GenDP: 3D Semantic Fields for Category-Level Generalizable Diffusion Policy

Yixuan Wang[1], **Guang Yin**[2], **Binghao Huang**[1], **Tarik Kelestemur**[3], **Jiuguang Wang**[3], **Yunzhu Li**[1]

[1]Columbia University, [2]University of Illinois, Urbana-Champaign, [3]Boston Dynamics AI Institute

Contents

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

1 Method

1.1 Detailed Comparisons with Prior Semantic Fields

Compared to prior works on semantic fields and representation, our paper offers following benefits:

- **Generalizable Representation**: Some existing semantic fields or representations are trained on small-scale datasets and are challenging to transfer to novel scenes or object categories. For example, Dense Object Nets (DON) [1] are trained via contrastive learning, which requires foreground and background separation and a careful selection of positive and negative pairs. This means that deploying a DON in a new environment or on new object categories requires additional training efforts. Similarly, DSR-Net [2] trains its representation on datasets collected by heuristic policy, requiring manual designs for different scenes. Therefore, deploying DSR-Net on new objects and scenarios also requires extra training efforts. In contrast, our pipeline does not need additional training and can be applied to a diverse set of object categories and scenes by leveraging visual foundational models.
- **Efficiency**: Several prior works build semantic fields using visual foundational models. However, some are not efficient enough for imitation learning, which needs to obtain visual feedback in real time. For example, F3RM [3] takes about 90 seconds for each scene, and Feature 3DGS [4] training time is around 200 seconds, according to its Supplementary Figure A. In contrast, our 3D semantic fields only need 0.166 seconds per frame, and the total inference time for each step, including the diffusion process, is 0.226 seconds.
- **Sparse Views**: Existing semantic fields need an extensive number of camera views to build quality fields, which limits their usage in imitation learning. For example, F3RM [3] requires 50 images to scan the workspace, and Feature 3DGS [4] needs 80 views, as specified in its Supplementary Section D. ConceptGraphs [5] uses the Replica dataset, which contains 2,000 views for each scene. In contrast, we use only four cameras placed around the workspace to build our 3D semantic fields, enabling us to obtain real-time visual feedback and achieve a control frequency of 4-5 Hz.

1.2 Detailed Comparisons with Prior Diffusion Policies

Compared to prior works on diffusion policies, our paper offers the following benefits:

- **Category-Level Generalization**: Although existing diffusion policies based on 3D point clouds, such as 3D Diffuser Actor [6], demonstrate certain levels of generalization capabilities, they do not yet exhibit robust enough category-level generalization capabilities. In our work, we introduce 3D semantic fields that can significantly boost performance and facilitate category-level generalization, as shown in Figure 4 (iv) of our paper.
- **Semantic Understanding**: Prior works, such as 3DP [7], have explored using raw 3D point clouds as inputs for diffusion policies. However, we found that geometric information can be ambiguous and miss important details, leading to failure in many daily tasks. In contrast, we provide additional semantic features to guide the policy to focus on semantically meaningful object parts to accomplish various tasks.
- **Flexible Action Representation**: We also note that 3D Diffuser Actor predicts end-effector keyposes instead of a continuous trajectory, requiring manual keypose definition for each task, as shown in its Appendix A.6. Additionally, keypose representation only specifies the starting and end poses, which may not suffice for tasks requiring the control over the whole trajectory, such as scooping coffee beans out and transporting them (our Use Spoon task). In our work, we predict actions over the whole trajectory, which is more flexible for different tasks.

1.3 Detailed Explanation on Geometric Ambiguity

In our paper, geometric ambiguity means that geometric information alone cannot distinguish between different underlying object states. For example, in the Open Pen task, the raw point cloud data of the pen is ambiguous because rotating the pen 180 degrees can result in a very similar point cloud. This makes it difficult to determine the position of the cap—a critical piece of state-related information for the success of the downstream cap-opening task.

One might consider incorporating RGB information into the point cloud. However, RGB information is sensitive to appearance variations, limiting its ability for category-level generalization.

Therefore, in our work, it is the semantic information that helps the policy to disambiguate geometric ambiguity and learn the correct correlation between observations and robotic actions. We combine geometric information with the semantic field—two necessary components for category-level generalization in complex manipulation tasks. Geometric information alone is ambiguous about the underlying states, while the semantic field helps identify where the caps are on different pens with varying appearances.

1.4 Detailed Explanation on Geometric Details

Compared to RGB data, 3D geometric data is robust to environmental and appearance changes, helping the diffusion policy better generalize to unseen instances, as shown in Figure 4 and Supplementary Figure 1. However, geometric information may not capture many important details, whereas semantic fields can highlight the right details and help the policy learn the correct correlation between observations and robot actions.

For example, in the Place Can task, even if there are slight geometric variations in the can tab, it can be difficult for the policy to attend to that location. Being aware of geometric details like the can tab, the trained policy can place the can in the correct pose instead of an upside-down position. In our experiments, we show that semantic fields can significantly boost the task success rate in the Place Can and Use Toothbrush tasks.

1.5 PointNet++ Details

We use PointNet++ to encode one latent vector from the point clouds [8]. There are several key details regarding its usage.

- No BatchNorm Layer: In the diffusion policy implementation, exponential moving averages (EMA) of the model parameters are used during the testing, which is not compatible with BatchNorm.
- Point Set Abstraction: Point set abstraction is an important operation in PointNet++. In our network, we deploy three layers of point set abstraction. For the first set abstraction, we have 512 groups with a radius of 0.04 and 32 points. For the second set abstraction, we have 128 groups with a radius of 0.08 and 64 points. For the last set abstraction, all sampled points are grouped to extract one latent vector.

1.6 Reference Descriptors

For one object category, we use four images to obtain reference descriptors, as shown in Figure 1. We start by acquiring four images containing the target object category using the Internet or any arbitrary phone camera. We then obtain a single descriptor by selecting one keypoint on each of the four images and averaging their corresponding features. Therefore, we only need to select $N \times 4$ keypoints for each object category, where N is the number of reference descriptors. These reference descriptors can be used for the entire task (e.g., 100 training demonstrations and testing scenes), making the labeling effort light.

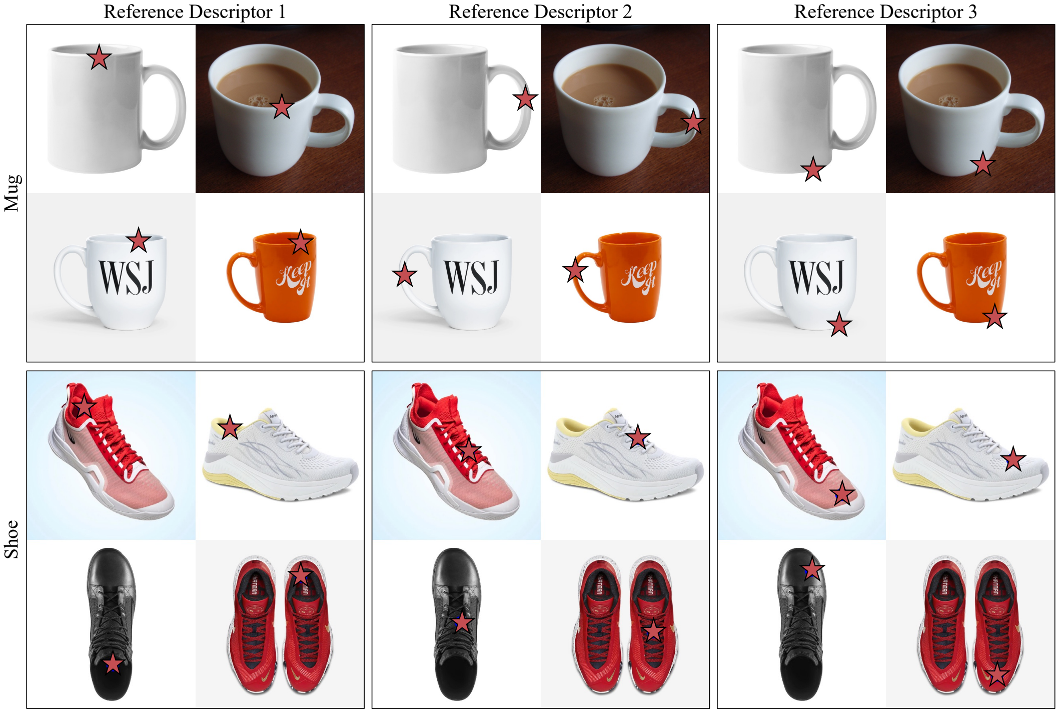

Figure 1: **Reference Images and Descriptors Example.** For one object category, we will obtain N reference descriptors, with $N = 3$ in these examples. For each descriptor, we will select one keypoint representing a specific object part from each figure. The keypoint is denoted as a red pentagram in this figure. We then average the four descriptors to obtain one reference descriptor.

2 Experiments Details

2.1 Setup

We summarize object categories and the number of object instances we considered in Table 1.

Task Category	Task Name	#Demo	Object	#Train Instances	#Test Instances
Sim.	Hang Mug	200	Mug	4	2
	Insert Pencil	100	Pencil	2	1
Geo. Details	Place Can	60	Soda Can	3	3
	Use Toothbrush	60	Toothbrush	1	4
Cat. Gen.	Align Shoes	60	Shoe	3	5
	Use Spoon	60	Spoon	2	2
Geo. Ambi.	Collect Knife	60	Knife	1	2
	Open Pen	60	Marker Pen	1	3

Table 1: **Task Details Summary.** This table summarizes our task categories, specific tasks, objects, the number of demonstrations, training instances, and testing instances. We evaluate our framework on eight tasks and eight object categories, where each object category includes several instances with diverse appearances and shapes. We divide our tasks into four categories to study our method's performance under various task types and scenarios, such as simulation and the real world.

Here is a list of task descriptions:

- **Hang Mug (Simulation)**: Hang a randomly placed mug on a fixed mug tree.
- **Pencil Insertion (Simulation)**: Pick up a pencil from the table and insert it into the pencil sharpener.

- **Collect Knife**: Collect a lying knife into an open container, and the knife direction is randomly chosen.
- **Open Pen**: Bimanually grasp the pen and open it.
- **Flip Can**: Flip a lying soda can and place it upright.
- **Use Toothbrush**: Grasp the toothbrush and spread the toothpaste on it.
- **Align Shoes**: Push shoes towards left.
- **Use Spoon**: Grasp spoon and scoop materials.

Due to space limitations, in all tables, we will use the following abbreviation:

- **Sim.**: Simulation
- **Geo. Details**: Geometry Details
- **Geo. Ambi.**: Geometry Ambiguity
- **Cat. Gen.**: Category Generalization

2.2 Demonstrations

In the simulation, we use SAPIEN to create simulation environments. We build an oracle policy to generate demonstrations in the simulation. The generated dataset contains robot states, multi-view RGBD observations, and robot actions.

In the real world, the data collection pipeline is similar, with additional calibration steps. We define the ChArUco board as the world coordinate and calibrate the camera poses to this coordinate. Then, we manually calibrate the transformation between the robot and the world coordinate. After the calibration phase, the data collection pipeline records multi-view RGBD observations, robot states, and robot actions.

2.3 Training

Before training, we preprocess multi-view RGBD observations into 3D semantic fields using the frozen DINOv2 encoder and save them to CPU memory. During training, the 3D semantic fields and corresponding ground truth actions are randomly sampled and fed into the diffusion policy. The training of the diffusion policy follows the practice outlined in the original Diffusion Policy paper [9].

2.4 Inference

During inference, the diffusion policy takes in one frame of multi-view RGBD observation and converts it into 3D semantic fields. Then, the 3D semantic fields are input into the diffusion policy to predict an action sequence.

We measure our inference time on the knife-collecting task, repeating the inference steps 10 times and computing the average time. Each inference step takes 0.226 seconds. The predicted action sequence is at 10 Hz, meaning the time difference between two consecutive actions in the action sequence is 0.1 seconds. This action sequence is then fed into the low-level controller, which operates at 50 Hz.

2.5 Evaluation

Here we list the evaluation criteria for each task:

- Hang Mug (Sim): If the final height of the mug is larger than 0.1m and lower than 0.3m, and the xy-space distance between the mug holder and the mug is lower than 0.1m, the reward is 1. Otherwise, the reward is 0.

- Pencil Insertion (Sim): If the pencil's final height is larger than 0.05m, and the xy-space angle between the pencil and the hole is smaller than $\arccos(0.8)$, the reward is 1. Otherwise, the reward is 0.

- Place Can: The policy rollout is only successful if the can is grasped and placed upright on the table.

- Use Toothbrush: The policy rollout is only successful if the toothbrush is grasped and the head of the toothbrush is aligned with the toothpaste tip.

- Align Shoes: The policy rollout is only successful if the shoe is rotated towards the left.

- Use Spoon: The policy rollout is only successful if the robot grasps the spoon, scoops the coffee beans, and pours into the bowl.

- Collect Knife: The policy rollout is only successful if the robot grasps the knife handle and puts it in the box.

- Open Pen: The policy rollout is only successful if the robot first grasps the pen body using one hand, and then pulls out the pen tap using another hand.

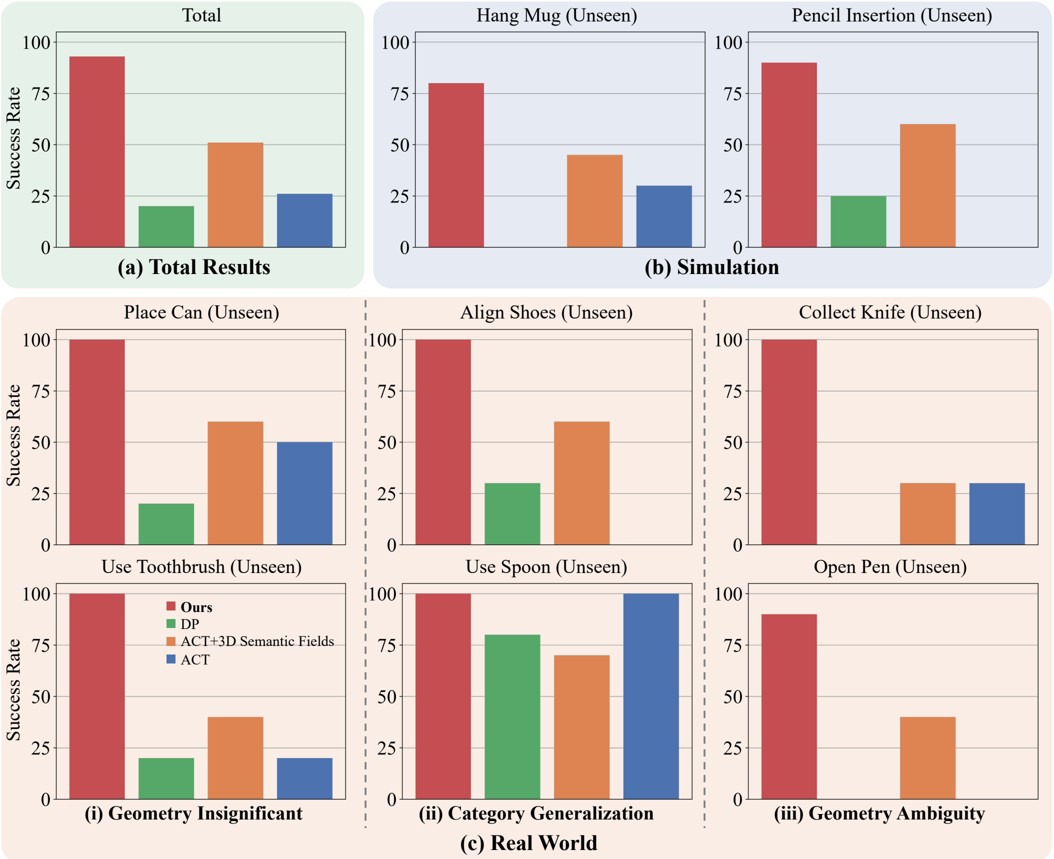

Figure 2: **Success Rate of Extension to ACT.** We measure the success rate of our method, diffusion policy, ACT with our 3D semantic fields, and ACT on unseen instances across different tasks. We found that our 3D semantic fields can help other imitation learning methods like ACT to generalize to novel instances as well. However, in practice, we choose the diffusion policy with 3D semantic fields since it performs the best across different tasks.

3 Additional Experiments

3.1 Quantitative Experiment Table

As mentioned in Section 4.2 of the main paper, we present a detailed table for Figure 4. This table provides more concrete results regarding our comparisons with the baseline methods. We can see that, although the diffusion policy performs well for the seen instances, it fails to generalize to novel instances. Compared with our method without semantics, our method also demonstrates the effectiveness of incorporating semantic information.

Task Category	Simulation			
Task Name	Hang Mug		Insert Pencil	
Instances	Seen	Unseen	Seen	Unseen
Ours	**95% (19/20)**	80% (16/20)	**95% (19/20)**	**90% (18/20)**
Ours w/o Semantics	65% (13/20)	**85% (17/20)**	40% (8/20)	40% (8/20)
Diffusion Policy	**95% (19/20)**	0% (0/20)	**95% (19/20)**	25% (5/20)
Diffusion Policy w/ RGBD	0% (0/20)	0% (0/20)	15% (3/20)	20% (4/20)

Task Category	Geometry Ambiguity		Geometry Details	
Task Name	Collect Knife	Open Pen	Place Can	Use Toothbrush
Instances	Unseen	Unseen	Unseen	Unseen
Ours	**100% (9/10)**	**90% (9/10)**	**100% (10/10)**	**100% (10/10)**
Ours w/o Semantics	50% (5/10)	50% (5/10)	60% (6/10)	30% (3/10)
Diffusion Policy	0% (0/10)	0% (0/10)	20% (2/10)	20% (2/10)
Diffusion Policy w/ RGBD	60% (6/10)	20% (2/10)	0% (0/10)	20% (2/10)

Task Category	Category Generalization		Total (Seen)	Total (Unseen)
Task Name	Collect Knife	Open Pen		
Instances	Unseen	Unseen		
Ours	**100% (10/10)**	**100% (10/10)**	**95% (38/40)**	**93% (93/100)**
Ours w/o Semantics	60% (6/10)	90% (9/10)	53% (21/40)	59% (59/100)
Diffusion Policy	30% (3/10)	80% (8/10)	**95% (38/40)**	20% (20/100)
Diffusion Policy w/ RGBD	10% (1/10)	70% (7/10)	8% (3/40)	22% (22/100)

Table 2: **Success Rate.** The method was evaluated across eight tasks. We observe that the diffusion policy performs similarly to our method in the seen environments but shows markedly worse performance in unseen instances. In addition, compared with ours without semantics, our method shows a better performance across different tasks and demonstrates the necessity of semantic information. These results underscore our policy's capability to achieve category-level generalization and encode semantic information.

3.2 Extension to ACT

We also extend our representation to other imitation learning methods, such as ACT [10]. Figure 2 shows the experiment results. First, we found that our representation helps other imitation learning algorithms generalize to novel instances. This is because raw RGB images are sensitive to environmental factors and object appearances, while our 3D semantic fields are robust against environmental changes and novel instances by explicitly using geometric and semantic information. Second, we found that our method outperforms ACT. We believe the main reason is that the diffusion policy is better at capturing multi-modal demonstration distribution compared to ACT, which is quite common in the real world. In summary, our representation can be applied to different imitation learning methods, but we choose the diffusion policy because it performs the best in practice.

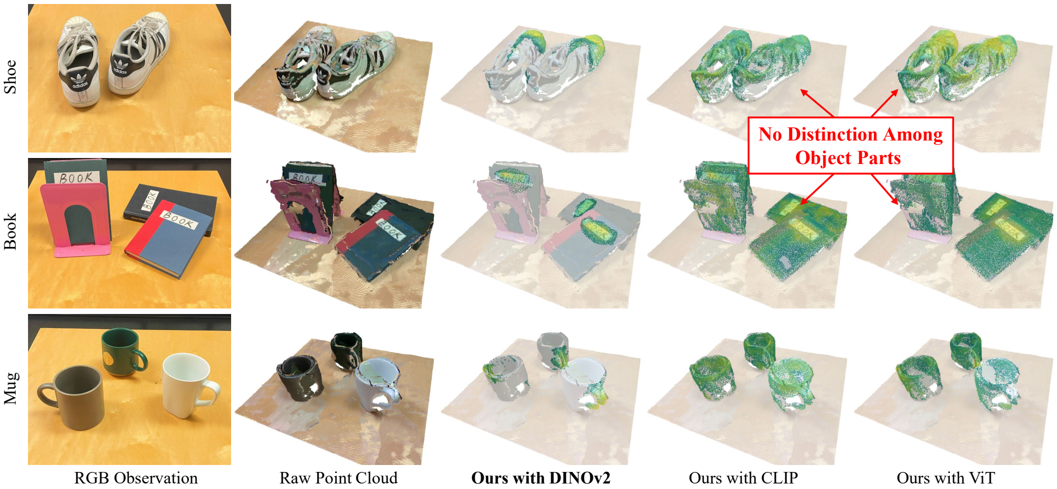

Figure 3: **3D Semantic Fields Visualization.** We visualize 3D semantic fields in the scene, similar to Figure 6 in the main paper. For ours with DINOv2, object parts from different instances are highlighted consistently. For example, shoe heads, book titles, and mug handles are highlighted in these scenes. In contrast, ours with CLIP and ours with ViT fail to highlight object parts consistently. Instead, their heatmap distributes over the scene and has no distinctions among object parts.

3.3 Ablation on Feature Extraction Backbone

Originally, we used DINOv2 as the feature backbone [11]. To study the influence of different feature backbones on performance, we replace DINOv2 with other feature backbones. Other than DINOv2, CLIP and Vision Transformer (ViT) are two of the most common image feature extractors [12, 13]. Therefore, we replace DINOv2 with CLIP and ViT and compare their performance. First, we try to qualitatively understand what the 3D semantic fields will look like with different feature backbones. Second, we quantitatively show the performance of the whole diffusion policy pipeline when the feature backbones are different.

Similar to Figure 6 in the main paper, we use the same 3D semantic field construction and visualization pipeline with different feature backbones. We visualize the 3D semantic fields of different object parts in Figure 3. We found that ours with DINOv2 can consistently highlight the object parts, such as shoe heads, book titles, and mug handles, while ours with CLIP and ours with ViT fail to do so. Without the capability to distinguish semantically meaningful parts of the object, ours with ViT and ours with CLIP fail to generalize to unseen instances, distinguish geometric ambiguities, and attend to subtle geometric details. This qualitative result justifies our usage of DINOv2 to generate useful 3D semantic fields.

Table 3 shows our quantitative results, where the experiment process is the same as the main paper. The table shows our method's average success rate is 94%, which is the best among all methods. As this table shows, ours with CLIP's average success rate is 57%, and ours with ViT's average success rate is 55%. This result is close to ours without semantics (58%) as shown in Table 1. This indicates that CLIP and ViT do not encode useful semantic information like object parts. In contrast, our method can highlight parts of the object that are critical for the task's success.

3.4 More Baselines

We compare with SORNet and DINOBot [14, 15]. We will describe how we compare our work with these two baselines in our setting respectively in Section 3.4.1 and Section 3.4.2. Then we present results and analyze them in Section 3.4.3.

3.4.1 SORNet

SORNet introduces an object-centric representation that can be generalized to objects unseen during training. Specifically, SORNet can take in observation from more than one view, where depth is

Task Category	Task Name	Instances	Ours	Ours with CLIP	Ours with ViT
Sim.	Hang Mug	Seen	**95% (19/20)**	55% (11/20)	80% (16/20)
		Unseen	**80% (16/20)**	50% (10/20)	70% (14/20)
	Insert Pencil	Seen	**95% (19/20)**	70% (14/20)	55% (11/20)
		Unseen	**90% (18/20)**	45% (9/20)	30% (6/20)
Geo. Ambi.	Collect Knife	Unseen	**100% (10/10)**	30% (3/10)	30% (3/10)
	Open Pen	Unseen	**100% (10/10)**	50% (5/10)	40% (4/10)
Geo. Details	Place Can	Unseen	**100% (10/10)**	40% (4/10)	50% (5/10)
	Use Toothbrush	Unseen	**90% (9/10)**	50% (5/10)	50% (5/10)
Cat. Gen.	Align Shoes	Unseen	**100% (10/10)**	90% (9/10)	40% (4/10)
	Use Spoon	Unseen	**100% (10/10)**	100% (10/10)	90% (9/10)
Total (Unseen)			**94% (131/140)**	57% (80/140)	55% (77/140)

Table 3: **Success Rate for Different Feature Backbones.** The method was evaluated across eight tasks and compared against different feature backbones. We use the average success rate as our evaluation metric. Compared with ours with CLIP and ours with ViT, our method consistently outperforms them. This demonstrates that DINOv2 could encode semantic information, such as parts of object that are important for task completion.

optional. It also takes in canonical object views and outputs corresponding embedding for each query object. For fair comparisons, we input RGBD observations from four viewpoints and one canonical object view into SORNet and obtain one embedding. Then we input this embedding to diffusion policy, the same as what did for our approach. We do not use pre-trained models provided by SORNet, since the observation modality and training domain are very different. Instead, we train the whole model from scratch.

3.4.2 DINOBot

DINOBot introduces a single-shot imitation learning framework that leverages visual features from foundational vision models. They first record the key end-effector pose and corresponding RGBD observations. Then they record a motion sequence. During testing, they will align current RGBD observations with recorded key RGBD observations to make sure the end-effector has arrived at the key pose. Then they will replay the motion sequence.

In our implementation, we choose one of our perspective views as the observation view. For the rest of the implementation, we follow the DINOBot.

3.4.3 Results

As shown in Table 4, SORNet performs badly for both seen and unseen instances. As mentioned by SORNet's authors, SORNet needs to be trained on a large-scale dataset with diverse object appearance and scene layouts. Specifically, they trained SORNet on two datasets, which contain 405 training objects and 330 training objects respectively. However, we only assume access to a demonstration dataset collected over one or a handful of objects. SORNet cannot effectively extract useful representation in this low-data regime, while our representation could generalize to unseen instances given the same amount of data.

As shown in Table 4, DINOBot cannot accomplish the tasks and fails in both seen and unseen scenarios. The reasons are twofold. First, DINOBot merely replays the recorded trajectory after the 6-DoF alignment, which limits its applicability in tasks that require adaptive adjustments based on the positions and shapes of objects. Second, DINOBot has only tackled single-arm manipulation tasks using a wrist-mounted camera, which restricts its extension to more complicated bimanual tasks (e.g., pen opening, as we demonstrated using our method). In contrast, our method represents

Task Category	Task Name	Instances	Ours	SORNet [14]	DINOBot [15]
Sim.	Hang Mug	Seen	**95% (19/20)**	0% (0/20)	0% (0/20)
		Unseen	**80% (16/20)**	0% (0/20)	0% (0/20)
	Insert Pencil	Seen	**95% (19/20)**	0% (0/20)	0% (0/20)
		Unseen	**90% (18/20)**	0% (0/20)	0% (0/20)
Geo. Ambi.	Collect Knife	Unseen	**100% (10/10)**	10% (1/10)	0% (0/10)
	Open Pen	Unseen	**100% (10/10)**	0% (0/10)	N/A% (N/A)
Geo. Details	Place Can	Unseen	**100% (10/10)**	20% (2/10)	0% (0/10)
	Use Toothbrush	Unseen	**90% (9/10)**	0% (0/10)	0% (0/10)
Cat. Gen.	Align Shoes	Unseen	**100% (10/10)**	0% (0/10)	0% (0/10)
	Use Spoon	Unseen	**100% (10/10)**	40% (4/10)	0% (0/10)
Total (Unseen)			**94% (131/140)**	5% (7/140)	0% (0/130)

Table 4: **Success Rate for Additional Representation and Imitation Learning Baselines.** The method was evaluated across eight tasks and compared with SORNet and DINOBot. We use the average success rate as our evaluation metric. The low success rate of SORNet implies that it needs large-scale data for pre-training, while we only assume access to demonstrations recorded in several hours. DINOBot's low success rate implies it is not as flexible as our method for action representation and task specification. There is no result for the opening pen task because of DINOBot's assumption of single-arm manipulation.

the action as a sequence of end-effector poses, which can be easily extended to a dual-arm setup and accomplish a more diversified set of tasks.

3.5 t-SNE

In addition, we also analyze whether our semantic fields could carry desired semantic information, as shown in Figure 4. We initially sample 3D grid points in the workspace and select the top 100 points with the highest similarity score for each semantic field. For the example shown in Figure 4, there are 500 points in total. Then, we query the descriptor fields and obtain semantic features corresponding to the selected points. We reduce the high-dimensional semantic features to a two-dimensional plane using t-distributed stochastic neighbor embedding (t-SNE). We observe that all points are distinctly separated, and each blob corresponds to one object part. This visualization shows that our semantic fields have a clear semantic meaning for each channel, which helps the policy to achieve category-level generalization.

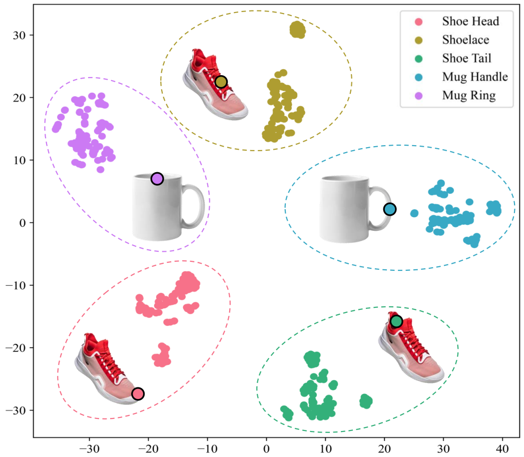

Figure 4: **t-SNE for Semantic Features with High Similarity.** Points in similarity fields are selected based on the top k similarity scores. We compute their corresponding semantic features and project them into two-dimensional space using t-SNE. It is clearly observed that all features are clustered and separated from each other. Each cluster corresponds to a feature that has semantic meaning, such as mug rings and mug handles.

3.6 Comparisons with Different RGBD Encoder

We also tried different RGBD encoders other than the standard encoder used for RGB. Specifically, we tested RD3D [16] as a different RGBD encoder (DP+RD3D) and evaluated it on two simulation tasks using the same dataset and task setup as the main paper. Our results are summarized in Table 5. We found that DP+RD3D fails in both tasks. We hypothesize that this is because RD3D was designed for object detection, which may not be particularly suitable for diffusion policy.

Method	Hang Mug (Unseen)	Insert Pencil (Unseen)	Total (Unseen)
DP	0% (0/20)	25% (5/20)	12.5% (5/40)
DP+RGBD	0% (0/20)	20% (4/20)	10% (4/40)
DP+RD3D	0% (0/20)	0% (0/20)	0% (0/40)
Ours w/o Semantics	**85% (17/20)**	40% (8/20)	62.5% (25/40)
Ours	80% (16/20)	**90% (18/20)**	**85% (34/40)**

Table 5: **Success Rate for DP+RD3D [16].** DP+RD3D was evaluated across two simulation tasks with the same setting as the main paper. We use the average success rate as our evaluation metric. We found that DP+RD3D fails in both tasks. We hypothesize that this is because RD3D was designed for object detection, which may not be particularly suitable for diffusion policy.

3.7 Comparisons with Semantic-Informed Encoder

Similar to our approach, we first extract DINOv2 feature maps from RGB and then build the semantic map by comparing to reference features, the same as in our paper. We then append the semantic map to the RGB and input it into the diffusion policy (DP+Semantic). Our results are summarized in Table 6

	Hang Mug (Unseen)	Insert Pencil (Unseen)	Total (Unseen)
DP	0% (0/20)	25% (5/20)	12.5% (5/40)
DP+RGBD	0% (0/20)	20% (4/20)	10% (4/40)
DP+Semantic	0% (0/20)	75% (15/20)	37.5% (15/40)
Ours w/o Semantics	**85% (17/20)**	40% (8/20)	62.5% (25/40)
Ours	80% (16/20)	**90% (18/20)**	**85% (34/40)**

Table 6: **Success Rate for DP+Semantic.** DP+Semantic was evaluated across two simulation tasks with the same setting as the main paper. We use the average success rate as our evaluation metric. We found that DP+Semantic performs better than DP, demonstrating that additional semantic information can guide the diffusion policy to focus on object parts important for task success. But it still performs worse than our approach.

We found that DP+Semantic performs better than DP, demonstrating that additional semantic information can guide the diffusion policy to focus on object parts important for task success. However, DP+Semantic still learns spurious correlations between RGB data and robotic actions, making it sensitive to appearance variations when encountering unseen instances from the same category. In contrast, our method utilizes 3D semantic fields that are robust to such variations and enable category-level generalization of the diffusion policy.

3.8 Ablation on Different Views

We provide additional experiments of training policy with different numbers of views, and our results are summarized in Table 7. We found that the performance of our method is not significantly affected by the number of camera views.

4 Detailed Comparisons with Prior Works

- *Functional Object-Oriented Network for Manipulation Learning* [17]: This work proposes a structured knowledge representation and generates motion sequences based on the representation. While this work represents the motion sequence as motion harmonics, our work is more flexible in action and task representation. In addition, this work does not show extensive real-world deployment results, while we focus more on the real-world deployment.
- *Affordance Detection for Task-Specific Grasping Using Deep Learning* [18]: This work uses convolutional neural networks to detect object affordance, class, and orientation for

Methods	Hang Mug (Unseen)	Pencil Insertion (Unseen)
1 View	85% (17/20)	90% (18/20)
2 Views	90% (18/20)	95% (19/20)
3 Views	75% (15/20)	85% (17/20)
4 Views	80% (16/20)	90% (18/20)
Total	82.5% (66/80)	90% (72/80)

Table 7: **Success Rate for Different Number of Views.** This table illustrates the success rates of our method with different numbers of views evaluated on two unseen tasks: Hang Mug and Pencil Insertion. We found that the performance of our method is not significantly affected by the number of camera views.

task-specific tasks. Although it shows generalization capabilities, it is still restricted to grasping tasks. In contrast, our framework supports a wider range of tasks.

- *SORNet: Spatial Object-Centric Representations for Sequential Manipulation* [14]: SORNet proposes an object-centric representation from RGB observations. We note three major differences from SORNet to our work. First, SORNet is object-centric, while ours is scene-representation, which needs less prior information. Second, SORNet builds representation from RGB, which makes it hard to generalize to novel domains, environments, and backgrounds, but our 3D semantic fields show category-level generalization capabilities. Lastly, through experiments, we found that SORNet might be better if pre-trained on a large dataset. In our work, we only assume access to a demonstration dataset of one or a handful of objects, while SORNet needs a larger dataset for pre-training to have a good performance.

- *Manipulation-Oriented Object Perception in Clutter Through Affordance Coordinate Frames* [19]: This work introduces the Affordance Coordinate Frame (ACF) for category-level pose estimation. It shows impressive category-level generalization capabilities. However, this work uses pre-defined motion primitives (grasp, pour, stir) for manipulation, which limits applicable tasks. In contrast, we do not assume any prior motion primitives, which is more flexible for action representation.

- *StructDiffusion: Object-Centric Diffusion for Semantic Rearrangement of Novel Objects* [20]: This work uses a diffusion model and object-centric transformer to construct goal structures given current point clouds and language inputs. They mainly focus on picking and placing tasks using modular methods, including the grasping planner and motion planner. Their modular method might not transfer to other tasks and other types of objects like deformable objects and granular objects easily. Without assumptions for target tasks and objects, our framework is flexible for object types and task specifications.

- *A Survey of Semantic Reasoning Frameworks for Robotic Systems* [21]: We position our work as instance-level semantic reasoning according to this survey. In our work, we propose 3D semantic fields, which can differentiate the object parts using foundational vision models.

- *Learning Generalizable Feature Fields for Mobile Manipulation* [22]: GeFF is a concurrent work using Generelizable NeRF for mobile manipulation [23]. There are two major differences between our work and GeFF. First, we do not distill feature fields from foundational vision models to avoid the loss in generalization capabilities. Second, GeFF uses open-push-close action sequences to accomplish tasks, while our policy predicts the end-effector trajectory, which is more flexible.

- *Evaluating Robustness of Visual Representations for Object Assembly Task Requiring Spatio-Geometrical Reasoning* [24]: This work mainly focuses on object assembly tasks by evaluating the robustness of visual representations from RGB observations. First, since RGB observation is sensitive to environment and instance variance, our method proposes to use 3D semantic fields to help the generalization of diffusion policy. Second, instead of focusing on assembly tasks, our framework is tested on a broader set of tasks.

- *You Only Demonstrate Once: Category-Level Manipulation from Single Visual Demonstration* [25]: You Only Demonstrate Once introduces the category-level generalizable last-inch policy using model-free pose estimation and trajectory transformation. There are two major differences. First, their behavior cloning algorithm is replaying the trajectory using the object's pose, which limits their applicability to tasks requiring adaptive adjustments based on the positions and shapes of objects. In contrast, we represent the trajectory as a sequence of end-effector poses, which is more flexible for task specification. In addition, they assume the object is rigidly attached to the end-effector, which is not applicable to some tasks like pushing the shoe. In summary, many of our tasks are not doable using this framework.
- *On the Effectiveness of Retrieval, Alignment, and Replay in Manipulation* and *DINOBot: Robot Manipulation via Retrieval and Alignment with Vision Foundation Models* [15, 26]: This line of work from the set of authors proposes a generalizable imitation learning framework, following phases of retrieval, alignment, and replay. DINOBot is the latest work enhancing the prior work using vision foundation models. First, DINOBot merely replays the recorded trajectory after the 6-DoF alignment, which limits its applicability in tasks that require adaptive adjustments based on the positions and shapes of objects. Second, DINOBot has only tackled single-arm manipulation tasks using a wrist-mounted camera, which restricts its extension to more complicated bimanual tasks (e.g., pen opening, as we demonstrated using our method). In contrast, our method represents the action as a sequence of end-effector poses, which can be easily extended to a dual-arm setup and accomplish a more diversified set of tasks.

References

[1] P. R. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018.

[2] Z. Xu, Z. He, J. Wu, and S. Song. Learning 3d dynamic scene representations for robot manipulation. *arXiv preprint arXiv:2011.01968*, 2020.

[3] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola. Distilled feature fields enable few-shot language-guided manipulation. *arXiv preprint arXiv:2308.07931*, 2023.

[4] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.

[5] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.

[6] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *Arxiv*, 2024.

[7] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[9] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[10] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[11] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[12] 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.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[14] W. Yuan, C. Paxton, K. Desingh, and D. Fox. Sornet: Spatial object-centric representations for sequential manipulation. In *5th Annual Conference on Robot Learning*, pages 148–157. PMLR, 2021.

[15] N. D. Palo and E. Johns. Dinobot: Robot manipulation via retrieval and alignment with vision foundation models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[16] Q. Chen, Z. Zhang, Y. Lu, K. Fu, and Q. Zhao. 3-d convolutional neural networks for rgb-d salient object detection and beyond. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3):4309–4323, 2022.

[17] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun. Functional object-oriented network for manipulation learning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2655–2662. IEEE, 2016.

[18] M. Kokic, J. A. Stork, J. A. Haustein, and D. Kragic. Affordance detection for task-specific grasping using deep learning. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 91–98. IEEE, 2017.

[19] X. Chen, K. Zheng, Z. Zeng, C. Kisailus, S. Basu, J. Cooney, J. Pavlasek, and O. C. Jenkins. Manipulation-oriented object perception in clutter through affordance coordinate frames. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 186–193. IEEE, 2022.

[20] W. Liu, Y. Du, T. Hermans, S. Chernova, and C. Paxton. Structdiffusion: Language-guided creation of physically-valid structures using unseen objects. In *RSS 2023*, 2023.

[21] W. Liu, A. Daruna, M. Patel, K. Ramachandruni, and S. Chernova. A survey of semantic reasoning frameworks for robotic systems. *Robotics and Autonomous Systems*, 159:104294, 2023.

[22] R.-Z. Qiu, Y. Hu, G. Yang, Y. Song, Y. Fu, J. Ye, J. Mu, R. Yang, N. Atanasov, S. Scherer, and X. Wang. Learning generalizable feature fields for mobile manipulation. *arXiv preprint arXiv:2403.07563*, 2024.

[23] J. Ye, N. Wang, and X. Wang. Featurenerf: Learning generalizable nerfs by distilling foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8962–8973, 2023.

[24] C. Ku, C. Winge, R. Diaz, W. Yuan, and K. Desingh. Evaluating robustness of visual representations for object assembly task requiring spatio-geometrical reasoning. *arXiv preprint arXiv:2310.09943*, 2023.

[25] B. Wen, W. Lian, K. Bekris, and S. Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. *Robotics: Science and Systems 2022*, 2022.

[26] N. Di Palo and E. Johns. On the effectiveness of retrieval, alignment, and replay in manipulation. *IEEE Robotics and Automation Letters*, 2024.