

Diff-LfD: Contact-aware Model-based Learning from Visual Demonstration for Robotic Manipulation via Differentiable Physics-based Simulation and Rendering

Xinghao Zhu¹ Jinghan Ke² Zhixuan Xu³ Zhixin Sun⁴ Bizhe Bai⁵ Jun Lv⁶
Qingtao Liu³ Yuwei Zeng⁷ Qi Ye³ Cewu Lu⁶ Masayoshi Tomizuka¹ Lin Shao⁷
¹UC Berkeley, ²USTC, ³Zhejiang University, ⁴Nanjing University
⁵University of Queensland, ⁶Shanghai Jiaotong University, ⁷National University of Singapore

Abstract: Learning from Demonstration (LfD) is an efficient technique for robots to acquire new skills through expert observation, significantly mitigating the need for laborious manual reward function design. This paper introduces a novel framework for model-based LfD in the context of robotic manipulation. Our proposed pipeline is underpinned by two primary components: *self-supervised pose and shape estimation* and *contact sequence generation*. The former utilizes differentiable rendering to estimate object poses and shapes from demonstration videos, while the latter iteratively optimizes contact points and forces using differentiable simulation, consequently effectuating object transformations. Empirical evidence demonstrates the efficacy of our LfD pipeline in acquiring manipulation actions from human demonstrations. Complementary to this, ablation studies focusing on object tracking and contact sequence inference underscore the robustness and efficiency of our approach in generating long-horizon manipulation actions, even amidst environmental noise. Validation of our results extends to real-world deployment of the proposed pipeline. Supplementary materials and videos are available on our webpage: <https://sites.google.com/view/diff-ldf>.

Keywords: Learning from Visual Demonstration, Model-based Robotic Manipulation, Differentiable Physics-based Simulation and Rendering

1 Introduction

Learning from Demonstration (LfD) empowers robots to acquire policies from expert demonstrations, such as those available on YouTube [1], which can reduce the human effort involved in robotic skill learning [2, 3]. This paper delves into the development of a model-based LfD pipeline that employs raw RGB videos as inputs. While model-based learning approaches have been acknowledged for their potential for superior sample-efficiency and generalization compared to model-free approaches [4–6], model-based LfD remains under-explored. Several major challenges hinder the wide application of model-based LfD in the physical world.

One challenge is how to automatically and efficiently develop a model that scales to high-dimensional input such as raw images or videos [7]. To tackle this, we introduce a self-supervised modeling pipeline that leverages recent advancements in differentiable rendering and signed distance functions. This pipeline estimates both the geometric shape of the object and its associated 6D poses, forming an explicit representation. A second challenge lies in enabling robots to effectively utilize physical models to generate efficient policies. This is particularly critical for robots operating in real-world contact-rich manipulation tasks where the physical interaction between the robot and its environment is a key factor [8, 9]. To address this, we develop a hierarchical LfD framework that integrates low-level modules for contact-point localization and contact-force optimization with

Correspondence to zhuxh@berkeley.edu and linshao@nus.edu.sg

a high-level module for contact sequence planning. These modules work in concert to plan manipulative actions. To ensure robust and real-time deployment, we further incorporate a neural policy designed to imitate the outcomes of planning algorithms. This enables the robot to execute complex tasks with high reliability and efficiency.

We have evaluated our pipeline on two datasets, including the sth-sth dataset [10] containing basic manipulation actions on various objects and a small recorded video dataset showing a human performing dexterous in-hand manipulation with primitive objects. The results, derived from rigorous simulation and real-world experiments, bear testament to the effectiveness of our proposed pipeline.

Our key contributions can be summarized as follows: 1) introduce a novel framework for model-based learning from visual demonstrations, 2) provide a self-supervised approach for pose estimation and shape reconstruction, utilizing the differentiable rendering, 3) develop a hierarchical policy that combines the low-level contact-point localization and contact-force optimization based on the differentiable simulation and high-level contact sequence planning, with neural imitation learning for efficient and robust real-world execution, 4) conduct comprehensive experimental validation of our algorithms in both simulated and real-world environments to demonstrate their efficacy and robustness.

2 Diff-LfD Framework

Overview. Given a demonstrated RGB video consisting of N frames denoted as $\mathcal{V} = \{\mathcal{I}_t\}_{t=1}^N$, we preprocess the video to segment and identify the most relevant objects with masks $\{\mathcal{M}_t\}_{t=1}^N$, exploiting the SAM [11]. The local frame of the object is randomly defined at the first frame. Our *Diff-LfD* calculates the object’s relative pose transformation in the demonstration and jointly estimates the object’s mesh \mathcal{O} and the associated 6D poses $\{\mathcal{P}_t\}_{t=1}^N$ at each frame. If the robot is provided with a similar but different object from the object recorded in the video, we align the pose of the provided manipulated object with the reconstructed object. Next, our pipeline infers the *wrench* (a combination of external forces and torques) required to complete the pose transformation across two consecutive time steps and generates feasible robot actions to accomplish the pose transformation. This planning includes both the low-level contact-point localization and contact-force optimization and high-level contact sequence planning to chain the whole (long-horizon) manipulation sequences. The manipulation actions are then utilized to train a neural network for robust real-world execution and generalization. We provide an explanation of the pipeline, *wrench*, and object alignment in [Appendix](#).

2.1 Pose and Shape Estimation with Differentiable SDF

This subsection introduces the pipeline for pose estimation and shape reconstruction from raw videos. We adopt the differentiable SDF (*Diff-SDF*) [12] to represent the object geometry, which has a large representation capability to model diverse objects with various topology structures. Moreover, Diff-SDF enables smooth image-based optimization to the images due to its inherent convexity [12]. The explicit surface mesh \mathcal{O} can be extracted from the SDF using the marching cube method [13].

Ideally, given an initialization of an SDF parameterized by ϕ and its associated 6D poses at each frame $\{\mathcal{P}_t\}_{t=1}^N$, the differentiable renderer \mathcal{R} produces a sequence of images $\{\mathcal{I}(\phi, \mathcal{P}_t)_t\}_{t=1}^N = \{\mathcal{R}(\phi, \mathcal{P}_t)\}_{t=1}^N$. *Diff-SDF* optimizes the SDF parameters to reconstruct the object shape by reducing the reconstruction loss:

$$\mathcal{L}^R = \sum_{t=1}^N \|\mathcal{I}(\phi, \mathcal{P}_t)_t - \mathcal{I}_t\| \quad (1)$$

However, this approach encounters difficulties when applied to real-world videos due to the following reasons: *unknown camera poses* and *lack of views for unseen regions*. Because the SDF optimization presumes that camera poses of \mathcal{I}_t are known in advance, which is not valid for real-world videos where camera poses are not provided. To estimate camera poses $\{\mathcal{P}_t^{-1}\}_{t=1}^N$, we employ differentiable rendering to hierarchically produce an explicit surface mesh $\hat{\mathcal{O}}$ with texture denoted

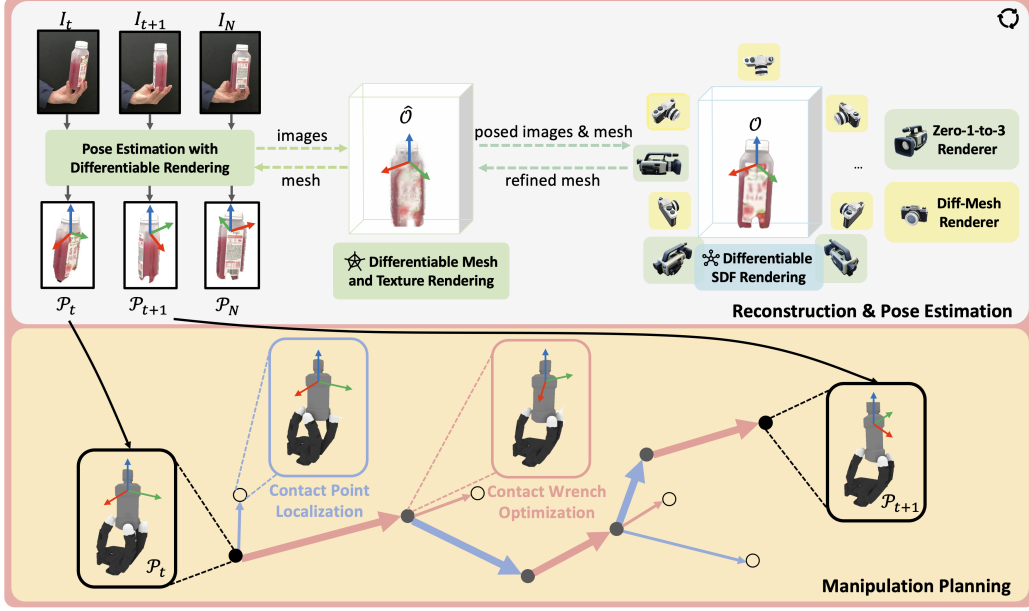


Figure 1: The proposed model-based learning from demonstration (LfD) pipeline can be divided into two primary components. The top part focuses on object shape reconstruction and pose estimation, employing differentiable mesh rendering and signed distance function (SDF) (Section 2.1). The bottom part illustrates the process of contact-aware hierarchical manipulation planning, involving contact point localization and differentiable wrench optimization (Section 2.2).

as $\hat{\mathcal{T}}$, and jointly estimate the objects poses $\{\hat{\mathcal{P}}_t\}_{t=1}^N$ over multiple images. Details are provided in the [Appendix](#).

We denote the optimized mesh, textures, and poses from differentiable rendering as $\hat{O}^*, \hat{\mathcal{T}}^*, \{\hat{\mathcal{P}}_t^*\}$, respectively, and their associated rendered images $\hat{\mathcal{I}}_t^* = \mathcal{R}(\hat{O}^*, \hat{\mathcal{T}}^*, \{\hat{\mathcal{P}}_t^*\})$ as:

$$\hat{O}^*, \hat{\mathcal{T}}^*, \{\hat{\mathcal{P}}_t^*\} = \arg \min_{\hat{O}, \hat{\mathcal{T}}, \{\hat{\mathcal{P}}_t\}} \sum_{t=1}^N \|\hat{\mathcal{I}}_t - \mathcal{I}_t\| \quad \text{where} \quad \hat{\mathcal{I}}_t = \mathcal{R}(\hat{O}, \hat{\mathcal{T}}, \{\hat{\mathcal{P}}_t\}) \quad (2)$$

The quality of mesh \hat{O}^* is usually not satisfying. We then optimize the Diff-SDF to get an optimized SDF ϕ^* by setting the camera pose to be $\{\hat{\mathcal{P}}_t^{**}\}$ to reduce the projection loss:

$$\phi^* = \arg \min_{\phi} \sum_{t=1}^N \|\mathcal{I}(\phi, \hat{\mathcal{P}}_t^*) - \hat{\mathcal{I}}_t^*\| \quad \text{where} \quad \hat{\mathcal{I}}_t^* = \mathcal{R}(\hat{O}^*, \hat{\mathcal{T}}^*, \{\hat{\mathcal{P}}_t^*\}) \quad (3)$$

After the Diff-SDF optimization, the resulting surface mesh \hat{O}^{**} is then extracted from the SDF ϕ^* . The \hat{O}^{**} is then leveraged to optimize the object poses $\{\hat{\mathcal{P}}_t^{**}\}$ as below. The process in Eqn 2-4 iterates until we get a small loss below a given threshold or reach the maximum iteration number.

$$\{\hat{\mathcal{P}}_t^{**}\}, \hat{\mathcal{T}}^{**} = \arg \min_{\{\hat{\mathcal{P}}_t\}, \hat{\mathcal{T}}} \sum_{t=1}^N \|\hat{\mathcal{I}}_t - \mathcal{I}_t\| \quad \text{where} \quad \hat{\mathcal{I}}_t = \mathcal{R}(\hat{O}^{**}, \hat{\mathcal{T}}, \{\hat{\mathcal{P}}_t\}) \quad (4)$$

Although each video contains multiple frames, there are still cases that lack sufficient views, resulting in poorly reconstructed unseen regions of the object. To address the incomplete views, we adopt a diffusion model [14] to infer the unseen areas. Our model takes the first real image \mathcal{I}_1 with a known camera pose and synthesizes images from different viewpoints around the object. We then combine these synthetic views with others for a complete object-shape reconstruction.

2.2 Contact-Aware Manipulation Policy

Building on the estimated object's pose and shape, this subsection delves into the process of manipulating an object between two consecutive poses \mathcal{P}_t and \mathcal{P}_{t+1} . If the robot is provided with a

similar but different object from the object recorded in the video, we align the pose of the provided manipulated object with the reconstructed object, with details in [Appendix](#).

Our framework employs a hierarchical structure consisting of low-level modules for contact-point localization and contact-force optimization, as well as high-level contact sequence planning. The low-level modules serve dual purposes: contact-point localization allows the robot to establish new contacts while keeping the object stationary, whereas contact-force optimization enables the robot to manipulate the object toward its target and maintain stable contact. These low-level actions are then orchestrated by the high-level contact sequence planning module to form a cohesive sequence of actions. To facilitate efficient and robust real-time deployment, we also incorporate a neural policy designed to imitate the planned trajectories.

Contact point localization. Contact point localization enables the robot to change contact points on the object, which encompasses two critical steps: *the generation of the transition target* and *the execution of the transition*. The transition target is calculated analytically with the desired object transformation wrench \mathcal{W} . Wrench \mathcal{W} is located at the objects’ center and represents the necessary wrench to facilitate the transformation from \mathcal{P}_t to \mathcal{P}_{t+1} . It is determined using a Proportional-Derivative (PD) controller: $\mathcal{W} = k_p * (\mathcal{P}_{t+1} - \mathcal{P}_t) - k_d \dot{\mathcal{P}}_t + g$, where k_p, k_d are the proportional and derivative gains, and g signifies the gravitational and external forces acting upon the object. Following this, we use an enumeration process to identify all plausible contact combinations that can generate the desired wrench \mathcal{W} . Initially, all potential contacts are assessed to single out those capable of producing the desired object wrench \mathcal{W} through contact points $\{p_i\}$ for $i \in [1..n]$. The number of contact points n is pre-determined based on the manipulation task. Further filtering processes are implemented to ascertain contacts that meet kinematic and stationary constraints: the inverse kinematics are solved to verify kinematic feasibility, and only one contact point can move at a given time while the remaining contact points hold the object immobile. Although multiple contact points could theoretically move while maintaining the object stationary, we found that planning is considerably more complex due to the enlarged search space, and the objects are prone to unintended movement due to execution errors. To execute the transition, we use the Rapidly-exploring Random Tree (RRT) motion planner to generate a feasible trajectory for the moving contact and the gravity compensation wrench on the remaining contacts to hold the object. More details of the contact point localization are provided in [Appendix](#).

Contact wrench optimization. Once the contact points $p = \{p_i\}$ are determined, the robot exerts contact wrenches $\mathcal{W}^p = \{\mathcal{W}^{p_i}\}$ at contact points to manipulate the object toward its target. The objective and loss functions to optimize the contact wrenches are defined in Eq. 5.

$$\min_{\mathcal{W}^p} \mathcal{L}(\mathcal{W}^p) = \lambda_{\mathcal{P}} \|\mathcal{P}' \ominus \mathcal{P}_{t+1}\| + \lambda_v \|\dot{\mathcal{P}}'\| + \lambda_{\mathcal{W}} \|\mathcal{W}^p\| \quad \text{subject to} \quad \mathcal{P}' = \mathcal{F}(\mathcal{P}, \mathcal{W}^p) \quad (5)$$

where \mathcal{F} represents the contact dynamics, \mathcal{P}' is the 6D object pose after applying the wrench \mathcal{W}^p from the initial object pose \mathcal{P} . \mathcal{P}_{t+1} is the target object pose, \ominus represents subtraction for 6D poses. $\dot{\mathcal{P}}'$ represents the object’s velocity and is added to damp the object’s speed, making the manipulation more stable [15]. $\lambda_{\mathcal{P}}, \lambda_v, \lambda_{\mathcal{W}}$ are hyperparameters standing for loss weights.

We propose using gradient-based methods to optimize the objective function, Eq. 5, with differentiable simulation in Nimble Physics[16] to approximate the forward dynamics \mathcal{F} . We use $s = (\mathcal{P}, p)$ as the concatenated state of the system in the following of this paper. The gradient of the objective can be computed as $\nabla = \frac{\partial \mathcal{L}}{\partial \mathcal{W}^p}$ from the simulation. However, gradients near the contact are often nonlinear, sensitive, and discontinuous, posing challenges for vanilla gradient descent optimization methods. To address this issue, this paper draws inspiration from [17–20] and proposes computing the gradient expectation at each point with Gaussian noises, as shown in Eq. 6. The contact wrench is then updated using a step size α along the gradient direction. In this paper, we use the analytical contact wrench computed during the contact point localization as the initial solution point for the optimization process.

$$\nabla = \mathbb{E}_{n_s, n_{\mathcal{W}} \sim \mathcal{N}} \left[\frac{\partial \mathcal{L}(\mathcal{F}(s + n_s, \mathcal{W}^p + n_{\mathcal{W}}), \mathcal{W}^p + n_{\mathcal{W}})}{\partial \mathcal{W}^p} \right] \quad (6)$$

High-level planning. Our global contact sequence planning, as detailed in Algorithm 1, employs hierarchical planning to identify viable manipulation sequences, utilizing the previously introduced contact point localization (`ContactLoc`) and contact wrench optimization (`OptWrench`). While the exertion of contact wrenches allows the robot to perform manipulation tasks involving nearby target poses, switching between multiple contacts is necessary when dealing with distant targets due to kinematic limitations.

Every node s in the planning tree encapsulates the object pose \mathcal{P} and contact p . The tree begins with a start node that represents the initial object pose and robot contact, with the goal

of reaching the target object pose \mathcal{P}_{t+1} . At each iteration, a node s is chosen and expanded using `ContactLoc` or `OptWrench` following A* search [21]. If the object has been successfully manipulated through the exertion of an optimized contact wrench (`OptIsSuccess`), the resulting node s' , containing the manipulated object pose and contact points, is expanded. Conversely, if the exertion fails, contact localization is performed by identifying a new set of contacts and expanding them in the tree. This planning algorithm continues until either the target object pose is reached (`IsSuccess`) or all nodes within the tree have been explored. The search procedure we propose focuses on optimizing contact wrenches at first and resorts to locating new contacts only if the exertion fails. Although this approach narrows down the search space, it may also prune potentially valid and optimal paths. For instance, transiting contacts before reaching the kinematic limit might result in a shorter trajectory with fewer contact switches. To address this, we introduce a random chance for each node to transition contacts, regardless of the wrench optimization outcome. This design promotes exploration within the planning process, enabling a more comprehensive discovery of the entire search space.

Sim2Real: closed-loop policy with domain randomization. Despite its efficacy in generating viable manipulation trajectories, the above-described planning algorithm is computationally demanding as it requires online enumeration of contacts and wrench optimization, rendering it unsuitable for real-time applications. To surmount this challenge, we utilize deep learning to approximate the manipulation policy. We leverage a fully connected network to learn the robot control commands that were derived from the high-level planning algorithm. This network ingests the object pose and joint angles as inputs and outputs of robot joint torques. These torques are obtained by mapping the contact wrenches into joint torques, courtesy of the Jacobian. Our training dataset is generated by solving the planning problem under conditions of noisy initial and target positions and perturbed system dynamics. This process results in a set of state-torque training pairs. We further augment each sample by introducing noise into the states and optimizing the joint torques to adhere to the planned trajectory. It’s noteworthy that for domain randomization, we optimize the contact wrench to reach the next state along the trajectory rather than solving the original planning problem with a distant target. This makes the data augmentation process more efficient. In a bid to further enhance performance, we fine-tune the network within a Markov Decision Process (MDP) framework using the REINFORCE algorithm [22]. During the fine-tuning process, the state and action spaces maintain the same setup as described earlier, while the reward function is defined as $r(s, \tau) = -\lambda_{\mathcal{P}} \|s \ominus \mathcal{P}_{t+1}\| - \lambda_v \|\dot{s}\| - \lambda_{\mathcal{W}} \|\tau\|$, where \mathcal{P}_{t+1} is the target object pose, and \dot{s} denotes the object velocity [15]. $\lambda_{\mathcal{P}}, \lambda_v, \lambda_{\tau}$ are hyperparameters.

Algorithm 1 Global Planning for Manipulation Sequences

```

1: Input:  $s_0 = (\mathcal{P}_t, p_t)$ , target object pose  $\mathcal{P}_{t+1}$ 
2: Output:  $\mathcal{R} = \{s\}$ 
3:  $\mathcal{Q} \leftarrow \{s_0\}, \mathcal{R} \leftarrow \{\emptyset\}$  ▷ Init.
4: while  $\mathcal{Q}$  is not empty do
5:    $s \leftarrow \text{SelectNode}(\mathcal{Q})$ 
6:   if IsSuccess( $s, \mathcal{P}_{t+1}$ ) then
7:     Return  $\mathcal{R}$  ▷ Exit if success
8:   end if
9:    $s' \leftarrow \text{OptWrench}(s, \mathcal{P}_{t+1})$  ▷ Opt. wrench
10:  if OptIsSuccess( $s, s'$ ) then
11:     $\mathcal{Q} \leftarrow \mathcal{Q} \cup s'; \mathcal{R} \leftarrow \mathcal{R} \cup s'$ 
12:  else
13:     $\mathcal{S} \leftarrow \text{ContactLoc}(s, \mathcal{P}_{t+1})$  ▷ Loc. contacts
14:    for  $s' \in \mathcal{S}$  do
15:       $\mathcal{Q} \leftarrow \mathcal{Q} \cup s', \mathcal{R} \leftarrow \mathcal{R} \cup s'$ 
16:    end for
17:  end if
18: end while
19: Return  $\mathcal{R}$ 

```

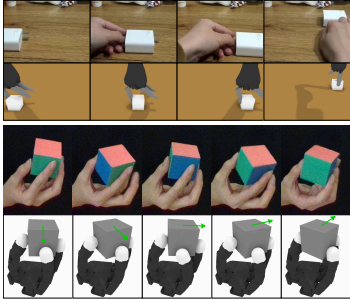


Figure 2: Experimental results from sth-sth (1st & 2nd rows) and in- (3rd & 4th rows). The first element in each cell is the mean/variance for the command object manipulation (3rd & 4th rows). The second is the difference between the target and final object rotation ($^{\circ}$).

	Pull Right	Pull Left	Push Right	Push Left
Baseline [23]	0.976	0.992	0.994	0.946
Ours	1.000	1.000	1.000	1.000

Figure 3: Baseline comparisons on LfD framework. Each cell represents the success rate of the manipulation.

	RRT		CITO		PGDM		iLQR		Ours	
Ball	122 \pm 20	7.2 $^{\circ}$	52 \pm 8	16.7 $^{\circ}$	2.14 \pm 0.4	2.4 $^{\circ}$	57 \pm 10	11.2 $^{\circ}$	62 \pm 12	2.6 $^{\circ}$
Cube	136 \pm 16	9.0 $^{\circ}$	60 \pm 7	18.5 $^{\circ}$	2.16 \pm 0.3	4.1 $^{\circ}$	70 \pm 19	13.3 $^{\circ}$	78 \pm 16	3.8 $^{\circ}$
Capsule	127 \pm 24	8.4 $^{\circ}$	63 \pm 4	15.2 $^{\circ}$	2.18 \pm 0.3	9.3 $^{\circ}$	80 \pm 6	12.2 $^{\circ}$	81 \pm 8	7.7 $^{\circ}$

3 Experiments

This section offers both quantitative and qualitative assessments of our proposed methodology. Our experiments are designed to address the following research questions: 1) How does our Diff-LfD framework compare to baselines that also rely on visual demonstrations? 2) What is the efficacy of our contact-aware manipulation algorithm in generating long-horizon trajectories? 3) Is our approach feasible for deployment in real-world scenarios? 4) How accurate is our self-supervised object reconstruction and tracking? 5) What is the utility of the views synthesized by the diffusion model? 6) What impact do gradient-based optimization, global planning, and random contact transition have on performance? 7) How robust are the generated trajectories and the closed-loop policy? We conducted evaluations in two distinct experimental settings: basic manipulation tasks involving primitive objects and more complex in-hand object manipulation tasks.

Experimental setups and ablation studies addressing questions 4-7 are elaborated in [Appendix](#).

Baseline comparisons on LfD framework. We compare our model-based approach with the method introduced in [23]. Petřík et al. [23] presents an optimization-based method to estimate a coarse 3D state representation, using a cylinder for the hand and a cuboid for the manipulated object(s). Such coarse approximation limits the representation capability and the quality of the state estimation. We utilize our object reconstruction and tracking to estimate the object trajectory and use contact planning to find a path. We select videos of 4 classes from the sth-sth dataset [10]: "Pull Right" with 164 videos, "Pull Left" with 130 videos, "Push Right" with 89 videos, and "Push Left" with 253 videos. We report the results as in Fig. 3. Our approach successfully finished all the classes and slightly outperformed the method introduced in [23]. One explanation is that these four types of videos are simple for our proposed pipeline to imitate. Thus, we also apply our method for in-hand manipulation tasks from raw videos to test the limits of our proposed framework. Fig. 2 shows the manipulation trajectory in two environments generated by our method.

Baseline comparisons on shape reconstruction and pose estimation. In contrast to other learning-based approaches for shape reconstruction and pose estimation, such as Neural Radiance Fields (NeRF), our perception module operates under a distinct task setting. Specifically, our input consists of a single-object RGB video featuring objects that undergo both rotation and translation. Most NeRF-based methods, on the other hand, rely on multiple static object poses with known camera positions. Some NeRF implementations utilize COLMAP [24] to initialize camera poses. However, this approach is less effective in our setting, where the background remains largely unchanged, and the frame count is limited. These factors hinder COLMAP's ability to accurately estimate object poses, leading to unstable object surface reconstructions from NeRF. Further experimental comparisons with Nope-NeRF [25], which also employs COLMAP for initialization, are available in the [webpage](#). Our findings indicate that Nope-NeRF fails to converge in more than half of the test cases (5 out of 9), resulting in empty reconstructions. The remaining cases yielded incorrect pose estimations and reconstructions when compared to our method.

Baseline comparisons on the contact-aware manipulation policy. In this study, we evaluate our contact-aware trajectory planning algorithm against four established baselines within the context of in-hand object rotation tasks. The baselines are as follows: 1) The Rapidly-Exploring Random Tree (RRT) planner, as outlined in [17], employs random sampling within a configuration space defined by both robot joint positions and object poses to identify feasible trajectories. 2) Contact-Implicit Trajectory Optimization (CITO) [26] first establishes a predefined trajectory for the object, then identifies optimal contact points along this path before calculating the requisite control inputs for trajectory tracking. 3) Pre-Grasp Informed Dexterous Manipulation (PGDM) [27] utilizes reinforcement learning to train manipulation agents, incorporating pre-computed grasp data to achieve the desired manipulation trajectory. 4) The Iterative Linear Quadratic Regulator (iLQR) [17] employs local approximations of the dynamical system to iteratively solve for optimal manipulation strategies through quadratic planning. For the purposes of this experiment, our algorithm operates without the closed-loop policy detailed in Sec. 2.2. We apply baselines on three in-hand manipulation tasks associated with the ball, the cube, and the capsule. We adopt two evaluation metrics: the averaging planning time and the difference between the target rotation and the final object rotation. Results are reported in Fig. 4.

While both PGDM and iLQR boast the quickest inference times, it’s crucial to highlight that PGDM requires approximately 5 hours of training for each task, and iLQR suffers from a higher tracking error compared to our method. The RRT approach uniformly expands its search tree, thereby increasing the probability of encountering unstable contacts and consequently requiring the most time to complete the task. In contrast, our algorithm and CITO focus on a more constrained search space where stable grasping is feasible, thereby simplifying the search complexity. Furthermore, our empirical results indicate that the final error rates for all baseline methods were consistently higher than our algorithm. Specifically, the RRT approach lacks a guarantee for optimal trajectory sampling, CITO overlooks physical dynamics during the planning phase, and iLQR struggles with optimization over nonlinear loss contours. These limitations render the baseline methods susceptible to failure due to dynamic uncertainties and execution errors.

Real-world experiments. We conducted real-world experiments for in-hand object manipulation. The experimental setups are illustrated in Fig. 5. We trained the closed-loop policy as discussed in Sec. 2.2 to imitate a human rotating a cube and deployed it as the robot controller. Experimental videos are available in the [webpage](#). This network receives the current joint angles of the robot, and the current object poses as input and outputs the joint torques. We performed the in-hand manipulation task with the Allegro Hand for different primitive objects and initial poses. During supervised learning, convergence of the policy is achieved in approximately 9.3 minutes, while fine-tuning takes an average of 50.1 minutes. The results of the difference between the target and final object rotation errors are Cylinder (3.8°), Ball (2.4°), Lemon (6.3°), and Avocado (5.9°), which further underscore the ability of our closed-loop policy to generalize across similar but distinct geometries.

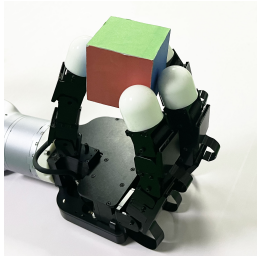


Figure 5: Allegro Hand performs in-hand object manipulations.

Additional experiments covering topics such as object reconstruction and tracking, the efficacy of the diffusion model, gradient-based optimization techniques, random contact transitions, and robustness analyses are provided in [Appendix](#).

4 Related Work

Learning from visual demonstration. We focus our review on the approaches that utilize visual data or adopt a model-based approach. For a broader review, we refer readers to [28]. One line of work in LfD [29–35] learns the cost and reward function from visual demonstrations. To extract knowledge from images/videos, various works [36–42] adopt representation-learning approaches to distill low-dimensional latent states and action representations. Developing explicit representation for LfD has received little attention due to the limited representation capability. Petrik et al. [23]

adopt coarse 3D cylinders and cuboids to present the hand and manipulated objects. To tackle the representation capability issue, we propose a self-supervised modeling pipeline that estimates the fine-level geometric shape of the object and its associated 6D pose sequences from raw videos. Moreover, our approach infers the contact forces underlie these poses, contributing to model-based learning algorithms.

Differentiable simulation and rendering in robotic manipulation. Recently, great progress has been made in the field of differentiable physics-based simulation and differentiable rendering [43–58], which for a broader review, please refer to [59, 60]. These differentiable tools have been applied in robotic manipulation tasks [20, 61–69]. We use the differentiable physics simulation to optimize the contact forces for in-hand manipulation tasks and proposed an iterative pose estimation and shape reconstruction pipeline from raw RGB videos via the differentiable rendering [70] and differentiable signed distance functions [12].

Model-based manipulation. The use of contact dynamics often leads to non-convex optimization problems, causing difficulties in finding local optima due to the discontinuity introduced by contact switching [19, 71, 72]. Contact-implicit trajectory optimization (CITO) [73–76] addresses this issue by planning manipulation actions without a pre-specified contact schedule. Chen et al. [26] further considers finger gaiting primitive in trajectory planning but assumes reachable states and pre-defined object trajectories, leading to potential failures due to ignorance of dynamical restrictions and control errors. Pang et al. [17] uses a convex quasi-dynamics model with a rapidly exploring random tree (RRT) to directly search feasible manipulation actions at the dynamical level, although the manipulated object lies on a surface and doesn’t require consideration of gravity. Our work plans manipulation actions directly at the dynamical level to address system noises and utilizes differentiable physics simulations for contact optimization and contact localization for efficient search.

5 Limitation

For the object shape reconstruction and pose estimation, we assume that the RGB videos are segmented, and the majority of the mass is concentrated as its geometry center. Our pipeline currently works only with rigid bodies, not with articulated rigid bodies or deformable objects. Although we leverage the diffusion model to mitigate the occlusion and reduce shape reconstruction uncertainty, the quality varies for real RGB videos when testing our pipeline in the wild. For contact-aware manipulation, we focus on tasks that require the object to move along a demonstrated trajectory. Generalizing the algorithm to other tasks with sparse rewards will be left for future work. Our approach relies on the differentiable physics-based simulation to generate the contact wrench with domain randomization to reduce the sim2real gap. Complicated physics interaction might fail to be captured by the differentiable physics simulation. We are interested in adding residual/learning layers to augment the differentiable physics simulation to align with the real world in future work.

6 Conclusion

This paper investigates the use of model-based learning from demonstrations for robotic manipulation tasks, contributing several significant aspects to the field. First, we introduce a new framework for learning from human visual demonstrations in a self-supervision manner, which has the potential to generate robot skills at a large scale. Second, we utilize differentiable rendering to track object poses in a self-supervised manner. Third, we design a high-level planning framework that employs differentiable simulations to generate long-horizon contact actions. This includes inferring and transitioning contact points, optimizing contact forces, and exerting them. The manipulation trajectories are then approximated by a neural network. Finally, we conduct experiments to evaluate the effectiveness of our approach from multiple angles. Our results demonstrate the robustness and efficiency of our proposed method to learn from human demonstrations and outperform existing approaches by a large margin.

Acknowledgments

Xinghao Zhu is supported by the UCB-FANUC Fellowship. This work was in part supported by a startup grant from the National University of Singapore.

References

- [1] X. Zhu, R. Tian, C. Xu, M. Huo, W. Zhan, M. Tomizuka, and M. Ding. Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot. <https://sites.google.com/berkeley.edu/fanuc-manipulation>, 2023.
- [2] L. Sun, H. Zhang, W. Xu, and M. Tomizuka. Efficient multi-task and transfer reinforcement learning with parameter-compositional framework. *IEEE Robotics and Automation Letters*, 8(8):4569–4576, 2023.
- [3] L. Sun, H. Zhang, W. Xu, and M. Tomizuka. Paco: Parameter-compositional multi-task reinforcement learning. In *NeurIPS*, 2022.
- [4] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.
- [5] S. Jin, X. Zhu, C. Wang, and M. Tomizuka. Contact pose identification for peg-in-hole assembly under uncertainties. In *2021 American Control Conference (ACC)*, pages 48–53. IEEE, 2021.
- [6] C. Wang, Y. Zhang, X. Zhang, Z. Wu, X. Zhu, S. Jin, T. Tang, and M. Tomizuka. Offline-online learning of deformation model for cable manipulation with graph neural networks. *IEEE Robotics and Automation Letters*, 7(2):5544–5551, 2022.
- [7] M. Huo, M. Ding, C. Xu, T. Tian, X. Zhu, Y. Mu, L. Sun, M. Tomizuka, and W. Zhan. Human-oriented representation learning for robotic manipulation. *arXiv preprint arXiv:2310.03023*, 2023.
- [8] X. Zhang, S. Jin, C. Wang, X. Zhu, and M. Tomizuka. Learning insertion primitives with discrete-continuous hybrid action space for robotic assembly tasks. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9881–9887. IEEE, 2022.
- [9] X. Zhang, C. Wang, L. Sun, Z. Wu, X. Zhu, and M. Tomizuka. Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning. In *7th Annual Conference on Robot Learning*, 2023.
- [10] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fründ, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thureau, I. Bax, and R. Memisevic. The “something something” video database for learning and evaluating visual common sense. *CoRR*, abs/1706.04261, 2017. URL <https://20bn.com/datasets/something-something>.
- [11] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.
- [12] D. Vicini, S. Speierer, and W. Jakob. Differentiable signed distance function rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)*, 41(4):125:1–125:18, July 2022. doi:10.1145/3528223.3530139.
- [13] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH ’87*, page 163–169, New York, NY, USA, 1987. Association for Computing Machinery. ISBN 0897912276. doi:10.1145/37401.37422. URL <https://doi.org/10.1145/37401.37422>.

- [14] R. Liu, R. Wu, B. V. Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv:2303.11328*, 2023.
- [15] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik. In-Hand Object Rotation via Rapid Motor Adaptation. In *Conference on Robot Learning (CoRL)*, 2022.
- [16] nimble. Nimble physics documentation. <https://nimblephysics.org/docs>.
- [17] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models, 2022.
- [18] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL <https://proceedings.mlr.press/v15/ross11a.html>.
- [19] R. Antonova, J. Yang, K. M. Jatavallabhula, and J. Bohg. Rethinking optimization with differentiable simulation from a global perspective. In *6th Annual Conference on Robot Learning*, 2022.
- [20] X. Zhu, W. Lian, B. Yuan, C. D. Freeman, and M. Tomizuka. Allowing safe contact in robotic goal-reaching: Planning and tracking in operational and null spaces. *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [21] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi: [10.1109/tssc.1968.300136](https://doi.org/10.1109/tssc.1968.300136). URL <https://doi.org/10.1109/tssc.1968.300136>.
- [22] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, may 1992. ISSN 0885-6125. doi: [10.1007/BF00992696](https://doi.org/10.1007/BF00992696).
- [23] V. Petrák, M. Tapaswi, I. Laptev, and J. Sivic. Learning object manipulation skills via approximate state estimation from real videos. In *Conference on Robot Learning*, pages 296–312. PMLR, 2021.
- [24] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [25] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2023.
- [26] C. Chen, P. Culbertson, M. Lepert, M. Schwager, and J. Bohg. Trajectorytree: Trajectory optimization meets tree search for planning multi-contact dexterous manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8262–8268, 2021. doi:[10.1109/IROS51168.2021.9636346](https://doi.org/10.1109/IROS51168.2021.9636346).
- [27] S. Dasari, A. Gupta, and V. Kumar. Learning dexterous manipulation from exemplar object trajectories and pre-grasps. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3889–3896. IEEE, 2023.
- [28] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.
- [29] N. Das, S. Bechtler, T. Davchev, D. Jayaraman, A. Rai, and F. Meier. Model-based inverse reinforcement learning from visual demonstrations. In *Conference on Robot Learning*, pages 1930–1942. PMLR, 2021.

- [30] A. Singh, L. Yang, K. Hartikainen, C. Finn, and S. Levine. End-to-end robotic reinforcement learning without reward engineering. *arXiv preprint arXiv:1904.07854*, 2019.
- [31] L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos, 2020.
- [32] N. Das, S. Bechtle, T. Davchev, D. Jayaraman, A. Rai, and F. Meier. Model based inverse reinforcement learning from visual demonstration. In *Conference on Robot Learning (CoRL)*, 2020.
- [33] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. *The International Journal of Robotics Research*, 40(12-14):1419–1434, 2021.
- [34] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 570–587. Springer, 2022.
- [35] K. Shaw, S. Bahl, and D. Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning*, pages 654–665. PMLR, 2023.
- [36] X. Zhu, L. Sun, Y. Fan, and M. Tomizuka. 6-dof contrastive grasp proposal network. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6371–6377. IEEE, 2021.
- [37] M. Yang and O. Nachum. Representation matters: offline pretraining for sequential decision making. In *International Conference on Machine Learning*, pages 11784–11794. PMLR, 2021.
- [38] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.
- [39] X. Zhu, Y. Zhou, Y. Fan, L. Sun, J. Chen, and M. Tomizuka. Learn to grasp with less supervision: A data-efficient maximum likelihood grasp sampling loss. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 721–727. IEEE, 2022.
- [40] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.
- [41] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.
- [42] X. Zhu, S. Jain, M. Tomizuka, and J. Van Baar. Learning to synthesize volumetric meshes from vision-based tactile imprints. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4833–4839. IEEE, 2022.
- [43] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter. End-to-end differentiable physics for learning and control. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL <https://proceedings.neurips.cc/paper/2018/file/842424a1d0595b76ec4fa03c46e8d755-Paper.pdf>.
- [44] J. Degraeve, M. Hermans, J. Dambre, et al. A differentiable physics engine for deep learning in robotics. *Frontiers in neurorobotics*, page 6, 2019.
- [45] Y. Hu, T.-M. Li, L. Anderson, J. Ragan-Kelley, and F. Durand. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)*, 38(6):201, 2019.

- [46] K. M. Jatavallabhula, M. Macklin, F. Golemo, V. Voleti, L. Petrini, M. Weiss, B. Considine, J. Parent-Lévesque, K. Xie, K. Erleben, et al. gradsim: Differentiable simulation for system identification and visuomotor control. *arXiv preprint arXiv:2104.02646*, 2021.
- [47] M. Geilinger, D. Hahn, J. Zehnder, M. Bächer, B. Thomaszewski, and S. Coros. Add: analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020.
- [48] T. Du, K. Wu, P. Ma, S. Wah, A. Spielberg, D. Rus, and W. Matusik. Diffpd: Differentiable projective dynamics. *ACM Trans. Graph.*, 41(2), nov 2021. ISSN 0730-0301. doi:10.1145/3490168. URL <https://doi.org/10.1145/3490168>.
- [49] J. Liang, M. Lin, and V. Koltun. Differentiable cloth simulation for inverse problems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/28f0b864598a1291557bed248a998d4e-Paper.pdf>.
- [50] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin. Scalable differentiable physics for learning and control. *arXiv preprint arXiv:2007.02168*, 2020.
- [51] Y. Li, T. Du, K. Wu, J. Xu, and W. Matusik. Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM Transactions on Graphics (TOG)*, 42(1):1–20, 2022.
- [52] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu. Fast and feature-complete differentiable physics for articulated rigid bodies with contact. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2021.
- [53] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane. Joint optimization of robot design and motion parameters using the implicit function theorem. In S. Srinivasa, N. Ayanian, N. Amato, and S. Kuindersma, editors, *Robotics, Robotics: Science and Systems, United States, 2017*. MIT Press Journals. doi:10.15607/rss.2017.xiii.003. Publisher Copyright: © 2017 MIT Press Journals. All rights reserved.; 2017 Robotics: Science and Systems, RSS 2017 ; Conference date: 12-07-2017 Through 16-07-2017.
- [54] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin. Efficient differentiable simulation of articulated bodies. In *International Conference on Machine Learning*, pages 8661–8671. PMLR, 2021.
- [55] K. Um, R. Brand, Y. R. Fei, P. Holl, and N. Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in Neural Information Processing Systems*, 33:6111–6122, 2020.
- [56] N. Wandel, M. Weinmann, and R. Klein. Learning incompressible fluid dynamics from scratch—towards fast, differentiable fluid models that generalize. *arXiv preprint arXiv:2006.08762*, 2020.
- [57] P. Holl, V. Koltun, and N. Thuerey. Learning to control pdes with differentiable physics. *arXiv preprint arXiv:2001.07457*, 2020.
- [58] T. Takahashi, J. Liang, Y.-L. Qiao, and M. C. Lin. Differentiable fluids with solid coupling for learning and control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7): 6138–6146, May 2021. doi:10.1609/aaai.v35i7.16764. URL <https://ojs.aaai.org/index.php/AAAI/article/view/16764>.
- [59] H. Kato, D. Beker, M. Morariu, T. Ando, T. Matsuoka, W. Kehl, and A. Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020.
- [60] S. Zhao, W. Jakob, and T.-M. Li. Physics-based differentiable rendering: from theory to implementation. In *ACM siggraph 2020 courses*, pages 1–30. 2020.

- [61] J. Lv, Q. Yu, L. Shao, W. Liu, W. Xu, and C. Lu. Sagci-system: Towards sample-efficient, generalizable, compositional, and incremental robot learning. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.
- [62] 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. 2022.
- [63] D. Turpin, L. Wang, E. Heiden, Y.-C. Chen, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg. Grasp’d: Differentiable contact-rich grasp synthesis for multi-fingered hands. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, pages 201–221. Springer, 2022.
- [64] 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. In *6th Annual Conference on Robot Learning*, 2022. URL <https://openreview.net/forum?id=tyxyBj2w4vw>.
- [65] K. M. Jatavallabhula, M. Macklin, F. Golemo, V. Voleti, L. Petrini, M. Weiss, B. Conside, J. Parent-Levesque, K. Xie, K. Erleben, L. Paull, F. Shkurti, D. Nowrouzezahrai, and S. Fidler. gradsim: Differentiable simulation for system identification and visuomotor control. *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=c_E8kFWfhp0.
- [66] P. Sundaresan, R. Antonova, and J. Bohg. Diffcloud: Real-to-sim from point clouds with differentiable simulation and rendering of deformable objects. *arXiv preprint arXiv:2204.03139*, 2022.
- [67] J. Zhang, Z. Wan, and J. Liao. Adaptive joint optimization for 3d reconstruction with differentiable rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2022.
- [68] J. Huang, J. Thies, A. Dai, A. Kundu, C. Jiang, L. J. Guibas, M. Nießner, T. Funkhouser, et al. Adversarial texture optimization from rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1559–1568, 2020.
- [69] J. Lv, Y. Feng, C. Zhang, S. Zhao, L. Shao, and C. Lu. Sam-rl: Sensing-aware model-based reinforcement learning via differentiable physics-based simulation and rendering. *arXiv preprint arXiv:2210.15185*, 2022.
- [70] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018.
- [71] H. J. Suh, M. Simchowitz, K. Zhang, and R. Tedrake. Do differentiable simulators give better policy gradients? In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20668–20696. PMLR, 17–23 Jul 2022.
- [72] Y. Fan, X. Zhu, and M. Tomizuka. Optimization model for planning precision grasps with multi-fingered hands. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1548–1554. IEEE, 2019.
- [73] I. Mordatch, Z. Popović, and E. Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’12, page 137–144. Eurographics Association, 2012. ISBN 9783905674378.
- [74] T. Marcucci, M. Gabiccini, and A. Artoni. A two-stage trajectory optimization strategy for articulated bodies with unscheduled contact sequences. *IEEE Robotics and Automation Letters*, 2(1):104–111, 2017. doi:10.1109/LRA.2016.2547024.

- [75] 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)*, pages 6520–6526, 2021. doi:[10.1109/ICRA48506.2021.9560766](https://doi.org/10.1109/ICRA48506.2021.9560766).
- [76] C. Wang, H.-C. Lin, S. Jin, X. Zhu, L. Sun, and M. Tomizuka. Bpomp: A bilevel path optimization formulation for motion planning. In *2022 American Control Conference (ACC)*, pages 1891–1897. IEEE, 2022.