O^{3}Afford: One-Shot 3D Object-to-Object Affordance Grounding for Generalizable Robotic Manipulation

Tongxuan Tian[1] **Xuhui Kang**[1] **Yen-Ling Kuo**[1]
[1]University of Virginia
{nua3jz,xuhui,ylkuo}@virginia.edu

Abstract: Grounding object affordance is fundamental to robotic manipulation as it establishes the critical link between perception and action among interacting objects. However, prior works predominantly focus on predicting single-object affordance, overlooking the fact that most real-world interactions involve relationships between pairs of objects. In this work, we address the challenge of object-to-object affordance grounding under limited data contraints. Inspired by recent advances in few-shot learning with 2D vision foundation models, we propose a novel one-shot 3D object-to-object affordance learning approach for robotic manipulation. Semantic features from vision foundation models combined with point cloud representation for geometric understanding enable our one-shot learning pipeline to generalize effectively to novel objects and categories. We further integrate our 3D affordance representation with large language models (LLMs) for robotics manipulation, significantly enhancing LLMs' capability to comprehend and reason about object interactions when generating task-specific constraint functions. Our experiments on 3D object-to-object affordance grounding and robotic manipulation demonstrate that our O^{3}Afford significantly outperforms existing baselines in terms of both accuracy and generalization capability. Project website: https://o3afford.github.io.

Keywords: Affordance Grounding, One-shot Learning, Robotic Manipulation

1 Introduction

Affordance grounding enables machines to identify the functional properties of objects or environments that indicate potential interactions. It has been used to effectively communicate how objects can be interacted with in numerous downstream tasks, including human-computer interaction [1], visual understanding [2], and robotic manipulation [3]. Many prior works have investigated affordance prediction in 2D pixel space [4, 5, 6, 7, 8]. However, these works predominantly focus on single-object affordance and neglect that many daily tasks involve object-to-object interactions.

Object-to-object affordance grounding addresses functional relationships in tasks that require meaningful interactions between pairs of objects. For instance, cutting typically requires two objects: a source (e.g., a knife) and a target (e.g., an apple) to interact in a specific spatial relation. A significant challenge here is the scarcity of annotated data for model training. It is hard to collect and annotate large-scale data for object-to-object affordance grounding. O2O-Afford [9] addresses this challenge in an annotation-free manner through automatic extraction of contacts in simulation, which renders it only applicable to simple affordances such as placing and fitting. Our work aims to infer complex affordances such as pouring and cutting, which are crucial in robotic manipulation.

We seek to develop a solution for generalizable object-to-object affordance grounding with minimal supervision. Recent advances in vision foundation models (VFMs) [10, 11, 12] have demonstrated strong zero-shot and few-shot generalization capabilities across various downstream tasks [13, 14, 15, 16, 17]. A recent study [4] has demonstrated VFMs' potential for 2D affordance prediction; however, the limited geometric information available in images restricts generalization

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

across different viewpoints and object categories, hindering practical application in robotic manipulation. We hypothesize that the geometric information provided by 3D representations such as point clouds, when combined with semantic features from VFMs, can enable robust generalization across varied geometries, unseen object instances, and entirely novel object categories.

In this paper, we introduce O^3Afford, a **O**ne-shot **O**bject-to-**O**bject **Afford**ance Grounding framework with 3D semantic point cloud distilled from VFMs. We first project features from DINOv2 [10] from multi-view RGB-D observations onto the point clouds of both the source object and the target object to be manipulated. The two object point clouds, enriched with part-aware semantic features, are then processed through our bi-directional affordance discovery module, which considers the geometric contexts of both objects in both directions to predict the final 3D affordance map. We integrate the output of our affordance grounding module with large language models (LLMs) to generate constraints for optimization-based robotic manipulation, enabling enhanced spatial understanding capabilities compared to reasoning solely from images or point clouds in robotic manipulation.

This paper makes the following contributions: 1) We introduce O^3Afford, a method for learning object-to-object affordance in 3D with one-shot examples. 2) We effectively integrate object-to-object affordance into optimization-based robotics manipulation with constraints generated by LLMs. 3) We show that O^3Afford outperforms baselines of affordance prediction and robotic manipulation across a range of simulated and real-world robotic manipulation tasks, exhibiting strong semantic and geometric generalization capabilities.

2 Related Work

Affordance Grounding. Several prior works have focused on learning 2D affordance, with established datasets and benchmarks [5, 7, 18, 19]. The primary goal is to predict a 2D functional affordance map under a finite set of affordance types [6, 20, 21] or language-conditioned settings [22, 23]. Recent efforts have extended it to weak supervision [21, 20], intra-class generalization [4, 24], and open-vocabulary [25, 22] settings. However, limited geometric information in 2D images poses challenges for downstream tasks like spatial understanding and manipulation, motivating efforts to ground affordance in 3D point cloud [26], from 2D images [27, 28], or language models [29, 30]. While effective, these works neglect the reality that many daily tasks involve object-to-object interactions in which the geometry of both objects should be considered. Furthermore, the challenge in this setting is the limited annotated data for training. O2O-Afford [9] addresses this issue through automatic annotation by extracting contact information from randomized configurations in simulated environments; however, this approach limits its scalability and generalizability to broader real-world scenarios. In comparison, our work aims to use minimal annotation (i.e., one-shot) for supervision while achieving better generalization, making it more suitable for real-world robotic manipulations.

Few-shot Learning with Foundation Models. Foundation models have demonstrated remarkable capabilities as few-shot learners [31, 32]. Pretraining on internet-scale datasets has endowed these models with common sense knowledge [11, 10, 33, 34], enabling them to adapt to new tasks with a limited number of demonstrations. ZegClip [15] extends CLIP's zero-shot prediction capability to zero-shot segmentation. Han and Lim [16] leveraged features from DINOv2 and utilized LLMs as few-shot learners to achieve few-shot object detection. Several works [35, 36, 37] harness the generative capabilities of latent diffusion models for few-shot semantic segmentation. In 3D, several studies [38, 39] explored pretraining on large-scale point cloud datasets, achieving promising results in downstream tasks, e.g., one-shot part segmentation and point cloud classification. While effective, these methods still lag significantly behind the successes demonstrated in image and language domains, due to limited 3D data. We investigate one-shot learning with point clouds for object-to-object affordance grounding using VFMs, demonstrating geometric and semantic generalization.

Affordance-Guided Robotic Manipulation. Affordance, as an intermediate representation that links perception and action, has been widely used for robotic manipulation. In reinforcement learning, several works have explored using affordance as guidance for sample-efficient policy learning [40], reward shaping [41, 42, 43], and sub-goal specification [44, 42]. In imitation learning, af-

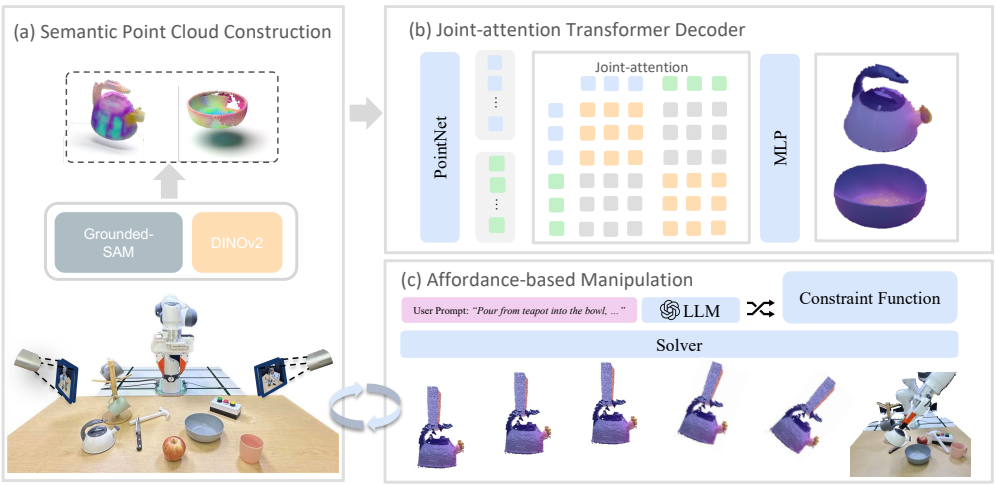

Figure 1: **O^{3}Afford framework.** O^{3}Afford first (a) constructs semantic point clouds; (b) employs joint cross-attention between the source and target objects for decoding; lastly, (c) leverages LLMs for task-relevant constraint function generation and optimizes the target pose for robot execution.

fordance is commonly used either as an additional conditional input [45, 46] or as an auxiliary learning objective [47, 48] to enhance generalization and improve sample efficiency. In planning-based manipulation, recent efforts utilized affordance as a versatile representation for low-level planning. RAM [3] adapt a retrieval-based approach for affordance transfer and action planning. [49, 50] used LLM-generated 3D affordance map or VLM-identified key points for low-level planning. RoboPoint [51] fine-tunes a VLM with affordance data to enable affordance reasoning capability for pick-and-place manipulation tasks. Our work, on the other hand, formulates planning as an optimization problem while employing LLMs to automatically create constraint functions that use the learned affordance maps, which effectively link affordance with actions.

3 Problem Formulation

We aim to address the problem of learning 3D object-to-object affordance under extreme data constraints, specifically, with only *one training sample* for each affordance category. We formulate object-to-object (O2O) affordance grounding as predicting a functional map between two object point clouds in which an O2O affordance category is defined as being uniquely identified by an interaction verb, e.g., put or pour, not by a specific object pair. Given a source object point cloud $P_s \in \mathbb{R}^{N_s \times (3+n)}$ and a target object point cloud $P_t \in \mathbb{R}^{N_t \times (3+n)}$, where N_s and N_t represent the number of points in each point cloud respectively, our goal is to predict affordance maps $A_s \in [0,1]^{N_s}$ and $A_t \in [0,1]^{N_t}$ that indicate the likelihood of O2O interaction at each point. Each point is represented by its 3D coordinates (x, y, z) and an n-dimensional semantic feature vector extracted from vision foundation models. Our model f_θ with parameters θ maps a pair of input point clouds to their respective affordance maps: $f_\theta : (\mathbf{P}^{src}, \mathbf{P}^{tgt}) \mapsto (\mathbf{A}^{src}, \mathbf{A}^{tgt})$

The training set consists of a set of K interacting object pairs, each corresponding to a distinct affordance category: $\mathcal{D}_{\text{train}} = \{(\mathbf{P}_i^{src}, \mathbf{P}_i^{tgt}, \mathbf{A}_i^{src}, \mathbf{A}_i^{tgt})\}_{i=1}^{K}$ where each $(\mathbf{P}_i^{src}, \mathbf{P}_i^{tgt})$ represents a *unique* object pair exhibiting the i-th category of affordance, and each affordance category appears *only once* in the training set. At test time, the model is evaluated in a zero-shot manner on novel object pairs exhibiting the same set of affordance categories $\mathcal{D}_{\text{test}} = \{(\mathbf{P}_j^{src}, \mathbf{P}_j^{tgt})\}_{j=1}^{K}$, but with entirely unseen objects: $\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \varnothing$.

4 Methodology

4.1 Overview

Our affordance grounding pipeline consists of three components as summarized in Fig. 1. First, we construct 3D semantic features from DINOv2 for object point clouds. Subsequently, our affor-

dance grounding module takes these semantically-enriched point clouds as input and predicts the corresponding affordance maps. To enable robotic manipulation, we leverage LLM for constraint function generation, which is integrated with optimization-based planning.

4.2 Semantic Point Cloud

To construct our 3D feature field for point cloud scenes, we draw inspiration from D^{3}Field [14] for projecting 2D semantic features into 3D space using DINOv2 [10]. In their framework, multi-view RGBD images are processed to extract DINOv2 features, which are then projected onto arbitrary 3D coordinates by mapping them to each camera's image space, interpolating features, and fusing them across views. Specifically, for a 3D point $\mathbf{x}$, we compute its projection $\mathbf{u}_i$ in the i-th camera view, determine the truncated depth difference $d_i = r_i - r_i'$ (where r_i is sensor-captured depth and r_i' is the interpolated depth), and assign visibility v_i and weight w_i to prioritize points near the surface. These weights guide the fusion of semantic features $\mathbf{f}_i$ and instance masks $\mathbf{p}_i$ across K views, yielding a unified 3D descriptor field. We adapt this method by aligning multi-view RGB observations with the point cloud geometry, projecting DINOv2 features onto the 3D points, and fusing them to encode semantic information directly onto the point cloud representation. This approach enables efficient and generalizable feature representation without additional training, supporting robust scene understanding in our work.

4.3 One-Shot Affordance Grounding

To enable effective affordance prediction from point clouds, we propose a joint-attention transformer decoder to predict affordance, which enables effective feature interaction across objects. Our network processes paired point clouds representing a source object $\mathbf{P}^{src} \in \mathbb{R}^{B\times N\times 3}$ and a target object $\mathbf{P}^{tgt} \in \mathbb{R}^{B\times N\times 3}$, along with corresponding visual features extracted from DINOv2 [10], denoted by $\mathbf{F}^{\text{src}}, \mathbf{F}^{\text{tgt}} \in \mathbb{R}^{B\times N\times 1024}$. The output consists of per-point affordance scores optimized via binary cross-entropy (BCE) loss against ground-truth annotations during training. Our model f_θ comprises two primary components: (1) a **Point Cloud Encoder**, (2) a **Joint-attention Transformer Decoder**.

Point Cloud Encoder. The point cloud encoder first jointly processes concatenated point clouds and their DINOv2-derived features. Given the combined input $\mathbf{P} = [\mathbf{P}^{src}, \mathbf{P}^{tgt}] \in \mathbb{R}^{2B\times N\times 3}$ and features $\mathbf{F} = [\mathbf{F}^{\text{src}}, \mathbf{F}^{\text{tgt}}] \in \mathbb{R}^{2B\times N\times 1024}$, we first tokenize the input using farthest point sampling (FPS) to select T patch centers, then employ k-nearest neighbors for each patch center to construct the corresponding patches. To effectively aggregate local geometric information into compact patch-level feature representations, we employ PointNet for feature extraction. This process finally produces tokenized input $\mathbf{Z} \in \mathbb{R}^{2B\times T\times 512}$. The object tokens are then concatenated with one-hot embeddings that distinguish between source object tokens and target object tokens for subsequent processing.

Joint-attention Transformer Decoder. We apply a joint attention transformer to effectively capture geometric and semantic contextual dependencies across source and target objects. Specifically, we implement a cross-attention mechanism to enable dynamic feature interaction between objects. Our multi-head attention module with 8 heads captures these contextual dependencies:

$$\mathbf{A}^{\text{src}} = \text{CrossAttention}(\mathbf{Z}^{\text{src}}, \mathbf{Z}^{\text{tgt}}, \mathbf{Z}^{\text{tgt}}), \mathbf{A}^{\text{tgt}} = \text{CrossAttention}(\mathbf{Z}^{\text{tgt}}, \mathbf{Z}^{\text{src}}, \mathbf{Z}^{\text{src}}).$$

This bidirectional interaction module effectively encodes the complementary affordance relationships between interacting objects.

To generate per-point predictions, we interpolate the final patch-level embeddings back to individual points in the original point clouds using nearest-neighbor interpolation based on distance to patch centroids. This process yields dense point-level embeddings $\mathbf{E}^{\text{src}}, \mathbf{E}^{\text{tgt}} \in \mathbb{R}^{B\times N\times 512}$. These dense embeddings are concatenated and processed through a lightweight MLP projection head to produce the final affordance map $\mathbf{A} \in [0,1]^{2B\times K\times N}$, representing predicted interaction probabilities, where the dimension K corresponds to separate channels for each affordance type prediction.

Training and Optimization. In training, we optimize our affordance grounding network using the binary cross-entropy (BCE) loss:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$$

During inference, the network directly outputs per-point affordance predictions, facilitating effective robotic manipulation planning and execution in real-world scenarios.

4.4 Affordance-Based Manipulation

Given the predicted affordance on the objects' point clouds, which clearly reveals the spatial relationship between the two objects (e.g., how they should make contact), we formulate affordance-based manipulation as a constraint-based optimization problem. Specifically, given a source object point cloud $\mathbf{P}^{src}$ and a target object point cloud $\mathbf{P}^{tgt}$, along with their respective affordance maps $\mathbf{A}^{src}$ and $\mathbf{A}^{tgt}$, we optimize a 6-DoF transformation $\mathbf{T} \in \text{SE}(3)$ applied to the source object that aligns the objects for the intended task while satisfying several constraints formulated as:

$$\min_{\mathbf{T}\in\text{SE}(3)} \quad \sum_{i=1}^{N} \lambda_i \cdot \mathcal{S}_i(\mathbf{P}^{src}, \mathbf{A}^{src}, \mathbf{P}^{tgt}, \mathbf{A}^{tgt}, \mathbf{T}) \tag{1}$$

where $\mathcal{S}_i$ represents the i-th constraint function evaluating specific aspects of the task requirements (e.g., collision avoidance, etc.) and λ_i is the weight reflecting the importance of each objective.

Constraint Generation with LLMs. To make our system versatile and scalable, we utilize LLMs to generate task-specific constraint functions instead of manually designed rules. Equipped with commonsense knowledge of daily tasks, LLMs can reason about how objects should interact and translate high-level semantics (e.g., "pouring", "cutting") into concrete geometric objectives. Specifically, we prompt the LLM with a task description, and it outputs Python functions that take the object point clouds and their affordance maps as input. These functions serve as score terms $\mathcal{S}_i$, capturing task-relevant spatial relationships such as alignment, contact, or insertion. The resulting constraint scores are then combined in the framework of Eq. 4.4 to optimize the final object pose $\mathbf{T} \in \text{SE}(3)$ with off-the-shelf solvers. We provide a detailed prompt template and example generated score functions in the Appendix B.3 and B.4.

5 Experiments

We evaluate our method for both affordance prediction and robotic manipulation tasks. We aim to answer three key research questions: **(1)** How effectively does our method perform in object-to-object affordance grounding tasks? **(2)** To what extent can our method generalize when training with only a single example for each affordance type? **(3)** How effectively can our method improve downstream robotic manipulation? We first demonstrate our experiment settings in Sec. 5.1. We then address the above three questions through two stages: evaluating the accuracy and generalization capability of affordance grounding (Sec. 5.2) and validating our approach in both simulation and real-world robotic manipulation (Sec. 5.3).

5.1 Experiment Setup

Given the absence of high-quality object-to-object affordance grounding datasets, we annotate and construct our own dataset in simulation using SAPIEN [52]. The affordance map is generated in two steps: first, we have several user-assigned contact points on the point cloud, and then propagate the affordance label to other points based on distance following [26].

We conduct robot manipulation experiments in both simulated environments using SAPIEN [52] and the real-world Franka Research 3. In simulation, we position four stereo-depth sensors [53] from different viewpoints around the workspace. In the real world, we position two Orbbec Femto

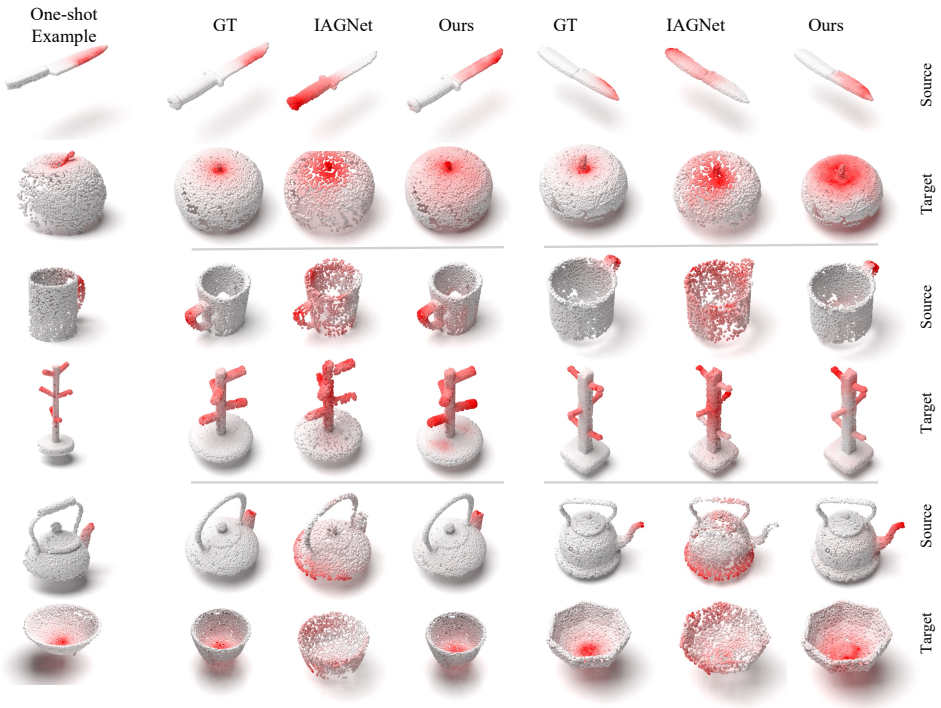

Figure 2: **Qualitative examples of the predicted object-to-object affordance in O^{3}Afford.**

Bolt cameras to get RGB-D observations. We employ the LLM GPT-4o [54] from OpenAI for constraint function generation in planning. We design five tasks that require two objects to interact meaningfully with each other: *pouring from teapot into bowl*, *inserting toast into toaster*, *pressing the button with hammer*, *hanging mug onto mug tree* and *cutting apple with knife* to evaluate our system's performance in varying contact geometries and spatial relationships between objects. Our training dataset consists of a single pair of interacting objects for each affordance type.

5.2 Affordance Grounding

5.2.1 Main Results

Baselines. We evaluate our affordance grounding module against 3 baselines: **O2O-Afford** [9], a training-based 3D object-to-object method; **IAGNet [27]**, a method learning from 2D images; and **RoboPoint [51]**, a VLM-based approach. We adopt four metrics during evaluation: **aIOU** [55], **SIMilarity** [56], **MAE** [57], and **AUC** [58], which systematically measure spatial overlap, visual similarity, pixel-wise error, and overall prediction quality.

Results. Tab. 1 shows the quantitative comparison. We also present our qualitative examples in Fig. 2. Each row in Fig. 2 represents a distinct affordance type, with three different affordance types illustrated. For each affordance type, we present two corresponding rows: the upper showing the source object and the lower displaying the target object. We observed that RoboPoint, as a vision-language model, demonstrates capability in object localization but lacks the precision to infer fine-grained affordance regions on objects. IAGNet exhibits the best performance among all baselines but suffers significantly from overfitting due to the one-shot training paradigm. Our method significantly outperforms all baselines and demonstrates robust generalization.

Method	↑ IOU	↑ SIM	↓ MAE	↑ AUC
O2O-Afford	14.31	0.5123	0.1219	74.29
IAGNet	16.89	0.5574	0.1402	73.30
RoboPoint	11.84	0.4376	0.3344	59.78
Ours	**26.19**	**0.6387**	**0.0612**	**96.00**

Table 1: **Quantitative comparison on object-to-object affordance grounding methods.**

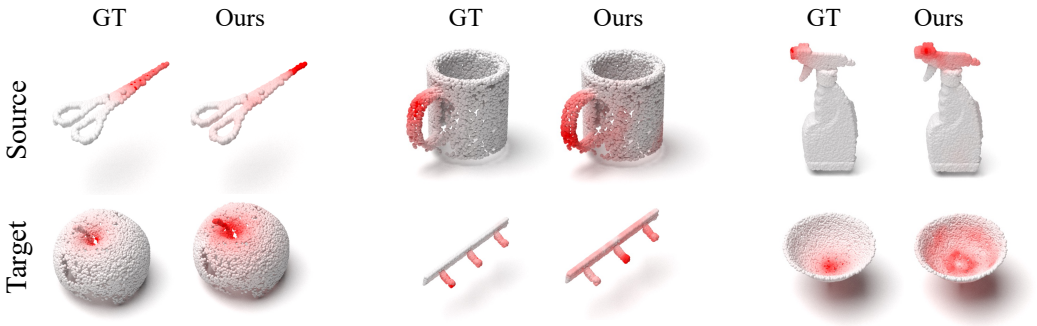

Figure 4: **Qualitative Results of Unseen Object Category.**

5.2.2 Generalization Analysis

Beyond the robust intra-class generalization under extreme data constraints demonstrated in Sec. 5.2.1, we also analyze the generalization capability of O^3Afford in this section. We study two forms of generalization: *generalization to different levels of occlusion* and *cross-category generalization*, both of which are prevalent and essential in real-world manipulation scenarios.

Occlusion Generalization. Rather than simulating occlusion by directly dropping points from the object point cloud, we evaluate generalization in a more realistic setting where objects are occluded by other objects (e.g., cubes). We exclude RoboPoint in the comparison, as it is not a training-based method. Fig. 3 shows how aIOU and SIM change at different levels of occlusion. Our method exhibits the least performance drop under varying levels of occlusion, whereas the two baselines suffer significant declines as occlusion increases due to larger geometric deviations from the training distribution.

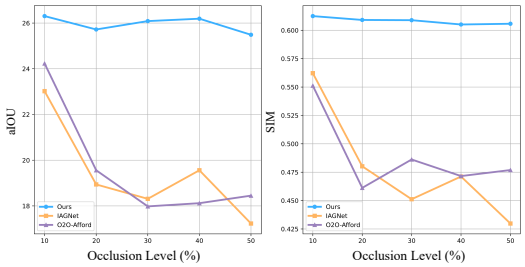

Figure 3: **Quantitative results under different occlusion levels**. Occlusion level (x-axis) ranges from 10% to 50% occluded point clouds.

Category-level Generalization. We consider object categories that are entirely unseen during training but may exhibit similar semantic or geometric functionality in manipulation tasks. Fig. 4 shows the example affordance predictions in a scissor for cutting, a coat rack for hanging, and a spray bottle for pouring. These are unseen categories but exhibit similar semantic functionality. Our model successfully predicts the affordance maps for these novel objects. This capability stems from both DINOv2's semantic features for identifying parts with similar semantics and the PointNet encoder for identifying similar local geometry, where we observe that latent patches with similar local geometries tend to be clustered together, as visualized in Appendix A.5.

Method	Pour			Hang			Press			Insert			Cut		
	2-view	3-view	4-view	2-view	3-view	4-view	2-view	3-view	4-view	2-view	3-view	4-view	2-view	3-view	4-view
Baseline	0/10	2/10	2/10	0/10	1/10	0/10	4/10	5/10	5/10	1/10	2/10	5/10	3/10	5/10	4/10
Rekep	2/10	3/10	3/10	1/10	0/10	2/10	4/10	4/10	6/10	3/10	2/10	3/10	4/10	4/10	4/10
Ours	**6/10**	**8/10**	**8/10**	**3/10**	**3/10**	**5/10**	**9/10**	**9/10**	**9/10**	**5/10**	**7/10**	**8/10**	**8/10**	**7/10**	**8/10**

Table 2: **Success rate of manipulation tasks in simulation.** Each method is evaluated under 2-view, 3-view, and 4-view setups over 10 trials.

5.3 Affordance-Guided Manipulation

Baselines. We evaluate our approach for robotic manipulation against two methods: (1) **ReKep**, which solves keypoint constraints through optimization for action planning; and (2) **Baseline, an ablated version** of our method that plans directly from object point clouds. The evaluation is conducted across 5 tasks, with ten trials per task. We record the success rate as the primary metric.

Figure 5: **Example execution of real-world manipulation tasks.** (Top) inserting, hanging, and cutting; (Bottom) pouring and pressing. The colors on the point clouds indicate the predicted affordance values as mapped to the colorbar on the right.

Results in simulation. Tab. 2 shows the success rate of each method. We evaluate in settings with different numbers of camera views, ranging from four views, representing full observation, to as few as two views, representing partial observation with significant occlusion challenges. Our method achieves the best overall performance, exhibiting the smallest performance drop when tested with fewer views, which highlights its robustness to occlusion. In contrast, the baseline planning approach without affordance performs the worst due to its lack of awareness of object relationships during manipulation, underscoring the importance of incorporating affordance. Compared to our method, ReKep exhibits two main drawbacks. First, although ReKep can propose keypoints on objects for planning, it does not guarantee the task-awareness of the proposed keypoints, as evidenced by its lower performance compared to ours under the full observation setting. Second, it is sensitive to occlusion, with its performance dropping significantly as the number of observation views decreases, making accurate keypoint proposal and tracking more challenging.

Results in real world. We evaluate our approach on 5 real-world manipulation tasks compared against the same baselines. Tab. 3 shows their success rates.

The result shows that affordance, as a mid-level representation, significantly enhances manipulation success rates. In common tasks requiring two-object interaction, our method achieves approximately an 80% success rate, while the baseline exhibits considerably poorer performance due to its inability to recognize functional properties from object point clouds. In more complex and extended-horizon tasks such as hanging, the baseline fails in all trials, whereas our method maintains approximately 50% success rate. We present qualitative examples in Fig. 5. The results demonstrate that affordance representation significantly enhances the feasibility of the generated constraint function for more reasonable interaction, as evidenced by the resulting coherent manipulation sequences.

Method	Pour	Hang	Press	Insert	Cut
Baseline	2/10	0/10	2/10	3/10	3/10
ReKep	3/10	2/10	5/10	4/10	5/10
Ours	**8/10**	**5/10**	**9/10**	**8/10**	**9/10**

Table 3: **Success rate for different tasks on the real robot.** Each task is evaluated over 10 trials. Orange indicates improvements over the baseline.

6 Conclusion

We introduced O^{3}Afford, a one-shot learning framework that addresses object-to-object affordance grounding under extreme data constraints. By distilling semantic features from VFMs onto point clouds, our approach achieves effective intra-class and category-level generalization while maintaining robustness to occlusion. When integrated with LLMs to generate constraints for optimization-based robotic manipulation, our system demonstrates exceptional performance in complex real-world manipulation tasks, highlighting O^{3}Afford's potential for broader scenarios. Future work includes incorporating language instructions into our object-to-object affordance prediction pipeline to enable open-vocabulary affordance grounding.

Limitations

While O^3Afford shows promise for figuring out where one 3D object can interact with another after seeing just one example, it still has some limitations. First, the system currently does not use language instructions to directly guide where it predicts the interaction should happen, which limits its ability to follow specific human directions. Second, the system's accuracy can decrease when parts of an object block the view of other parts (self-occlusion). This can result in an incomplete 3D model of the object and lead to incorrect predictions about interaction points on hidden parts, even if those hidden parts are important for the task. Finally, the practical performance of O^3Afford is constrained by the inherent limitations and noise present in current RGB-D sensing technologies, whether based on stereo vision, Time-of-Flight (ToF), structured light, or others. These sensors often produce imperfect depth data under various real-world conditions. For instance, stereo systems can struggle with textureless or reflective surfaces, while ToF sensors' accuracy can be affected by the target object's surface material (impacting light reflection) and potentially suffer from multipath interference in cluttered scenes. This underlying noise and potential inaccuracies in the input depth data can degrade the quality of the 3D point clouds used by our method, consequently limiting the precision and reliability of the final affordance predictions.

7 Acknowledgments

This research was partly supported by Delta Electronics Inc. and NSF CMMI-2443076. We acknowledge Research Computing at the University of Virginia for providing the computational resources that made the results in this work possible. We would like to thank Zezhou Cheng and Henry Kautz for their valuable feedback on the early draft of the paper.

References

[1] V. Kaptelinin and B. Nardi. Affordances in hci: toward a mediated action perspective. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 967–976, 2012.

[2] A. Delitzas, A. Takmaz, F. Tombari, R. Sumner, M. Pollefeys, and F. Engelmann. Scenefun3d: fine-grained functionality and affordance understanding in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14531–14542, 2024.

[3] Y. Kuang, J. Ye, H. Geng, J. Mao, C. Deng, L. Guibas, H. Wang, and Y. Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation. *arXiv preprint arXiv:2407.04689*, 2024.

[4] G. Li, D. Sun, L. Sevilla-Lara, and V. Jampani. One-shot open affordance learning with foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3086–3096, 2024.

[5] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao. Learning affordance grounding from exocentric images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2252–2261, 2022.

[6] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao. Leverage interactive affinity for affordance learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6809–6819, 2023.

[7] W. Zhai, H. Luo, J. Zhang, Y. Cao, and D. Tao. One-shot object affordance detection in the wild. *International Journal of Computer Vision*, 130(10):2472–2500, 2022.

[8] X. Zhao, Y. Cao, and Y. Kang. Object affordance detection with relationship-aware network. *Neural Computing and Applications*, 32(18):14321–14333, 2020.

[9] K. Mo, Y. Qin, F. Xiang, H. Su, and L. Guibas. O2o-afford: Annotation-free large-scale object-object affordance learning. In *Conference on robot learning*, pages 1666–1677. PMLR, 2022.

[10] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[14] Y. Wang, M. Zhang, Z. Li, T. Kelestemur, K. Driggs-Campbell, J. Wu, L. Fei-Fei, and Y. Li. D^3fields: Dynamic 3d descriptor fields for zero-shot generalizable rearrangement. In *8th Annual Conference on Robot Learning*, 2024.

[15] Z. Zhou, Y. Lei, B. Zhang, L. Liu, and Y. Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[16] G. Han and S.-N. Lim. Few-shot object detection with foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28608–28618, June 2024.

[17] J. Zhang, C. Herrmann, J. Hur, L. Polania Cabrera, V. Jampani, D. Sun, and M.-H. Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems*, 36:45533–45547, 2023.

[18] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 1374–1381. IEEE, 2015.

[19] J. Sawatzky, A. Srikantha, and J. Gall. Weakly supervised affordance detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2795–2804, 2017.

[20] J. H. Jang, H. Seo, and S. Y. Chun. Intra: Interaction relationship-aware weakly supervised affordance grounding. In *European Conference on Computer Vision*, pages 18–34. Springer, 2024.

[21] G. Li, V. Jampani, D. Sun, and L. Sevilla-Lara. Locate: Localize and transfer object parts for weakly supervised affordance grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10922–10931, 2023.

[22] S. Qian, W. Chen, M. Bai, X. Zhou, Z. Tu, and L. E. Li. Affordancellm: Grounding affordance from vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7587–7597, 2024.

[23] C. Chen, Y. Cong, and Z. Kan. Worldafford: Affordance grounding based on natural language instructions. In *2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 822–828. IEEE, 2024.

[24] Y. Ju, K. Hu, G. Zhang, G. Zhang, M. Jiang, and H. Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation. In *European Conference on Computer Vision*, pages 222–239. Springer, 2024.

[25] E. Tong, A. Opipari, S. Lewis, Z. Zeng, and O. C. Jenkins. Oval-prompt: Open-vocabulary affordance localization for robot manipulation through llm affordance-grounding. *arXiv preprint arXiv:2404.11000*, 2024.

[26] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia. 3d affordancenet: A benchmark for visual object affordance understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1778–1787, June 2021.

[27] Y. Yang, W. Zhai, H. Luo, Y. Cao, J. Luo, and Z.-J. Zha. Grounding 3d object affordance from 2d interactions in images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10905–10915, October 2023.

[28] Y. Yang, W. Zhai, H. Luo, Y. Cao, and Z.-J. Zha. Lemon: Learning 3d human-object interaction relation from 2d images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16284–16295, 2024.

[29] H. Chu, X. Deng, Q. Lv, X. Chen, Y. Li, J. Hao, and L. Nie. 3d-affordancellm: Harnessing large language models for open-vocabulary affordance detection in 3d worlds. *arXiv preprint arXiv:2502.20041*, 2025.

[30] Y. Li, N. Zhao, J. Xiao, C. Feng, X. Wang, and T.-s. Chua. Laso: Language-guided affordance segmentation on 3d object. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14251–14260, June 2024.

[31] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[32] H. Song, L. Dong, W.-N. Zhang, T. Liu, and F. Wei. Clip models are few-shot learners: Empirical studies on vqa and visual entailment. *arXiv preprint arXiv:2203.07190*, 2022.

[33] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[34] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[35] R.-Z. Qiu, Y.-X. Wang, and K. Hauser. Aligndiff: aligning diffusion models for general few-shot segmentation. In *European Conference on Computer Vision*, pages 384–400. Springer, 2024.

[36] M. Zhu, Y. Liu, Z. Luo, C. Jing, H. Chen, G. Xu, X. Wang, and C. Shen. Unleashing the potential of the diffusion model in few-shot semantic segmentation. *arXiv preprint arXiv:2410.02369*, 2024.

[37] W. Tan, S. Chen, and B. Yan. Diffss: Diffusion model for few-shot semantic segmentation. *arXiv preprint arXiv:2307.00773*, 2023.

[38] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang. Uni3d: Exploring unified 3d representation at scale. In *International Conference on Learning Representations (ICLR)*, 2024.

[39] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li. Pointclip: Point cloud understanding by clip. *arXiv preprint arXiv:2112.02413*, 2021.

[40] J. Borja-Diaz, O. Mees, G. Kalweit, L. Hermann, J. Boedecker, and W. Burgard. Affordance learning from play for sample-efficient policy learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6372–6378. IEEE, 2022.

[41] O. Y. Lee, A. Xie, K. Fang, K. Pertsch, and C. Finn. Affordance-guided reinforcement learning via visual prompting. *arXiv preprint arXiv:2407.10341*, 2024.

[42] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023.

[43] Y. Geng, B. An, H. Geng, Y. Chen, Y. Yang, and H. Dong. Rlafford: End-to-end affordance learning for robotic manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5880–5886. IEEE, 2023.

[44] K. Fang, P. Yin, A. Nair, H. R. Walke, G. Yan, and S. Levine. Generalization with lossy affordances: Leveraging broad offline data for learning visuomotor tasks. In *Conference on Robot Learning*, pages 106–117. PMLR, 2023.

[45] S. Nasiriany, S. Kirmani, T. Ding, L. Smith, Y. Zhu, D. Driess, D. Sadigh, and T. Xiao. Rt-affordance: Affordances are versatile intermediate representations for robot manipulation. *arXiv preprint arXiv:2411.02704*, 2024.

[46] S. Wu, Y. Zhu, Y. Huang, K. Zhu, J. Gu, J. Yu, Y. Shi, and J. Wang. Afforddp: Generalizable diffusion policy with transferable affordance. *arXiv preprint arXiv:2412.03142*, 2024.

[47] Y. Zha, S. Bhambri, and L. Guan. Contrastively learning visual attention as affordance cues from demonstrations for robotic grasping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7835–7842. IEEE, 2021.

[48] O. Mees, J. Borja-Diaz, and W. Burgard. Grounding language with visual affordances over unstructured data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[49] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 540–562. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/huang23b.html.

[50] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024.

[51] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

[52] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[53] I. Liu, E. Yang, J. Tao, R. Chen, X. Zhang, Q. Ran, Z. Liu, and H. Su. Activezero: Mixed domain learning for active stereovision with zero annotation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13033–13042, 2022.

[54] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[55] M. A. Rahman and Y. Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International symposium on visual computing*, pages 234–244. Springer, 2016.

[56] M. J. Swain and D. H. Ballard. Color indexing. *International journal of computer vision*, 7(1): 11–32, 1991.

[57] C. J. Willmott and K. Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1): 79–82, 2005.

[58] J. M. Lobo, A. Jiménez-Valverde, and R. Real. Auc: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, 17(2):145–151, 2008.

[59] Y. Tang, W. Huang, Y. Wang, C. Li, R. Yuan, R. Zhang, J. Wu, and L. Fei-Fei. Uad: Unsupervised affordance distillation for generalization in robotic manipulation. *arXiv preprint arXiv:2506.09284*, 2025.

[60] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

Appendices

A Additional Results

A.1 Ablation

We present ablation results on the model architecture to clarify the specific contributions of each module within our affordance model in Table 4. Specifically, we ablated three components: DINOv2 features (w/o DINOv2), PointNet encoder (w/o PointNet), and joint-attention (w/o Joint-Attn).

Method	↑ IOU	↑ SIM	↓ MAE	↑ AUC
w/o DINOv2	6.59	0.5261	0.0384	69.16
w/o PointNet	20.06	0.5923	0.0541	98.90
w/o Joint-Attn	17.15	0.5717	0.0707	95.55
Ours	**26.19**	**0.6387**	**0.0612**	**96.00**

Table 4: **Model Ablations.**

A.2 Occlusion Qualitative Results

Fig. 6 presents the qualitative examples of our occlusion experiment in the simulated environment. We show that our method, leveraging the generalizable semantic features of DINOv2, demonstrates robustness under varying levels of occlusion. This includes extreme case with up to 50% point cloud occluded which is considered the most difficult scenario in real-world applications.

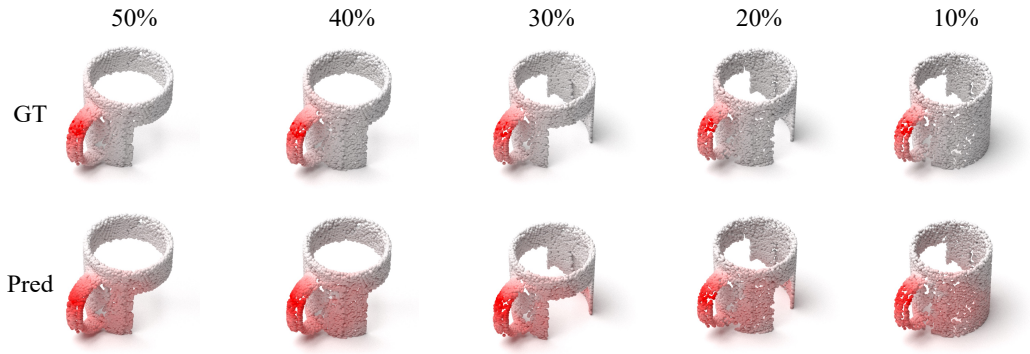

Figure 6: **Qualitative examples of different occlusion level.**

A.3 RoboPoint Qualitative Results

Fig. 7 shows the affordance prediction from RoboPoint [51]. While RoboPoint. As a vision-language model, RoboPoint demonstrates strong object localization capabilities, but it lacks the precision required to infer fine-grained affordance regions on objects, as illustrated in Fig. 7. Most of its predictions successfully fall onto the area of the object, but don't onto the correct part of the object which hinders its application for more complex manipulation tasks.

A.4 OOAL Qualitative Results

We highlight the limitations of recent work OOAL [4], which employs vision foundation models (VFMs) for 2D affordance grounding. OOAL exhibits two main drawbacks: (1) sensitivity to viewpoint changes, and (2) a noticeable sim-to-real performance gap. We show the qualitative examples

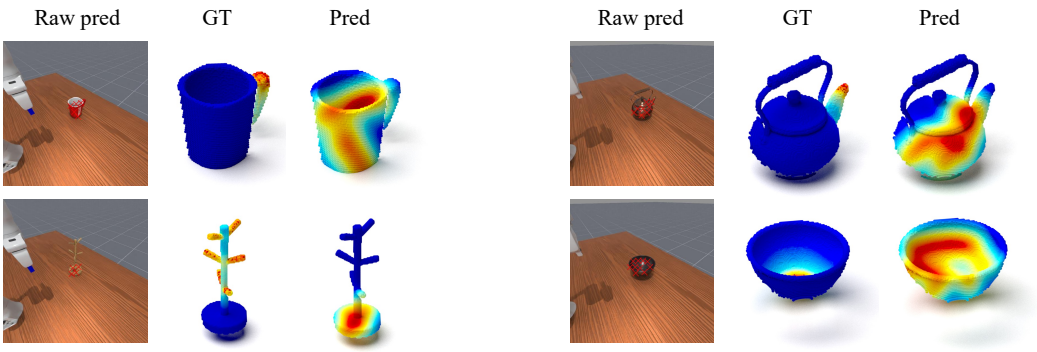

Figure 7: **Qualitative examples of RoboPoint predictions.**

in Fig. 8. We evaluate OOAL on data collected in the simulation environment using multiple viewpoints. The results reveal inconsistent predictions across viewpoints and a significant performance drop, despite the data is collected in the simulation. This highlights the benefit of affordance grounding in the 3D point cloud space, which reduces the sim-to-real gap and provides geometric awareness and robustness to viewpoint variations.

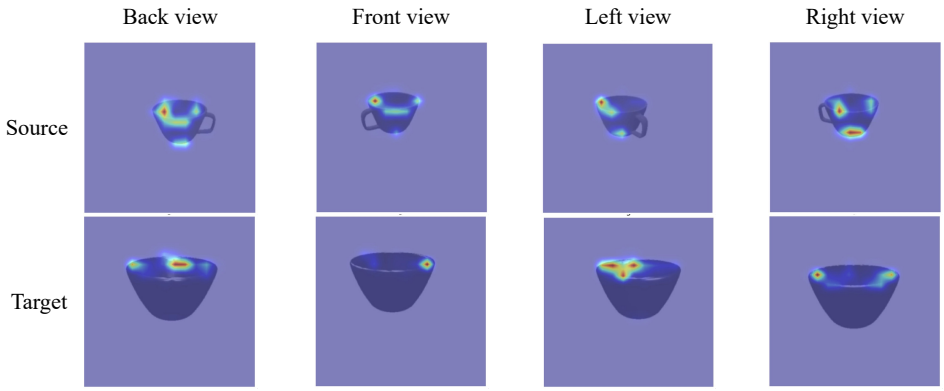

Figure 8: **Qualitative examples of OOAL predictions for four different views.** Each row renders the affordance heatmaps of the same source/target object across views.

A.5 Visualization of Latent Patch Embedding

For the PointNet-encoded patch embeddings, we observe that they encode the reusable geometric primitives across different objects as shown in our t-SNE plot (Figure 9), which shows that patches with similar local geometries tend to cluster together. We hypothesize that these patch embeddings contribute to the model's robustness to occlusion, as partial occlusion often leaves the local, functional parts unaffected.

A.6 Manipulation Qualitative Results

Fig. 10 shows the example executions of the different types of actions in the simulator using our proposed method.

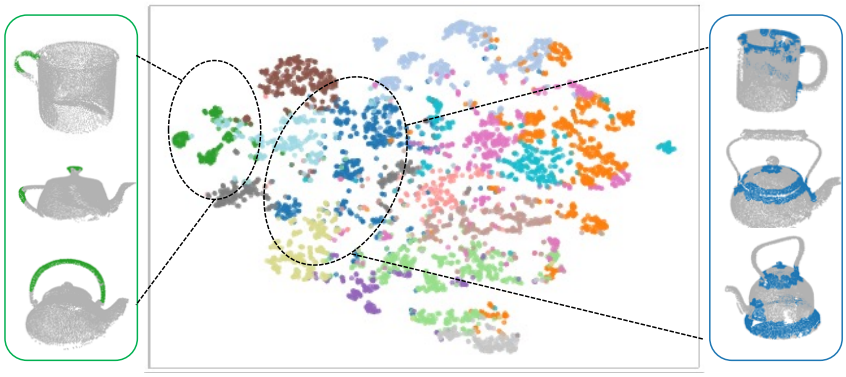

Figure 9: **t-SNE visualization of latent patch embeddings.**

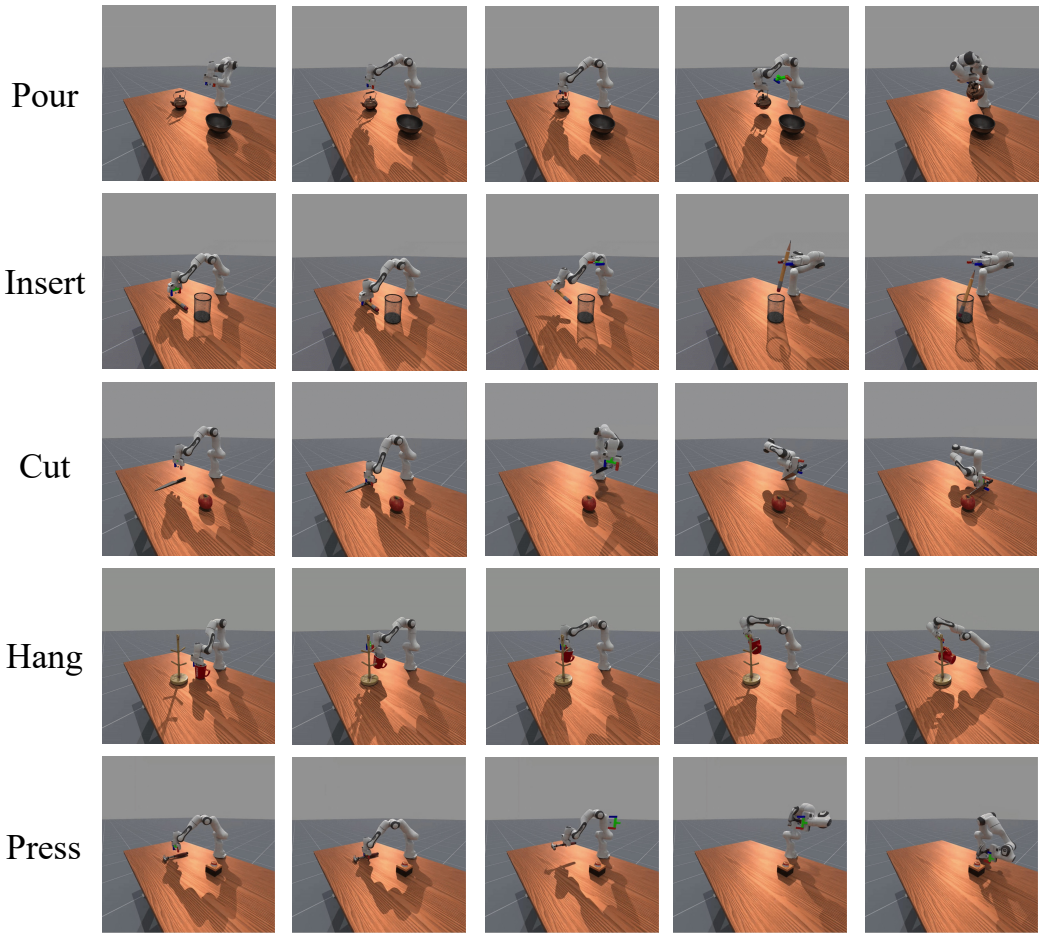

Figure 10: **Qualitative results in simulation environment.**

B Implementation Details

B.1 Baselines

RoboPoint RoboPoint [51] is a vision-language model finetuned on an image-language affordance dataset. To apply it to a point cloud, we first prompt RoboPoint to predict affordance pixels in the image, which are then projected onto the point cloud using the corresponding depth information. Across all views, we aggregate these predictions by averaging the affordance scores assigned from different views weighted by depth consistency across views.

IAGNet IAGNet [27] is a training-based method which takes an image, with an annotated object bounding box, and an object point cloud as input. To adapt the model for the object-to-object setting, we use the same interaction image, with bounding boxes annotated for both the source and target objects during training and testing, for each interacting object pair. This modification preserves the core idea of extracting semantic interaction information from 2D images without altering the model architecture.

UAD UAD [59] is a concurrent work that also projects features from DINOv2 onto the point cloud to segment out the parts and uses vision-language models for affordance auto-annotation. Our work diverges from UAD in task settings.

B.2 Model Details

Point Cloud Encoder We employ a patch-based architecture for point cloud encoding that processes the input through local grouping and feature extraction. The encoder first groups points using kNN, then processes each local patch through a specialized patch encoder, and finally incorporates positional information through learnable embeddings. This design enables effective capture of both local geometric structures and global spatial relationships.

Hyperparameter	Value
Output dimension	512
Number of groups	256
Group size	64
Group radius	0.15
Position embedding dimension	512
Patch encoder hidden dims	[784, 512]

Table 5: Hyperparameters of the point cloud encoder.

Decoder We employ a transformer-based decoder which additionally adopts joint-attention to enable feature communication between the source and target objects. We provide detailed hyperparameters in Table 6.

Hyperparameter	Value
Number of attention heads	8
Attention head dimension	512
Dropout	0.1
Cross attention dimension	512
Point cloud embedding dimension	512
Activation function	GELU
Output MLP dimensions	[512, 256]
Normalization type	LayerNorm
Normalization epsilon	1e-5

Table 6: Hyperparameters of the decoder.

B.3 Prompt Template

We use GPT-4o for constraint function generation. Note that our pipeline can be integrated with any advanced LLMs for constraint function generation. We provide an example prompt template below.

Prompt Template for Certain Constraint

You are given two 3D objects represented as point clouds. Each point is associated with an affordance score predicted by a perception model. Your task is to propose a constraint function for the specified affordance type that evaluates how well the source object can interact with the target object.

Inputs:

- `Source Object Name`: [SRC_OBJECT_NAME]
- `Target Object Name`: [TGT_OBJECT_NAME]
- `Interaction Type`: [AFFORDANCE] (e.g., pour, hang, press, cut, put, plug in)
- `Source Point Cloud`: $\{(x_i, y_i, z_i)\}_{i=1}^{N}$, with affordance scores $\{a_i\}_{i=1}^{N}$
- `Target Point Cloud`: $\{(x_j, y_j, z_j)\}_{j=1}^{M}$, with affordance scores $\{b_j\}_{j=1}^{M}$

Task: Generate a constraint function that evaluates the quality of an affordance-specific interaction between source and target objects. The function should consider high-affordance regions, interaction-specific spatial constraints, and physical plausibility.

Code Skeleton:

```
def compute_alignment_score(src_aff, tgt_aff, src_pcd, tgt_pcd):
    score = 0
    """
    # TODO: Implement affordnace alignment contraint

    return score
```

Constraints:

- The function should use the information from high-affordance regions of both objects
- The evaluation must reflect the semantic meaning of the specified affordance type
- Consideration should be given to the physical feasibility of the interaction
- All constraints should be combined into a single cost value (lower is better)

B.4 Example Constraint Functions

We provide an example of a generated constraint function in Fig. 11. A list of LLM-generated constraint functions with its corresponding weights is listed in Table 7.

```python
def compute_alignment_score(src_aff, tgt_aff, src_pcd, tgt_pcd):

        # Get the center of source object affordance regions
        src_aff_center = src_aff[0]['centroid']

        # Get the center of target object affordance regions
        tgt_aff_center = tgt_aff[0]['centroid']

        # Affordance alignment score
        # The part to be inserted should be aligned with the container
        horizontal_alignment = np.linalg.norm(
                src_insert_center[:2] - tgt_container_center[:2]
        )
        aff_alignment_score = horizontal_alignment * 10

        return aff_alignment_score
```

Figure 11: Example of the generated function for object alignment.

Task	Primary Constraints	Evaluation Components	Weights
Pour	• Position (above) • Orientation (tilted 30-60°) • Distance (medium clearance)	• Affordance alignment • Position constraint • Orientation constraint • Clearance	• 0.3 • 0.2 • 0.3 • 0.1
Hang	• Position (contact) • Orientation (hooked) • Stability (CoM below support)	• Affordance alignment • Contact quality • Stability • Collision	• 0.3 • 0.3 • 0.3 • 0.1
Cut	• Position (aligned) • Orientation (perpendicular) • Distance (contact)	• Affordance alignment • Position constraint • Collision	• 0.4 • 0.4 • 0.2
Press	• Position (aligned) • Orientation (perpendicular) • Distance (contact)	• Affordance alignment • Position constraint • Orientation constraint • Collision	• 0.4 • 0.3 • 0.2 • 0.1
Insert	• Position (inside) • Orientation (aligned) • Distance (inside container)	• Affordance alignment • Position constraint • Orientation constraint • Collision	• 0.3 • 0.4 • 0.2 • 0.1

Table 7: Generated Constraint Functions for Different Manipulation Tasks

C Failure Cases Analysis

We present an empirical analysis of failure cases in this section. Our manipulation pipeline involves three key stages: constructing a semantic point cloud from multiview RGB-D observations, grounding affordance on the partially reconstructed point cloud, and executing actions via a low-level motion planner. As a result, failure cases primarily fall into three categories: (1) inaccurate point cloud reconstruction due to sensor noise, (2) incorrect affordance predictions caused by object self-occlusion and unobservable regions, and (3) inverse kinematics (IK) failure during motion planning.

Sensor noise We use the Orbbec Femto Bolt camera, a Time-of-Flight-based sensor, for capturing RGB-D observations. It provides significantly higher-quality depth compared to stereo-based sensors. However, it may still produce erroneous depth measurements when object materials are highly reflective or when objects are too dark in color, preventing the sensor from receiving sufficient reflected light to compute accurate depth. We provide an example of an incorrectly reconstructed point cloud in Fig. 12.

Figure 12: Example of incorrect point cloud.

Object Self-occlusion Our pipeline assumes that affordance regions, such as the mouth of a teapot, are not occluded. However, in practice, these regions can be self-occluded by the object itself. For example, the bottom of a bottle is difficult to observe even with a top-down camera view. This limitation could potentially be addressed by recent advances in single-view or sparse-view reconstruction methods (e.g., TRELLIS [60]), which are orthogonal to our approach. We leave the integration of such techniques for future work.

IK Failure Since we directly optimize the final interaction pose of the object, the resulting pose may be infeasible for execution due to IK failure in the motion planner. We use a screw-based planner, and most IK failures occur when the final pose requires drastic rotational movements. We provide some visualizations of IK failure cases in Fig. 14.

Figure 13: Example of IK failure.

D Environment Setup

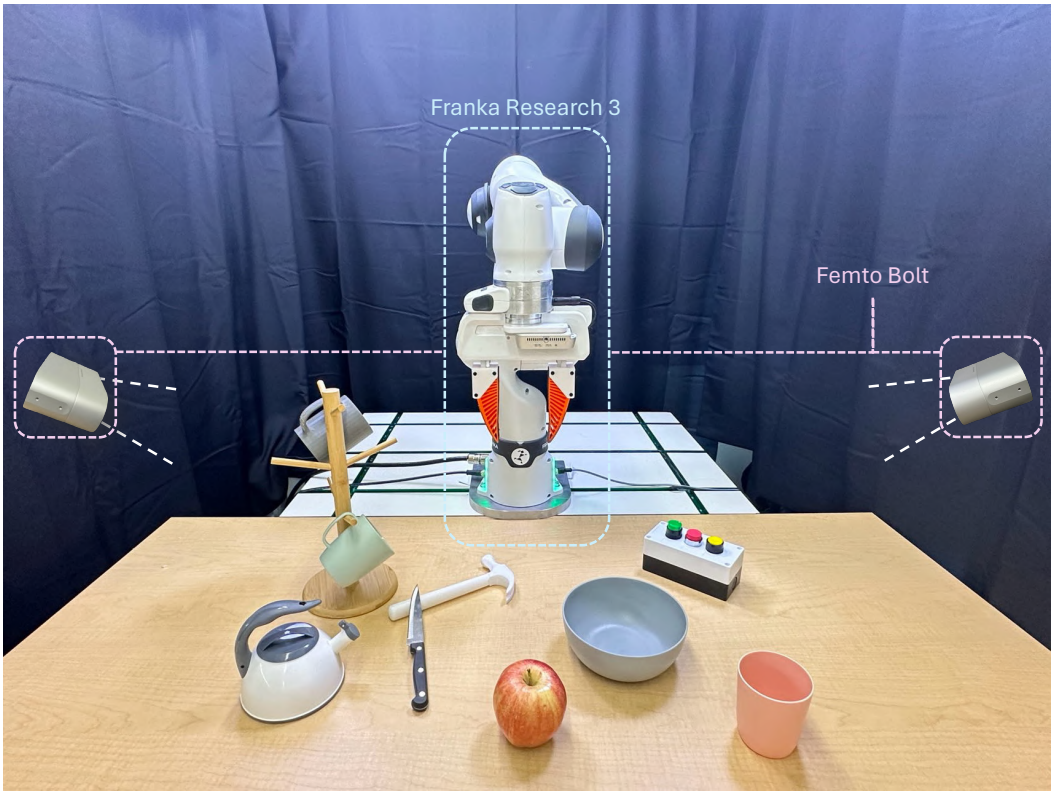

Figure 14: Real robot experiment setup.

Our task uses a Franka Research 3 robot, a 7-DOF manipulator. To capture comprehensive visual information and mitigate occlusions, we employ two Femto Bolt RGB-D cameras. These Time-

of-Flight (ToF) based cameras are positioned on the left and right sides in front of the robot's workspace, providing robust depth perception across the scene.

We utilize a diverse set of objects for our challenging object-to-object interaction tasks. These tasks require the model to not only identify the correct manipulation point on the source object but also to understand the intended interaction point on the target object. This necessitates a deep understanding of object-to-object affordance. Our tasks include:

- **Pouring water from a teapot into a bowl or:** In this task, the functional affordance is the teapot's spout, which must be correctly aligned with the receiving area of the bowl (e.g., its bottom center) or the plate. The model must comprehend this specific object-to-object affordance for pouring. For instance, while the edge of the bowl might offer a grasping affordance, it is not the correct target for the pouring task. This task critically evaluates the model's ability to understand task-specific affordance.
- **Cutting fruit with a knife:** The model must identify the sharp edge of the knife (source object affordance) and the appropriate surface of the fruit (target object affordance) to perform a cutting action. This involves understanding the functional relationship between the knife's blade and the fruit's body.
- **Pressing a button with a hammer:** This task requires the model to recognize the head of the hammer as the striking surface and the button as the target. The interaction involves a precise contact between the hammer's affordance point and the button's surface.
- **Hanging a cup onto a mugtree:** The model needs to identify the handle of the cup (source object affordance) and a free peg on the mug tree (target object affordance). Successful execution depends on aligning the cup's handle to engage with the mugtree's peg.

These tasks are designed to test the model's understanding of object affordance and how they relate to each other in the context of specific actions.