2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

DAP: Diffusion-based Affordance Prediction for Multi-modality Storage

Haonan Chang, Kowndinya Boyalakuntla, Yuhan Liu, Xinyu Zhang, Liam Schramm, Abdeslam Boularias

***Abstract*— Solving storage problems—where objects must be accurately placed into containers with precise orientations and positions—presents a distinct challenge that extends beyond traditional rearrangement tasks. These challenges are primarily due to the need for fine-grained 6D manipulation and the inherent multi-modality of solution spaces, where multiple viable goal configurations exist for the same storage container. We present a novel Diffusion-based Affordance Prediction (DAP) pipeline for the multi-modal object storage problem. DAP leverages a two-step approach, initially identifying a placeable region on the container and then precisely computing the relative pose between the object and that region. Existing methods either struggle with multi-modality issues or computation-intensive training. Our experiments demonstrate DAP's superior performance and training efficiency over the current state-of-the-art RPDiff, achieving remarkable results on the RPDiff benchmark. Additionally, our experiments showcase DAP's data efficiency in real-world applications, an advancement over existing simulation-driven approaches. Our contribution fills a gap in robotic manipulation research by offering a solution that is both computationally efficient and capable of handling real-world variability. Code and supplementary material can be found at: https://github.com/changhaonan/DPS.git.**

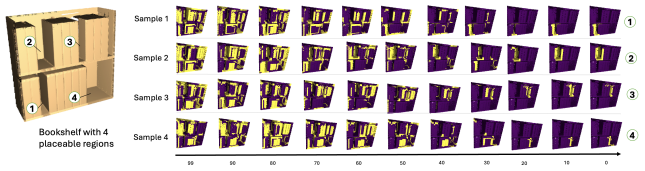

Fig. 1: Visualization of the backward diffusion process in affordance prediction. Rows represent different samples, and columns represent diffusion steps, from 99 to 0. Yellow indicates placeable regions, while purple indicates non-placeable areas. Initially, the scene shows random segmentation, which gradually converges to four placeable regions as the process progresses.

I. INTRODUCTION

Storage tasks like placing a plate in a dishwasher or a book on a shelf involve positioning objects within containers requires00 meeting strict geometric criteria. These tasks present unique challenges: they require collision-free placement in stable, contact-based configurations and often allow for multiple geometrically distinct yet functionally correct solutions. This multi-modality significantly impacts regression-based models, such as Coarse-to-fine Q-attention [1], Relational Neural Descriptor Fields [2], Neural Shape Mating [3], or Structformer [4].

Diffusion models address multi-modality issues, as seen in StructDiffusion [5] and RPDiff [6], but these approaches have limitations, such as inaccurate pose predictions and the need for extensive training in simulated environments.

We introduce the **D**iffusion-based **A**ffordance **P**rediction (DAP) method to tackle storage problems by separately addressing geometric constraints and multi-modality. DAP first identifies a placeable region within the container using diffusion-based affordance prediction, then derives the goal pose by matching the object to this region, free from interference by other regions. For instance, when placing a plate in a dishwasher, DAP models valid slots, samples one, and solves for its goal pose.

Our contributions are: (1) DAP efficiently predicts accurate goal poses for storage tasks by generating a multi-modal affordance distribution; (2) DAP outperforms previous S.O.T.A method, RPDiff in both accuracy and training efficiency on the RPDiff benchmark, training in just 2 hours; (3) We demonstrate DAP's effectiveness in real-world storage tasks with noisy observations and minimal training data.

II. RELATED WORKS

A. Pair-wise Object Manipulation

Storage tasks require precise transformation between the moving object and the stationary container. Traditional pair-wise object manipulation methods use point cloud registration to identify task-relevant regions, followed by relative transformation estimation. Tax-Pose [7] and R-NDF [2] utilize transformers and neural descriptor fields for correspondence and transformation calculations. Neural Shape Mating [3] learns transformations directly without point cloud registration. However, these methods struggle with multi-modal tasks, a challenge addressed by RPDiff's [6] diffusion-based pose refinement model, which incurs high computational costs and extensive training time (several days of training on an advanced GPU such as NVIDIA V100).

Our approach addresses these computational and multi-modality challenges by employing a diffusion-based affordance prediction method. This approach identifies one task-relevant region among many and then establishes correspondences between this region and the moving object to accurately estimate the transformation.

B. Affordance Prediction & Point Cloud Segmentation

In 3D point cloud segmentation, methods are categorized into: (1) Semantic segmentation for broad classes, (2) Instance segmentation for individual entities, and (3) Affordance segmentation for interaction regions (e.g., pushing, storing). Initial methods focused on MLP/CNN architectures (e.g., PointNet [8], PointNet++ [9], PointGroup [10]), but the trend has shifted towards transformer-based models (e.g.,

The authors are with the Department of Computer Science, Rutgers University, 08854 New Brunswick, USA. This work is supported by NSF awards 1846043 and 2132972.

Superpoint [11], Point Transformer [12], Mask3D [13], OneFormer3D [14]), which enhance performance in semantic and instance segmentation. In affordance prediction, 3D AffordanceNet [15] provides benchmarks across 18 categories, while 3DAPNet [16] predicts affordance regions and generates 6DoF poses.

Our task is to segment a suitable storage region among many, adding complexity beyond traditional segmentation. Semantic segmentation cannot distinguish multiple viable regions, and instance segmentation is impractical due to varying storage spaces. While our task aligns with affordance prediction, existing methods fail to handle multi-modality and cannot select a single region from multiple options. We overcome this by integrating a diffusion model with Point Transformer architecture to model the distribution of placeable storage regions within a container's point cloud.

C. Diffusion Model

Diffusion models have been successful in various generative tasks, such as image generation [17], [18], imitation learning [19], and offline reinforcement learning [20]. These models use two processes: a noising forward process that adds Gaussian noise iteratively to data samples, and a denoising backward process where a model learns to predict and remove this noise to reconstruct the original data. Training involves minimizing the mean-squared error between predicted and actual noise [18].

Two key milestones in diffusion models are Denoising Diffusion Probabilistic Models (DDPM) [18], which use the Rao-Blackwell theorem for faster training, and Diffusion Transformers (DiT) [21], which leverage transformer architectures for improved scaling and handling variable-length inputs. Our approach uses the diffusion transformer architecture with the DDPM loss function.

III. Problem formulation

We address the challenge of multi-modality storage. Our objective is to position a target object O inside a bigger container C, considering that there are multiple viable placements for O within C. We represent the relative transformation between O and C as $\mathbf{T}_{OC} \in \mathbb{SE}(3)$. The storage is successful when $\mathbf{T}_{OC}$ falls in the support of a multi-modal distribution $\mathscr{D}$. The goal is to, given the point cloud observations of O and C, $\mathbf{P}_O$ and $\mathbf{P}_C$, in the world coordinate system W, calculate a transformation for O, denoted as $\mathbf{T}_{WO} = (\mathbf{R}_{WO} \in \mathbb{SO}(3), \mathbf{t}_{WO} \in \mathbb{R}^3)$. Applying this transformation to object O should result in the relative pose of O and C falling into the distribution $\mathscr{D}$. Point cloud $\mathbf{P}$ consists of point vertexes $\{v_i\}_{i=1}^N$ and normals $\{n_i\}_{i=1}^N$. We assume a small set of M demonstrations $\{\mathbf{P}_O^j, \mathbf{P}_C^j, \mathbf{T}_{WO}^j\}_{j=1}^M$ is provided.

IV. Method

We tackle this problem using a two-stage method. Initially, we employ a diffusion-based affordance prediction to identify the placeable regions within the container, given the target object. Unlike conventional affordance prediction methods, which return all placeable regions simultaneously without distinction, our diffusion-based approach singles out one focused region in each sample. Upon identifying the placeable region, we proceed to compute the relative pose between the placeable region and the target object. Rather than directly calculating the $\mathbb{SE}(3)$ transformation, we first establish a point-wise correspondence between the container's local region and the target object's point cloud. This correspondence predicts which parts of the container and target should be in contact. We then utilize the algorithm in [22] to determine the pose from this correspondence.

A. Diffusion-based Affordance Prediction

The primary challenges in the multi-modal storage problem are twofold: (1) The model must have high enough accuracy that the generated poses are stable and avoid collisions, and (2) The multi-modal nature of the task presents multiple viable solutions, making it difficult for learning-based methods to separate them. To address the first issue, we adopt a coarse-to-fine strategy, proven by prior research [1], [23] to enhance pose prediction accuracy effectively. In tackling the second challenge of ambiguity of viable solutions, we introduce a diffusion-based affordance prediction method. This method serves as a critical step in our coarse-to-fine strategy, effectively narrowing down the possibilities by focusing on placeable regions within the container. Specifically, we aim to predict a score $\mathbf{S} = (s_1, s_2, \ldots, s_{N_C}), s_i \in [-1, 1]$ for each point in the container point cloud $\mathbf{P}_C$, where a higher score signifies a more suitable placement area. After we obtain the affordance prediction $\mathbf{S}$, we crop the container based on this prediction, and then perform pose-relevant computation on that local geometry. This prediction is framed as a generative task, aiming to model the conditional distribution of score $\mathbf{S}$ over container geometry $\mathbf{P}_C$.

Data labeling: As outlined in the problem formulation, our data comprises $\{\mathbf{P}_C, \mathbf{P}_O, \mathbf{T}_{WO}\}$. From this, we need to generate labels for placeable affordance. We apply the transformation $\mathbf{T}_{WO} = (\mathbf{R}_{WO}, \mathbf{t}_{WO})$ to $\mathbf{P}_O$ using the formula:

$$v_i' = \mathbf{R}_{WO} v_i + \mathbf{t}_{WO},\ n_i' = \mathbf{R}_{WO} n_i. \tag{1}$$

This results in the transformed point cloud $\mathbf{P}_O' = \{(v_i', n_i')\}_{i=1}^{N_O}$, which represents the goal object point cloud. Next, we identify points on the container $\mathbf{P}_C$ whose minimal distance to the transformed object $\mathbf{P}_O'$ is smaller than a threshold ε_{place}. These nearby points to the target point cloud on the container point cloud indicate the placeable region on $\mathbf{P}_C$. We assign a score of 1 to these points and -1 to the rest. Formally, this labeling is defined as:

$$s_i = \begin{cases} 1 & \text{if } \min\limits_{v_j \in \mathbf{P}_C} ||v_i' - v_j||_2 < \varepsilon_{place},\ v_i' \in \mathbf{P}_O' \\ -1 & \text{else} \end{cases} \tag{2}$$

Here, ε_{place} serves as a hyper-parameter to adjust the size of the placeable region, enabling us to mitigate the ambiguity inherent in the multi-modality storage challenge.

Training: Based on our label generation method, the score $\mathbf{S}$ will be a distribution conditioned on the container's geometry $\mathbf{P}_C$, denoted as $\mathscr{D}_S = p(\mathbf{S}|\mathbf{P}_C)$. To capture $\mathscr{D}_S$, we utilize a

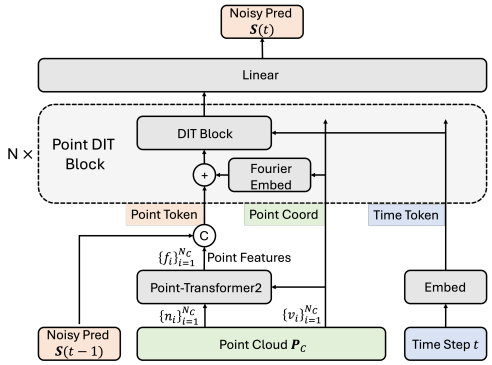

Fig. 2: The Diffusion Affordance Prediction Architecture.

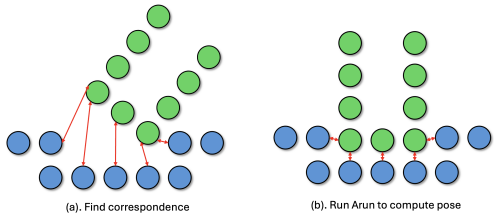

Fig. 3: Illustration of the correspondence and pose computation on a 2D toy example. The green points are the target object, and the blue points are the container.

denoising diffusion probabilistic model (DDPM) [18]. We construct a continuous diffusion process $\{\mathbf{S}(t)\}_{t=0}^{T}$ indexed by time-variable t. $\mathbf{S}(0)$ originates from the demonstration data, representing the ground-truth affordance score. As the time-step t progresses from 1 to T (the total number of diffusion steps), $\mathbf{S}(t)$ is progressively perturbed by Gaussian noise,

$$p(\mathbf{S}(t)|\mathbf{S}(t-1),\mathbf{P}_C) := \mathcal{N}(\mathbf{S}(t);\sqrt{1-\beta_t}\mathbf{S}(t-1),\beta_t I). \quad (3)$$

Here β_t follows the notation in [18]. The training goal is to learn a network $\mu_\theta(\mathbf{S}(t),t,\mathbf{P}_C)$, which is able to backward the diffusion process, estimating $\mathbf{S}(t-1)$ from $\mathbf{S}(t)$:

$$p_\theta(\mathbf{S}(t-1)|\mathbf{S}(t),\mathbf{P}_C) := \mathcal{N}(\mathbf{S}(t-1);\mu_\theta(\mathbf{S}(t-1),t,\mathbf{P}_C),\sigma_t). \quad (4)$$

According to [18], rather than directly estimating μ_θ, we can express μ_θ as:

$$\mu_\theta(\mathbf{S}(t),t,\mathbf{P}_C) = \frac{1}{\sqrt{\alpha}}(\mathbf{S}(t) - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(\mathbf{S}(t),t,\mathbf{P}_C). \quad (5)$$

Thus, the training objective for the DDPM can be simplified to:

$$L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T],\mathbf{S}_0,\varepsilon_t}\left[\|\varepsilon_t - \varepsilon_\theta(\mathbf{S}(t),t,\mathbf{P}_C)\|^2\right]. \quad (6)$$

The parameters $\alpha_t,\bar{\alpha}_t,\beta_t,\varepsilon_t$ adhere to the definitions provided in [18]. This training objective is equal to minimizing a variational lower bound over the KL-divergence between a learned distribution $\mathscr{D}_\theta$ and the goal distribution $\mathscr{D}_S$. After training, we can sample from the learned distribution $\mathscr{D}_\theta$ with the learned network $\varepsilon_\theta(\mathbf{S}(t),t,\mathbf{P}_C)$. We start from a pure Gaussian noise $\mathbf{S}(T)\sim\mathcal{N}(0,I)$, and then perform the denoising steps from $t=T$ to $t=1$ using:

$$\mathbf{S}(t-1) = \frac{1}{\sqrt{\alpha_t}}(\mathbf{S}(t) - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(\mathbf{S}(t),t,\mathbf{P}_C)) + \sigma_t z. \quad (7)$$

Here $z\sim\mathcal{N}(0,I)$ if $t>1$ else 0. And we select $\sigma_t^2=\beta_t$. After iterating from $t=T$ to $t=1$, we get an affordance prediction $\mathbf{S}(0)$. Fig. 1 provides an illustrative visualization for this sampling process.

Architecture: For the network $\varepsilon_\theta(\mathbf{S}(t-1),t,\mathbf{P}_C)$, we adopt a diffusion-transformer (DiT) architecture as introduced in [21]. A major distinction is that, whereas the original DiT was designed for 2D tasks, our task is inherently 3D. We detail our architecture in Fig. 2. $\varepsilon_\theta(\mathbf{S}(t-1),t,\mathbf{P}_C)$ takes as input the point cloud $\mathbf{P}_C$, the noisy prediction $\mathbf{S}(t-1)$, and the time-step t. As illustrated in Fig. 2, the point cloud $\mathbf{P}_C$ is input into the network at two different positions: one part uses the point coordinates $\{v_i\}_{i=1}^{N_C}$, and the other utilizes per-point features $\{f_i\}_{i=1}^{N_C}$. The Point-Transformer2 [24] serves as the backbone to extract per-point features $\{f_i\}_{i=1}^{N_C}$ from coordinates $\{v_i\}_{i=1}^{N_C}$ and normals $\{n_i\}_{i=1}^{N_C}$. These per-point features $\{f_i\}_{i=1}^{N_C}$ are concatenated with the noisy scores $\{s_i\}_{i=1}^{N_C}$ to form the point tokens. The time-step t is processed through an embedding layer, generating the time token. These point tokens, point coordinates, and the time token are then fed into the Point-DiT block. Within the Point-DiT block, we apply a Fourier position embedding [25] to encode the point-wise positional information. Notably, unlike in traditional transformer architectures where positional encoding is applied only at the first layer, we implement this encoding at every layer. As demonstrated in [13], applying positional encoding at each transformer layer proves advantageous for segmentation tasks. Subsequently, the position-encoded point tokens and time-token are processed by the DiT Block, which retains the structure described in [21]. The output refined point tokens are then used as input for the next Point-DiT layer, while the point coordinates and time-token remain unchanged. Finally, a linear layer projects the latent embeddings back to an $N_C\times 1$ vector with a range of $[-1,1]$.

After obtaining the final affordance prediction $\mathbf{S}$, we crop point cloud $\mathbf{P_C}$ by removing all points with negative scores. We use $\mathbf{P}_C^*$ to denote the cropped point cloud in Section IV-B.

B. Pose estimation

A key challenge in multi-modality storage is achieving high accuracy in placement pose generation, particularly in compact spaces like placing a book on a shelf, where gaps are minimal, and the pose must be both physically plausible and collision-free. Previous works on pairwise object manipulation [2], [7] demonstrate that decomposing pose estimation—first finding point-wise correspondence in the point cloud, then computing the 6D pose—yields more stable and accurate results than direct pose prediction. We therefore utilize a similar pipeline in our method. We train a network $\mathbf{C}_\phi(\mathbf{P}_C^*,\mathbf{P}_O)$ to predict the correspondence matrix $\mathbf{C}$ between two geometries $\mathbf{P}_C^*$ and $\mathbf{P}_O$. This correspondence $\mathbf{C}$ models which point on $\mathbf{P}_C^*$ should be in contact with which point on $\mathbf{P}_O$. Then, we apply Arun's algorithm, which

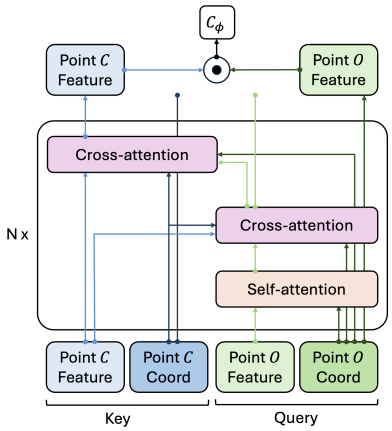

Fig. 4: The correspondence prediction architecture inspired by IMOP [23].

is a least squares optimization method that minimizes the distance between the corresponding points. Arun's algorithm returns the goal pose, $\mathbf{T}_{WO}$. We present a toy 2D example in Fig. 3 to illustrate how our pose estimation pipeline looks.

Data labeling: We sample a random size bounding box around the demonstrated storage location. We crop $\mathbf{P}_C$ using this bounding box to get $\mathbf{P}_C^*$. The ground-truth correspondence identifies which parts of $\mathbf{P}_C^*$ and $\mathbf{P}_O$ should be in contact. We apply $\mathbf{T}_{WO}$ to $\mathbf{P}_O$ using Eq. 1, resulting in $\mathbf{P}'_O$. Subsequently, we calculate the pairwise distance between all points in $\mathbf{P}'_O$ and all points in $\mathbf{P}_C^*$. For any two points in $\mathbf{P}_C^*$ and $\mathbf{P}_O$, their correspondence value is set to 1 if their distance is less than a threshold ε_{corr}, and 0 otherwise. Correspondence matrix $\mathbf{C}$'s shape is $(N_O \times N_C)$. Mathematically, $\mathbf{C}$ is defined as follows:

$$\mathbf{C}(i,j) = \begin{cases} 1, & ||v'_i - v_j||_2 < \varepsilon_{corr}, v'_i \in \mathbf{P}'_O, v_j \in \mathbf{P}_C^* \\ 0, & else \end{cases} \tag{8}$$

Training: The training for correspondence is conducted through pure supervised learning. We assume that the multi-modality problem has been addressed by the diffusion-based affordance prediction, leading to the existence of only one optimal correspondence for given $\mathbf{P}_C^*$ and $\mathbf{P}_O$. To this end, we train a network $\mathbf{C}_\phi(\mathbf{P}_C^*, \mathbf{P}_O)$ to approximate $\mathbf{C}$. We employ a focal loss between $\mathbf{C}_\phi(\mathbf{P}_C^*, \mathbf{P}_O)$ and the ground-truth $\mathbf{C}$ as training objective:

$$L^{corr} = \sum_{i=1}^{N_O} \sum_{j=1}^{N_C} \log\left(\mathbf{C}(i,j)\mathbf{C}_\phi(i,j)\right) \cdot \left(1 - \mathbf{C}(i,j)\mathbf{C}_\phi(i,j)\right)^\gamma \tag{9}$$

The focal loss is specifically chosen to mitigate the imbalance in data distribution. γ is a hyper-parameter to tune the balancing strength.

Architecture: We employ Point-Transformer2 [24] as the 3D backbone network to extract point-wise features $\{f_i\}_{i=1}^N$ for both $\mathbf{P}_C^*$ and $\mathbf{P}_O$. Point-Transformer2 introduces an efficient attention mechanism termed Grouped Vector Attention (GVA). Unlike classical attention mechanisms that calculate the attention between all key tokens and query tokens, GVA computes attention within predefined groups, necessitating the establishment of these groups beforehand. In 3D problems, where each token is associated with 3D points, we can utilize K-nearest-neighbors (KNN) to determine the attention groups. For instance, to compute a KNN-based GVA between two point clouds $\mathbf{P}_1$ and $\mathbf{P}_2$, where $\mathbf{P}_1$ serves as the query point cloud and $\mathbf{P}_2$ as the key point cloud, we determine the K-nearest neighbors of each point in $\mathbf{P}_1$ within $\mathbf{P}_2$. The attention logit for each point in $\mathbf{P}_1$ is then calculated using this point-token and its K-nearest neighbor point tokens in $\mathbf{P}_2$. Due to space constraints, we refer readers to [24] for the complete definition of GVA.

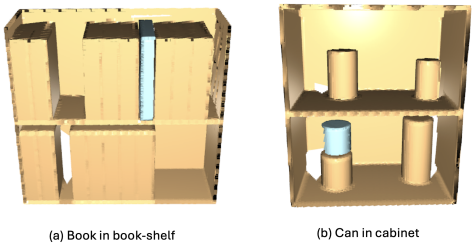

Fig. 5: Samples from RPdiff benchmark. We show two sample scenes from the RPdiff benchmark: one is placing a book into the bookshelf and the other is stacking a can inside a cabinet.

In our approach, we use KNN-GVA for efficient self-attention and cross-attention processing on point cloud data. The full correspondence prediction pipeline is depicted in Fig. 4. $\mathbf{P}_C^*$ and $\mathbf{P}_O$ are fed into the backbone point network to extract point-wise features. Object point tokens $\{f_i\}_{i=1}^{N_O}$ act as query tokens, while container point tokens $\{f_i\}_{i=1}^{N_C}$ serve as key tokens. These query tokens are processed through a KNN-GVA layer for self-attention. Subsequently, cross-attention is performed between the query tokens and key tokens to refine the query tokens. This process is followed by cross-attention between key tokens and query tokens to refine the key tokens. The refined query and key tokens are then used as inputs for the next block. Finally, a dot-product operation is employed to predict the correspondence between points:

$$\mathbf{C}_\phi(i,j) = f_i \cdot f_j, \; i \in \mathbf{P}_O, j \in \mathbf{P}_C^* \tag{10}$$

Pose solving & Ranking: After getting the point-correspondence between $\mathbf{P}_O$ and $\mathbf{P}_C^*$, we can analytically compute the goal pose $\mathbf{T}_{WO}$ using Arun's algorithm [22]. While the pose estimation step is a deterministic process, the previous step, diffusion-based affordance prediction and cropping, is a sampling process. We sample K candidate poses each time, where K is a hyper-parameter. We perform simple collision checking between the resulting $\mathbf{P}'_O$ and $\mathbf{P}_C^*$: counting how many points of $\mathbf{P}_C^*$ fall within the bounding box of $\mathbf{P}'_O$. Candidates are ranked based on this collision estimation.

V. EXPERIMENTS

A. Simulation Experiment

We evaluate our method using the benchmark from RPDiff [6] (check Fig. 5), which provides a challenging

Method	Book/Shelf	Can/Cabinet
C2F Q-attn	57%	51%
R-NDF-base	00%	14%
NSM-base	02%	08%
NSM-base + CVAE	17%	19%
RPDiff	94%	85%
DAP (ours)	**98%**	**94%**

TABLE I: Performance on RPDiff benchmark (Success rate).

Method	Book/Shelf	Can/Cabinet
CAP	24%	36%
DAP (ours)	**98%**	**94%**

TABLE II: Ablation study on RPDiff benchmark.

simulation environment for addressing the multi-modal rearrangement problem. This environment includes tasks such as book shelving, can stacking, and cup hanging, all of which highlight the benchmark's complexity due to the variability in container and object geometries within each task. This variability demands a model's ability to generalize across different geometric configurations. We exclude the cup hanging task from evaluation as it does not match our problem requirement that the object is to be placed in a bigger container. We use the same baselines as RPDiff, comparing our method against five approaches, each offering different solutions to multi-modal rearrangement challenges.

Coarse-to-Fine Q-attention (C2F-QA): Adapted from classification, predicts a score distribution over a voxelized scene to identify object centroid translations, refining predictions to higher resolutions before predicting object rotation. The highest-scoring transformation is executed.

Relational Neural Descriptor Fields (R-NDF): R-NDF uses a neural field shape representation to match local coordinate frames with category-level 3D models for relational tasks. The "R-NDF-base" version lacks the refinement energy-based model from the original.

Neural Shape Mating (NSM) + CVAE: NSM aligns paired point clouds using a Transformer. "NSM-base" trains on large perturbations without local cropping and makes a single prediction. Enhanced with a Conditional Variational Autoencoder (CVAE), NSM can predict multiple transforms, selecting the top-scoring one for execution. "NSM-base" and "NSM-base + CVAE" are considered separate baselines.

Relational Pose Diffusion (RPDiff): RPDiff operates on 3D point clouds, generalizing across new geometries, poses, and layouts. It tackles multiple similar rearrangement solutions through an iterative pose de-noising strategy, producing precise, multi-modal outputs. It was the leading method on RPDiff's benchmark until this work.

TABLE I compares DAP with baselines, showing that while RPDiff outperforms other methods, DAP surpasses RPDiff significantly, demonstrating superior capability. DAP's efficiency is evident as it requires only one hour each for training its affordance prediction and pose estimation modules on a 3090 GPU, compared to RPDiff's eight days of training on a V100 GPU, highlighting DAP's remarkable efficiency.

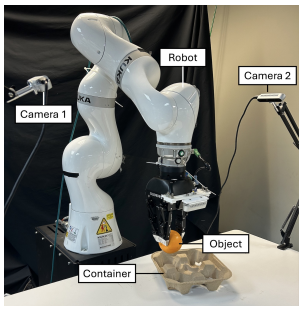

Fig. 6: Robot setup: a Kuka robot equipped with two RealSense D415 cameras and a three-finger Robotiq hand.

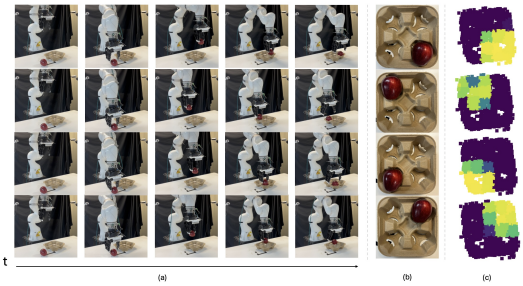

Fig. 7: Real experiment on fruit storage. From left to right are: (a) robot execution recording; (b) final storage result; and (c) placeable affordance prediction.

B. Ablation Study

To analyze the impact of diffusion-based affordance prediction, we conducted an ablation study upon RPDiff benchmark. We compare DAP with a variant framework without using DDPM loss:

Classification Affordance Prediction (CAP): Instead of treating affordance prediction as a generative task, we approach it as a classification problem using cross-entropy loss. The architecture remains the same as DAP. During inference we do not perform iterative de-noising, but provide the classification in one step.

The results of our ablation study are depicted in TABLE II. Classification Affordance Prediction (CAP) significantly underperforms compared to the complete DAP. This finding confirms our hypothesis that diffusion-based affordance prediction effectively addresses multi-modality issues.

C. Real world Experiment

We conducted a qualitative real-world experiment to evaluate DAP's performance using a real-to-real setup for both data collection and deployment, unlike previous sim-

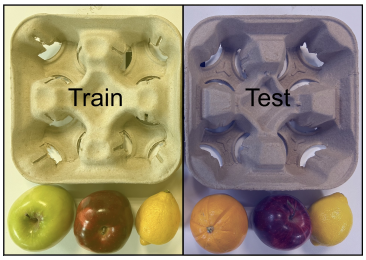

Fig. 8: Task objects: the fruits and storage racks used in the real-world fruit storage task for training (left) and testing (right).

to-sim or sim-to-real setups [5], [6]. This approach faces challenges from noisier and scarcer data compared to the clean, abundant data available in simulations. DAP is the first to address the multi-modality storage problem in a real-to-real context, advancing beyond methods like RPDiff [6] and StructDiffusion [5] that rely on sim-to-real setups. While frameworks like Transporter networks [26] and CLIPort [27] handle real-to-real setups, they do not address multi-modality issues, which necessitates a delicate balance between the model's representational capacity and data efficiency.

Robot Setup: We used a Kuka IIWA 14 robot arm with a Robotiq 3-finger adaptive gripper, and two RealSense D415 cameras to observe the container and object (refer Fig. 6).

Task: We trained and tested DAP on a real-world fruit storage task, where the robot arm picks and places a fruit into one of four rack slots, based on initial point clouds. We collected 80 demonstrations using unseen fruits and storage racks during testing. Segment-Any-Thing (SAM) [28], [29] was used for segmentation. As shown in Fig. 7, DAP successfully detected all four placeable regions, demonstrating its ability to generalize to unseen objects and containers.

VI. CONCLUSION

We introduce DAP, a diffusion-based affordance prediction pipeline for multi-modality storage problems, focused on placing a target object into a larger container. DAP involves two steps: diffusion-based prediction and pose estimation. First, it samples a placeable region in the container using a diffusion model, then computes point-wise correspondence between the target object and the cropped container region to identify contact areas. Arun's algorithm is used to determine the object's goal pose relative to the container. Our experiments, in both simulation and real-world scenarios, show that DAP outperforms previous methods in performance and training efficiency. We believe DAP will inspire further research in multi-modality pair-wise object manipulation.

REFERENCES

[1] S. James, K. Wada, T. Laidlow, and A. J. Davison, "Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 739–13 748.

[2] A. Simeonov, Y. Du, Y.-C. Lin, A. R. Garcia, L. P. Kaelbling, T. Lozano-Pérez, and P. Agrawal, "Se (3)-equivariant relational rearrangement with neural descriptor fields," in *Conference on Robot Learning*. PMLR, 2023, pp. 835–846.

[3] Y.-C. Chen, H. Li, D. Turpin, A. Jacobson, and A. Garg, "Neural shape mating: Self-supervised object assembly with adversarial shape priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 724–12 733.

[4] W. Liu, C. Paxton, T. Hermans, and D. Fox, "Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6322–6329.

[5] W. Liu, T. Hermans, S. Chernova, and C. Paxton, "Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects," in *Workshop on Language and Robotics at CoRL 2022*, 2022.

[6] A. Simeonov, A. Goyal, L. Manuelli, Y.-C. Lin, A. Sarmiento, A. R. Garcia, P. Agrawal, and D. Fox, "Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement," in *Conference on Robot Learning*. PMLR, 2023, pp. 2030–2069.

[7] C. Pan, B. Okorn, H. Zhang, B. Eisner, and D. Held, "Tax-pose: Task-specific cross-pose estimation for robot manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 1783–1792.

[8] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[9] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[10] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia, "Pointgroup: Dual-set point grouping for 3d instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, 2020, pp. 4867–4876.

[11] J. Sun, C. Qing, J. Tan, and X. Xu, "Superpoint transformer for 3d scene instance segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 2393–2401.

[12] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.

[13] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3d: Mask transformer for 3d semantic instance segmentation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8216–8223.

[14] M. Kolodiazhnyi, A. Vorontsova, A. Konushin, and D. Rukhovich, "Oneformer3d: One transformer for unified point cloud segmentation," *arXiv preprint arXiv:2311.14405*, 2023.

[15] 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*, 2021, pp. 1778–1787.

[16] T. Nguyen, M. N. Vu, B. Huang, T. Van Vo, V. Truong, N. Le, T. Vo, B. Le, and A. Nguyen, "Language-conditioned affordance-pose detection in 3d point clouds," *arXiv preprint arXiv:2309.10911*, 2023.

[17] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[19] 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.

[20] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion policies as an expressive policy class for offline reinforcement learning." International Conference on Learning Representations, 2023.

[21] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4195–4205.

[22] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[23] X. Zhang and A. Boularias, "One-shot imitation learning with invariance matching for robotic manipulation," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[24] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 330–33 342, 2022.

[25] Y. Li, S. Si, G. Li, C.-J. Hsieh, and S. Bengio, "Learnable fourier features for multi-dimensional spatial positional encoding," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 816–15 829, 2021.

[26] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 726–747.

[27] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on robot learning*. PMLR, 2022, pp. 894–906.

[28] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[29] X. Zhang, Y. Wang, and A. Boularias, "Detect every thing with few examples," 2023.