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 through contact is central to many robotic tasks, such as manipulation and locomotion. However, controlling hybrid systems is challenging due to the interplay between discrete contact events and continuous motion. In cable routing, for example, the robot must generate smooth motions to initiate and maintain contact between the cable and fixtures (Fig. 1). A break in contact may cause failure, and the robot must adapt to disturbances to ensure 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.

We propose a learning-based model predictive control (MPC) framework for contact-rich tasks. Central to our approach is a latent diffusion model that generates future contact sequences as subgoals, guiding an MPC controller to produce motions that realize the desired contact relations. A key question, however, is determining the best representation for these contact relationships.

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.

To address these limitations, we propose to encode contact relationships using a modified version of Neural Descriptor Fields (NDF) [8]. 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.

* equal contribution

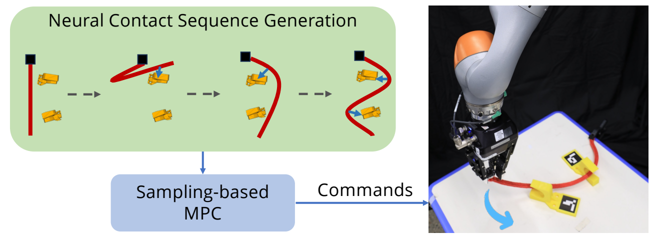

Fig. 1: By predicting future contact sequences using a latent diffusion model, we enable long-horizon contact-rich deformable object manipulation such as cable routing using a sampling-based MPC controller.

To capture the contact switching required to reach a goal, we train a latent diffusion model to predict the contact sequence represented by neural descriptors. 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 long-horizon, contact-rich tasks including cable routing and notebook folding. ICD consistently matches or outperforms baselines that plan to fixed positions or directly predict actions, and naturally adapts target contact relations across environments.

II. Preliminaries

A. Problem Statement

In this paper, we consider long-horizon contact-rich manipulation problems that involve changing contacts. The goal is specified by a pair of point clouds $(\boldsymbol{P}_{o_g}, \boldsymbol{P}_s)$, where $\boldsymbol{P}_{o_g}$ and $\boldsymbol{P}_s$ represent the object in its goal state and the scene respectively.

Rather than matching the exact goal shape or pose of the object, our objective is to match the **contact relationship** between the object and the scene, so that the object is in contact with the scene in the appropriate locations. For example, in cable routing, the objective is to route the cable through the opening of the hook, the cable must pass through

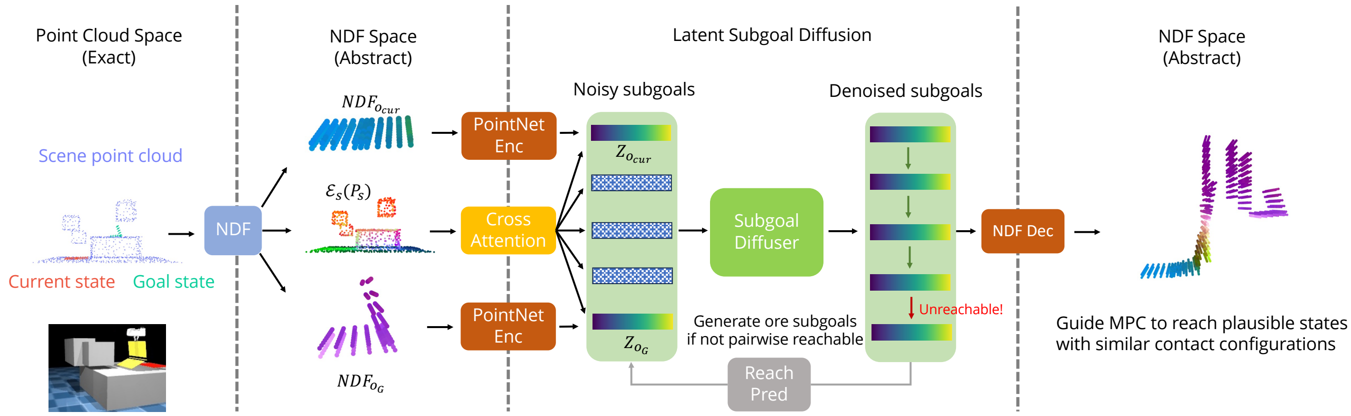

Fig. 2: System overview, with the notebook folding task as an example. First, ICD transforms the scene, current object, and goal object point cloud, into an implicit contact representation using a modified NDF model. The NDF model can be used to extract point-wise contact relationships of the object, 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.

a hook such that it touches the front but not the back. This type of problem presents significant challenges due to the need for joint reasoning over both continuous motion and discrete contact switching.

III. Method

We introduce *Implicit Contact Diffuser*, a method that enables the joint reasoning of discrete change of contact and continous motion in long-horizon manipulation problems. *Implicit Contact Diffuser* captures the object-environment contact relationship using a dense neural contact representation. Then, *Implicit Contact Diffuser* leverage a latent diffusion model to predict the future contact sequence tracked by a sampling-based MPC.

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 [9], [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) [8], [10], [11] 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 [12] 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. [8], 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 uses a $\boldsymbol{SE}(3)$-invariant Vector Neuron architecture [13] to improve descriptor generalization. However, full $\boldsymbol{SE}(3)$ invariance can produce unrealistic results—for example, treating contact with the floor and ceiling as equivalent. To address this, we constrain invariance to gravity-aligned rotations ($\boldsymbol{SE}(3)^z$) by adding a small constant to the z-axis of point features, making descriptors sensitive to orientation relative to gravity.

The original NDF model [8] encodes the entire point cloud into a single global feature vector by averaging over $\mathcal{E}_s(\boldsymbol{P}_s)$. In contrast, we aggregate the local features of nearby contact candidates for each query point using K-nearest neighbors (Fig. 3) 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

Building on the dense contact-aware representation, we introduce **Implicit Contact Diffuser**, a diffusion-based model that generates a sequence of subgoals $\boldsymbol{\tau}_{ndf} = [\boldsymbol{P}_{ndf_0}, \ldots, \boldsymbol{P}_{ndf_M}]$, where each $\boldsymbol{P}_{ndf} \in \mathbb{R}^{N\times d}$ is an NDF point cloud representing contact features.

Unlike prior works [14], [15] which only generate individual point cloud $\boldsymbol{P} \in \mathbb{R}^{N\times 3}$ with diffusion model, we model sequential latent point clouds to capture contact switching.

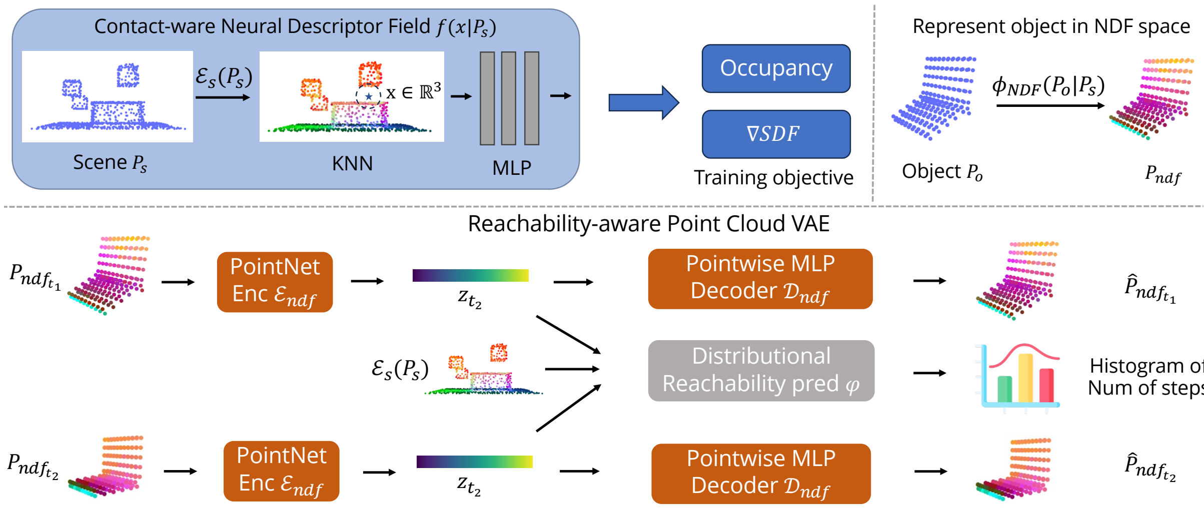

Fig. 3: As shown in the the upper figure, the NDF model is trained to encode local geometries of the scene by predicting occupancy and gradient direction of the Signed Distance Function (SDF) of the scene. Given an object point cloud $\boldsymbol{P}_o$, such as that of a notebook, we transform it into a contact-aware latent representation $\boldsymbol{P}_{ndf}$. In the bottom figure, we show how the reachability-aware point cloud VAE is trained. In addition 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.

To address this, we adopt Latent Diffusion Models (LDM) [16]. First, We train a Variational Autoencoder (VAE) [17] 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. [18].

Reachability-aware Point Cloud VAE. The VAE comprises three components: a PointNet++ encoder $\mathcal{E}_{ndf}(\boldsymbol{z}_t|\boldsymbol{P}_{ndf_t})$ [19], 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. [14]. 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 farther apart), in line with previous work [20], [21]. Given 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 [18], we discretize the reachability into R bins and frame the reachability prediction problem as a classification problem. An MLP $\varphi(\boldsymbol{r}|\boldsymbol{z}_{t_1}, \boldsymbol{z}_{t_2}, \mathcal{E}_s(\boldsymbol{P}_s))$ with cross-entropy loss.

Latent Point Cloud Diffusion Model Given current state, goal specification, and the scene, the objective of the latent diffusion model is to generate a sqeuence of NDF subgoals $\boldsymbol{\tau}_{ndf}$, . With the point cloud VAE described above, the diffusion model only needs to model the distribution of the condensed latent vectors, denoted as $p(\boldsymbol{\tau}_z|\boldsymbol{z}_{cur}, \boldsymbol{z}_{goal}, \mathcal{E}_s(\boldsymbol{P}_s))$. The diffusion model reasons about the contact interaction between predicted subgoals and the scene using cross-attention. Following Subgoal Diffuser [18], 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 Implicit Contact Subgoals

We use Model Predictive Path Integral (MPPI)[22] to plan robot actions that track contact subgoals. Actions are evaluated via rollouts in MuJoCo[23], then transformed into NDF space using ϕ_{ndf}. The cost $\mathcal{J}MPPI$ is computed as the sum of Euclidean distance to the ndf subgoals and a collision cost. By minimizing $\mathcal{J}MPPI$, the planner generates actions that achieve the desired contact relationships defined by $\hat{\boldsymbol{\tau}}ndf$.

IV. Experiments

A. Simulation Experiments

1) Tasks: We evaluate our method on two long-horizon manipulation tasks that involve changing contact (Fig. 5).

Cable routing. The goal is to route a rope through two randomly placed fixtures. One end is fixed; the other is held by a floating gripper. Success requires passing through both fixtures. We also report the "complete rate"—the percentage of individual fixtures routed correctly. This task is challenging due to: (1) high-dimensional rope dynamics, (2) precise

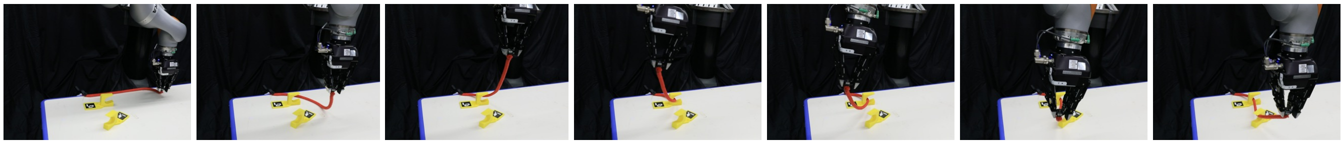

Fig. 4: Physical demonstration with a 7-DoF Kuka arm on cable routing with 3 different cables for a total of 10 runs. Videos are available on our website.

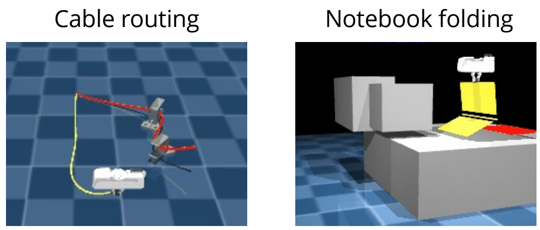

Fig. 5: We evaluate our methods on two long-horizon contact-rich tasks in simulation: cable routing and notebook folding. Goals are visualized in red.

Method	Cable Routing Success ↑	Cable Routing Complete ↑	Notebook Success ↑
Implicit Contact Diffuser	**90**	**95**	95
Subgoal Diffuser [18]	65	80	**100**
Diffusion Policy [24]	30	40	70
3D Diffusion Policy [25]	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.

control to maintain contacts, and (3) long-horizon planning to avoid local minima.

Notebook folding. The task involves lifting a notebook from the floor onto a table, laying it flat, and folding it—each phase involving distinct contact modes. Table positions and obstacle layouts are randomized. The notebook is grasped at its edge center by a floating gripper. Success is defined by reaching a goal point cloud within a set distance threshold.

For both tasks, the goal specification is provided as a point cloud. We evaluated each method on 10 test cases for 2 seeds. The environments are built in the MuJoCo [23] 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 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 [26] with 100 diffusion steps.

1) Baselines: 1) **MPPI**: 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** [18]: A modified 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** [24]: 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** [25]: 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 [8] 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 long-horizon reasoning (diffusion policy and 3D diffusion policy) and MPC methods that plan directly to the goal (PC-MPPI and NDF-MPPI).

Contact-aware state representation is critical for long-horizon contact reasoning. 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 example, the cable may appear close to the goal but be on the wrong side of a fixture. In contrast, our NDF-based subgoals encode geometric contact relationships, offering more reliable guidance for achieving correct contact configurations.

C. Physical Demonstration

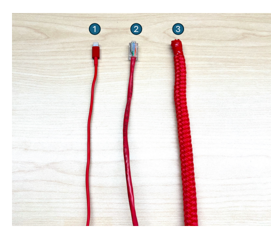

Fig. 6: Cables used in the physical experiments.

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 [27] 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.

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] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.