NeRF-based-Action-Evaluation-for-Robotic-Grasping

Shenrui Wu **Yukai Zhou** **Weihan Xu**

Equal Contribution(for CS182 grading)

1 Introduction

Our work introduces a novel framework for robotic manipulation that leverages learned 3D scene representations to solve complex 6-DoF tasks from only RGB inputs. Prior approaches in vision-based manipulation often struggle with generalizing to unseen object orientations or require specialized depth sensors, which can fail on challenging materials. A key difficulty lies in equipping agents with the spatial reasoning needed to handle significant out-of-plane rotations. Our method addresses these challenges by first building a Neural Radiance Field (NeRF) of the scene from multi-view images. This NeRF model then acts as a form of "mental imagery," allowing us to synthesize novel orthographic views that are more suitable for learning affordances. From these synthesized views, a dual-component policy, consisting of separate attention and transport models, predicts pixel-wise picking and placing affordances. By searching for the optimal action across this imagined space of novel views, our agent can effectively reason about 6-DoF poses. This approach enables robust generalization to challenging object configurations, a significant step beyond prior works. In summary, our main contributions are the successful adaptation of a NeRF-based view synthesis pipeline for robust affordance prediction and the demonstration of its effectiveness on challenging manipulation tasks. Our code is available at here!

2 Related Works

2.1 Vision-based Manipulation

Object-centric vs. Action-centric. Previous robotic manipulation research has largely followed two paradigms. Object-centric methods represent scenes with 6-DoFposes [1, 2, 3] or dense descriptors [4, 5], but can struggle with deformable objects or require specialized data collection. In contrast, action-centric, end-to-end approaches [6, 7] offer greater flexibility but are known to be sample inefficient. To bridge this gap, some previous works incorporate spatial structure to improve efficiency [8, 9, 10].

2.2 Neural Fields for Robotics

For robotic manipulation, several methods have leveraged neural fields, but with some limitations. For instance, GIGA [11] and NDF [12] depend on depth cameras, which our method does not require. Other works focus primarily on grasping, such as Dex-NeRF [13] for 3-DoF poses and NeRF-Supervision [14] for learning pick-point descriptors. In contrast, our method is capable of predicting both 6-DoFpicking and pick-conditioned placing. Furthermore, unlike model-based [15] or reinforcement learning [16] approaches, our method uses imitation learning to bypass the need for complex reward function design, improves sample efficiency via approximate rotational equivariance.

3 Method

Our goal is to train a robotic agent capable of performing complex 6-DoF manipulation tasks from visual inputs. To achieve this, our methodology is structured into three key stages discussed below:

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

(Section 3.1) a comprehensive data generation pipeline that produces expert demonstrations enriched with synthesized visual data; (Section 3.2) an imitation learning framework to train the agent's policy, which consists of separate models for picking and placing affordances; and (Section 3.3) a rigorous evaluation protocol to assess the agent's performance and generalization capabilities on unseen scenarios.

3.1 Data Generation

Our approach is founded on imitation learning, for which a high-quality dataset of expert demonstrations is a crucial prerequisite. We have developed a systematic data generation pipeline utilizing the PyBullet [17] physics simulator, built upon the Ravens [18] benchmark framework. This process consists of two main stages for each demonstration episode: task execution by an oracle agent and scene reconstruction via Neural Radiance Fields (NeRF) [19].

Oracle-based(expert) Demonstration For each task, a scripted oracle agent with access to the ground-truth state of the simulation (e.g., object poses) determines and executes an optimal action sequence. The oracle logic has been adapted to compute a grasp pose (T_{pick}) by selecting a stable point on the target object's surface and determines the place pose (T_{place}). The resulting successful demonstrations, comprising sequences of observations and actions ($a_t = \{T_{\text{pick}}, T_{\text{place}}\}$), form the basis of our dataset.

Mental Imagery via NeRF A core component of our methodology is the use of "mental imagery" [20, 21] to create standardized and information-rich visual representations. Instead of directly using raw perspective images from the simulator's cameras, we generate novel orthographic views from a learned 3D scene representation. This is accomplished as follows for each successful demonstration:

- **Multi-view Image Capture:** At the key moment before the robot executes the pick action, we capture a set of RGB images of the scene from multiple pre-defined camera viewpoints.
- **NeRF Scene Reconstruction:** These multi-view images and their corresponding camera poses are used to train a scene-specific NeRF model. We employ instant-NGP [22], which rapidly learns a high-fidelity 3D representation of the scene.
- **Orthographic View Synthesis:** Once the NeRF model is trained, we perform orthographic ray casting to render a set of novel, standardized views. These orthographic images are more amenable to learning translation-equivariant policies with ConvNets.
- **Data Storage:** The final rendered orthographic color maps and height maps are saved and we ensure that when an episode is loaded for training, these corresponding NeRF-generated views are provided to the agent.

3.2 Training

The agent's policy is factorized into two main components: an attention model for picking (T_{pick}) and a transport model for placing (T_{place}). Both models are trained via imitation learning using the expert demonstrations generated in the previous stage.

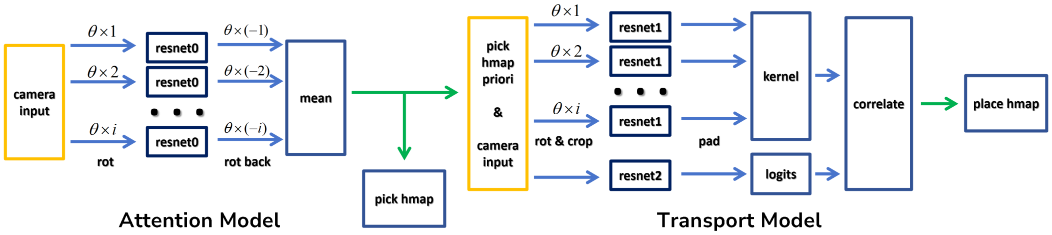

Figure 1: **Model structure**.(a)Attention model rotates and averages ResNet0 feature maps to produce a pick heatmap.(b)Transport model uses that pick heatmap with ResNet1/ResNet2–derived kernels and logits to compute a place heatmap via correlation.

Attention Model (Picking) The attention model is a fully convolutional ResNet [23] that takes a NeRF-rendered orthographic view of the scene as input. It outputs a pixel-wise affordance heatmap, where the highest value indicates the optimal picking location. The model is trained with a softmax cross-entropy loss against a one-hot target map, where the single "hot" pixel corresponds to the expert's demonstrated pick location.

Transport Model (Placing) The transport model is also based on a two-stream ResNet architecture. Its goal is to predict the placing affordance conditioned on the previously selected pick action. The training process involves:

- **Kernel Generation:** A small image patch centered at the expert's pick location is cropped from the *pick position image* and encoded by one of the ResNet streams to produce a "visual kernel". This kernel represents the visual features of the object being held.
- **Correlation(Affordance Matching via Visual Kernels):** This visual kernel is then used as a convolutional filter and is applied across the feature map of the target placement scene *place position image*, which is generated by the second ResNet stream. This operation computes a dense affordance map indicating the quality of placing the object at every pixel with various discrete rotations.

Data Augmentation To improve model robustness and generalization, we apply random 2D augmentations (translations and rotations) to the NeRF-rendered images and their corresponding pixel-based labels during training.

3.3 Evaluation

To assess the performance of our trained agent, the evaluation is conducted on a set of unseen test episodes, which are generated using the same pipeline as the training data but with different random seeds to ensure no overlap. The evaluation process for a single test episode is as follows:

Environment Setup The simulation environment is initialized to the starting state of the test episode.

Mental Imagery Generation As with the training data, a scene-specific NeRF is constructed from multi-view RGB images. A dense set of V candidate orthographic views are rendered from this NeRF model.

Action Selection as Optimization The agent performs a search over all rendered views to find the optimal action. This involves a two-step process:

- **Pick Selection:** The attention model processes the *pick position image* to predict a pixel-wise affordance map. The pixel with the highest value is selected as the picking point, u^*_{pick}.
- **Place Selection:** A visual kernel is extracted from the *pick position image* at u^*_{pick}. This kernel is then convolved with all V candidate placement views to generate V distinct placement affordance maps. The agent selects the pixel and view $(u^*_{\text{place}}, v^*_{\text{place}})$ that yield the highest affordance value across all maps.

Action Execution The selected 2D pixel coordinates $(u^*_{\text{pick}}, u^*_{\text{place}})$ and their associated view orientations $(T^*_{v_{\text{pick}}}, T^*_{v_{\text{place}}})$ are converted back into 3D world coordinates $(T_{\text{pick}}, T_{\text{place}})$ using the depth information from the NeRF and the known camera parameters. This final 6-DoF action is then executed in the PyBullet environment.

4 Experiments

4.1 Environment

We conduct our simulation experiments within the Ravens [18] benchmark, a suite of challenging robotic manipulation tasks built in the PyBullet [17] physics simulator. To address tasks requiring full spatial reasoning, we evaluate our method on four 6-DoF manipulation tasks extended from this benchmark: **block-insertion**, **hanging-disks**, **stacking-objects**, and **place-red-in-green**. These tasks present significant challenges, such as dealing with cluttered environments(place-red-in-green) and generalizing to novel objects not seen during training (hanging-disks, stacking-objects).

All simulated experiments are conducted using a Universal Robot UR5e arm equipped with a suction gripper. The agent's input observations for NeRF reconstruction consist of 30 RGB images, captured from different cameras pointing toward the center of the workspace.

4.2 Evaluation Protocol

To rigorously assess our agent's capabilities, we perform evaluations under two distinct settings: **Normal** and **Challenging**. This evaluation protocol allows us to measure not only the model's performance on scenarios similar to its training data but also its crucial ability to generalize to novel and more challenging situations.

Normal: In this setting, test episodes are generated with object poses sampled from the same distribution as the training set. For 6-DoF tasks, this typically involves random rotations where roll and pitch angles (θ_x, θ_y) are within a smaller range, such as $[-\frac{\pi}{6}, \frac{\pi}{6}]$, while the yaw angle (θ_z) is sampled from $[-\pi, \pi]$.

Challenging: To create a more challenging test of generalization, this setting uses object poses with roll and pitch angles sampled from a range outside the training distribution. For instance, angles are sampled from $[-\frac{\pi}{4}, -\frac{\pi}{6}] \cup [\frac{\pi}{6}, \frac{\pi}{4}]$, representing significantly larger and previously unseen rotations. However, we only apply this setting to task **place-red-in-green** due to the suitability of task settings.

Metric We measure the task success rate over lots of evaluation runs for each task. A run is considered a success if the task goal is achieved, which is determined by the reward function of the specific task returning a value close to 1.0.

4.3 Baseline Methods

To contextualize the performance of our method, we benchmark it against 4 published baselines. Form2Fit [24] predicts the placing action by estimating dense descriptors of the scene for geometric matching. Transporter-$SE(2)$ and Transporter-$SE(3)$ are both introduced in [18]. GT-State MLP that assumes perfect object poses takes ground truth state (object poses) as inputs and trains an MLP to regress two $SE(3)$ poses for $\mathcal{T}_{\text{pick}}$ and $\mathcal{T}_{\text{place}}$.

4.4 Results

Method	block-insertion	place-red-in-greens	hanging-disks	stacking-kits	place-red-in-greens
	Normal	Normal	Normal	Normal	Chanllenging
Training iterations	40000	24000	9000	20000	40000
GT-State MLP	0	1	0	0	1
Form2Fit [24]	0	79	1	4	30
Transporter-$SE(2)$ [18]	21	74	32	18	18
Transporter-$SE(3)$ [18]	20	77	32	16	20
Ours	**80**	**86**	**40**	**32**	**50**

Table 1: **Quantitative results (trained on 10 generated senarios)**. Task success rate (%) under out-of-distribution rotations for four tasks. All models are trained on 10 expert demonstrations.

References

[1] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019.

[2] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category level object pose estimation via neural analysis-by-synthesis. In *ECCV*, 2020.

[3] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *CVPR*, 2020.

[4] Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *CoRL*, 2018.

[5] Peter Florence, Lucas Manuelli, and Russ Tedrake. Self-supervised correspondence in visuomotor policy learning. *RA-L*, 2019.

[6] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *ICRA*, 2018.

[7] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[8] Haojie Huang, Dian Wang, Robin Walter, and Robert Platt. Equivariant transporter network. In *RSS*, 2022.

[9] Dian Wang, Robin Walters, and Robert Platt. So (2) equivariant reinforcement learning. In *ICLR*, 2022.

[10] Xupeng Zhu, Dian Wang, Ondrej Biza, Guanang Su, Robin Walters, and Robert Platt. Sample efficient grasp learning using equivariant models. In *RSS*, 2022.

[11] Zhenyu Jiang, Yifeng Zhu, Maxwell Svetlik, Kuan Fang, and Yuke Zhu. Synergies between affordance and geometry: 6-dof grasp detection via implicit representations. In *RSS*, 2021.

[12] 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. In *ICRA*, 2022.

[13] Jeffrey Ichnowski*, Yahav Avigal*, Justin Kerr, and Ken Goldberg. Dex-NeRF: Using a neural radiance field to grasp transparent objects. In *CoRL*, 2020.

[14] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. NeRF-Supervision: Learning dense object descriptors from neural radiance fields. In *ICRA*, 2022.

[15] Danny Driess, Zhiao Huang, Yunzhu Li, Russ Tedrake, and Marc Toussaint. Learning multi-object dynamics with compositional neural radiance fields. *arXiv preprint arXiv:2202.11855*, 2022.

[16] Danny Driess, Ingmar Schubert, Pete Florence, Yunzhu Li, and Marc Toussaint. Reinforcement learning with neural radiance fields. *arXiv preprint arXiv:2206.01634*, 2022.

[17] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016.

[18] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, and Johnny Lee. Transporter networks: Rearranging the visual world for robotic manipulation. *CoRL*, 2020.

[19] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[20] Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971.

[21] Alan Richardson. *Mental imagery*. Springer, 2013.

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. In *SIGGRAPH*, 2022.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[24] Kevin Zakka, Andy Zeng, Johnny Lee, and Shuran Song. Form2fit: Learning shape priors for generalizable assembly from disassembly. In *ICRA*, 2020.