Shelving, Stacking, Hanging: Relational Pose Diffusion for Multi-modal Rearrangement

Anonymous Author(s)
Affiliation
Address
email

Abstract: We propose a system for rearranging objects in a scene to achieve a desired object-scene placing relationship, such as a book inserted in an open slot of a bookshelf. The pipeline generalizes to novel geometries, poses, and layouts of both scenes and objects, and is trained from demonstrations to operate directly on 3D point clouds. Our system overcomes challenges associated with the existence of many geometrically-similar rearrangement solutions for a given scene. By leveraging an iterative pose de-noising training procedure, we can fit multi-modal demonstration data and produce multi-modal outputs while remaining precise and accurate. We also show the advantages of conditioning on relevant local geometric features while ignoring irrelevant global structure that harms both generalization and precision. We demonstrate our approach on three distinct rearrangement tasks that require handling multi-modality and generalization over object shape and pose in both simulation and the real world.

Keywords: Object Rearrangement, Multi-modality, Manipulation, Point Clouds

1 Introduction

Consider Figure 1, which illustrates (1) placing a book on a partially-filled shelf and (2) hanging a mug on one of the multiple racks on a table. These tasks involve reasoning about geometric interactions between an object and the scene to achieve a goal, which is a key requirement in many cleanup and de-cluttering tasks of interest to the robotics community [1]. In this work, we enable a robotic system to perform one important family of such tasks: 6-DoF rearrangement of rigid objects [2]. Our system uses point clouds obtained from depth cameras, allowing real-world operation with unknown 3D geometries. The rearrangement behavior is learned from a dataset of examples that show the desired object-scene relationship – a scene and (segmented) object point cloud are observed and a demonstrator transforms the object into a final configuration.

Real-world scenes are often composed of objects whose shapes and poses can vary independently. Such composition creates scenes that (i) present combinatorial variation in geometric appearance and layout (e.g., individual racks may be placed anywhere on a table) and (ii) offer many locations and geometric features for object-scene interaction (e.g., multiple slots for placing the book and multiple racks for hanging the mug). These features of real-world scenes bring about two key challenges for learning that go hand-in-hand: multi-modal placements and generalization to diverse scene layouts.

- **Multi-modality** appears in the *output* space of rearrangement prediction. There may be many scene locations to place an object, and these multiple placement possibilities create difficulties during both learning and deployment. Namely, a well-known challenge in *learning* from demonstrations is fitting a dataset containing similar inputs that have different associated targets (modes). Moreover, during deployment, predicting multiple candidate rearrangements can help the robot choose the ones that also satisfy any additional constraints, such as workspace limits and collision avoidance. Therefore, the system must *predict* multi-modal outputs that span as many different rearrangement solutions as possible.
- **Generalization** must be addressed when processing the *inputs* to the system. A scene is composed of many elements that vary in both shape and layout. For example, a shelf can be located anywhere in the environment, and there are many possible book arrangements within a shelf. The point clouds

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

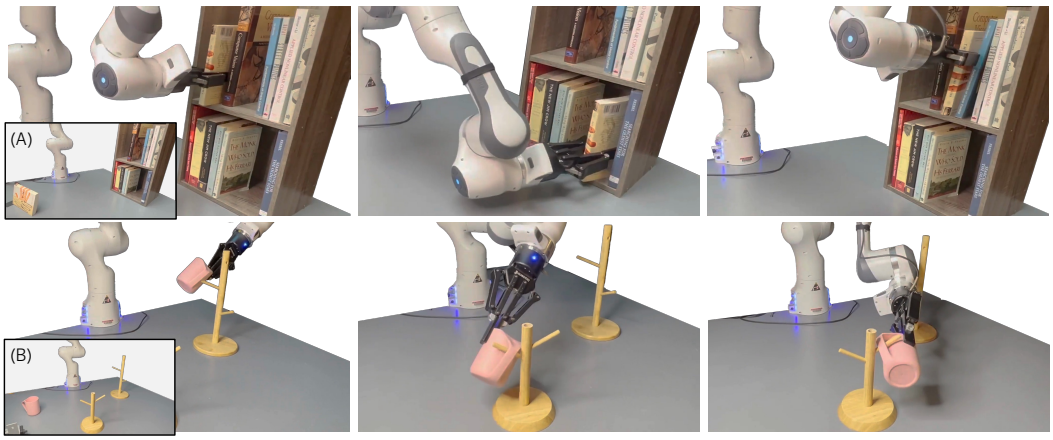


Figure 1: By learning from a set of demonstrations of a rearrangement task, such as *place the book in the shelf* (A) and *hang the mug on the rack* (B), Relational Pose Diffusion (RPDiff) can produce *multiple* transformations that achieve the same object-scene relationship for new object/scene pairs.

that are presented to the model reflect this diversity. Generalizing to such input variability is harder than generalizing to shape and pose variations for a single object, due to the combinatorially many arrangements and layouts of scenes. Moreover, the system must also generalize to any possible initial configuration of the object.

Given a dataset of final object-scene point clouds (obtained by transforming the observed object point cloud into its resultant configuration at the end of the demo), we can synthesize many initial object configurations as perturbations of the final point clouds. Using this data, we can naturally cast rearrangement prediction as *point cloud pose de-noising*. From a final object-scene point cloud, we create a "noised" point cloud by randomly transforming the object and train a neural network to predict how to transform the noised point cloud back into the original configuration (using the known perturbation for ground truth supervision). During deployment, we similarly predict a de-noising object transformation that satisfies the learned relation with the scene and use this predicted transformation as the rearrangement action. The robot executes the predicted rearrangement using a combination of grasp sampling, inverse kinematics, and motion planning.

Unfortunately, learning to de-noise from large perturbations in one step can be ineffective when considering multi-modality [3] – creating similar-looking noised point clouds with prediction targets that differ can lead the model to learn an average solution that fits the data poorly. We overcome this difficulty by training the predictor as a diffusion model [4, 5] to perform *iterative* de-noising. By creating a *multi-step* noising process, diffusion models are trained to *incrementally* reverse the process one step at a time. Intuitively, early steps in this reverse process are closer to the ground truth and the associated prediction targets are more likely to be unique across samples – the prediction "looks more unimodal" to the model. The model similarly generates the test-time output in an iterative fashion. By starting this inference procedure from a diverse set of initial guesses, the predictions can converge to a diverse set of final solutions.

While iterative de-noising helps with multi-modality, we must consider how to support generalization to novel scene layouts. To achieve this, we propose to *locally encode* the scene point cloud by cropping a region near the object. Locally cropping the input helps the model generalize by focusing on details in a local neighborhood and ignoring irrelevant and distant distractors. The features for representing smaller-scale patches can also be re-used across different spatial regions and scene instances [6–9]. We use a larger crop size on the initial iterations because the inference procedure starts from random guesses that may be far from a good solution. As the solution converges over multiple iterations, we gradually reduce the crop size to emphasize a more local scene context.

In summary, we present Relational Pose Diffusion (RPDiff), a method that performs 6-DoF relational rearrangement conditioned on an object and scene point cloud, that (1) generalizes across shapes, poses, and scene layouts, and (2) gracefully handles scenarios with multi-modality. We evaluate our approach in simulation and the real world on three tasks, (i) comparing to existing methods that either struggle with multi-modality and complex scenes or fail to achieve precise rearrangement, and (ii) ablating the various components of our overall pipeline.

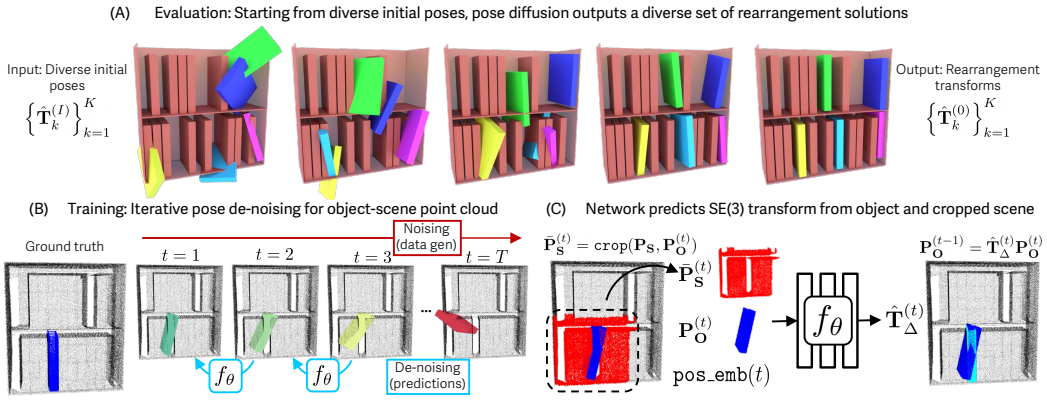


Figure 2: **Method Overview.** (A) Starting from an object and scene point cloud $\mathbf{P_O}$ and $\mathbf{P_S}$, we transform $\mathbf{P_O}$ to a diverse set of initial poses. RPDiff takes the initial object-scene point clouds as input, iteratively updates the object pose, and outputs a *set* of object configurations that satisfy a desired relationship with the scene. This enables integrating RPDiff with a planner to search for a placement to execute while satisfying additional system constraints. (B) The model is trained to perform *iterative pose de-noising*. Starting from object-scene point clouds that satisfy the desired task, we apply a sequence of perturbations to the object and train the model to predict SE(3) transforms that remove the noise one step at a time. (C) To facilitate generalization to novel scene layouts, we crop the scene point cloud to the region near the object point cloud.

2 Problem Setup

Our goal is to predict a set of SE(3) transformations $\{\mathbf{T}_k\}_{k=1}^K$ that accomplish an object rearrangement task given the scene $\mathbf{S}$ and the object $\mathbf{O}$, represented as 3D point clouds ($\mathbf{P_S} \in \mathbb{R}^{M\times 3}$ and $\mathbf{P_O} \in \mathbb{R}^{N\times 3}$, respectively). By selecting (i.e., via a learned scoring function) and applying one transformation from this set, we can place the object in a manner that fulfills the desired geometric relationship with the scene. We assume the object point cloud is segmented from the whole scene, which does not have any additional segmented objects (e.g., we cannot segment any individual books on the shelf). We also assume a training dataset $\mathcal{D} = \{(\mathbf{P_O}, \mathbf{P_S})\}_{l=1}^L$ where each data point represents an object placed at the desired configuration. For example, $\mathcal{D}$ could include point clouds of books and bookshelves (with different shapes, poses, and configurations of books on the shelf), and SE(3) transformations that place the books in one of the available slots. These demonstrations could come from a human or a scripted algorithm with access to ground truth object states in simulation.

Critically, depending on constraints imposed by other system components (e.g., available grasps, robot reachability, collision obstacles), the system must be capable of producing *multi-modal* output transformations. Predicting diverse outputs enables searching for a placement that can be feasibly executed. For execution on a robot, the robot has access to a grasp sampler [10], inverse kinematics (IK) solver, and motion planner to support generating and following a pick-and-place trajectory.

3 Method

The main idea is to iteratively de-noise the 6-DoF pose of the object until it satisfies the desired geometric relationship with the scene point cloud. An overview of our framework is given in Fig. 2.

3.1 Object-Scene Point Cloud Diffusion via Iterative Pose De-noising

We represent a rearrangement action $\mathbf{T}$ as the output of a multi-step de-noising process for a combined object-scene point cloud, indexed by discrete time variable $t = 0, ..., T$. This process reflects a transformation of the object point cloud in its initial noisy configuration ${\mathbf{P_O}}^{(T)}$ to a final configuration ${\mathbf{P_O}}^{(0)}$ that satisfies a desired relationship with the scene point cloud $\mathbf{P_S}$, i.e., ${\mathbf{P_O}}^{(0)} = \mathbf{T}{\mathbf{P_O}}^{(T)}$. To achieve this, we train neural network $f_\theta : \mathbb{R}^{N\times 3} \times \mathbb{R}^{M\times 3} \to \text{SE}(3)$ to predict an SE(3) transformation from the combined object-scene point cloud at each step. The network is trained as a diffusion model [4, 5] to incrementally reverse a manually constructed noising process that gradually perturbs the object point clouds until they match a distribution ${\mathbf{P_O}}^{(T)} \sim p_\mathbf{O}^{(T)}(\cdot \mid \mathbf{P_S})$, which we can efficiently sample from during deployment to begin de-noising at test time.

Test-time Evaluation. Starting with $\mathbf{P_O}$ and $\mathbf{P_S}$, we sample K initial transforms $\{\hat{\mathbf{T}}_k^{(I)}\}_{k=1}^K$ * and apply these to $\mathbf{P_O}$ to create initial object point clouds $\{\hat{\mathbf{P}}_{\mathbf{O},k}^{(I)}\}_{k=1}^K$ where $\hat{\mathbf{P}}_{\mathbf{O},k}^{(I)} = \hat{\mathbf{T}}_k^{(I)}\mathbf{P_O}$. For each of the K initial transforms, we then perform the following update for I steps.† At each iteration i:

$$\mathbf{T}_\Delta^{(i)} = \mathbf{T}_\Delta^{\text{Rand}} f_\theta\Big(\hat{\mathbf{P}}_{\mathbf{O}}^{(i)}, \mathbf{P_S}, \texttt{pos_emb}\big(t\big)\Big) \qquad t = \texttt{i_to_t}(i) \tag{1}$$

$$\hat{\mathbf{T}}^{(i-1)} = \mathbf{T}_\Delta^{(i)}\hat{\mathbf{T}}^{(i)} \qquad \hat{\mathbf{P}}_{\mathbf{O}}^{(i-1)} = \mathbf{T}_\Delta^{(i)}\hat{\mathbf{P}}_{\mathbf{O}}^{(i)} \tag{2}$$

The update $\mathbf{T}_\Delta^{(i)}$ is formed by multiplying the denoising transform predicted by our model f_θ with a perturbation transform $\mathbf{T}_\Delta^{\text{Rand}}$ that is sampled from an iteration-conditioned normal distribution which converges toward deterministically producing an identify transform as i tends toward 0. In the de-noising process, $\mathbf{T}_\Delta^{\text{Rand}}$ helps each of the K samples converge to different multi-modal pose basins (analogously to the perturbation term in Stochastic Langevin Dynamics [11]). The function `pos_emb` represents a sinusoidal position embedding. Since f_θ is only trained on a finite set of t values (i.e., $t = 1, ..., 5$) but we might want to perform the update in Eq. 2 for a larger number of steps, we use the function `i_to_t` to map the iteration i to a timestep value t that the model has been trained on. Details on external noise scheduling and mapping i to t can be found in Appendix A3.

Generally, we search through K solutions $\{\hat{\mathbf{T}}_k^{(0)}\}_{k=1}^K$ for one that can be executed while satisfying all other constraints (e.g., collision-free trajectory). However, we also want a way to select a single output to execute assuming there are no other constraints to satisfy. We may also want to reject "locally optimal" solutions that fail to complete the desired task. To achieve this, we use a separate classifier h_ϕ to score the predicted poses (i.e., $s_k = h_\phi(\mathbf{P_O}_k^{(0)}, \mathbf{P_S})$ where $s \in [0, 1]$), such that the sample indexed with $k_{\text{exec}} = \texttt{argmax}\ \{s_k\}_{k=1}^K$ can be selected for execution‡.

Training. Given a dataset sample $(\mathbf{P_O}, \mathbf{P_S})$, we start with final "placed" object point cloud $\mathbf{P_O}^{(0)} = \mathbf{P_O}$ and randomly sampled timestep $t \in [1, T]$. We then obtain a perturbation transform $\mathbf{T}_{\text{noise}}^{(t)}$ from a timestep-conditioned distribution with appropriately scaled variance and create a noised point cloud $\mathbf{P_O}^{(t)} = \mathbf{T}_{\text{noise}}^{(t)}\mathbf{P_O}$. The task is to predict a transformation that takes one de-noising step as $\hat{\mathbf{T}}_\Delta^{(t)} = f_\theta(\mathbf{P}_{\mathbf{O}}^{(t)}, \mathbf{P_S}, \texttt{pos_emb}(t))$. Network parameters θ are trained to minimize a loss between the prediction $\hat{\mathbf{T}}_\Delta^{(t)}$ and a ground truth target $\mathbf{T}_{\Delta,\text{GT}}^{(t)}$. The loss is composed of the mean-squared translation error, a geodesic rotation distance error [12, 13], and the chamfer distance between the point cloud obtained by applying the predicted transform and the ground-truth next point cloud.

A natural target for f_θ to predict is the inverse of the perturbation, i.e., $\mathbf{T}_{\Delta,\text{GT}}^{(t)} = \mathbf{T}_{\text{noise,inv}}^{(t)} = \left[\mathbf{T}_{\text{noise}}^{(t)}\right]^{-1}$, to encourage recovering the original sample. However, as the perturbation magnitude varies across timesteps, this requires output predictions of different scales for different timesteps. In supervised learning with neural networks, it is advisable to keep the magnitudes of both input and output signals consistent in order to minimize large fluctuations in gradient magnitudes between samples [14]. For this reason, an alternative approach is to encourage the network to take shorter "unit steps" in the *direction* of the original sample. We achieve this by uniformly interpolating the full inverse perturbation as $\{\mathbf{T}_{\text{interp}}^{(s)}\}_{s=1}^t = \texttt{interp}(\mathbf{T}_{\text{noise, inv}}^{(t)}, t)$ and training the network to predict one interval in this interpolated set, i.e., $\mathbf{T}_{\Delta,\text{GT}}^{(t)} = [\mathbf{T}_{\text{interp}}^{(t-1)}]^{-1}\mathbf{T}_{\text{interp}}^{(t)}$ (details in Appendix A2 and A7).

For the success classifier, we generate positive and negative rearrangement examples, where positives use the final demonstration point cloud, $\mathbf{P_O}^{(0)}$, and negatives are obtained by sampling diverse perturbations of $\mathbf{P_O}^{(0)}$. The classifier weights ϕ (separate from weights θ) are trained to minimize a binary cross-entropy loss between the predicted likelihood and the ground truth success labels.

3.2 Architecture

We use a Transformer [15] for processing point clouds and making pose predictions. We use a Transformer because it learns to (i) identify important geometric parts *within* the object and the scene and (ii) capture relationships that occur *between* the important parts of the object and the

* Initial rotations are drawn from a uniform grid over SO(3) , and we uniformly sample translations that position the object within the bounding box of the scene point cloud.

† We denote application of SE(3) transform $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ to 3D point $\mathbf{x}$ as $\mathbf{Tx} = \mathbf{Rx} + \mathbf{t}$

‡ See Appendix A7 for results showing that scoring with h_ϕ performs better than, e.g., uniform output sampling

scene. Starting with $\mathbf{P_O}$ and $\mathbf{P_S}$, we tokenize the point clouds to obtain input features. This can be performed by passing through a point cloud encoder [16, 17], but we simply downsample and append a one-hot feature to each point indicating whether it is part of the object or the scene. We then pass these input tokens through a Transformer encoder and decoder, which performs self-attention on the scene point cloud, and cross-attention between the scene and the object. This produces output features for each point, which are mean-pooled to obtain a global feature vector. The global feature is passed to a set of MLPs which predict the rotation $\mathbf{R} \in \mathrm{SO}(3)$ and a translation $\mathbf{t} \in \mathbb{R}^3$. As in [10, 18], we represent the rotation by predicting vectors $a \in \mathbb{R}^3$ and $b \in \mathbb{R}^3$, finding the component of b that is orthogonal to a, and normalizing to obtain $\hat{a}$ and $\hat{b}$. We then take a cross product to obtain $\hat{c} = \hat{a} \times \hat{b}$, and construct $\mathbf{R}$ as $\begin{bmatrix}\hat{a} & \hat{b} & \hat{c}\end{bmatrix}$. We incorporate iteration t by passing $\texttt{pos_emb}(t)$ as a global token in the decoder and adding it to the global output feature. To predict success likelihood, we process point clouds with the same Transformer but output a single scalar followed by a sigmoid.

3.3 Local Conditioning

The approach described above conditions the transform regression on both the object and the scene. However, distant global information can act as a distraction and hamper both precision and generalization. Prior work has also observed this and suggested hard attention mechanisms on the input observation like cropping task-relevant regions to improve generalization by ignoring irrelevant distractors [8, 9]. Building on this intuition, we modify the scene point cloud by cropping $\mathbf{P_S}$ to only include points that are near the current object point cloud $\mathbf{P_O}^{(i)}$. Our modified pose prediction thus becomes $\hat{\mathbf{T}}_{\Delta}^{(i)} = f_\theta\Big(\hat{\mathbf{P}}_{\mathbf{O}}^{(i)}, \bar{\mathbf{P}}_{\mathbf{S}}^{(i)}, \texttt{pos_emb}\big(\texttt{i_to_t}(i)\big)\Big)$ where $\bar{\mathbf{P}}_{\mathbf{S}}^{(i)} = \texttt{crop}(\hat{\mathbf{P}}_{\mathbf{O}}^{(i)}, \mathbf{P_S})$. The function $\texttt{crop}$ returns the points in $\mathbf{P_S}$ that are within an axis-aligned box centered at the mean of $\hat{\mathbf{P}}_{\mathbf{O}}^{(i)}$. We try one variant of the $\texttt{crop}$ function that returns a fixed-size crop, and another that adjusts the crop size depending on the iteration variable i (the size starts large and gradually decreases for later iterations).

4 Experiments: Design and Setup

Our quantitative experiments in simulation are designed to answer the following questions:

1. How well does RPDiff achieve the desired tasks compared to other methods for rearrangement?
2. How successful is RPDiff in producing a diverse set of transformations compared to baselines?
3. How does our performance change with different components modified or removed?

We also demonstrate RPDiff within a pick-and-place pipeline in the real world to further highlight the benefits of multi-modal generation and our ability to transfer from simulation to the real world.

4.1 Task Descriptions and Training Data Generation

We evaluate our method on three tasks that emphasize multiple available object placements: (1) placing a book on a partially-filled bookshelf, (2) stacking a can on a stack of cans or an open shelf region, and, (3) hanging a mug on one of many racks with many hooks. As a sanity check for our baseline implementations, we also include two easier versions of "mug on rack" tasks that are "less multi-modal". These consist of (i) hanging a mug on one rack with a single hook and (ii) hanging a mug on one rack with two hooks. We programmatically generate ∼1k-3k demonstrations of each task in simulation with a diverse set of procedurally generated shapes (details in Appendix A2). We use each respective dataset to train both RPDiff and each baseline (one model for each task). For our real-world experiments, we directly transfer and deploy the models trained on simulated data.

4.2 Evaluation Environment Setup

Simulation. We conduct quantitative experiments in the PyBullet [19] simulation engine. The predicted transform is applied to the object by simulating an insertion controller which directly actuates the object's center of mass (i.e., there is no robot in the simulator). The insertion is executed from a "pre-placement" pose that is offset from the predicted placement. This offset is obtained using prior knowledge about the task and the objects and is not predicted (see Appendix A6 for details). To quantify performance, we report the success rate over 100 trials, using the final simulator state to compute success. We also quantify coverage by comparing the set of predictions to a ground truth set of feasible solutions and computing the corresponding precision and recall. Details on the insertion controller, computation of $\mathbf{T}^{\text{pre-place}}$, and the task success criteria can be found in the Appendix.

Method	Mug/EasyRack	Mug/MedRack	Book/Shelf	Mug/Multi-MedRack	Can/Cabinet
C2F Q-attn	0.31	0.31	0.57	0.26	0.51
R-NDF-base	0.75	0.29	0.00	0.00	0.14
NSM-base	0.83	0.17	0.02	0.01	0.08
NSM-base + CVAE	–	0.39	0.17	0.27	0.19
RPDiff (**ours**)	**0.92**	**0.83**	**0.94**	**0.86**	**0.85**

Table 1: **Rearrangement success rates in simulation.** On tasks with a unimodal solution space and simpler scene geometry, each method performs well (see **Mug/EasyRack** task). However, on tasks involving more significant shape variation and multi-modality, RPDiff works better than all other approaches.

Real World. We also apply RPDiff to object rearrangement in the real world using a Franka Panda robotic arm with a Robotiq 2F140 parallel jaw gripper. We use four calibrated depth cameras to observe the tabletop environment. From the cameras, we obtain point clouds $\mathbf{P_O}$ and $\mathbf{P_S}$ of object $\mathbf{O}$ and scene $\mathbf{S}$ and apply our method to predict transformation $\mathbf{T}$. $\mathbf{T}$ is applied to $\mathbf{O}$ by transforming an initial grasp pose $\mathbf{T}_{\text{grasp}}$ (using a separate grasp predictor [10]) by $\mathbf{T}$ to obtain a placing pose $\mathbf{T}_{\text{place}} = \mathbf{T}\mathbf{T}_{\text{grasp}}$, and inverse kinematics and motion planning is used to reach $\mathbf{T}_{\text{grasp}}$ and $\mathbf{T}_{\text{place}}$.

4.3 Baselines

Coarse-to-Fine Q-attention (C2F-QA). This method adapts the classification-based approach proposed in [8] to relational rearrangement. We train a fully convolutional network to predict a distribution of scores over a voxelized representation of the scene, denoting a heatmap over candidate translations of the object centroid. The model runs in a "coarse-to-fine" fashion by performing this operation multiple times over a smaller volume at higher resolutions. On the last step, we pool the voxel features and predict a distribution over a discrete set of rotations to apply to the object. We use our success classifier to rank the predicted transforms and execute the output with the top score.

Relational Neural Descriptor Fields (R-NDF). R-NDF [20] uses a neural field shape representation trained on category-level 3D models as a feature space wherein local coordinate frames can be matched via nearest-neighbor search. R-NDFs have been used to perform relational rearrangement tasks via the process of encoding and localizing task-relevant coordinate frames near the object parts that must align to achieve the desired rearrangement. We call this method "R-NDF-base" because it does not feature the additional energy-based model for refinement proposed in the original work.

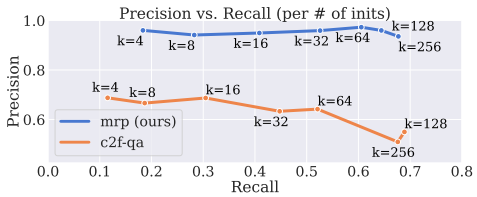
Neural Shape Mating (NSM) + CVAE. Neural Shape Mating (NSM) [3] uses a Transformer to process a pair of point clouds and predict how to align them. Architecturally, NSM is the same as our relative pose regression model, with the key differences of (i) being trained on arbitrarily large perturbations of the demonstration point clouds, (ii) not using local cropping, and (iii) only making a single prediction. We call this baseline "NSM-base" because we do not consider the auxiliary signed-distance prediction and learned discriminator proposed in the original approach [3]. While the method performs well on unimodal tasks, the approach is not designed to handle multi-modality. Therefore, we modify NSM to act as a conditional variational autoencoder (CVAE) [21] to better enable learning from multi-modal data. We use NSM+CVAE to predict multiple transforms and execute the output with the top score produced by our success classifier.

5 Results

5.1 Simulation: Success Rate Evaluation

Table 1 shows the success rates achieved by each method on each task and highlights that our method performs best across the board. The primary failure mode from C2F-QA is low precision in the rotation prediction. Qualitatively, the C2F-QA failures are often close to a successful placement but still cause the insertion to fail. In contrast, our refinement procedure outputs very small rotations that can precisely align the object relative to the scene.

Similarly, we find R-NDF performs poorly on more complex scenes with many available placements. We hypothesize this is because R-NDF encodes scene point clouds into a global latent representation. Since the single set of latent variables must capture all possible configurations of the individual scene components, global encodings fail to represent larger-scale scenes with significant geometric variability [6, 7]. For instance, R-NDF can perform well with individual racks that all have a single hook, but fails when presented with multiple racks.



(a)

Crop Method	Success Rate		
	Mug/Rack	Book/Shelf	Can/Cabinet
None	0.58	0.62	0.42
Fixed	0.76	0.92	0.75
Varying	0.86	0.94	0.85

(b)

Figure 3: (a) **Coverage evaluation in simulation.** Both RPDiff and C2F-QA achieve high placement coverage, but the prediction quality of C2F-QA reduces with an increase in coverage, while RPDiff produces outputs that remain precise while achieving high coverage. (b) **Cropping ablations.** Success rate of RPDiff with different kinds of scene point cloud conditioning. The increased success rate achieved when using local scene cropping highlights the generalization and precision benefits of focusing on a local spatial region.

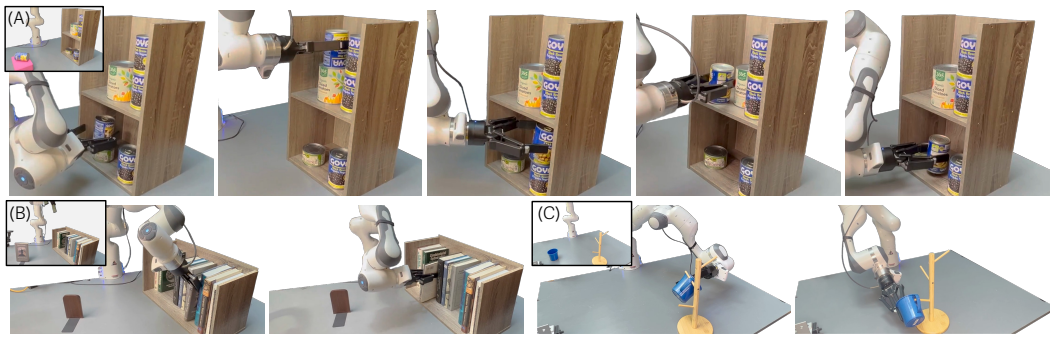


Figure 4: **Real-world multi-modal rearrangement.** Executing **Can/Cabinet** (A), **Book/Shelf** (B), and **Mug/Rack** (C) in the real world. For each task, the initial object-scene configuration is shown in the top-left image, and examples of executing multiple inferred placements are shown in the main image sequence.

Finally, while NSM+CVAE improves upon the unimodal version of NSM, we find the generated transforms vary too smoothly between the discrete modes (e.g., book poses that lie in between the available shelf slots), an effect analogous to the typical limitation of VAE-based generators producing blurry outputs in image generation. We hypothesize this over-smoothing is caused by trying to make the approximate posterior match the unimodal Gaussian prior. This contrasts RPDiff's ability to "snap on" to the available placing locations in a given scene. More discussion on the performance obtained by the baseline methods and how they are implemented can be found in Appendix A6.

5.2 Simulation: Coverage Evaluation

Next, we evaluate the ability to produce multi-modal outputs that cover the space of rearrangement solutions and examine the tradeoff between prediction quality and coverage. Since coverage is affected by the number of parallel runs we perform, we compute average recall and average precision for different values of K (the number of initial poses that are refined). Precision and recall are computed with respect to a set of ground truth rearrangement solutions for a given object-scene instance. We consider positive predictions as those that are within a 3.5cm position and 5-degree rotation threshold of a ground truth solution.

Fig. 3a shows the results for our approach along with C2F-QA, the best-performing baseline. We observe a trend of better coverage (higher recall) with more outputs for both approaches. For a modest value of $K = 32$, we observe RPDiff is able to cover over half of the available placement solutions on average, with C2F-QA achieving slightly lower coverage. However, we see a stark difference between the methods in terms of precision as the number of outputs is increased. C2F-QA suffers from more outputs being far away from any ground truth solution, while our approach maintains consistently high generation quality even when outputting upwards of 200 rearrangement poses.

5.3 Simulation: Local Cropping Ablations and Modifications

Finally, we evaluate the benefits of introducing local scene conditioning into our relative pose regression model. Table 3b shows the performance variation of our method with different kinds of scene point cloud conditioning. We achieve the best performance with the version of local

conditioning that varies the crop sizes on a per-iteration basis. Using a fixed crop size marginally reduces performance, while conditioning on the whole uncropped scene point cloud performs much worse. This highlights the generalization and precision benefits of focusing on a local spatial region near the object in its imagined configuration. It also suggests an advantage of using a coarse-to-fine approach that considers a larger region on earlier iterations. Additional results examining the effect of the success classifier, external noise, and parameterization of `i_to_t` can be found in Appendix A7.

5.4 Real World: Object rearrangement via pick-and-place

Finally, we use RPDiff to perform relational rearrangement via pick-and-place on real-world objects and scenes. Fig. 1 and Fig. 4 show the robot executing *multiple* inferred placements on our three tasks. We relied on our approach's ability to output multiple solutions, as some geometrically valid placements were not kinematically feasible for the robot based on its workspace limits and the surrounding collision geometry. Please see the supplemental video for real-world execution.

6 Related Work

Object Rearrangement from Perception. Object rearrangement that operates with unknown objects in the real world by operating from perceptual input has been an area of growing interest [2, 3, 8, 20, 22–48]. One straightforward method is end-to-end training to directly regress the relative transformation, as in Neural Shape Mating (NSM) [3]. Others have explored identifying task-relevant object parts and then solving for the desired alignment, as in TAX-Pose and R-NDF [20, 40, 48]. However, many of these approaches in their naive form struggle when there is *multi-modality* (NSM and TAX-Pose can only output a single solution). There has been success addressing multi-modality by performing classification over a discretized version of the search space [8, 42, 44, 46, 47, 49], but these methods are typically less precise.

Denoising Diffusion and Iterative Regression. Diffusion models [4, 50] use an iterative de-noising process to perform generative modeling. While they were originally designed for generating images, they have been extended to other domains including waveforms [51, 52], 3D shapes [53, 54], and decision-making[55–57]. Several approaches have applied diffusion models (and related energy-based models) to a variety of robotics domains, including policy learning [58, 59], motion planning/trajectory optimization [60–62], grasping [56], and object rearrangement [20, 41]. The use of iterative regression has also been successful in other domains such as pose estimation [63–66].

SE(3)-DiffusionFields [56] integrate learned 6-DoF grasp distributions within a trajectory optimization framework, and LEGO-Net [57] employs iterative de-noising to generate realistic-looking room layouts. Our work differs in that we do not assume known object states or 3D models. Most similar to our work, StructDiffusion [41] uses a diffusion model to perform language-conditioned object rearrangement with point clouds. While the focus in [41] is to rearrange multiple objects into abstract structures (e.g., circles, lines) specified via natural language, we emphasize covering all rearrangement modes and integrating with sampling-based planners.

7 Limitations and Conclusion

Limitations. The amount of demonstration data we use can only be easily obtained via scripted policies in simulation. Future work could explore pre-trained representations and multi-task learning to reduce data requirements for new tasks. We also suffer from some amount of sim2real gap due to training on simulated point clouds. Incorporating broader sim2real transfer techniques and finetuning on real-world data would be useful to investigate. Finally, we execute the predicted placement open loop. These types of rearrangement tasks would benefit from a closed-loop policy that can track progress toward the predicted rearrangement goal and react to/recover from disturbances.

Conclusion. This work presents an approach for rearranging objects in a scene to achieve a desired placing relationship, while operating with novel geometries, poses, and scene layouts. Our system can produce multi-modal distributions of object transformations for rearrangement, overcoming the difficulty of fitting multi-modal demonstration datasets and facilitating integration with planning algorithms that require diverse actions to search through. Our results illustrate the capabilities of our framework across a diverse range of rearrangement tasks involving objects and scenes that present a large number of feasible rearrangement solutions.

References

[1] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023.

[2] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020.

[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*, pages 12724–12733, 2022.

[4] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[5] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[6] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022. ISSN 1467-8659. doi:10.1111/cgf.14505.

[7] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger. Convolutional occupancy networks. In *Proc. ECCV*, 2020.

[8] 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*, pages 13739–13748, 2022.

[9] S. James and A. J. Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):1612–1619, 2022.

[10] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021.

[11] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[12] J. J. Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 4, pages 3993–3998. IEEE, 2004.

[13] D. Q. Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[14] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao. Normalization techniques in training dnns: Methodology, analysis and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[16] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[17] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[18] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[19] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. *GitHub repository*, 2016.

[20] A. Simeonov, Y. Du, L. Yen-Chen, , A. Rodriguez, , L. P. Kaelbling, T. L. Perez, and P. Agrawal. Se(3)-equivariant relational rearrangement with neural descriptor fields. In *Conference on Robot Learning (CoRL)*. PMLR, 2022.

[21] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[22] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *The International Symposium of Robotics Research*, pages 132–157. Springer, 2019.

[23] S. Cheng, K. Mo, and L. Shao. Learning to regrasp by learning to place. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=Qdb1ODTQTnL.

[24] R. Li, C. Esteves, A. Makadia, and P. Agrawal. Stable object reorientation using contact plane registration. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6379–6385. IEEE, 2022.

[25] S. Thompson, L. P. Kaelbling, and T. Lozano-Perez. Shape-based transfer of generic skills. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5996–6002. IEEE, 2021.

[26] A. Simeonov, Y. Du, B. Kim, F. R. Hogan, J. Tenenbaum, P. Agrawal, and A. Rodriguez. A long horizon planning framework for manipulating rigid pointcloud objects. In *Conference on Robot Learning (CoRL)*, 2020. URL https://anthonysimeonov.github.io/rpo-planning-framework/.

[27] S. Lu, R. Wang, Y. Miao, C. Mitash, and K. Bekris. Online object model reconstruction and reuse for lifelong improvement of robot manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1540–1546. IEEE, 2022.

[28] M. Gualtieri and R. Platt. Robotic pick-and-place with uncertain object instance segmentation and shape completion. *IEEE robotics and automation letters*, 6(2):1753–1760, 2021.

[29] P. Florence, L. Manuelli, and R. Tedrake. Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters*, 2019.

[30] A. Curtis, X. Fang, L. P. Kaelbling, T. Lozano-Pérez, and C. R. Garrett. Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1940–1946. IEEE, 2022.

[31] C. Paxton, C. Xie, T. Hermans, and D. Fox. Predicting stable configurations for semantic placement of novel objects. In *Conference on Robot Learning*, pages 806–815. PMLR, 2022.

[32] 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.

[33] A. Goyal, A. Mousavian, C. Paxton, Y.-W. Chao, B. Okorn, J. Deng, and D. Fox. Ifor: Iterative flow minimization for robotic object rearrangement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14787–14797, 2022.

[34] A. H. Qureshi, A. Mousavian, C. Paxton, M. Yip, and D. Fox. NeRP: Neural Rearrangement Planning for Unknown Objects. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021. doi:10.15607/RSS.2021.XVII.072.

[35] D. Driess, J.-S. Ha, and M. Toussaint. Learning to solve sequential physical reasoning problems from a scene image. *The International Journal of Robotics Research*, 40(12-14):1435–1466, 2021.

[36] D. Driess, J.-S. Ha, and M. Toussaint. Deep visual reasoning: Learning to predict action sequences for task and motion planning from an initial scene image. In *Robotics: Science and Systems 2020 (RSS 2020)*. RSS Foundation, 2020.

[37] 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)*, pages 6322–6329. IEEE, 2022.

[38] W. Goodwin, S. Vaze, I. Havoutis, and I. Posner. Semantically grounded object matching for robust robotic scene rearrangement. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11138–11144. IEEE, 2022.

[39] M. Danielczuk, A. Mousavian, C. Eppner, and D. Fox. Object rearrangement using learned implicit collision functions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6010–6017. IEEE, 2021.

[40] C. Pan, B. Okorn, H. Zhang, B. Eisner, and D. Held. Tax-pose: Task-specific cross-pose estimation for robot manipulation. In *6th Annual Conference on Robot Learning*.

[41] W. Liu, T. Hermans, S. Chernova, and C. Paxton. Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects. *arXiv preprint arXiv:2211.04604*, 2022.

[42] L. Yen-Chen, P. Florence, A. Zeng, J. T. Barron, Y. Du, W.-C. Ma, A. Simeonov, A. R. Garcia, and P. Isola. MIRA: Mental imagery for robotic affordances. In *Conference on Robot Learning (CoRL)*, 2022.

[43] K. Wada, S. James, and A. J. Davison. ReorientBot: Learning object reorientation for specific-posed placement. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[44] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee. Transporter networks: Rearranging the visual world for robotic manipulation. *Conference on Robot Learning (CoRL)*, 2020.

[45] H. Huang, D. Wang, R. Walters, and R. Platt. Equivariant Transporter Network. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022. doi:10.15607/RSS.2022.XVIII.007.

[46] K. Mo, Y. Qin, F. Xiang, H. Su, and L. Guibas. O2O-Afford: Annotation-free large-scale object-object affordance learning. In *Conference on Robot Learning (CoRL)*, 2021.

[47] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[48] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[49] N. M. M. Shafiullah, Z. J. Cui, A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=agTr-vRQsa.

[50] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[51] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

[52] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.

[53] S. Luo and W. Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[54] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[55] M. Janner, Y. Du, J. Tenenbaum, and S. Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

[56] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki. Se(3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[57] Q. A. Wei, S. Ding, J. J. Park, R. Sajnani, A. Poulenard, S. Sridhar, and L. Guibas. Lego-net: Learning regular rearrangements of objects in rooms. *arXiv preprint arXiv:2301.09629*, 2023.

[58] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2023.

[59] P. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. *Conference on Robot Learning (CoRL)*, 2021.

[60] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine. Planning with diffusion for flexible behavior synthesis, 2022.

[61] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal. Is conditional generative modeling all you need for decision-making?, 2022.

[62] Y. Du, T. Lin, and I. Mordatch. Model based planning with energy based models, 2021.

[63] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742, 2016.

[64] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.

[65] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 574–591. Springer, 2020.

[66] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic. Megapose: 6d pose estimation of novel objects via render & compare. *arXiv preprint arXiv:2212.06870*, 2022.

[67] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[68] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

[69] T. Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

[70] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[71] J. Sola, J. Deray, and D. Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018.

[72] T. Chen, A. Simeonov, and P. Agrawal. AIRobot. https://github.com/Improbable-AI/airobot, 2019.

[73] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[74] 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*, pages 652–660, 2017.

[75] K. Murphy, C. Esteves, V. Jampani, S. Ramalingam, and A. Makadia. Implicit-pdf: Non-parametric representation of probability distributions on the rotation manifold. *arXiv preprint arXiv:2106.05965*, 2021.

[76] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 IEEE international conference on robotics and automation*, pages 1–8. IEEE, 2018.

[77] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. In *ICCV*, 2021. URL https://arxiv.org/abs/2104.12229.

[78] A. Fishman, A. Murali, C. Eppner, B. Peele, B. Boots, and D. Fox. Motion policy networks. In *Conference on Robot Learning*, pages 967–977. PMLR, 2023.