

D³Fields: Dynamic 3D Descriptor Fields for Zero-Shot Generalizable Rearrangement

Anonymous Author(s)
Affiliation
Address
email

1 Contents

2	1 Method Details	1
3	1.1 Notation: Camera Transformation and Projection	1
4	1.2 Grounded-SAM Masks Association	2
5	1.3 Keypoints Tracking Initialization	2
6	1.4 Model-Predictive Control (MPC) Details	2
7	2 Additional Experiments	3
8	2.1 Implementation Details	3
9	2.2 Keypoint Tracking Results	4
10	2.3 Mesh Comparisons with FeatureNeRF and F3RM	4
11	2.4 Quantitative Correspondence Comparisons with FeatureNeRF and F3RM	5
12	2.5 Ablation Study: Qualitative Correspondence Comparisons	6
13	2.6 Ablation Study: Quantitative Correspondence Comparisons	8
14	2.7 Abaltion Study: Quantitative Manipulation Results	8

15 1 Method Details

16 1.1 Notation: Camera Transformation and Projection

17 We assume that there are multiple RGBD cameras with fixed viewpoints. We assume all cameras'
18 intrinsic parameters \mathbf{K} and extrinsic parameters \mathbf{T} are known. The i th camera's extrinsic parameters
19 are defined as follows:

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ 0^T & 1 \end{bmatrix} \in \mathbb{SE}(3), \quad (1)$$

20 where Euclidean group $\mathbb{SE}(3) := \{\mathbf{R}, \mathbf{t} \mid \mathbf{R} \in \mathbb{SO}^3, \mathbf{t} \in \mathbb{R}^3\}$. For a 3D point \mathbf{x} in the world frame,
21 we could obtain the pixel \mathbf{u}_i projected in i th camera and distance to i th camera r_i as follows:

$$\mathbf{u}_i = \text{Proj}(\mathbf{K}_i (\mathbf{R}_i \mathbf{x} + \mathbf{t}_i)), \quad r_i = [0, 1]^T (\mathbf{R}_i \mathbf{x} + \mathbf{t}_i), \quad (2)$$

22 where Proj performs perspective projection, mapping a 3D vector $p = [x, y, z]^T$ to a 2D vector
23 $q = [x/z, y/z]^T$.

Algorithm 1 Fusion Process

```
1: procedure FUSION( $\mathbf{x}$ ) ▷ Input 3D point
2:    $\mathbf{u}_i, r_i \leftarrow \text{Project}(\mathbf{x}, i), r'_i \leftarrow \mathcal{R}_i[\mathbf{u}_i]$  ▷ 3D projection and depth reading
3:    $d_i \leftarrow r'_i - r_i, d'_i \leftarrow \text{Truncate}(d_i, \mu)$  ▷ Truncated depth difference using Eq. 3 in main paper
4:    $v_i, w_i \leftarrow \text{Weights}(d_i)$  ▷ Assign weights to each view using Eq. 4 in main paper
5:    $\mathbf{f}_i, \mathbf{p}_i \leftarrow \text{Interpolate}(\mathcal{W}_i^f, \mathcal{W}_i^p, \mathbf{u}_i)$  ▷ Interpolate features using Eq. 5 in main paper
6:    $\mathbf{f}, \mathbf{p} \leftarrow \text{Fuse}(\mathbf{f}_i, \mathbf{p}_i, v_i, w_i)$  ▷ Fuse features using Eq. 6 in main paper
```

1.2 Grounded-SAM Masks Association

Using Grounded-SAM [1, 2], we could extract instance segmentation masks from each view. However, masks from different views can contain a different number of instances, and the instance IDs may not be consistent. To tackle this problem, we need to post-process the instance segmentation results. The high-level idea is merging instances from different viewpoints given their geometric distance.

We will save all merged instances to a list. Specifically, we will first start from the first viewpoint. For each instance mask, we map them to 3D point clouds and save them into a list. Then we move to the next viewpoint and map all instance masks to 3D point clouds. We will compare each instance with the merged instances in the list. If they have significant overlapping, which is measured by the Intersection of Union (IoU) of two point clouds, we will merge the instance in the new viewpoint to the merged instances in the list. This process will continue until all viewpoints are iterated.

After merging all instances, we will filter instances not stably detected. Specifically, instances that meet one of the following criteria will be filtered out:

- The instance has little point cloud.
- The instance is known to be a background, such as the table.
- The instance overlaps with other instances, while other instances have a higher confidence.

After the filtering, we will assign consistent instance IDs to the instance mask in each viewpoint.

1.3 Keypoints Tracking Initialization

To initialize keypoint tracking, we first densely sample points, which can either come from grid sampling or instances' 3D point clouds. Then we evaluate these points using our D³Fields and mask out those points not belonging to the desired instance. Then we downsample these points to the desired number using the farthest point sampling.

1.4 Model-Predictive Control (MPC) Details

As described in Section 3.4 of the main paper, our MPC framework needs a reference camera to bridge the gap between 3D representation and 2D representation. In our work, the reference camera's extrinsic parameters are defined according to tasks. Typically, it looks down at the workspace from the top of the workspace. Its intrinsic parameters are the same as the real camera's intrinsic parameters. The detailed MPC algorithm we used is described in Algorithm 2.

For the pick-and-place tasks, our dynamics model is simplified as the object is rigidly attached to the end-effector.

Algorithm 2 Trajectory optimization at each MPC step

Input: Current state s_0 , goal s_{goal} , time horizon T , gradient descent iteration N
the perception module h , and the dynamics module f

Output: Actions $a_{0:T-1}$

Sample current action sequence $\hat{a}_{0:T-1}^*$

for $i = 1, \dots, N$ **do**

 Sample M action sequences $\hat{a}_{0:T-1}^{1:M}$ near current action sequence

for $m = 1, \dots, M$ **do**

for $t = 0, \dots, T - 1$ **do**

 Predict the next step $s_{t+1} \leftarrow f(s_t, \hat{a}_t^m)$

 Calculate the task loss $c^m \leftarrow c(s_T, y_g)$

 Calculate the current action sequence $\hat{a}_{0:T-1}^*$ using the task loss $c^{1:M}$

Return $\hat{a}_{0:T-1}^*$

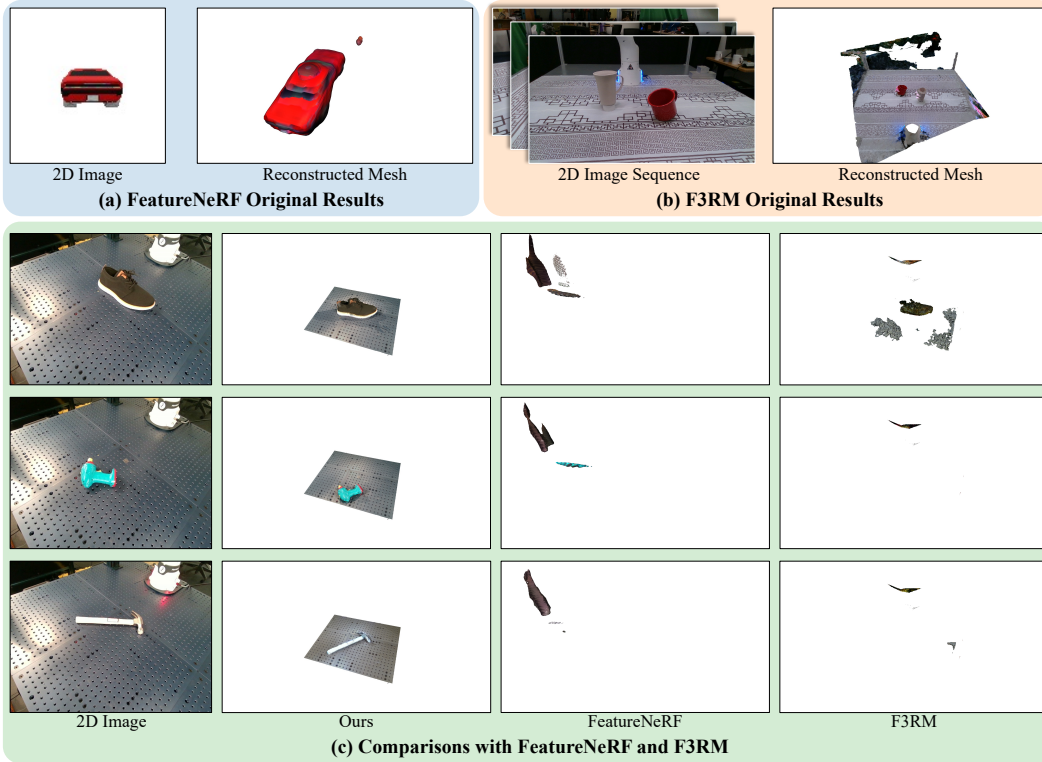


Figure 1: Mesh Reconstruction Comparison. (a) shows the reconstructed mesh of the FeatureNeRF, given a 2D image from the training distribution. This reflects that our mesh extraction process can work well when the input image is within the training distribution. Given a sequence of 2D images densely scanning the workspace, (b) also shows good reconstruction quality of the scene. However, when given sparse views containing novel instances, both FeatureNeRF and F3RM fail to generate accurate meshes for the scene, which demonstrates the effectiveness of our method.

2 Additional Experiments

2.1 Implementation Details

For the truncation threshold μ , we set it as 0.02 across all experiments. For the prompts used for Grounded-SAM [1, 2] in Figure 7 in the main paper, from left to right, they are “shoe”, “mug”, “spoon”, “can”, and “toothpaste” respectively. Their confidence threshold used for Grounded-SAM is 0.2.

61 We also compare with several baselines, with details listed below:

- 62 • Dense Object Nets (DON) [3]: We compare the effects of using different feature backbones,
63 where DON is one of the baseline backbones. We use the pre-trained DON model since our
64 method also uses off-the-shelf models with no re-training and additional data. Specifically,
65 we use the model trained on shoe class saved [here](#).
- 66 • DINO [4]: Another backbone feature backbone we baseline on is DINO, which is the prior
67 work of DINOv2. We use the code provided by [5] to extract dense DINO features.
- 68 • RGBD+DINOv2: We also compare our method to simply merging point clouds with DI-
69 NOv2 features from multiple viewpoints [6].
- 70 • FeatureNeRF [7]: We compare our representation with other state-of-the-art 3D implicit
71 semantic representation including FeatureNeRF. We trained the model on the car example
72 dataset provided by the authors. To compare with our model, we do not distill the model
73 from the DINO model, but DINOv2. It is worth noting that FeatureNeRF only uses one
74 RGB image to generate the neural fields. We found that giving more views to the FeatureN-
75 eRF model during inference time will lead to worse performance. Therefore, we keep its
76 input as a single-view RGB image.
- 77 • Distilled Feature Fields (F3RM) [8]: Another 3D implicit semantic representation we com-
78 pare to is F3RM. We use the same training code as provided by the authors, except that we
79 distill from DINOv2 models to make it comparable with our model.

80 For the dynamics training of shoe pushing example, we collect 20 episodes of pushing one shoe.
81 Then we train a dynamics model that can take in current particles and a pushing action and predict
82 particles in the next step.

83 For the evaluation in the real world, we summarize details regarding our tasks in Table 1.

Environment	Task Name	Objects
Real World	Organize Shoes	Shoe
	Collect Debris	Almonds
	Organize Office Table	Mouse, Pen, Mug
	Organize Utensils	Knife, Spoon, Fork, Bowl
	Organize Fruits	Apple, Banana
	Push Shoes	Shoe
Simulation	Serve Food	Cupcake, Bread, Tomato, Lemon, Banana
	Organize Mugs	Mug
	Organize Shoes	Shoe
	Organize Utensils	Knife, Spoon, Fork

Table 1: **Task Details Summary.** This table summarizes our task environment, specific tasks, and objects. We evaluate our framework on eight tasks and fifteen object categories, where each object category covers several object instances with diverse appearances and shapes.

84 2.2 Keypoint Tracking Results

85 We show two examples of 3D keypoint tracking in Figure 2. In the first scenario, we track a shoe
86 as it is pushed and subsequently flipped. The second example demonstrates tracking a shoe that is
87 lifted and then placed down. Our system robustly tracks the shoe in the 3D space. These examples
88 underscore the effectiveness of D³Fields in maintaining accurate tracking in dynamic scenarios,
89 which enables our dynamics learning capabilities.

90 2.3 Mesh Comparisons with FeatureNeRF and F3RM

91 We qualitatively compare the descriptor fields
 92 generated by the three methods. We extract the
 93 mesh from these fields using marching cubes,
 94 as shown in Figure 1. We observe that our
 95 D³Fields could generate accurate color meshes
 96 given sparse views. FeatureNeRF could recon-
 97 struct a reasonable mesh given the single 2D
 98 image from the training distribution, as shown
 99 in Figure 1 (a). However, when it encoun-
 100 ters a new object out of the training distribu-
 101 tion, the reconstructed mesh will be totally off,
 102 even when we apply the image preprocessing
 103 to align the testing images with the training set
 104 in terms of image sizes, data range, and back-
 105 ground color. Although Figure 1 (b) shows that
 106 we can reconstruct a clear mesh with dense im-
 107 age sequences the same as the original paper,
 108 its color mesh is quite inaccurate given sparse
 109 viewpoints in our experiment setting, except for
 110 the shoe case.

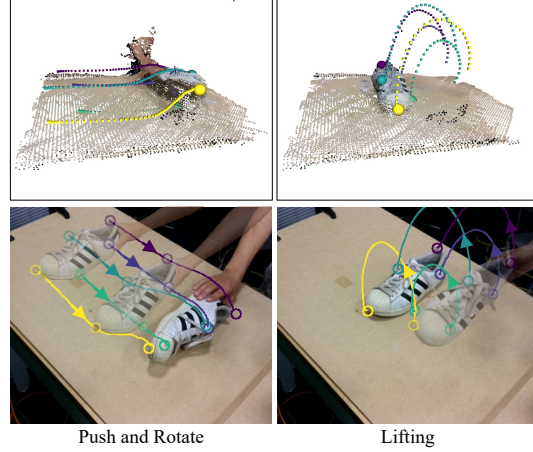


Figure 2: **Keypoint Tracking.** We apply D³Fields to tracking tasks and showcase two tracking examples, both of which involve 3D motions and partial observations from single viewpoints. This shows that our representation is 3D, dynamic, and semantic.

111 2.4 Quantitative Correspondence Comparisons with FeatureNeRF and F3RM

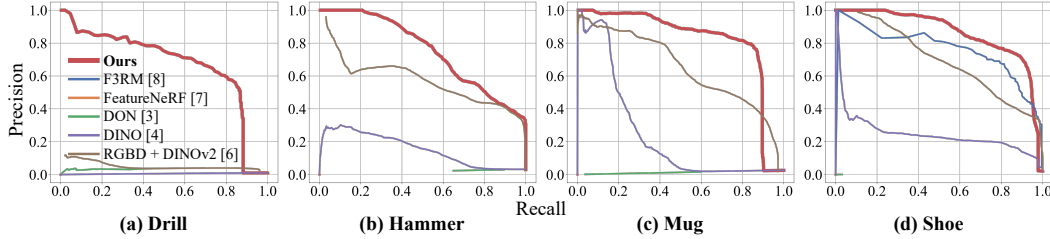


Figure 3: **Precision-Recall of Various Thresholds for Different Instances.** The curves show how D³Fields compares with 3 baseline methods in terms of matching quality, tested on 4 different instances: mug, bag, pan, and shoe. We use the precision-recall curve to measure the correspondence quality. Our method shows to consistently exceeds the performance of the baseline approaches, which demonstrates our method’s capability to encode semantic information accurately and establish precise correspondences using the semantic information.

112 In addition, we also measure the quantitative correspondence accuracy of our method, FeatureNeRF,
 113 and F3RM. We manually label the ground truth correspondence keypoints on the source image and
 114 the target descriptor fields. We measure the correspondence quality using the precision-recall curve,
 115 as shown in Figure 3. A larger area under the curve indicates better correspondence quality. Details
 116 regarding the precision-recall curve are provided later. We could see that F3RM and FeatureNeRF
 117 collapse to the origin point except for the shoe example for F3RM. This is because these two methods
 118 fail to reconstruct meshes given sparse observations and unseen instances. In contrast, our method
 119 shows a much better correspondence quality.

120 To generate the precision-recall curve, we manually label one point \mathbf{x}_{src} on the 2D source image, and
 121 a set of corresponding 3D points \mathbf{x}_{tgt} . For 2D points, we can get the associated semantic feature \mathbf{f}_{src} .
 122 For vertices on the reconstructed mesh, we can obtain a set of semantic features $\{\mathbf{f}_{\text{tgt},0}, \dots, \mathbf{f}_{\text{tgt},N}\}$.
 123 Then we compute the cosine similarity between features \mathbf{f}_{src} and $\{\mathbf{f}_{\text{tgt},0}, \dots, \mathbf{f}_{\text{tgt},N}\}$. For one similarity
 124 threshold τ , we can filter out a set of points \mathbf{F}_{τ} with similarity scores higher than τ . In addition, we
 125 can get the set of points \mathbf{G} that are close to \mathbf{x}_{tgt} . Then we can define precision P_{τ} and R_{τ} as follows:

$$P_{\tau} = \frac{|\mathbf{F}_{\tau} \cap \mathbf{G}|}{|\mathbf{F}_{\tau}|}, \quad R_{\tau} = \frac{|\mathbf{F}_{\tau} \cap \mathbf{G}|}{|\mathbf{G}|}. \quad (3)$$

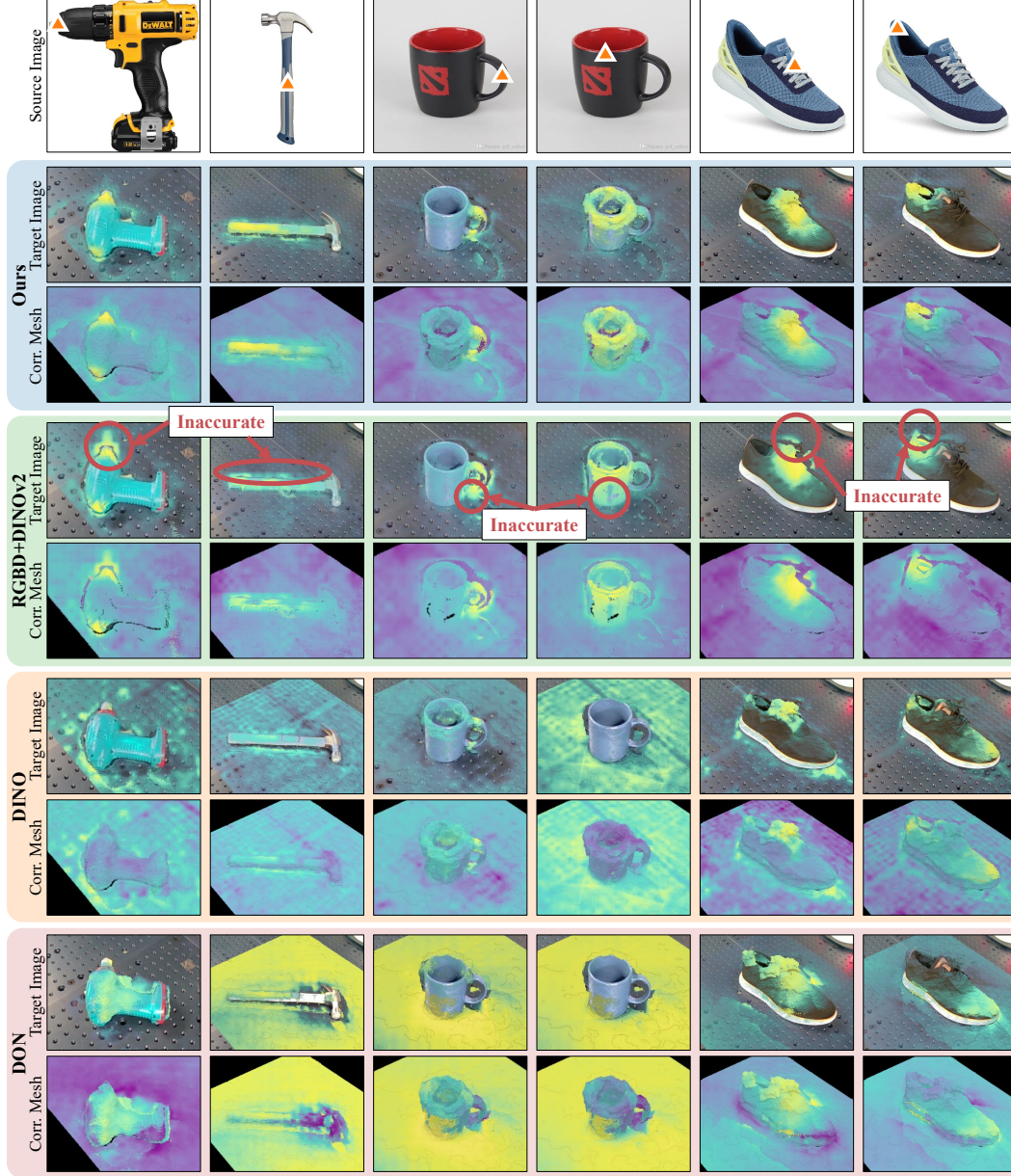


Figure 4: **Correspondence Quantitative Comparison.** The top row shows the selected pixel from the source image, and the following rows show the corresponding areas for different methods. While our method could have accurate correspondence, RGBD+DINOv2 corresponds to some points in the background. For example, the drill tip is not accurately highlighted in the RGBD+DINOv2 example, while ours can accurately highlight the drill tip. Ours with DINO feature backbones fail to identify objects accurately, while ours with DON fail to generalize to novel scenes and novel instances.

By varying τ , we can plot the precision-recall curve as shown in Figure 3.

2.5 Ablation Study: Qualitative Correspondence Comparisons

In this section, we first study how different feature backbones could affect the correspondence quality. Then we show the qualitative correspondence results of our method. As mentioned in Section 2.1, we substitute our method’s backbone with other pre-trained models, like DON and DINO [3, 4]. We also compare with RGBD+DINOv2 to show the effectiveness of our method.

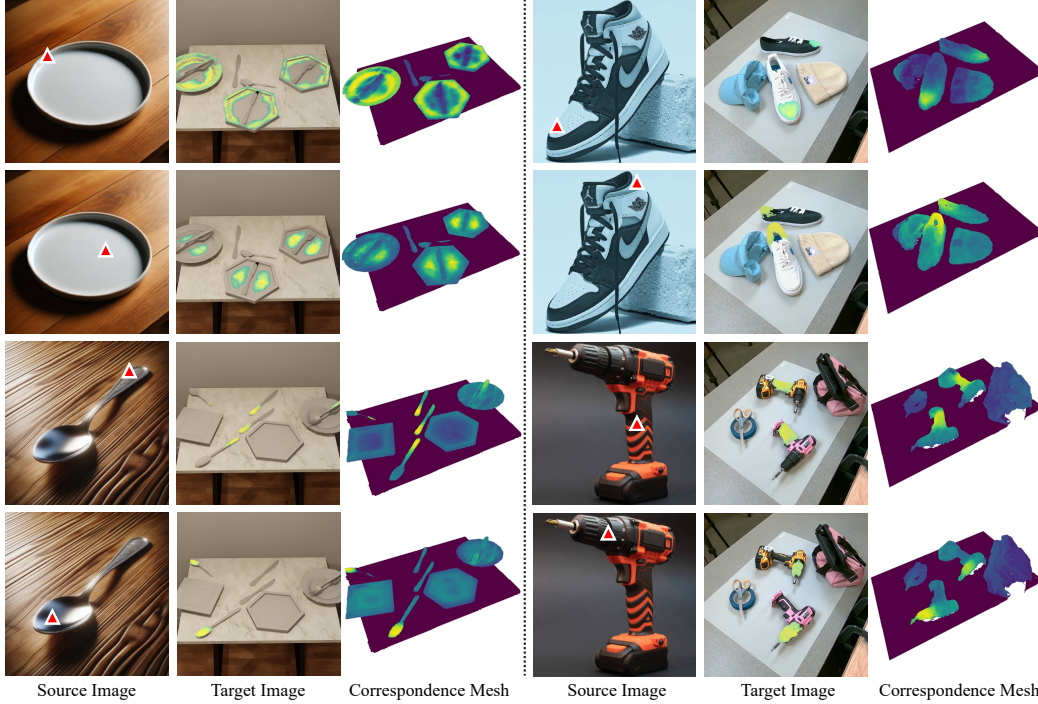


Figure 5: **Cross-Domain Correspondence.** The red triangles represent query points in the source image, and the corresponding areas are highlighted in the 3D mesh. First, we observe that our representation can encode features for object parts and establish the correspondence, such as spoon tips and spoon handles. In addition, we found the correspondence can be multimodal. When the shoe head is selected, multiple shoe heads in the workspace are highlighted. At last, the correspondence is generalizable across different contexts, instances, and domains, which demonstrates our method’s generalization capabilities.

Figure 4 shows the qualitative correspondence results of all ablation baselines. There are three key messages we observe from this figure.

- Compared with RGBD+DINOv2, our method’s correspondence quality is better. We have more accurate correspondence since our representation can amortize noises from single views by considering 3D consistency. Although, RGBD+DINOv2 could have certain spatial consistency, there are still variances in results from different viewpoints, while our method can guarantee spatial consistency.
- Compared with DINO, our correspondence is more fine-grained and accurate. Thanks to the advancement of foundation visual models, DINOv2 encodes more fine-grained features and enables correspondence with higher accuracy.
- DON struggles to generalize to novel scenes and unseen object categories. Although the original DON shows good correspondence quality, it is trained on one type of object with a relatively small dataset. Compared with visual foundational models, it shows limited generalization capabilities in terms of scenes and object categories.

We also visualize the correspondence from 2D images to our workspace as shown in Figure 5. Specifically, we extract the DINOv2 feature of the selected pixel in the 2D image. Then we highlight the part of the 3D mesh with features close to the query feature. There are two observations regarding the qualitative correspondence results. First, the semantically similar parts are correctly matched across different instances and contexts. For example, when we select the rim of the plate in the 2D image, the corresponding part in the 3D mesh is highlighted. This matching is consistent across different object parts, such as the head and tail of the shoe, the handle and blade of the knife, and the tip and bar of the drill. Second, the correspondence is multimodal when there are multiple semantically similar object parts in the workspace. For example, when we select the spoon handle in the 2D image, multiple utensil handles in the workspace are highlighted. The correspondence

156 qualitative results show that our D³Fields could establish meaningful correspondences across differ-
 157 ent instances and contexts, so that we can rely on correspondence to define the planning objective
 158 function.

159 2.6 Ablation Study: Quantitative Correspondence Comparisons

160 Similar to Section 2.4, we generate the precision-recall curve to quantitatively compare the corre-
 161 spondence quality with ablation baselines. We can make the following observations regarding the
 162 baseline correspondence results.

- 163 • Compared with RGBD+DINOv2, our method shows to have more accurate correspondence
 164 results. This is because our D³Fields can average out noises from each viewpoint, while
 165 RGBD+DINOv2 will accumulate noises.
- 166 • DINO faces challenges in accurately distinguishing specific object components. This limi-
 167 tation results in less precise correspondence, as shown in Figure 3.
- 168 • Although DON can encode semantic features in seen environments and instances, it fails
 169 to generalize to novel environments and object categories. Therefore, its correspondence
 170 results are even worse than DON, as shown in Figure 3.

171 2.7 Ablation Study: Quantitative Manipulation Results

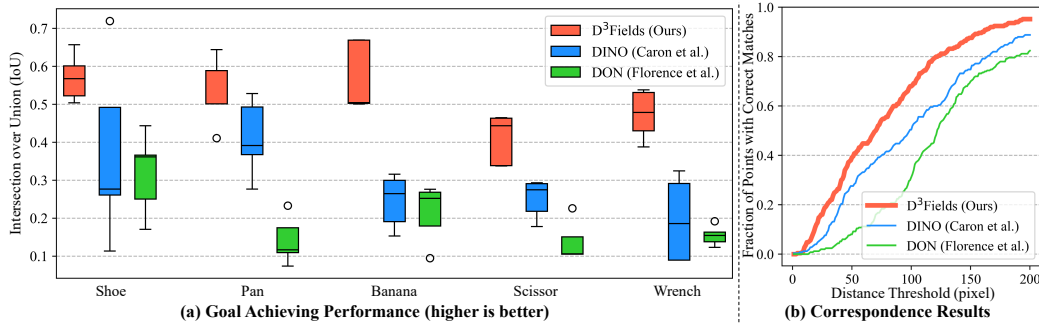


Figure 6: **Quantitative Evaluation.** We perform real-world quantitative evaluations by measuring final goal-achieving performance and keypoints correspondence accuracy. (a) We use IoU to measure goal-achieving performance. Results indicate that our method aligns with the goal configurations much better than DON and DINO across various object categories and scenarios. (b) We measure the keypoints correspondence accuracy according to the fraction of points with accurate matches, with correct matches determined by a distance threshold. Our method is consistently better at aligning with the goal image, regardless of the chosen threshold.

172 In Figure 6 (a), we measure performance using the IoU between the mask of the goal image and the
 173 mask of the final state post-manipulation. Higher IoU values indicate a greater degree of alignment
 174 between the intended and achieved configurations. Our method demonstrates superior performance
 175 across five distinct object categories, consistently outshining the baseline methods. For each cate-
 176 gory, we performed 5 experiments for the evaluation results. This not only highlights its exceptional
 177 manipulation accuracy but also its robust generalization capabilities. While the DINO model ex-
 178 hibits some struggles, particularly in distinguishing specific object components and consequently
 179 yielding less precise results, it still performs better than DON. Although DON shows commendable
 180 results with familiar objects and configurations, its performance dips in novel scenarios, revealing a
 181 lack of generalization. These results collectively emphasize the significant advantages of our method
 182 in diverse and accurate object manipulation.

183 In Figure 6 (b), we present the correspondence results. We label 10 corresponding keypoint pairs on
 184 both the goal image and the final manipulation result to sufficiently evaluate the correspondence ac-
 185 curacy. The accuracy of correspondence was determined by calculating the proportion of keypoints
 186 that were accurately matched, using a predefined distance threshold as the criterion. If the dis-
 187 tance between corresponding keypoints exceeds this threshold, they are determined as unmatched.

Our method shows superior performance across various thresholds, consistently outperforming the baseline models. DINO emerges as the second-best in terms of performance, exhibiting broad applicability but with a lower precision compared to our method. Meanwhile, DON lags in performance, primarily due to its struggles with generalization in novel scenarios. These results, in conjunction with those from Figure 6(a), reiterate our method’s outstanding capabilities in both generalization and accuracy. While DINO provides reasonable applicability, it lacks the precision of our approach, and the performance of DON is hindered by its limited adaptability.

References

- [1] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- [2] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.
- [3] P. R. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In A. Billard, A. Dragan, J. Peters, and J. Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 373–385. PMLR, 29–31 Oct 2018.
- [4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [5] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021.
- [6] 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.
- [7] 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.
- [8] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola. Distilled feature fields enable few-shot manipulation. In *7th Annual Conference on Robot Learning*, 2023.