Trajectory Descriptor Fields: An Extension of Neural Descriptor Fields for Trajectory Imitation

Rahul Aggarwal, Prakhar Mittal, Pujith Kachana, Luke Jones
Georgia Institute of Technology
Atlanta, GA, United States
{rahulaggarwal965, prakhar, pkachana3, ljones96}@gatech.edu

Abstract

Motion planning is a resource-intensive aspect of robotic manipulation, especially when trajectories must fit additional task-specific constraints. Previous work introduced Neural Descriptor Fields as object representations that allow few-shot imitation learning of object manipulation tasks [5]. We propose a novel application of Neural Descriptor Fields to transfer full trajectories from object manipulation demonstrations. As laid out in previous work, we utilize an object representation that describes the points of the object and target, but also the poses relative to the object and target [5]. However, we extend previous work by applying the alignment of objects with the demonstrated data across entire trajectories via energy function integration and trajectory optimization. Our method is able to learn generalized representations of the demonstrated trajectories and successfully replicates them in simulation.

1. Background

Pick-and-place tasks are a common use case for modern robots, especially in warehouse and industry domains. Currently, such robot manipulation tasks are carried out via traditional controls-based motion planning algorithms. These methods work for continuous and non-changing tasks and environments, but are very limited for trajectory generation when it comes to generalizing to any initial object orientation, inconsistent trajectories, or varying placement locations. Recent work has shown much success in finding effective pick and place poses through robot learning approaches using human demonstration (imitation learning) [5]. The use of dense correspondences allows for demonstrated pick and place poses on a reference object to be generalized to new instances (e.g. finding the handle on a new mug given a labeled reference mug) [5]. However, there is still room for improvement in trajectory generation from demonstrations. Although new pick and place poses can be found, there is no guarantee that the trajectory will follow the demonstration. This can be problematic in scenarios where the trajectory characterization is important, such as moving a mug from a bottom shelf to a top shelf. Simply performing a pose-to-pose transformation to compute the trajectory without learning from the demonstration could lead to collisions. In this study, we extend latent-descriptor-based representations for trajectories to provide a more robust and efficient method for learning generalized pick-and-place trajectories from human demonstrations.

1.1. Related Works

Our approach relies heavily on previous work carried out in "Neural Descriptor Fields: SE(3)-Equivariant Object Representations for Manipulation" [5]. In this work, the authors show how Neural Descriptor Fields (NDFs) can be used to accurately and quickly learn representations of not only the points for an object (eg. mug) and a target (eg. rack for the mug or the gripper of the robot), but also the relative poses for this object-target pair, in as little as 5-10 demonstrations. The authors are successful in applying this approach to situations with objects different from those used in demonstration (eg. new mug) by using an optimization-based approach in which they search for the"best" pose i.e. the pose that most closely aligns with the data from the demonstration. As for training the NDFs, the authors use self-supervised learning. Finally, the authors show that NDFs are SE(3)-equivariant allowing for object manipulation despite an unseen rotation and/or translation applied to the object [5].

Other work that forms an important part of the foundation for our work, and also for the work on NDFs, is "Occupancy Networks: Learning 3D Reconstruction in Function Space" [4]. Here, the authors present a novel method for 3D reconstruction rooted in a learning-based approach that is able to solve the issues that plagued previous methods, including the inability to represent geometries more complex than coarse or restricted geometries. The authors show how Occupancy Networks implicitly represent 3D surfaces

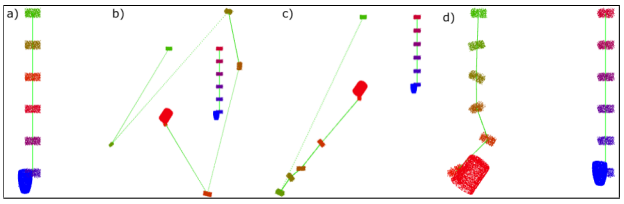

Figure 1: **a**) Reference Trajectory **b**) Randomly initialized test trajectory **c**) Test trajectory optimized over trajectory descriptor loss ℓ_t **d**) Test trajectory optimized over the combined trajectory descriptor loss ℓ_t and correlation loss ℓ_c.

as continuous decision boundaries of a deep neural network classifier, and how they can be used to generate 3D reconstructions of "infinite" resolutions without heavy memory usage [4].

1.2. Key Contributions

In this study, we present Trajectory Descriptor Fields as a robust representation for object motion and provide a framework for learning from demonstration trajectories with few-shot learning. The learned representations generalize to different environments with varying initial object locations, placement locations, and movement around obstacles, and to completely new trajectories. This approach has many potential applications for situations in which pick-and-place robots operate in diverse and changing environments or have additional constraints such as physical obstacles or other humans/robots (eg. robots in the fast food industry).

2. Approach

We use a concatenation of Pose Descriptor Fields to obtain naive Trajectory Descriptor Fields that uniformly parameterize trajectories across a target object class [5]. Next, we optimize the trajectory descriptors such that the refined descriptor is "closer" to the the reference descriptor. This approach capitalizes on the fact that trajectories can be discretely represented as a concatenation of poses.

The repository "ndf_robot" can be found here [1] and is associated with the original NDF paper [5]. We have made changes to "ndf_robot" according to our approach that is described thoroughly in the following subsections. Our code can be found here. [2]

2.1. Trajectory Descriptors

A trajectory can be represented as a continuous set of intermediate poses between the pick pose and the place pose. In this work, we choose to parameterize trajectories as discrete, evenly-spaced intermediate poses along the trajectory.

[1] https://github.com/anthonysimeonov/ndf_robot
[2] https://github.com/mittalprakhar/trajectory-descriptor-fields

The number of poses used to parameterize a trajectory is a hyperparameter, n, and we can show that the trajectory becomes better parameterized as n grows larger: the trajectory asymptotically approaches continuous parameterization as $n \to \infty$.

Using these intermediate poses, we compute a trajectory descriptor by concatenating the corresponding pose descriptors using the pre-trained occupancy network for the object class. This descriptor encodes the locations of numerous query points relative to the input object's point cloud to create n pose descriptors, thereby representing the trajectory followed by the object.

2.2. Trajectory Optimization

Our objective is to recreate the demonstration trajectory on new object instances with varying initial and final poses. To do so, first, we compute a trajectory descriptor for the demonstrated reference trajectory, and then, we optimize over randomly sampled poses for the new object instance to find a matching trajectory descriptor. Finally, the poses from the pose descriptors corresponding to the obtained trajectory descriptor are used to recreate the reference trajectory on the new object.

To elaborate, the reference trajectory consists of n reference poses, $\hat{\mathcal{T}} = \{\hat{\mathbf{T}}_i \in SE(3) : i \in [n]\}$ where each reference pose is represented by a corresponding rotation, $\hat{\mathbf{R}}$, and translation, $\hat{\mathbf{t}}$ from the zero-centered object point-cloud $\hat{\mathbf{P}}$. Query points are sampled along these poses and n pose descriptors are created, similar to the original work [5]. These reference pose descriptors are concatenated to create a reference trajectory descriptor, $\phi(\hat{\mathcal{T}}|\hat{\mathbf{P}})$. This trajectory descriptor is used to represent the reference trajectory and the rotations and translations of poses along new trajectories will be optimized such that the computed trajectory descriptor matches the reference.

When computing a trajectory for a new object, $n-1$ poses are initialized with random rotations and translations, and the last pose is set to be the desired place pose (Fig 1). The query points are fed through the neural pose descriptor model to find new pose descriptors, $F(\mathbf{T}_i|\mathbf{P}) : i \in [n]$ which are then concatenated to a new trajectory descriptor, $\phi(\mathcal{T}|\mathbf{P})$ where $\mathbf{P}$ is the new object point cloud.

$$\phi(\mathcal{T}|\mathbf{P}) = \bigoplus_{i \in [n]} F(\mathbf{T}_i|\mathbf{P}) \tag{1}$$

This new descriptor is compared to the reference descriptor, and the error is backpropagated all the way to the new rotations and translation to optimize the new poses to match the reference poses, thereby matching the trajectories. The initial loss function is the ℓ_1-loss between the reference and new trajectories:

$$\ell_t = \|\phi(\hat{\mathcal{T}}|\hat{\mathbf{P}}) - \phi(\mathcal{T}|\mathbf{P})\|_1 \tag{2}$$

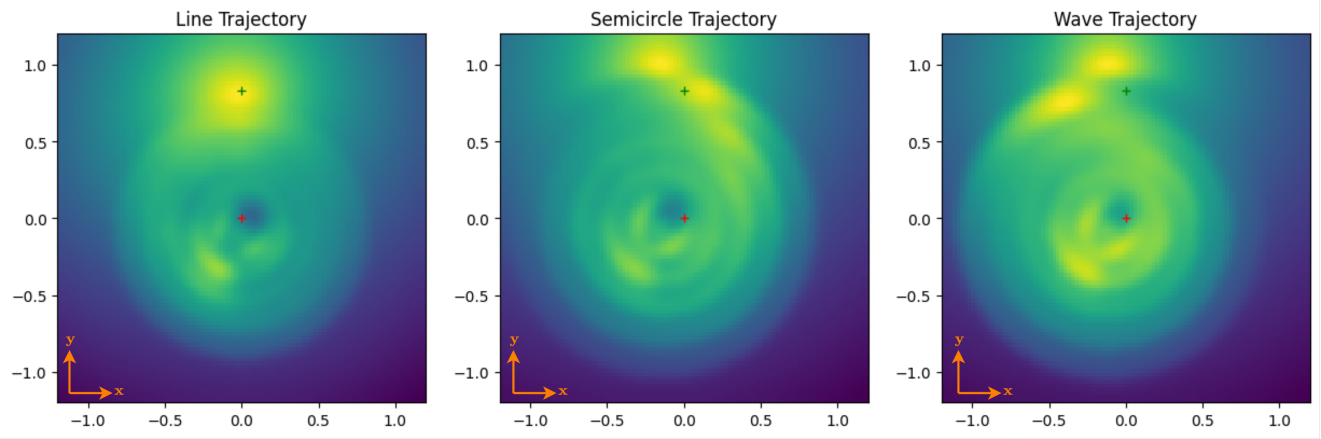

Figure 2: The energy landscape induced by the combination of the trajectory descriptor loss ℓ_t and correlation loss ℓ_c. The red plus and green plus represent the starting and desired end positions of the object (in this case a mug). *Left*: The energy field induced by the line trajectory in Fig 4a. *Middle*: The energy field induced by the semicircle trajectory in Fig 4c. *Right*: The energy field induced by the wave trajectory in Fig 4e.

Because the model to generate the descriptors is fully differentiable, this loss induces an energy field over the trajectory space that can be optimized, similar to how pose energies are optimized in the original work [5]. This approach yields a trajectory that is relatively similar to the demonstration, but does not generalize to cases with a new pick pose but the same place pose. Because the descriptors overfit to represent spatial relationships with an object point cloud, the generated trajectory is the same as the demonstrated trajectory with the trajectory orientation matching the point cloud orientation (Fig 1), possibly leading the trajectory away from the place point.

To make the model invariant to the initial pose of the object point cloud, we introduce a second loss element to bring the new poses closer to the demonstrated poses, the correlation loss, ℓ_c. Our final loss is a linear combination of the trajectory descriptor loss and the correlation loss:

$$\ell = \ell_t + \ell_c = \|\phi(\hat{\mathcal{T}}|\hat{\mathbf{P}}) - \phi(\mathcal{T}|\mathbf{P})\|_1 + k\|\hat{\mathcal{T}} - \mathcal{T}\| \quad (3)$$

where k is a tunable hyperparameter. A high k value enforces pose similarity too strongly, limiting the model's generalizability. A low k value leads to weak pose constraints, generating trajectories that drift away from the place pose. With a well tuned k, however, this new loss function has empirically shown to produce generalizable trajectories from expert demonstrations (Fig 1).

To validate the effectiveness of our trajectory descriptor and associated loss functions, we construct the following energy field:

$$E(\mathbf{x}|\hat{\mathbf{P}}, \mathbf{P}, \hat{\mathcal{T}}) = \|\phi(\hat{\mathcal{T}}|\hat{\mathbf{P}}) - \phi(\mathcal{X}|\mathbf{P})\|_1 + k\|\hat{\mathcal{T}} - \mathcal{X}\| \quad (4)$$

where $\mathcal{X}$ is a trajectory defined by a series of sampled points $\mathbf{x}_i$ with a fixed orientation $I \in SO(3)$. We omit variance in rotation for ease of visualization. The sequence of optimal minimizers $\mathcal{X}^*$ is obtained by creating a strict sampling pattern of minimizers:

$$x_i^* \in \mathcal{X}^* = \arg\min_{x_i} E(\mathbf{x}|\hat{\mathbf{P}}, \mathbf{P}, \hat{\mathcal{T}}) \quad (5)$$

In Fig 2, we plot the energy for points on a densely sampled grid with variation across reference mug instance, mug pose, and mug trajectory. The lighter colors in the plot represent that the lowest energy points (corresponding to the minimizers) imitate the trajectory of the reference configuration. The circular artifacts that can be seen are side effects of the sampling patterns for the minimizers but the overall trajectory minimizer $\mathcal{X}^*$ can be interpreted as the lowest cost path through the associated energy field. These visualizations validate that our proposed solution transfers across entirely different trajectories.

2.3. Additional Challenges

One of the first problems we faced was how to augment the trajectory descriptor loss ℓ_t to balance between generalizable trajectories and expert imitation. We initially introduced a pairwise compensator loss:

$$\ell_p = \sum_{i=1}^{n-1} \|\mathbf{T}_i^{-1}\mathbf{T}_{i+1} - I\|_F \quad (6)$$

where $\|\cdot\|_F$ is the Frobenius matrix norm. The motivation behind ℓ_p was to constraint the subsequent pose difference using a metric called the Absolute Pose Error (APE). However, after implementing and testing the network training using the loss $\ell = \ell_t + \ell_p$, we observed empirically that the resulting trajectories, while accurate in generalization and

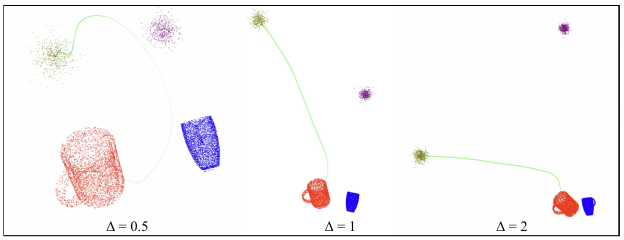

Figure 3: Testing the effect of increasing the distance of queried points from the object's point cloud center on the learned trajectory.

Expert Trajectory	Endpoint Displacement	Loss
Line	$(0, 0, 0)$	0.006797
Line	$(0, 0, 0.4)$	0.011077
Line	$(-0.2, 0.1, 0.3)$	0.011253
Semicircle	$(0, 0, 0)$	0.004429
Semicircle	$(0, 0.1, 0)$	0.005685
Semicircle	$(-0.2, 0.1, 0.3)$	0.006624
Wave	$(0, 0, 0)$	0.005342
Wave	$(-0.2, 0, 0)$	0.005408
Wave	$(-0.2, 0.1, 0.3)$	0.007355

Table 1: Final loss values for different trajectories and end points. The model was trained using $k = 0.02$ and epochs $= 100$.

transferring over initial object pose, were poor in trajectory imitation. To be specific, we observed that ℓ_p tended to minimize heavily each pair of poses except for a single pair $\mathbf{T}_j, \mathbf{T}_{j+1}$, which would have a very high loss. Attempts to modify the loss by penalizing larger values by changing the matrix norm to one of higher power (e.g $\|\cdot\|_\infty$) only served to overpower the trajectory descriptor loss ℓ_t and would require extensive hyperparameter tuning for proper normalization.

A secondary problem that we faced later in development was related to how the distance from the queried points to the object point cloud's center affected the quality of our learned trajectory from the demonstration. With an initial set of query points closely paired to the surface of the mug, we observe very accurate corresponding poses for the test mug; however, as we extended the distance of the query points further from the object's center and surface, we notice an increasing divergence from the demonstration's point cloud, as can be seen in Fig 3. This is a result of the occupancy network encoding a limited volume of space around the input object point cloud, resulting in less salient pose descriptors for query point clouds further away.

3. Experiments and Results

3.1. Experimental Setup

Our experiments were conducted in simulation using Google Colab. We use mugs as our target object and utilize an occupancy network pre-trained on mugs to generate descriptor fields. Our data consists of several expert demonstrations of trajectory with mugs including line, semi-circle, and wave trajectories. Test mugs were trained to imitate the demonstrated trajectories from the same starting point, a starting point offset in one dimension, and a starting point offset in all dimensions. Finally, we generate test trajectories, energy landscapes, and loss graphs to qualitatively and quantitatively analyze learning and generalization of trajectories.

3.2. Results

Our success was mostly measured through qualitative means, comparing target trajectories to those generated by our model. Overall this analysis was verified by the energy landscape. However, from a more quantitative lens, the decrease in loss also shows that learning was taking place here. This quantitative measure of loss also allows us to observe how individual parameters affect the model: for example we are able to observe how changing starting poses affects the loss.

The best results were obtained for $k = 0.02$. Quantitative results are summarized in Table 1. The loss rapidly declined in the first few epochs and plateaued around epoch 40 for all the trajectories that we tested (Fig 5). The loss was lower for simpler trajectories such as line than for more complex ones like wave and semicircle. The loss was also slightly higher when the test endpoint was not the same as the reference endpoint but was displaced.

Qualitatively, the test trajectories produced by the model had a visual resemblance to the reference trajectories (Fig 4). This confirmed that the model had learnt a generalized representation of the reference trajectories and it can successfully imitate them as long as the test endpoints are close to the reference endpoints. Compare and contrast the semicircle test trajectory (c) with and (d) without the endpoint displacement. In (c), the trajectory was a perfect semicircle whereas in (d), the displacements in all three dimensions were gradually accounted for across the entire trajectory, leading to a semicircular arc.

Moreover, the start orientation of the mug was different for the test trajectories compared to the orientation in the reference trajectories, but the end orientation was expected to be the same for both. The model accounted for this as is evident from (a). The mug was gradually flipped over the entire course of the test trajectory such that it ended up at the expected orientation irrespective of the start orientation. This showed that the model had not only learned the gen-

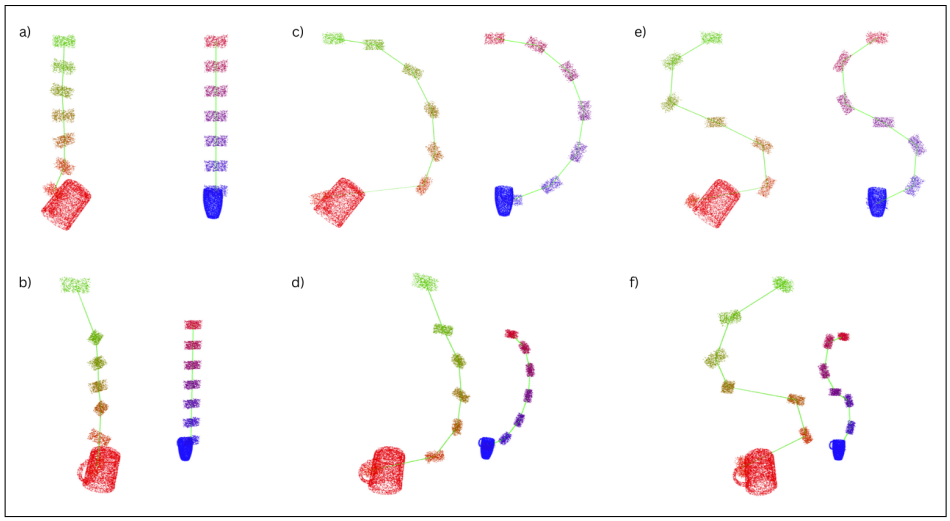

Figure 4: Side-by-side comparison of test trajectories (red) and reference trajectories (blue) for **a**) Line with no endpoint displacement, **b**) Line with endpoint displacement $(-0.2, 0.1, 0.3)$ **c**) Semicircle with no endpoint displacement, **d**) Semicircle with endpoint displacement $(-0.2, 0.1, 0.3)$ **e**) Wave with no endpoint displacement, **f**) Wave with endpoint displacement $(-0.2, 0.1, 0.3)$

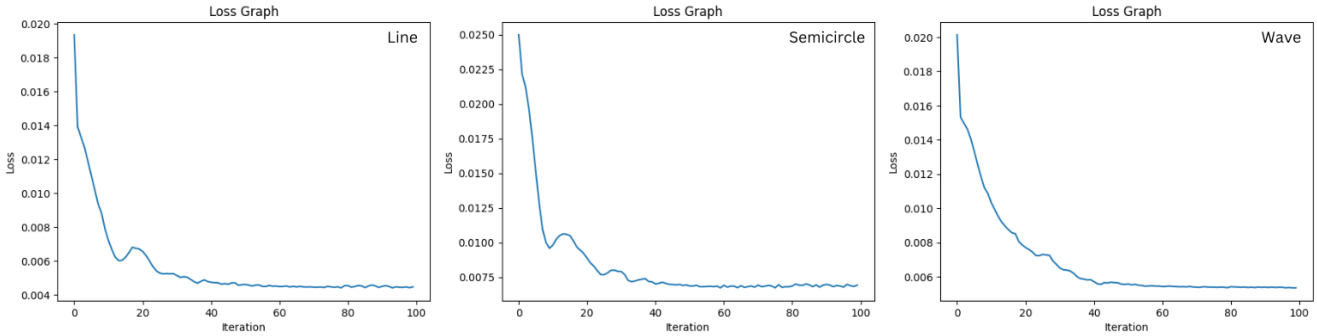

Figure 5: Loss graphs for different trajectories with zero endpoint displacement when $k = 0.02$ and epochs $= 100$. *Left*: Line. *Middle*: Semicircle. *Right*: Wave.

eralized positions associated with a trajectory but also the orientation changes.

4. Conclusion

We present Trajectory Descriptor Fields, an application of Neural Descriptor Fields [5], to define transferable energy landscapes over full object manipulation trajectories. We use this novel representation to allow few-shot and zero-shot imitation learning for pick-and-place manipulation tasks. We build upon the pre-trained category-specific occupancy networks that provide instance and pose equivariant point descriptors by constructing a new type of trajectory descriptor that generalizes across pick and place frames and imitates expert demonstrations.

There are several limitations to this work that may provide directions for future research. Notably, Point Descriptor Fields and the subsequent Pose Descriptor Fields and Trajectory Descriptor Fields do not generalize across class barriers. A possible extension of our work is to use contrastive learning as in [2] [3] to align the latent vectors of classes to obtain generalization across pick-and-place for disparate objects. Another avenue of research would be to predict sequential pose descriptors using transformers in a manner similar to [1]. This would allow for novel trajectories to be stochastically generated for better generalization on unseen data.

Our findings could also be further explored through real-world experiments. The original work on Neural Descriptor Fields [5] tests their pick-and-place imitation on a Franka Panda robot using expert manipulation demonstrations. A similar real-world experiment could be conducted with expert trajectory demonstrations to obtain additional benchmarks and reveal any possible sim-to-real gaps.

References

[1] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting, 2020. 5

[2] Thomas Kipf, Elise Van der Pol, and Max Welling. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019. 5

[3] Xiao Ma, David Hsu, and Wee Sun Lee. Learning latent graph dynamics for deformable object manipulation. *arXiv preprint arXiv:2104.12149*, 2, 2021. 5

[4] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space, 2019. 1, 2

[5] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B. Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se(3)-equivariant object representations for manipulation. *arXiv preprint arXiv:2112.05124*, 2021. 1, 2, 3, 5