Implicit Contact Diffuser: Sequential Contact Reasoning with Latent Point Cloud Diffusion

Zixuan Huang[1], Yinong He*[1], Yating Lin*[1], Dmitry Berenson[1]

***Abstract*— Long-horizon contact-rich manipulation has long been a challenging problem, as it requires reasoning over both discrete contact modes and continuous object motion. We introduce *Implicit Contact Diffuser* (ICD), a diffusion-based model that generates a sequence of neural descriptors that specify a series of contact relationships between the object and the environment. This sequence is then used as guidance for an MPC method to accomplish a given task. The key advantage of this approach is that the latent descriptors provide more task-relevant guidance to MPC, helping to avoid local minima for contact-rich manipulation tasks. Our experiments demonstrate that ICD outperforms baselines on complex, long-horizon, contact-rich manipulation tasks, such as cable routing and notebook folding. Additionally, our experiments also indicate that ICD can generalize a target contact relationship to a different environment. More visualizations can be found on our website https://implicit-contact-diffuser.github.io**

I. Introduction

Interacting with the environment through contact is central to many robotic tasks, such as manipulation and locomotion. Despite the ubiquity of contact interactions, controlling these hybrid systems poses significant challenges due to the complex interplay between discrete contact events and continuous motion. For instance, in cable routing, the robot must generate smooth motions to initiate and maintain contact between the cable and the fixtures (Fig. 1). If the contact breaks at any point, the cable could slip off the fixtures. Moreover, when model errors or external disturbances occur, the robot must adjust its actions accordingly to maintain task success.

A large body of work has attempted to tackle these challenges by planning [1], [2], [3], [4] or trajectory optimization [5], [6], [7] through contact. However, these methods are typically limited to rigid objects, or face limitations in online replanning due to the high computational costs involved.

In this paper, we introduce a learning-based model predictive control (MPC) framework to address this class of problems. In particular, we train a latent diffusion model to generate future contact sequences as subgoals, which guide a MPC controller to generate robot motions that establish the desired contact relationships. A key question, however, is determining the best representation for these contact relationships.

One approach is to use binary contact states. Wi et al. [8] propose to specify desired contact locations by predicting a heatmap over the environment. However, this approach lacks critical information regarding which part of the object should be in contact, a crucial factor for tasks where maintaining precise object-environment interactions is important. Additionally, it cannot capture the dynamic contact switching required in certain tasks.

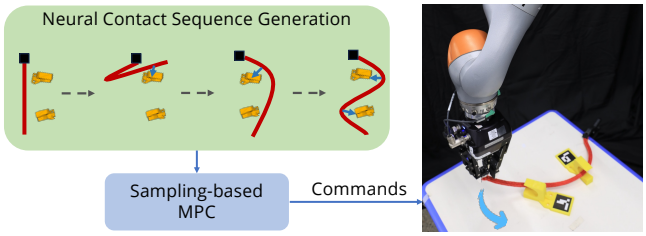

Fig. 1: By predicting future contact sequences using a latent diffusion model, we enable long-horizon contact-rich deformable object manipulation such as cable routing using a sampling-based MPC controller.

To overcome these limitations, we leverage recent advancements in implicit neural representations and encode contact relationships using a modified version of Neural Descriptor Fields (NDF) [9]. We train a scene-level NDF to capture geometric information by predicting occupancy and gradient direction of the signed distance function. By querying the scene NDF with the object's point cloud, we compute a dense, contact-aware representation of the object. Our experiments show that these neural descriptors capture task-relevant geometric relationships (e.g., left or right of a fixture) rather than specific locations, providing more flexible guidance. This allows us to transfer goal contact relationships across different environments at test time.

To capture the contact switching required to reach a goal, we train a latent diffusion model to predict the contact sequence represented by neural descriptors. We also learn a reachability function, similar to Subgoal Diffuser [10], to determine the required sequence length. The key contributions of this paper are: 1) a latent diffusion model that reasons about evolving contact relationships in long-horizon manipulation tasks; 2) an MPC framework that plans motions based on desired contact relationships rather than precise locations. 3) a scene-level neural descriptor field that provides local contact representations, enabling greater generalization across environments.

We validate our method on challenging long-horizon contact-rich manipulation tasks, including cable routing and notebook folding. Our results show that ICD outperforms or is on par with baselines that plan to exact locations rather than focusing on contact relationships, as well as baselines that directly predict actions without planning. ICD can also adapt a target contact relationship to a different environment naturally.

[1] University of Michigan, Ann Arbor
* equal contribution

II. Related work

A. 3D Representation for Object Manipulation

Prior works studies different representations for object manipulations, such as key points [11], RGB image [12], [13], point cloud [14], [15] or mesh [16], [17], [18]. Recently, Neural Descriptor Fields (NDF) [9], [19], [20] demonstrates itself as an effective implicit representation for category-level generalization. In this work, we propose a variant of NDF where spatial structure is preserved. We show that compared to explicit representations such as point cloud, the NDF better captures the soft contact relationships between object and environment.

B. Contact Reasoning for Robot Manipulation

Controlling the robot to make and break contacts purposefully has been one of the key challenges for robotics, since it involves optimizing over a hybrid system that contains both continuous (robot motion) and discrete variables (contact). One common approach [1], [2], [3], [4] is to find object motions using a sampling-based motion planner guided by a high-level search for contact modes. However, these methods are typically limited to rigid objects. Recently, learning-based methods have been introduced to detect or control contact [21], [22], [23], [24], [25] for complaint tools such as spatulas. Wi et al. [8] designs a framework for contact-rich manipulation that predicts the target contact patch over the environment conditioned on the language. However, the predicted contact patch does not specify which part of the object should make contact, and does not model a sequence of changing contacts. In this paper, we propose to use a contact-aware neural representation and a diffusion-based architecture to model future contact sequences for highly deformable objects, such as cables.

C. Diffusion Models for Robotics

Diffusion models have also been applied to robot manipulation, either as a policy class that predicts action directly from observation [13], [26], [27], [28], [29], [30], [31], or as a learned planner to generate future trajectories [32], [33], [34], [35], [10]. Although some existing diffusion-based methods have been shown to work on certain contact-rich manipulation tasks, such as planar pushing [13], dumpling making [15] or book shelving [36], our experiment suggests that they struggle with tasks that involve long-horizon reasoning of changing contacts. Similarly to us, the Subgoal Diffuser [10] generates future subgoals using a diffusion model to guide an MPC controller. However, Subgoal Diffuser represents the subgoals using locations of key points, which can be overly constrained and does not reason about the contact interaction between object and environments explicitly.

D. Long-horizon reasoning for robot manipulation

Long-horizon manipulation tasks usually contain several distinct stages and contain a lot of local optima. One way to tackle this is to plan over skill abstractions [37], [38], [39], [40], [41], [42], [43] learned with imitation learning or reinforcement learning. Another way is to decompose tasks into multiple subgoals [44], [45], [46], [47], [10], [35], which can be used to guide a low-level policy. We propose a method to generate subgoals represented by neural descriptors, which will highlight the contact relationships bwtween the objects and environment. While NOD-TAMP [48] uses a similar representation for long-horizon reasoning, it adapts a given demonstration trajectory to a new situation by optimization while we directly learn the distribution of the trajectory using a latent diffusion model. Also, NOD-TAMP cannot handle deformable objects.

III. Preliminaries

A. Problem Statement

In this paper, we consider long-horizon contact-rich manipulation problems of deformable object that involve changing contacts. We denote the robot state by $\boldsymbol{s}_t$ and the action by $\boldsymbol{a}_t$. The goal specification is represented as a pair of point clouds $(\boldsymbol{P}_{o_g}, \boldsymbol{P}_s)$, where $\boldsymbol{P}_{o_g}$ is the point cloud of the object in a goal state and $\boldsymbol{P}_s$ is the point cloud of the scene. However, the goal is not to match the shape and pose of the object exactly but to match the **contact relationship** between the object and the scene, so that the object is in contact with the scene in the correct locations. For example, in a cable routing task, the objective is to route the cable through the opening of the hook, ensuring that the cable touches the front side of the hook but not the back. It is important to note that we focus solely on the geometric aspect of the contact, without differentiating between the contact modes such as sticking or sliding contact.

This type of problem presents significant challenges due to the need for joint reasoning over both continuous motion and discrete contact switching, particularly for high-dimensional deformable objects. Additionally, long-horizon reasoning is crucial for generating effective contact-switching behavior while avoiding local minima, ensuring that the robot can progress toward the final goal without becoming stuck in suboptimal configurations.

Our objective is to learn a dense object-centric representation of contact relationship, which describes how each point of the object interacts with the environment. Next, we learn a generative model that, given the current state, scene, and goal specification, predicts a sequence of contact subgoals using the learned representation. These subgoals guide an MPC method to sequentially make and break contact and ultimately reach a goal state that conforms to the goal specification.

We assume access to an offline dataset $\mathcal{D}$, which contains N different trajectories of object point clouds and the corresponding scene point cloud $(\boldsymbol{\tau}^i, \boldsymbol{P}_s^i)$, where $\boldsymbol{\tau}^i = [P_{o_0}^i, P_{o_1}^i, \cdots, P_{o_L}^i]$. The offline dataset is collected with a scripted policy that does not guarantee task completion. We also assume access to the full point cloud for both the object and the scene, and the order of points in $\boldsymbol{P}_o$ does not change between states.

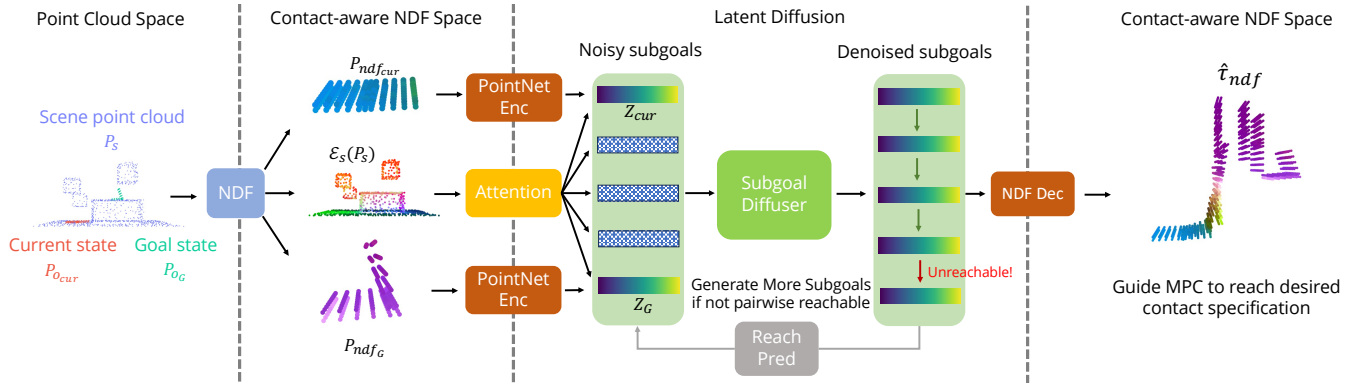

Fig. 2: System overview, with the notebook folding task as an example. First, ICD transforms the scene, current object, and goal object point cloud, into an implicit contact representation using a modified NDF model. The NDF model can be used to extract point-wise contact relationships of the object, shown by the color. Next, we project the dense NDF point clouds into low-dimensional latent vectors and utilize a latent diffusion model to generate a sequence of contact subgoals. The latent diffusion model generates subgoals recursively from coarse to fine, depending on a reachability measure. Finally, we track these predicted subgoals using a sampling-based MPC method, ensuring that the object reaches the desired contact specification.

B. Diffusion Models

Implicit Contact Diffuser is largely based on diffusion models [49], [50], which are a powerful class of generative models that frame data generation as a K-step iterative denoising procedure. To sample a noise-free output $\boldsymbol{\tau}*^0$ from a diffusion model, it starts by sampling $\boldsymbol{\tau}^K$ from a Gaussian noise distribution. Then we perform K iterations of stochastic Langevin Dynamics [51] with the update rule $\boldsymbol{\tau}^{k-1} = \alpha^k(\boldsymbol{\tau}^k - \gamma^k\epsilon_\theta(\boldsymbol{\tau}^k, k) + N(0, \sigma^2 I))$. α^k and γ^k are both hyperparameters related to the noise schedule and $N(0, \sigma^2 I))$ denotes Gaussian noise added at each iteration. ϵ_θ is parameterized by a neural network to estimate the noise that can be used to recover the original data. DDPMs [50] propose to train diffusion model using the variational lower-bound on $\log p_\theta(\boldsymbol{\tau})$: $\mathcal{L}_{DDPM}(\theta) = ||\epsilon^k - \epsilon_\theta(\boldsymbol{\tau}^k, k)||^2$.

IV. Method

In this section, we introduce *Implicit Contact Diffuser*, a method designed to capture and reason about contact switching in long-horizon deformable object manipulation. In Section IV-A, we discuss how to represent the object-environment contact relationships of deformable objects using an implicit neural representation. In Section IV-B, we describe how to train a latent point cloud diffusion model to predict the contact sequence. Finally, in Section IV-C, we discuss how to follow the predicted contact sequence using a sampling-based MPC planner.

A. Contact-aware Neural Descriptor Field

Finding a suitable contact representation that facilitates planning is a challenging problem. If we naively represent contact with a binary discrete representation, planning over the contact space can quickly become combinatorially expensive, which is one of the reasons why prior methods [52], [4] struggle with deformable objects. Our key insight is that we can capture the soft object-environment contact relationships using a continuous implicit neural representation. We build upon Neural Descriptor Fields (NDF) [9], [19], [20] to develop a contact-aware neural representation for deformable objects, utilizing a scene NDF. Given a scene point cloud $\boldsymbol{P}_s$, we learn a function f to map a 3D coordinate $x \in \mathbb{R}^3$ to a latent neural descriptor in $\mathbb{R}^d$:

$$f(\boldsymbol{x}|\boldsymbol{P}_s) = f(\boldsymbol{x}|\mathcal{E}_s(\boldsymbol{P}_s)) \tag{1}$$

where $\mathcal{E}_s(\boldsymbol{P}_s)$ is a PointNet [53] model. Given an object point cloud $\boldsymbol{P}_o$, the state of the object can be described as the concatenation of all point descriptors:

$$\boldsymbol{P}_{ndf} = \phi_{NDF}(\boldsymbol{P}_o|\boldsymbol{P}_s) = \bigoplus_{\boldsymbol{x}_i \in \boldsymbol{P}_o} f(\boldsymbol{x}_i|\boldsymbol{P}_s) \tag{2}$$

Since the function f is trained to predict the geometric features of the scene, the NDF point cloud $\boldsymbol{P}_{ndf} \in \mathbb{R}^{N \times d}$ can be interpreted as an encoding of point-wise geometric relations with the scene for every point on the object.

We make several key design choices to adapt NDF, ensuring it better suits the tasks we are dealing with. Similar to Simeonov et al. [9], we train $f(\boldsymbol{x}|\boldsymbol{P}_s)$ using occupancy prediction. Additionally, we incorporate an auxiliary loss on the gradient direction of the signed distance function (SDF): $\mathcal{J}_{grad} = (\nabla SDF(x) - \hat{\nabla} SDF(x))^2$, where $\nabla SDF(x)$ and $\hat{\nabla} SDF(x))$ refer to ground-truth and predicted gradients of the SDF. This helps the descriptors encode not only whether a point is in contact (occupied), but also how to make contact for points that are not yet in contact.

NDF adopts a $\boldsymbol{SE}(3)$-invariant neural network architecture, Vector Neuron [54], to enhance the generalizability of the descriptors. While the descriptors remain unchanged when a transformation $T \in \boldsymbol{SE}(3)$ is applied to the object and the scene simultaneously, this can sometimes lead to unrealistic outcomes. For example, the object will have similar NDF features whether it contacts the floor or the ceiling. To mitigate this, we modify the Vector Neuron to be invariant only to the rotations along the direction of gravity (as gravity plays a large part in determining the configuration of a deformable object) , which we define as $\boldsymbol{SE}(3)^z$. Specifically, we add a small constant value to the z-axis of the point features, ensuring that rotations not aligned with the z-axis produce distinct latent features.

The original NDF model [9] encodes the entire point cloud into a single global feature vector by averaging over

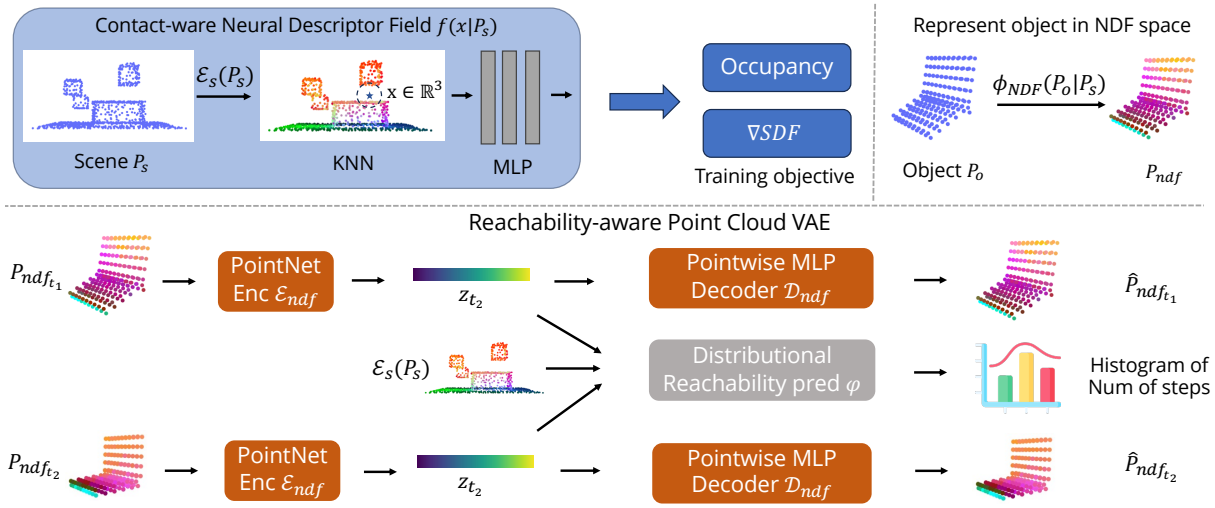

Fig. 3: shown in the the upper figure, the NDF model is trained to encode local geometries of the scene by predicting occupancy and gradient direction of the Signed Distance Function (SDF) of the scene. Given an object point cloud $\boldsymbol{P}_o$, such as that of a notebook, we transform it into a contact-aware latent representation $\boldsymbol{P}_{ndf}$. In the bottom figure, we show how the reachability-aware point cloud VAE is trained. In additional to the regular reconstruction and KL divergence loss, we introduce a distributional reachability prediction loss to encourage temporal consistency in the latent space. The reachability predictor is also used in the latent diffusion model to decide the number of subgoals required for the tasks, as shown in Fig. 2.

$\mathcal{E}_s(\boldsymbol{P}_s)$. In contrast, we aggregate the local features of nearby contact candidates for each query point using K-nearest neighbors (Fig. 2) based on the intuition that the object is more likely to make contact with spatially closer points. Our experiments indicate that incorporating these local NDF features is important for improving task performance.

B. Implicit Contact Diffuser

In the previous section, we describe a dense contact-aware neural representation for deformable objects. Now we will use this representation to tackle long-horizon contact-rich manipulation problems with contact switching. We introduce **Implicit Contact Diffuser**, a diffusion-based architecture that generates a sequence of subgoals $\boldsymbol{\tau}_{ndf} = [\boldsymbol{P}_{ndf_0}, \boldsymbol{P}_{ndf_1}, \boldsymbol{P}_{ndf_2}, \ldots, \boldsymbol{P}_{ndf_M}]$, represented as NDF point clouds $\boldsymbol{P}_{ndf} \in \mathbb{R}^{N\times d}$.

While diffusion models have been applied to point cloud generation, prior works [55], [56] only generate individual point clouds $\boldsymbol{P} \in \mathbb{R}^{N\times 3}$. In our case, in order to capture contact switching, we need to generate a sequence of coherent latent point clouds consisting of high-dimensional point features.

To tackle this sequential point cloud generation problem, we propose using Latent Diffusion Models (LDM) [57]. We begin by training a Variational Autoencoder (VAE) [58] to project the high-dimensional point cloud $\boldsymbol{P}_{ndf}$ into low-dimensional vectors. Next, we train a hierarchical diffusion model to recursively generate subgoals from coarse to fine, following Huang et al. [10].

Reachability-aware Point Cloud VAE. The VAE comprises three components: a PointNet++ encoder $\mathcal{E}_{ndf}(\boldsymbol{z}_t|\boldsymbol{P}_{ndf_t})$ [59], a point-wise MLP decoder $\mathcal{D}_{ndf}(\hat{\boldsymbol{P}}_{ndf_t}|\boldsymbol{P}_o^{canon}, \boldsymbol{z}_t)$, and a distributional reachability prediction MLP $\varphi(\hat{\boldsymbol{r}}|\boldsymbol{z}_{t_1}, \boldsymbol{z}_{t_2}, \mathcal{E}_s(\boldsymbol{P}_s))$, as visualized in Fig. 3. The encoder $\mathcal{E}_{ndf}$ compresses the NDF point cloud $\boldsymbol{P}_{ndf_t}$ into a latent vector $\boldsymbol{z}_t$. The pointwise MLP decoder $\mathcal{D}_{ndf}$ is adapted from Luo et al. [55]. Given $\boldsymbol{z}_t$ and the canonical object point cloud $\boldsymbol{P}_o^c$, an implicit decoder $\mathcal{D}_{ndf}$ reconstructs the NDF point cloud from the latent vector. The query coordinates $\boldsymbol{P}_o^{canon}$ are predefined, i.e., a straight rope or a magazine that is laid flat.

The VAE is trained by three different losses:

$$\mathcal{L}_{vae} = \lambda_1 \mathcal{L}_{recon}(\boldsymbol{P}_{ndf}, \hat{\boldsymbol{P}}_{ndf}) \tag{3}$$
$$+ \lambda_2 D_{\mathrm{KL}}(\mathcal{E}_{ndf}(\boldsymbol{z}_t|\boldsymbol{P}_{ndf_t}), \boldsymbol{N}(z)) \tag{4}$$
$$+ \lambda_3 \mathcal{L}_{Reach}(\boldsymbol{r}, \varphi(\boldsymbol{r}|\boldsymbol{z}_{t_1}, \boldsymbol{z}_{t_2}, \mathcal{E}_s(\boldsymbol{P}_s))) \tag{5}$$

In addition to the regular reconstruction loss and KL regularization loss, we introduce a reachability loss $\mathcal{L}_{reach}$ to encourage temporal consistency in the learned latent space.

During training, we sample pairs of states in the same trajectory using the discounted state occupancy measure (lower probability for states that take more steps to reach) in line with previous work [60], [61]. For a pair of NDF point clouds $(\boldsymbol{P}_{ndf_{t_1}}, \boldsymbol{P}_{ndf_{t_2}})$, we define reachability as the minimum number of steps to travel between them. Following Subgoal Diffuser [10], we discretize the reachability into K bins and frame the reachability prediction problem as a classification problem and train an MLP $\varphi(\boldsymbol{r}|\boldsymbol{z}_{t_1}, \boldsymbol{z}_{t_2}, \mathcal{E}_s(\boldsymbol{P}_s))$ with cross-entropy loss. Since we do not assume that the training data are high-quality demonstrations, and there might exist multiple paths of different lengths to travel between two states, $\varphi(\boldsymbol{r}|\boldsymbol{z}_{t_1}, \boldsymbol{z}_{t_2}, \mathcal{E}_s(\boldsymbol{P}_s)$ will capture the distribution of reachability between two states. During test time, we use "softmin" to estimate shortest distance (highest reachability), which is used to determine the number of subgoals for the latent diffusion model.

Latent Point Cloud Diffusion Model The objective of the latent diffusion model is to generate a sqeuence of NDF subgoals $\boldsymbol{\tau}_{ndf}$, given current state, goal specification, and the scene. With the point cloud VAE described above, the diffusion model only needs to model the distribution of the condensed latent vectors, denoted as $p(\boldsymbol{\tau}_z|\boldsymbol{z}_{cur}, \boldsymbol{z}_{goal}, \mathcal{E}_s(\boldsymbol{P}_s))$.

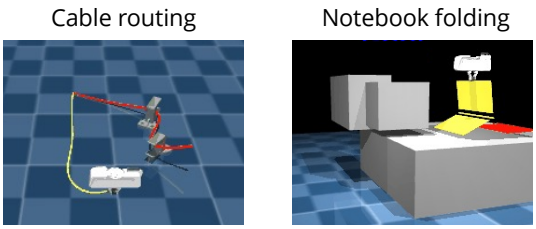

Fig. 4: We evaluate our methods on two long-horizon contact-rich tasks in simulation: cable routing and notebook folding. Goals are visualized in red.

Following Subgoal Diffuser [10], we generate subgoal sequences recursively in a coarse to fine manner. Starting from $\boldsymbol{\tau}_z^0 = [\boldsymbol{z}_{cur}, \boldsymbol{z}_{goal}]$, in each iteration, the number of subgoals in $\boldsymbol{\tau}_z^{l+1}$ increases by $|\boldsymbol{\tau}_z^{l+1}| = |\boldsymbol{\tau}_z^l| \times 2 - 1$. Instead of generating from scratch, the latent diffusion model predicts the next level of subgoals $\boldsymbol{\tau}_z^{l+1}$ conditioned on the previous ones $\boldsymbol{\tau}_z^l$. Hence, the latent diffusion model can be written as $p(\boldsymbol{\tau}_z^{l+1}|\boldsymbol{\tau}_z^l, \mathcal{E}_s(\boldsymbol{P}_s))$.

C. MPPI with Neural Contact Subgoals

Every T steps, Neural Contact Diffuser generates a sequence of contact subgoals $\hat{\boldsymbol{\tau}}_{ndf}$. We use a sampling-based MPC method, Model Predictive Path Integral (MPPI) [62], to plan a sequence of robot actions to track the subgoals. We define robot actions $\boldsymbol{a} \in \mathbb{R}^3$ as the delta translation of the end effector. At each step, MPPI samples K action sequences of length H, where H is the planning horizon.

The sampled actions are evaluated by rolling out in the MuJoCo [63] simulator with the following cost:

$$\mathcal{J}_{MPPI} = \sum_{t=0}^{H-1} \left(\min_{\hat{\boldsymbol{P}}_{ndf_i} \in \hat{\boldsymbol{\tau}}_{ndf}} (\boldsymbol{P}_{ndf_t} - \hat{\boldsymbol{P}}_{ndf_i})^2 + \lambda_{col} \max(-SDF(\mathbf{r}_t), 0) \right)$$

$\hat{\boldsymbol{\tau}}_{ndf}$ is the desired NDF subgoals predicted by the diffusion model. The rollouts from the simulator are transformed to NDF space using ϕ_{ndf}, which we denote as $\boldsymbol{P}_{ndf_t}$. The first cost term is the Euclidean distance to the closest NDF subgoal. A subgoal will be removed from the goal chain $\hat{\boldsymbol{\tau}}_{ndf}$ once the current state is within predefined distance threshold. The second cost is to prevent the robot from colliding with the environment, represented as the scene SDF. The robot geometry is approximated by a set of spheres as in [64]. By minimizing $\mathcal{J}_{MPPI}$, MPPI generates robot actions that manipulate the object to achieve the desired contact relationships described by $\hat{\boldsymbol{\tau}}_{ndf}$.

V. Experiments

Our experiments aim to show that 1) the scene NDF is a good representation for capturing contact relationships and 2) *Implicit Contact Diffuser* is capable of long-horizon contact reasoning and generating contact sequences to guide an MPC controller to reach the desired contact relationship. We also demonstrate our method on a physical robot and the videos can be found on our website.

Method	Cable Routing Success ↑	Cable Routing Complete ↑	Notebook Success ↑
Implicit Contact Diffuser	**90**	**95**	95
Subgoal Diffuser [10]	65	80	**100**
Diffusion Policy [13]	30	40	70
3D Diffusion Policy [15]	15	40	5
PC-MPPI	25	55	50
NDF-MPPI	55	70	10
Global NDF	50	75	75

TABLE I: We evaluate every method on 10 test cases for 2 seeds (20 runs in total) and report the success rate. For the cable routing task, success is defined as the cable being routed through both fixtures. Additionally, we report the "complete rate," which represents the percentage of fixtures successfully routed by the cable.

A. Simulation Experiments

1) Tasks: We evaluate our method on two long-horizon manipulation tasks that involve changing contact (Fig. 4).
Cable routing. The goal is to route the rope through two randomly placed fixtures on a table. One end of the cable is fixed and the other is grasped by a floating gripper. The task is considered successful if the rope is routed through *both* fixtures. We also consider the "complete rate"—the percentage of successfully routed individual fixtures. This task is challenging due to: 1) The high-dimensional state space and complex rope dynamics; 2) The need for precise control of a deformable object (ensuring the cable stays inside the first fixture when routing the second); and 3) Long-horizon reasoning (to avoid local minima).
Notebook folding. The goal is to move notebook from the ground to the table, lay it on the table, and fold it. Each stage can be characterized by distinct contact mode. The positions and sizes of the tables and obstacles are randomized. The floating gripper grasps the notebook in the middle of its edge. The task is considered success if the pairwise distance to the goal object point cloud is below a threshold.

For both tasks, the goal specification is provided as point cloud. We evaluated each method on 10 test cases for 2 seeds. The environments are built in the MuJoCo [63] simulator.

B. Implementation Details

We collected 5,000 trajectories of length 200 for cable routing and 10,000 trajectories of length 100 for notebook using scripted policies. The scene point clouds contains 1000 points and the object point clouds contain around 200 points. The NDF model is trained using equal weights for occupancy prediction and SDF gradient prediction. For VAE, the loss weights for reconstruction, KL-divergence and reachability are 1, $1e^{-6}$ and $1e^{-5}$. For the diffusion model, we follow the training scheme of DDPMs [50] with 100 diffusion steps. MPPI samples 80 trajectories with a horizon of 10. We use a noise scale of 0.001 for action sampling and a temperature of 0.005 for cost computation.

1) Baselines: 1) **MPPI**: We evaluate MPPI without the subgoals for guidance. We explore two different object representations for cost computation, referred to as PC-MPPI and NDF-MPPI; In PC-MPPI, the cost is computed as the distance in point cloud space, while NDF-MPPI computes cost in NDF space. 2) **Subgoal Diffuser** [10]: A modified

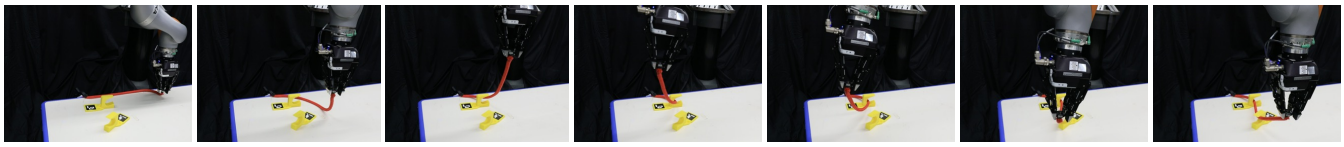

Fig. 5: Physical demonstration with a 7-DoF Kuka arm on cable routing with 3 different cables for a total of 10 runs. Videos are available on our website.

version of Subgoal Diffuser that predicts a sequence of object point clouds using the same latent diffusion model as our method. The predicted subgoals are also tracked by the same MPPI planner. 3) **Diffusion Policy** [13]: We adapt the official implementation to make the policy goal-conditioned. This version uses a keypoints-based object representation, while the scene information is encoded using the PointNet encoder from the NDF model. 4) **3D Diffusion Policy** [15]: This baseline takes as input the point clouds of the object and the scene, and directly predicts the actions for the robot to execute. 5) **Global NDF**. Instead of retrieving local features using KNN, this baseline follows the original NDF [9] to compute a global feature vector for the entire scene.

2) Results: The quantitative results can be found in Table I, and here we discuss our main findings.

Subgoal generation is critical for long-horizon reasoning. We observe that the subgoal-based methods outperform both model-free methods that do not have explicit global reasoning (diffusion policy and 3D diffusion policy) and MPC methods that plan directly to the goal (PC-MPPI and NDF-MPPI). This result shows the importance of reasoning over the intermediate contact sequences explicitly.

Contact-aware state representation is critical for long-horizon contact reasoning. We observe that while subgoal diffuser performs well on notebook folding, its success rate drops significantly on cable routing. Upon inspection, we found that the primary failure mode is that the point cloud-based subgoal tends to lead the MPC to local minima since it does not capture the contact relationship. For instance, it may lead to a state where the cable is spatially close to the goal configuration but is positioned incorrectly, such as being on the wrong side of the fixtures. In contrast, our method leverages NDF to capture the geometric relationships between the rope and the fixtures. The NDF-based subgoals provide better guidance for the MPC to reach desired contact relationships. We also observe that the modified locally-conditioned NDF better captures the contact relationships compared to the global NDF.

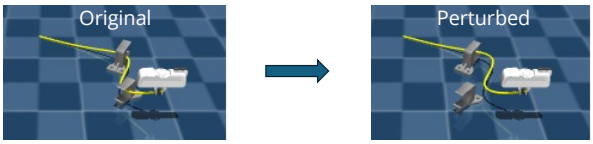

Fig. 6: Examples for adaptation test.

3) Adaptation test: Our previous experiment assumes the goal specification is provided for each task. However, obtaining the exact goal specification can be challenging in practice. It would be beneficial if we could reuse a previous goal specification $(\boldsymbol{P}_{o_g}, P_s)$ in a different environment P_s'. To explore this, we conduct an additional experiment, called the adaptation test. In this experiment, we randomly perturb the positions and orientations of the fixtures. As shown in Fig. 6, planning to the original goal using a point cloud-based representation is likely to fail, due to the change of fixture locations. Our key insight is that, even when the scene is altered, the desired contact relationships between the object and the scene remain consistent.

Method	Success rate ↑	Complete Rate ↑
Ours	90	90
Subgoal Diffuser [10]	25	47.5

TABLE II: Results of adaptation test on cable routing

As illustrated in Table II, planning in the NDF space allows our method to successfully route the cable in the perturbed scene, whereas the baseline, which relies on precise positional subgoals, struggles to adapt to the changes in the environment.

C. Physical Demonstration

We deployed *Implicit Contact Diffuser* on a 7 DoF Kuka LBR iiwa arm for a real-world version of the cable-routing task. We used a Zivid 2 camera and CDCPD [65] to track the point cloud of the cable. We tested on 3 different cables, one soft, thin charging cable, one stiff ethernet cable, and a thick rope, for a total of 10 trials. While our method succeeds 9 / 10 runs, challenges such as perception errors from the tracker and the limited workspace of the robot affected the overall reliability of the method. Please see our website for the videos.

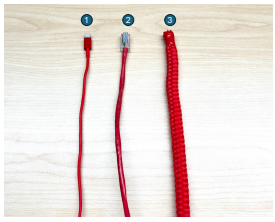

Fig. 7: Cables used in the physical experiments.

VI. Conclusion and Future Work

We introduce a novel framework that enables the robot to reason about changing contacts between environments and objects. Our approach captures object-environment interactions using a smooth, continuous implicit representation. We then use a latent point cloud diffusion model to generate future contact subgoals using this representation. When integrated with an MPC method, the robot can intelligently initiate and break contacts to manipulate the object to satisfy a desired contact specification. However, the method has limitations: 1) It assumes access to full object and environment point clouds, which are often unavailable in real-world scenarios. Shape completion methods [66], [17] could be applied to address this issue. 2) While replanning helps address model and perception errors, these errors are not considered during subgoal generation—a gap that could be addressed with online learning through interaction.

REFERENCES

[1] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided sampling-based planning for quasistatic dexterous manipulation in 2d," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6520–6526.

[2] ——, "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2730–2736.

[3] X. Cheng, S. Patil, Z. Temel, O. Kroemer, and M. T. Mason, "Enhancing dexterity in robotic manipulation via hierarchical contact exploration," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 390–397, 2023.

[4] B. Aceituno and A. Rodriguez, "A hierarchical framework for long horizon planning of object-contact trajectories," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 189–196.

[5] J. Park, J. Haan, and F. C. Park, "Convex optimization algorithms for active balancing of humanoid robots," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 817–822, 2007.

[6] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, 2012, pp. 137–144.

[7] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[8] Y. Wi, M. Van der Merwe, P. Florence, A. Zeng, and N. Fazeli, "Calamari: Contact-aware and language conditioned spatial action mapping for contact-rich manipulation," in *7th Annual Conference on Robot Learning*, 2023.

[9] 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)*. IEEE, 2022, pp. 6394–6400.

[10] Z. Huang, Y. Lin, F. Yang, and D. Berenson, "Subgoal diffuser: Coarse-to-fine subgoal generation to guide model predictive control for robot manipulation," *arXiv preprint arXiv:2403.13085*, 2024.

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

[12] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for physical sequential fabric manipulation," *Autonomous Robots*, vol. 46, no. 1, pp. 175–199, 2022.

[13] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[14] J. Yang, C. Deng, J. Wu, R. Antonova, L. Guibas, and J. Bohg, "Equivact: Sim (3)-equivariant visuomotor policies beyond rigid object manipulation," *arXiv preprint arXiv:2310.16050*, 2023.

[15] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024.

[16] X. Lin, Y. Wang, Z. Huang, and D. Held, "Learning visible connectivity dynamics for cloth smoothing," in *Conference on Robot Learning*. PMLR, 2022, pp. 256–266.

[17] Z. Huang, X. Lin, and D. Held, "Mesh-based dynamics with occlusion reasoning for cloth manipulation," *arXiv preprint arXiv:2206.02881*, 2022.

[18] ——, "Self-supervised cloth reconstruction via action-conditioned cloth tracking," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7111–7118.

[19] 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.

[20] E. Chun, Y. Du, A. Simeonov, T. Lozano-Perez, and L. Kaelbling, "Local neural descriptor fields: Locally conditioned object representations for manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1830–1836.

[21] Y. Wi, P. Florence, A. Zeng, and N. Fazeli, "Virdo: Visio-tactile implicit representations of deformable objects," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3583–3590.

[22] Y. Wi, A. Zeng, P. Florence, and N. Fazeli, "Virdo++: Real-world, visuo-tactile dynamics and perception of deformable objects," *arXiv preprint arXiv:2210.03701*, 2022.

[23] M. Van der Merwe, D. Berenson, and N. Fazeli, "Learning the dynamics of compliant tool-environment interaction for visuo-tactile contact servoing," in *Conference on Robot Learning*. PMLR, 2023, pp. 2052–2061.

[24] M. Van der Merwe, Y. Wi, D. Berenson, and N. Fazeli, "Integrated object deformation and contact patch estimation from visuo-tactile feedback," *arXiv preprint arXiv:2305.14470*, 2023.

[25] C. Higuera, S. Dong, B. Boots, and M. Mukadam, "Neural contact fields: Tracking extrinsic contact with tactile sensing," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 576–12 582.

[26] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki, "Act3d: Infinite resolution action detection transformer for robotic manipulation," *arXiv preprint arXiv:2306.17817*, 2023.

[27] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *Conference on Robot Learning*. PMLR, 2023, pp. 3766–3777.

[28] L. Chen, S. Bahl, and D. Pathak, "Playfusion: Skill acquisition via diffusion from language-annotated play," in *Conference on Robot Learning*. PMLR, 2023, pp. 2012–2029.

[29] L. X. Shi, A. Sharma, T. Z. Zhao, and C. Finn, "Waypoint-based imitation learning for robotic manipulation," *arXiv preprint arXiv:2307.14326*, 2023.

[30] U. A. Mishra, S. Xue, Y. Chen, and D. Xu, "Generative skill chaining: Long-horizon skill planning with diffusion models," in *Conference on Robot Learning*. PMLR, 2023, pp. 2905–2925.

[31] P. M. Scheikl, N. Schreiber, C. Haas, N. Freymuth, G. Neumann, R. Lioutikov, and F. Mathis-Ullrich, "Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects," *IEEE Robotics and Automation Letters*, 2024.

[32] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," *arXiv preprint arXiv:2205.09991*, 2022.

[33] I. Kapelyukh, V. Vosylius, and E. Johns, "Dall-e-bot: Introducing web-scale diffusion models to robotics," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3956–3963, 2023.

[34] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision-making?" *arXiv preprint arXiv:2211.15657*, 2022.

[35] W. Li, X. Wang, B. Jin, and H. Zha, "Hierarchical diffusion for offline decision making," 2023.

[36] A. Simeonov, A. Goyal, L. Manuelli, L. Yen-Chen, A. Sarmiento, A. Rodriguez, P. Agrawal, and D. Fox, "Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement," *arXiv preprint arXiv:2307.04751*, 2023.

[37] B. Eysenbach, R. R. Salakhutdinov, and S. Levine, "Search on the replay buffer: Bridging planning and reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[38] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu, "Robocook: Long-horizon elasto-plastic object manipulation with diverse tools," *arXiv preprint arXiv:2306.14447*, 2023.

[39] X. Lin, Z. Huang, Y. Li, J. B. Tenenbaum, D. Held, and C. Gan, "Diffskill: Skill abstraction from differentiable physics for deformable object manipulations with tools," *arXiv preprint arXiv:2203.17275*, 2022.

[40] X. Lin, C. Qi, Y. Zhang, Z. Huang, K. Fragkiadaki, Y. Li, C. Gan, and D. Held, "Planning with spatial-temporal abstraction from point clouds for deformable object manipulation," *arXiv preprint arXiv:2210.15751*, 2022.

[41] S. Cheng and D. Xu, "League: Guided skill learning and abstraction for long-horizon manipulation," *IEEE Robotics and Automation Letters*, 2023.

[42] S. Jin, W. Lian, C. Wang, M. Tomizuka, and S. Schaal, "Robotic cable routing with spatial representation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5687–5694, 2022.

[43] J. Luo, C. Xu, X. Geng, G. Feng, K. Fang, L. Tan, S. Schaal, and S. Levine, "Multi-stage cable routing through hierarchical imitation learning," *IEEE Transactions on Robotics*, 2024.

[44] T. Jurgenson, O. Avner, E. Groshev, and A. Tamar, "Sub-goal trees

a framework for goal-based reinforcement learning," in *International conference on machine learning*. PMLR, 2020, pp. 5020–5030.

[45] S. Nair and C. Finn, "Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation," *arXiv preprint arXiv:1909.05829*, 2019.

[46] K. Fang, P. Yin, A. Nair, and S. Levine, "Planning to practice: Efficient online fine-tuning by composing goals in latent space," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4076–4083.

[47] S. Nasiriany, V. Pong, S. Lin, and S. Levine, "Planning with goal-conditioned policies," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[48] S. Cheng, C. Garrett, A. Mandlekar, and D. Xu, "Nod-tamp: Multi-step manipulation planning with neural object descriptors," *arXiv preprint arXiv:2311.01530*, 2023.

[49] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

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

[51] 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)*. Citeseer, 2011, pp. 681–688.

[52] Z. Xian, N. Gkanatsios, T. Gervet, T.-W. Ke, and K. Fragkiadaki, "Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation," in *7th Annual Conference on Robot Learning*, 2023.

[53] 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.

[54] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 200–12 209.

[55] 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*, 2021, pp. 2837–2845.

[56] A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, K. Kreis *et al.*, "Lion: Latent point diffusion models for 3d shape generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10 021–10 039, 2022.

[57] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[58] D. P. Kingma, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

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

[60] B. Eysenbach, V. Myers, R. Salakhutdinov, and S. Levine, "Inference via interpolation: Contrastive representations provably enable planning and inference," *arXiv preprint arXiv:2403.04082*, 2024.

[61] B. Eysenbach, T. Zhang, S. Levine, and R. R. Salakhutdinov, "Contrastive learning as goal-conditioned reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 603–35 620, 2022.

[62] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1433–1440.

[63] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[64] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos *et al.*, "Curobo: Parallelized collision-free robot motion generation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8112–8119.

[65] Y. Wang, D. McConachie, and D. Berenson, "Tracking partially-occluded deformable objects while enforcing geometric constraints," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 199–14 205.

[66] C. Chi and S. Song, "Garmentnets: Category-level pose estimation for garments via canonical space shape completion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3324–3333.