rate. For objects in arbitrary poses (bottom row), DON’s performance suffers, while NDFs maintains higher success rates due to their equivariance to  $SE(3)$  transformations.

to achieve success rate above 10%.

**Evaluation Metrics.** To quantify the capabilities of each method, we measure success rates for grasping (stable object contact after grasp close) and placing (stable contact with placement surface), along with overall success, corresponding to both grasp and placement success.

**Training Details.** To pretrain DON [10] and NDF, we generate a dataset of 100,000 objects of mug, bowl and bottle categories at random tabletop poses. For each object, 300 RGB-D views with labeled dense correspondences are used to train DON, while we train NDF with point clouds captured from four static depth cameras. RGB-D images of the objects are rendered with PyBullet. While DON requires separate models for shapes in each category, we train a single instance of NDF on shapes across all categories. We train NDF using an occupancy network  $\Phi(\mathbf{x}, \mathcal{E}(\mathbf{P}))$  to reconstruct 3D shapes given the captured depth maps and train DON utilizing the author’s provided codebase.

#### IV. EXPERIMENTS: RESULTS

We conduct experiments in simulation to compare the performance of NDFs with Dense Object Nets (DON) [10] on three different object classes, and different pose configurations of each object. We then conduct ablation studies of the choice of parameterizing NDFs as the concatenation of pretrained

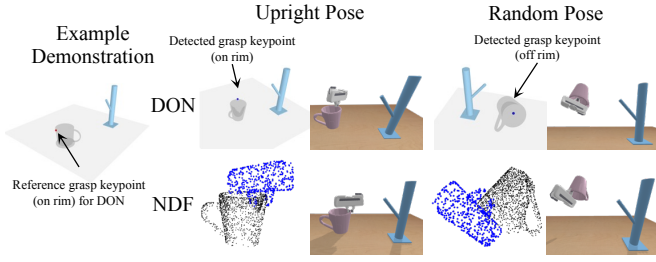


Fig. 8: **Qualitative Examples of Grasp Predictions** – Both DON and NDF predict successful grasps on upright mugs. When mugs exhibit arbitrary poses, DON fails to detect the correct keypoint for grasping, while our method still successfully infers a grasp.

occupancy network activations, as well as the effect of the number of demonstrations. Finally, we apply our full model to a real robot, and validate that the proposed method generalizes to out-of-distribution object configurations.

### A. Simulation Experiments

**Upright Pose.** First, we consider the ability to transfer manipulation skills to novel objects in different upright poses. We find that across mugs, bowls, and bottles, NDFs dramatically outperform DON on placing, and perform significantly better on grasping (Table I, top). We find that DON’s failures are usually a function of either insufficient precision in keypoint predictions, or failed registration of test-time keypoints to the demonstration keypoints. We find that even if predicted keypoints locations are semantically correct, the place may still fail when the *relative locations* of keypoints to each other are too different from the demonstration objects. This may happen, for instance, if the object is significantly smaller, or the shape is otherwise significantly different. In contrast, the proposed method matches descriptors in a learned, highly over-parameterized latent space, and is significantly more robust in solving for placement poses.

**Arbitrary Pose.** Next, we consider a harder setting: while the demonstrations are all performed on upright-posed objects, the robot must subsequently execute the task on objects in *arbitrary* SE(3) poses. In this setting, we find that the performance of DON suffers significantly, even though we trained DON on a large dataset of images of objects in different poses. In contrast, we find that NDF’s performance, while not at the same level as in the upright task, suffers dramatically less, maintaining a high pick-and-place success rate (Table I, bottom). Fig. 8 highlights an example to illustrate this performance gap. The drop in our method’s performance can be attributed to the fact that while provably equivariant to rotations and translations, the PointNet encoder is not perfectly robust to unobserved occlusions and disocclusions of the object point cloud: pointclouds might be missing parts previously observed, or contain parts that were previously unobserved. For instance, if only upright mugs were observed, the encoder has not previously seen the bottom of a mug.

### B. Analysis

We now analyze NDF’s dependence on the occupancy network parameterization, the number of demonstrations, and

| Random NDF |       |         | Last Layer OccNet |       |         | First Layer OccNet |       |         | All Layer OccNet |             |             |
|------------|-------|---------|-------------------|-------|---------|--------------------|-------|---------|------------------|-------------|-------------|
| Grasp      | Place | Overall | Grasp             | Place | Overall | Grasp              | Place | Overall | Grasp            | Place       | Overall     |
| 0.42       | 0.00  | 0.00    | 0.75              | 0.88  | 0.65    | 0.77               | 0.84  | 0.65    | <b>0.96</b>      | <b>0.92</b> | <b>0.88</b> |

TABLE II: Success rates of NDFs with different descriptors.

| 0.1  |      |      | 0.5  |      |      | 1.0         |             |             | 2.0  |      |      | 10.0 |      |      |
|------|------|------|------|------|------|-------------|-------------|-------------|------|------|------|------|------|------|
| G    | P    | O    | G    | P    | O    | G           | P           | O           | G    | P    | O    | G    | P    | O    |
| 0.77 | 0.53 | 0.39 | 0.77 | 0.85 | 0.63 | <b>0.96</b> | <b>0.92</b> | <b>0.88</b> | 0.78 | 0.56 | 0.42 | 0.73 | 0.39 | 0.27 |

TABLE III: Effect of representing pose descriptors with differently scaled query point clouds  $\mathcal{X}$ .

the size of the query point cloud used for encoding pose descriptors. We run our analysis on the upright mug task.

**Neural Descriptors.** Full NDFs are parameterized as the concatenation of the activations of all layers of an occupancy network trained for 3D reconstruction. In Table II, we analyze the effect of parameterizing NDFs with features from a randomly initialized occupancy network, as well as with only the first- or last-layer activations of a trained occupancy network. We find that utilizing all activations obtains the best performance by a large margin. This validates our assumptions on occupancy networks as a hierarchical feature extractor, and the task of 3D reconstruction as an important part of learning informative features.

**Query Point Cloud Scaling.** We further study the effect of the scale of the query point cloud  $\mathcal{X}$  for representing the grasping and placing pose descriptors. In Table III we show that our choice of sampling in the bounding box of the rigid body that interacts with the object is a robust heuristic, while scaling  $\mathcal{X}$  up or down reduces the performance.

**Number of Demonstrations.** We also analyze the impact of demonstration number on the performance of NDFs and DON on the upright mug pick-and-place task. Please see Table IV for quantitative results. We find that while the performance of NDFs decreases significantly in the single-demonstration case, it still significantly outperforms DON, and more demonstrations yield significant performance gains.

### C. Real World Execution

Finally, we validate that NDFs enable manipulation of novel object instances in novel poses on the real robot. We record ten pick-and-place demonstrations on mugs, bowls, bottles in *upright poses*. We then execute the same pick-and-place task on novel instances of real mugs, bowls, and bottles in a variety of different, often challenging, configurations. Please see Fig. 9 both for a visualization of the demonstrations and the qualitative results, as well as the **supplementary video** for sample videos of each of the real world task executions.

## V. RELATED WORK

### A. Generalizable Manipulation

Our work builds upon a rich line of research on imitation learning for manipulation. For known objects, one may rely on pose estimation [36, 46, 49], however, this does not enable category-level manipulation. Template-matching with coarse 3D primitives [15, 23, 42] or non-rigid registration [36] can

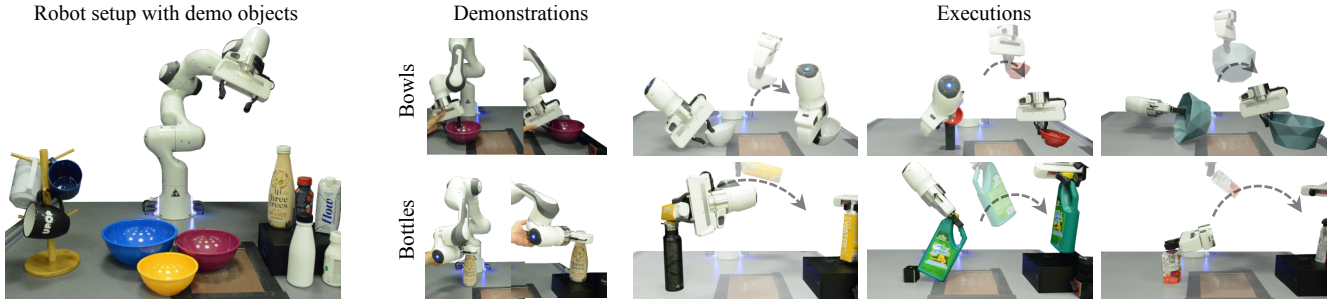


Fig. 9: Example executions of NDF on real bottles and bowls. Ten demonstrations with an object in an upright pose were used per category. NDFs enable inferring 6-DoF poses for both picking and placing *unseen* object instances in *out-of-distribution* poses.

| Model    | 1           | 5           | 10          |
|----------|-------------|-------------|-------------|
| DON [10] | 0.32        | 0.36        | 0.45        |
| NDF      | <b>0.46</b> | <b>0.70</b> | <b>0.88</b> |

TABLE IV: Overall success of NDFs and DON as a function of total example demonstrations.

enable generalization across changes in shape and pose, but suffers when objects deviate significantly from the primitive or test and reference scene are too different. Direct learning of pick-and-place policies meanwhile requires large amounts of data from demonstrations [2, 14, 40].

Our work is closely related to recent work leveraging category-level keypoints as an object representation for transferrable robotic manipulation. Keypoints can either be predicted directly [12, 13, 20], requiring a large, human-annotated dataset, or can be chosen among a set of self-supervised category-level object correspondences [10, 41]. However, keypoints must be carefully chosen to properly constrain manipulation poses, with outcomes sensitive to both keypoint choice and accuracy. Both approaches use 2D convolutional neural networks (CNNs) for prediction. As 2D CNNs are only equivariant to shifts of the object parallel to the image plane, these methods require observing images of objects from *all possible* rotations and translations at training time, and even then do not guarantee that keypoints are consistent across 6-DoF configurations of object instances. Transporter Nets [48] predict manipulation poses via a CNN over orthographic, top-down views, equipping the model with equivariance to in-plane, 2D translations of objects. However, this approach struggles to predict arbitrary 6-DoF poses, and is not equivariant to full 3D rotations and translations.

Neural Descriptor Fields enable transferring observed manipulation poses across an object category using task-agnostic, self-supervised pre-training, without human-labeled keypoints, and are fully equivariant to  $SE(3)$  transformations. We demonstrate imitation of full pick-and-place tasks for unseen object configurations from a small handful of demonstrations, and significantly outperform baselines based on correspondence predicted in 2D.

### B. Neural Fields and Neural Scene Representations

Our approach leverages neural implicit representations to parameterize a continuous descriptor field which represents a manipulated object. Most saliently, such fields have been proposed to represent 3D geometry [5, 25, 27, 29, 33], appearance [22, 24, 34, 37, 38, 44, 47], and tactile properties [11]. They offer several benefits over conventional

discrete representations: due to their continuous nature, they parameterize scene surfaces with “infinite resolution”. Furthermore, their functional nature enables the principled incorporation of symmetries, such as  $SO(3)$  equivariance [7, 50]. Their functional nature further enables the construction of latent spaces that encode class information as well as 3D correspondence [8, 17, 39]. Lastly, neural fields have been leveraged to find unknown camera poses in 3D reconstruction tasks [19, 45].

## VI. DISCUSSION AND CONCLUSION

Several limitations and avenues for future work remain. While this approach is in principle applicable to non-rigid objects, this remains to be tested, and extensions based on recent work on non-rigid scenes in 3D reconstruction and novel view synthesis [9, 18, 26, 28, 31, 43] might be necessary. Further, NDFs only define transferable energy landscapes over *poses* and *points*: future work may explore integrating such energy functions with trajectory optimization to enable NDFs to transfer to full trajectories. Furthermore, we assume the placement target remains static: future work may explore similarly inferring an object-centric representation of the placement target.

In summary, this work introduces Neural Descriptor Fields as object representations that allow few-shot imitation learning of manipulation tasks, with only task-agnostic pre-training in the form of 3D geometry reconstruction, and without the need for further training at imitation learning time. We build on prior work using dense descriptors for robotics, neural fields, and geometric machine learning to develop dense descriptors that both generalize across instances and provably generalize across  $SE(3)$  configurations, which we show enables our approach to apply to novel objects in both novel rotations and translations, where 2D dense descriptors are insufficient.

## VII. ACKNOWLEDGEMENT

This work is supported by DARPA under CW3031624 (Transfer, Augmentation and Automatic Learning with Less Labels) and the Machine Common Sense program, Singapore DSTA under DST00OECI20300823 (New Representations for Vision), Amazon Research Awards, and the NSF AI Institute for Artificial Intelligence and Fundamental Interactions (IAIFI). Anthony and Yilun are supported in part by NSF Graduate Research Fellowships. We would like to thank Leslie Kaelbling and Tomas Lozano-Perez for helpful discussions.

## REFERENCES

- [1] Brenna D Argall et al. “A survey of robot learning from demonstration”. In: *Robotics and autonomous systems* (2009).
- [2] Lars Berscheid, Pascal Meißner, and Torsten Kröger. “Self-supervised learning for precise pick-and-place without object model”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4828–4835.
- [3] Angel X Chang et al. “Shapenet: An information-rich 3d model repository”. In: *arXiv preprint arXiv:1512.03012* (2015).
- [4] Yinbo Chen, Sifei Liu, and Xiaolong Wang. “Learning continuous image representation with local implicit image function”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8628–8638.
- [5] Zhiqin Chen and Hao Zhang. “Learning implicit fields for generative shape modeling”. In: *Proc. CVPR*. 2019, pp. 5939–5948.
- [6] Erwin Coumans and Yunfei Bai. “Pybullet, a python module for physics simulation for games, robotics and machine learning”. In: *GitHub repository* (2016).
- [7] Congyue Deng et al. “Vector Neurons: A General Framework for SO (3)-Equivariant Networks”. In: *arXiv preprint arXiv:2104.12229* (2021).
- [8] Yu Deng, Jiaolong Yang, and Xin Tong. “Deformed implicit field: Modeling 3d shapes with learned dense correspondence”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 10286–10296.
- [9] Yilun Du et al. “Neural Radiance Flow for 4D View Synthesis and Video Processing”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.
- [10] 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.
- [11] Ruohan Gao et al. “ObjectFolder: A Dataset of Objects with Implicit Visual, Auditory, and Tactile Representations”. In: *arXiv preprint arXiv:2109.07991* (2021).
- [12] 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.
- [13] Wei Gao and Russ Tedrake. “kPAM-SC: Generalizable Manipulation Planning using KeyPoint Affordance and Shape Completion”. In: *arXiv preprint arXiv:1909.06980* (2019).
- [14] Marcus Gualtieri, Andreas ten Pas, and Robert Platt. “Pick and place without geometric object models”. In: *2018 IEEE International Conference on Robotics and Automation*. IEEE. 2018, pp. 7433–7440.
- [15] 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.
- [16] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [17] Amit Pal Singh Kohli, Vincent Sitzmann, and Gordon Wetzstein. “Semantic implicit neural scene representations with semi-supervised training”. In: *2020 International Conference on 3D Vision (3DV)*. IEEE. 2020, pp. 423–433.
- [18] Zhengqi Li et al. “Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.
- [19] Chen-Hsuan Lin et al. “BARF: Bundle-Adjusting Neural Radiance Fields”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2021.
- [20] Lucas Manuelli et al. “kpam: Keypoint affordances for category-level robotic manipulation”. In: *arXiv preprint arXiv:1903.06684* (2019).
- [21] Lars Mescheder et al. “Occupancy Networks: Learning 3D Reconstruction in Function Space”. In: *Proc. CVPR*. 2019.
- [22] Ben Mildenhall et al. “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”. In: *Proc. ECCV*. 2020.
- [23] 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.
- [24] Michael Niemeyer et al. “Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision”. In: *Proc. CVPR*. 2020.
- [25] Michael Niemeyer et al. “Occupancy Flow: 4D Reconstruction by Learning Particle Dynamics”. In: *Proc. ICCV*. 2019.
- [26] Michael Niemeyer et al. “Occupancy flow: 4d reconstruction by learning particle dynamics”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 5379–5389.
- [27] Jeong Joon Park et al. “DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation”. In: *Proc. CVPR*. 2019.
- [28] Keunhong Park et al. “Deformable Neural Radiance Fields”. In: *arXiv preprint arXiv:2011.12948* (2020).
- [29] Songyou Peng et al. “Convolutional occupancy networks”. In: *Proc. ECCV*. 2020.
- [30] Dean A Pomerleau. “ALVINN: An autonomous land vehicle in a neural network”. In: *NIPS*. 1989.
- [31] Albert Pumarola et al. “D-NeRF: Neural Radiance Fields for Dynamic Scenes”. In: *arXiv preprint arXiv:2011.13961* (2020).
- [32] Charles Ruizhongtai Qi et al. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space”. In: *Advances in neural information processing systems*. 2017, pp. 5099–5108.
- [33] Daniel Rebain et al. “Deep Medial Fields”. In: *arXiv preprint arXiv:2106.03804* (2021).
- [34] Shunsuke Saito et al. “Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization”. In: *Proc. ICCV*. 2019, pp. 2304–2314.
- [35] Stefan Schaal. “Is imitation learning the route to humanoid robots?”. In: *Trends in cognitive sciences* (1999).
- [36] 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.
- [37] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. “Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations”. In: *Proc. NeurIPS* 2019. 2019.
- [38] Vincent Sitzmann et al. “Light Field Networks: Neural Scene Representations with Single-Evaluation Rendering”. In: *arXiv*. 2021.
- [39] Vincent Sitzmann et al. “MetaSdf: Meta-learning signed distance functions”. In: *Proc. NeurIPS* (2020).
- [40] Shuran Song et al. “Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4978–4985.
- [41] Priya Sundareshan et al. “Learning Rope Manipulation Policies Using Dense Object Descriptors Trained on Synthetic Depth Data”. In: *arXiv preprint arXiv:2003.01835* (2020).
- [42] 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.
- [43] Wenqi Xian et al. “Space-time Neural Irradiance Fields for Free-Viewpoint Video”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 9421–9431.
- [44] Lior Yariv et al. “Multiview neural surface reconstruction by disentangling geometry and appearance”. In: *Proc. NeurIPS* (2020).
- [45] 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.
- [46] Youngrook 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.
- [47] Alex Yu et al. “pixelNeRF: Neural Radiance Fields from One or Few Images”. In: *Proc. CVPR* (2020).
- [48] Andy Zeng et al. “Transporter Networks: Rearranging the Visual World for Robotic Manipulation”. In: *Conference on Robot Learning (CoRL)* (2020).
- [49] 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.
- [50] Minghan Zhu, Maani Ghaffari, and Huei Peng. “Correspondence-Free Point Cloud Registration with SO (3)-Equivariant Implicit Shape Representations”. In: *arXiv preprint arXiv:2107.10296* (2021).