TieBot: Model-based Learning to Knot a Tie from Visual Demonstration via Differentiable Physics-based Simulation

Weikun Peng[1], Jun Lv[2], Haonan Chen[3], Siheng Zhao[3], Jichen Sun[1], Cewu Lu[2], and Lin Shao[1]

***Abstract*— The tie-knotting task is highly challenging due to the tie's high deformation and long-horizon manipulation actions. This work presents a model-based learning from demonstration system called *TieBot* for the robots to learn to knot a tie. We introduce the iterative keypoint learning and the hierarchical matching approach to estimate the tie's shape from the demonstrated video. With these estimated shapes in a differentiable cloth simulation used as subgoals, we propose to combine model-free RL and differentiable simulation to generate the action sequences to achieve these subgoals. Then, an imitation learning approach is used to learn the action sequences from raw point clouds in the simulation. Lastly, our pipeline learns a residual policy when the imitated policy is applied to real-world execution, mitigating the sim2real gap. We demonstrate the effectiveness of *TieBot* in simulation and the real world. In the real-world experiment, a MOVO robot successfully knots the tie. Supplemental materials and videos are available on our project webpage at https://sites.google.com/view/tiebot**

I. Introduction

Learning deformable object manipulation holds great utility across a wide range of applications. One intriguing domain is robotic tie knotting. Service robots must be adept at tasks like aiding the elderly or individuals with disabilities in dressing for certain social events. Teaching robots to knot ties, as a special case of deformable object manipulation, typically push the limits of robotic manipulation. This offers valuable insights not only for tie knotting but also for the broader field of object manipulation.

Deformable object manipulation presents challenges for robots due to its high-dimensional state and complex dynamics. Extracting state information and modeling are difficult problems. In contrast, humans have accumulated extensive knowledge about manipulating deformable objects. For instance, humans have devised over a dozen distinct methods to knot a tie. These priors makes learning from demonstarion (LfD) a promising direction. LfD empowers a robot to acquire a policy from expert demonstrations, significantly reducing the need for manual design of task-specific reward functions. Consequently, LfD stands as a potent and efficient framework for instructing robots in the execution of complex skills.

[1]Weikun Peng, Jichen Sun, and Lin Shao are with the Department of Computer Science, National University of Singapore, Singapore. [peng.weikun,e1139713]@u.nus.edu, and linshao@nus.edu.sg

[2]Jun Lv and Cewu Lu are with Department of Computer Science, Shanghai Jiao Tong University, China. [lyujune_sjtu, lucewu]@sjtu.edu.cn

[3]Siheng Zhao and Haonan Chen are with the Department of Computer Science and Technology, Nanjing University, China. [zhaosh,chenhn]@smail.nju.edu.cn

Fig. 1. Our proposed *TieBot* performs tie-knotting task, which is high-deformation and sequential multiple-steps manipulation task. We leverage differentiable physics simulation to recovery the object's deformable state from human demonstration and learn a goal-condition policy to accomplish the tie-knotting task.

However, existing LfD methods cannot handle the tie-knotting. Knotting a tie is much more complex than previous deformable object tasks such as produce a knot in 1D rope. Ties are 2D deformable objects with a high-dimensional state space which makes visual perception challenging. Knotting a tie contains multiple complex robotic actions such as the dual-arm grasping and pulling. Such a large action space and under-actuated dynamics, making common algorithms either non-applicable or computationally expensive.

In this work, a systematic framework for manipulating tie is proposed, which aims to address all the challenges discussed above. We develop a model-based learning from demonstration pipeline for the robots to learn to knot a tie. First, we propose an iterative keypoint learning and hierarchical matching method to estimate the tie's shape with extremely large deformation from the demonstrated video. Here, We adopt a differentiable cloth simulator called *DiffClothAI* [1]. The simulation supports intersection-free contact for cloth, which is important for the tie knotting

task. These estimated shapes from the demonstrated video are used as subgoals. To tackle the high-dimensional state and action space, we propose to combine model-free RL and differentiable simulation to produce the action sequences to achieve these subgoals. Then an imitation learning approach is used to learn the action sequences from raw point clouds in the simulation which generalizes to different but similar tie shape positions and sizes. Lastly, our pipeline learns a residual policy when applying the imitated policy to the real-world execution, mitigating the sim2real gap. To the best of our knowledge, it's the first time to demonstrate the possibility of guiding a robot to accomplish the extremely complex tie-knotting task.

In summary, we make the following contributions: 1) We introduced a systematic model-based LfD framework for a dual-arm robot to learn to knot to tie. 2) We proposed an iterative oriented keypoints learning and a hierarchical matching approach to estimate the tie's shape with high deformation from the demonstrated RGB-D video. 3) We presented an integrated pipeline to combine differentiable physics simulation and reinforcement learning to generate the robotic action from a high-dimensional state and action space. 4) We conducted extensive experiments to demonstrate the effectiveness of our proposed pipeline both in simulation and in real-robot experiments.

II. Related Work

We review related literature on key components in our approach, including deformable object manipulation, learning from demonstration, and model-based robot learning via differentiable simulations. We describe how we are different from previous work.

A. Deformable Object Manipulation

Deformable Objects contain 1D structures such as cables and ropes, 2D structures such as fabrics, and 3D structures such as plasticine [2, 3]. Ties are 2D deformable objects, however, they are similar to 1D deformable objects but can produce extremely complex topological structures. Here we review related manipulation tasks for 1D and 2D deformable objects. For a broader review, please refer to [4].

1D Cable/Rope. Knotting is a typical manipulation task for 1D deformable object with a rich literature [5]. Various approaches were proposed to tackle problems in planning [6, 7] and state estimation [8]. **2D Cloth**. Garment folding is a classical manipulation task for 2D deformable objects [9–14]. A line of recent methods have been focusing on learning goal-conditioned policies in simulation [15–18]. Avigal Avigal et al. [19] develop a framework to learning Efficient bimanual folding policy. In this work, we study the tie-knotting task for robotic manipulation.

B. Learning from Demonstration

Here we focus our review on approaches developed for deformable objects. For a broader review, we refer readers to [20]. A line of research developed LfD for 1D deformable objects. Schulman et al. [21] introduced a LfD pipeline for adapting trajectories to new situations, based on non-rigid registration between the training scene and the test scene [22]. This work has also been extended to jointly optimize the transformation function and the trajectory in a unified optimization [23], to incorporate normals to find better registrations [24], and to use trajectory-aware method with multiple demonstration for large deviation from training scene [25]. Lee et al. [26] present a method that combines geometric warping with statistical learning to compute target poses, forces, and time-varying gains. For typical 2D/3D deformable objects such as garments or fabrics, Seita et al. [27] proposed to train a neural network model to generate grasping point for bed making based depth images, utilizing human demonstrated grasping data. Joshi et al. [28] used Dynamic Movement Primitives to learn cloth manipulation from demonstrations. Salhotra et al. [29] proposed a LfD approach which takes raw images as inputs and test their approach only in the simulation. Seita et al. [30] study the problem of Sequential Fabric Smoothing with a deep imitation learning approach in fabric simulator. They use DAgger [31] to train policies and test the learned policies in physical robot with domain randomization [32]. Here we develop a model-based LfD leveraging the differentiable cloth simulation.

C. Robot Learning via Differentiable Physics Simulation

Recently, great progress has been made in the field of differentiable physics-based simulation designed for rigid bodies [33–35], soft bodies [36–40], cloth [1, 41–43], articulated bodies [44–46], and fluids [47–50]. Differentiable simulation have been applied in various robotic manipulation tasks [51, 52]. Jatavallabhula et al. [53] first proposed a pipeline to leverage differentiable simulation and rendering for system identification and visuomotor control. Ma et al. [54] introduced a rendering-invariant state prediction network that maps images into states that are agnostic to rendering parameters. Li et al. [55] develop physics augmented continuum neural radiance fields to estimate both the unknown geometry and physical parameters of highly dynamic objects from multi-view videos. Recently, Zhu et al. [56] developed a model-based learning from visual demonstration for rigid body manipulation. In this work, we present a model-based learning from visual demonstration for deformable objects.

III. Technical Approach

This work presents a model-based LfD framework called *TieBot* to guide a dual-arm robot shown in Fig. 1 with an RGB-D camera to knot the tie from an RGB-D demonstration video. An overview of our proposed method is in Fig. 2. We first briefly introduce the differentiable cloth simulation used in this work (Sec. III-A). Then we describe the procedures to estimate the tie's shape sequences from the demonstrated video (Sec. III-B). Using the tie's shape sequences as subgoals, we introduce a pipeline to generate robot actions to manipulate the tie, combining model-free reinforcement

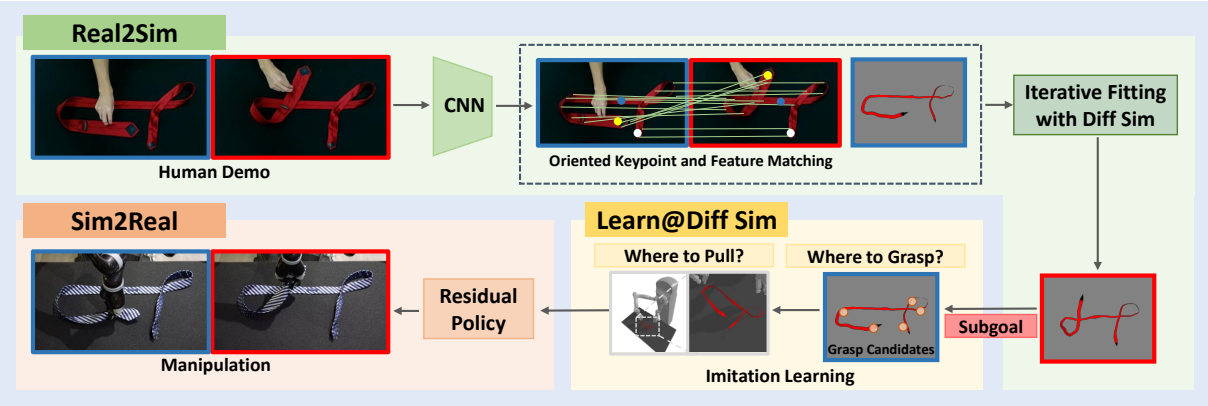

Fig. 2. *TieBot* utilizes differentiable simulation to estimate the tie's shapes from the demonstrated video. Then using the sequences of tie shapes as sub-goals, we introduce how to generate the robot actions to manipulate the tie. The pipeline also has a residual policy to reduce the sim2real gap.

learning (RL) and the differentiable simulation (Sec. III-C). Lastly, we discuss how to transfer the learned policy in simulation to the real knotting procedures (Sec. III-D).

A. Differentiable Cloth Simulation

We use *DiffClothAI* [1], a differentiable cloth simulation that maintains intersection-free property, which is crucial for tie-knotting. We use the positions of tie's vertices $\mathcal{X}_t^S$ to describe the tie shape. The action is applied by grasping certain vertices $\mathcal{V}$ to pull them towards new positions $a_t = \mathcal{P}^{a_t}(\mathcal{V})$. *DiffClothAI* simulates the tie's deformed shape at the next time step $\mathcal{X}_{t+1}^S = \mathcal{F}^{sim}(\mathcal{X}_t^S, a_t)$, where $\mathcal{F}^{sim}$ denotes the simulation. *DiffClothAI* provides the gradients $\partial \mathcal{X}_{t+1}^S / \partial a_t$, which are used to guide robot action updation. For detailed information about *DiffClothAI*, we refer to [1]. Note that a_t is executed on the selected vertices of the tie's mesh and we cannot change the index of selected vertices during the forward simulation from t to $t+1$. Thus, we need to select these vertices and discuss the grasping point generation in Sec. III-C.1.

B. Real2Sim

Given a demonstrated RGB-D video, our pipeline first segments the tie using the *Track-Anything* [57] and transforms the segmented depth images into point clouds. Meanwhile, a tie's mesh is loaded into the *DiffClothAI*. At time step t, we use the tie mesh's vertices denoted as $\mathcal{X}_t^S$ to describe the tie shape. From the demonstrated RGB images and segmented point clouds denoted as $\{\mathcal{I}_t^D\}$ and $\{\mathcal{X}_t^D\}$, Real2Sim pipeline estimates tie shape sequences $\{\mathcal{X}_t^S\}$ in simulation.

The pipeline first aligns the tie in the simulation with the initial demonstration by minimizing the Chamfer Distance [58] between $\mathcal{X}_0^S$ and $\mathcal{X}_0^D$. We assume the initial alignment is good.

1) Local Feature Matching: If $\mathcal{X}_{t-1}^S$ and $\mathcal{X}_{t-1}^D$ are aligned and there are correspondences between $\mathcal{X}_{t-1}^D$ and $\mathcal{X}_t^D$, we can build up the correspondences from the tie shape in the simulation to the next demonstrated point cloud $\mathcal{X}_t^D$ and optimize the tie's vertices to align $\mathcal{X}_t^S$ towards $\mathcal{X}_t^D$. The iteration continues.

Here we adopt the feature matching called *LoFTR* [59] to build up correspondences between two RGB images $\mathcal{I}_{t-1}$ and $\mathcal{I}_t$ as shown in Fig 3. From the correspondences between $\mathcal{I}_{t-1}$ and $\mathcal{I}_t$, we can calculate the correspondences between the segmented point clouds $\mathcal{X}_{t-1}^D$ and $\mathcal{X}_t^D$, which are used to align $\mathcal{X}_t^S$ towards $\mathcal{X}_t^D$. We select these corresponding vertices from $\mathcal{X}_t^S$ as the control points $\mathcal{V}$ defined in III-A in *DiffClothAI* to minimize the distances between these points in $\mathcal{X}_t^S$ and the corresponding positions in $\mathcal{X}_t^D$.

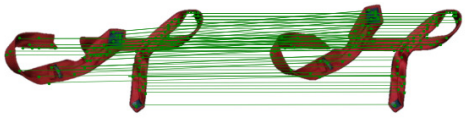

Fig. 3. Local Feature matching between two images.

2) Oriented Keypoints: Another method is to directly provide the correspondences from the tie's shape $\mathcal{X}^S$ in the simulation to the demonstrated point cloud $\mathcal{X}_t^D$ at each time step. This approach avoids generating correspondences between two demonstrated point clouds $\mathcal{X}_{t-1}^D$ and $\mathcal{X}_t^D$. We first define five keypoints on the tie's mesh and set up the local frames on each keypoints since the tie is a 2D deformable object with positive and negative side. Then we train a neural network model to predict the keypoints and associated local frames on the demonstrated point clouds. So that we could align $\{\mathcal{X}_t^S\}$ to $\{\mathcal{X}_t^D\}$ by sequentially optimizing the keypoint vertices on the tie mesh and their neighbor vertices (to calculate the local frames) to match the corresponding keypoints and local frames on the demonstrated point cloud.

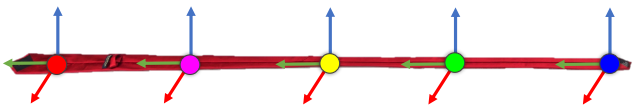

Fig. 4. The oriented keypoints to represent the state of the tie.

Definition. Selecting keypoints on deformable objects remains an open-ended problem, which falls outside this work's scope. We uniformly define five keypoints along the tie's surface in our setting. For each keypoint as the origin, we define the local frame as follows. The z direction is the surface normal from the tie's positive side to the negative side. The x direction is the direction of the tie's middle skeleton. The y direction is derived using the right-hand rule. These five keypoints in a predefined order play the role of

the skeleton as shown in Fig. 4.

Deepnet Model. We train a deep neural network $f_{\mathcal{K}}$ to predict the keypoints and the local frame for K keypoints $\{\mathcal{K}_j\}_{j=1}^{K}$. The neural network tasks as input a point cloud denoted as $\mathcal{X}_t = \{\mathcal{X}_{t,i}\}_{i=1}^{M}$ where M is the number of point clouds, and decodes to $M \times 7$ for each keypoint $\mathcal{K}_j$. For each point's $X_{t,i}$, the associated seven dimensions' output contains the probability score $S_{t,i}^k$ (1D) that the point $X_{t,i}$ to be the nearest point to the keypoint $\mathcal{K}_j$, the offset vector $D_{t,i}^k$ (3D) is the translation vector from the point $X_{t,i}$ to $\mathcal{K}_j$, the normal $\mathcal{F}_{t,i}^k$ (6D) is the normalized z and x direction of the local frame. Given these results, we are able to select the point in $\mathcal{X}_t$ with highest score $S_{t,i}^k$ and derive the position of the $\mathcal{K}_j$'s position, and the local frame described in the world coordinate. Due to page limit, the detailed description about the neural network architecture, training data generation in simulation, and loss function is put on the supplementary.

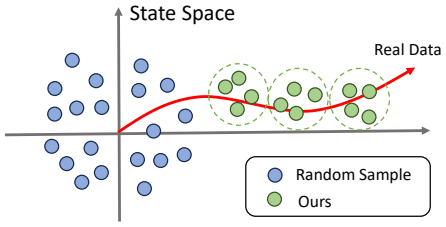

Fig. 5. Compared to random sampling, our iteratively sampling can select more data point from the tie-knotting distribution.

3) Iteratively Learning Keypoints: For the knotting-tie task, the high-dimensional state in deformable objects makes it challenging to generate sufficient training data to cover all the states encountered in the knotting procedure. As illustrated in Fig 5, the policy of successfully knotting a tie occupies only a small portion of the state space. Thus, uniformly applying random actions from the initial tie state in the simulation to produce training data fails to cover these states.

The question is how to generate the training data to cover all the states of the tie-knotting process. We use an iterative learning keypoints approach. When we detect the differences between $\mathcal{X}_t^S$ and $\mathcal{X}_t^D$ is larger than a threshold, we backtrack to the previous time step, gather the tie's shape $\mathcal{X}_{t-1}$ and apply random actions to the tie's mesh at $t-1$ in the simulation to generate annotated training data and retrain the keypoint network.

4) Hierarchical Matching: We combine the oriented keypoints and feature matching to generate a hierarchical correspondences for deformable objects. The oriented keypoints produce a global skeleton matching, while the feature matching plays the local dense matching. We use the hierarchical matching approach to estimate tie's shape from the demonstration. We denote the results of these estimated tie's shapes in the simulation as $\{\mathcal{X}_t^{\mathcal{G}}\}$, which are used as sub-goals to guide robot actions.

C. Learn@Sim

Given these subgoals $\{\mathcal{X}_t^{\mathcal{G}}\}$, our pipeline begins to sequentially generate feasible robotic actions in the simulation to guide the tie $\{\mathcal{X}_t^S\}$ towards these subgoals. Unlike Sec. III-B, where we can directly control arbitrary vertices on the tie, now our robot can at most grasp **two** regions to pull the tie. The robot's two arms need to cooperate. Note that the manipulating process to knot a tie is a long-horizon manipulation task.There are multiple grasping and pulling iterations. The ending position from the previous iteration also affects the grasping choices of the current iteration.

Here the robot motion trajectory is defined in Eqn. 1, which contains starting and ending poses for each hand.

$$a_t = \{\mathcal{T}_{t,i}^A, \mathcal{T}_{t,i}^B\}_{i=L,R} \tag{1}$$

Here $\{\mathcal{T}_{t,i}^A\}_{i=L,R}$ and $\{\mathcal{T}_{t,i}^B\}_{i=L,R}$ represent the starting and ending poses of the motion trajectory for the left and right gripper. We adopt $\{\mathcal{V}_{t,i}\}_{i=L,R}$ to represent the grasping vertex of the tie's mesh for the left and right gripper. The gripper can choose to grasp one vertex of the tie's mesh or choose to not grasp the tie (We add a NULL to represent this case). For the $\{\mathcal{V}_{t,i}\}_{i=L,R}$, the starting poses $\{\mathcal{T}_{t,i}^A\}_{i=L,R}$ need to enable the robot to grasp the vertices with closed finger tips and the gripper palms are collision-free (with the tie, the environment, and the robot).

1) Combined Model-free RL with Differentiable Simulation: For the action generation, our pipeline needs to consider 1) where to grasp? 2) where to pull?

WhereToPull. To pull the previous tie's shape $\mathcal{X}_{t-1}^S$ towards the subgoal $\mathcal{X}_t^G$, assume we have identified the grasping vertices on the tie $\{\mathcal{V}_{t,i}\}_{i=L,R}$ and have moved the grippers to starting poses $\{\mathcal{T}_{t,i}^A\}_{i=L,R}$ to grasp the tie. Our pipeline generates the feasible pulling trajectories for each gripper (satisfying kinematics constraints, collision-free) so that the tie moves towards the subgoal in the simulation.

In the differentiable cloth simulation, we formulate the pulling action generation as an optimization problem.

$$\begin{aligned} &\mathcal{L} = \|\mathcal{X}_N - \mathcal{X}_t^{\mathcal{G}}\|^2 \\ \text{s.t.}\quad &\mathcal{X}_{n+1} = \mathcal{F}^{sim}(\mathcal{X}_n, \mathcal{P}^{a_n}(\{\mathcal{V}_{t,i}\}_{i=L,R})) \\ &\mathcal{X}_0 = \mathcal{X}_{t-1}^S \\ &\{\mathcal{T}_{n,i}\}_{i=L,R} \in \mathcal{C} \\ &\{\mathcal{T}_{0,i}\}_{i=L,R} = \{\mathcal{T}_{t,i}^A\}_{i=L,R} \end{aligned} \tag{2}$$

Here $\mathcal{P}^{a_n}(\{\mathcal{V}_{t,i}\}_{i=L,R})$ denotes the action for *DiffClothAI*, which is the new positions of selected vertices to drag the ties towards the next shape. $\{\mathcal{T}_{n,i}\}_{i=L,R}$ are the gripper 6D poses in the world frame, which need to satisfy kinematic constraints and collision-free constraints $\{\mathcal{T}_{n,i}\}_{i=L,R} \in \mathcal{C}$. The optimized final tie's shape $\mathcal{X}_N$ is set to be the initialization of the next optimization. Due to the page limit, we put detailed descriptions to get the solution for this optimization problem leveraging the differentiable simulation in the supplementary.

Denote the above optimization process as a function $\mathscr{P}$, which tasks as input a initial mesh state $\mathcal{X}_{t-1}^S$ and a subgoal state $\mathcal{X}_t^G$, and returns the resulting mesh state in Eqn. 3.

$$\mathcal{X}_t^S = \mathscr{P}(\mathcal{X}_{t-1}^S, \mathcal{X}_t^{\mathcal{G}}, \{\mathcal{T}_{t,i}^A\}_{i=L,R}, \{\mathcal{V}_{t,i}\}_{i=L,R}) \tag{3}$$

WhereToGrasp. Given the previous ending shape of tie $\mathcal{X}_{t-1}^S$, the previous ending poses of the robot's grippers

$\{\mathcal{T}^B_{t-1,i}\}_{i=L,R}$, the subgoal $\mathcal{X}^{\mathcal{G}}_t$ for the current iteration, we need to figure out which vertices to grasp so that the robot can drag the whole mesh model to its target position accurately. We assume that grasping points cannot change once selected within one process from $t-1$ to t.

We formulate the grasping vertices selection sequences as a Markov Decision Processes (MDP). The state s contains the previous tie's vertices $\mathcal{X}^S_{t-1}$, the point-wise displacement for each tie vertices $\mathscr{D}_t = \mathcal{X}^{\mathcal{G}}_t - \mathcal{X}^S_{t-1}$, the previous ending poses of the robot's grippers $\{\mathcal{T}^B_{t-1,i}\}_{i=L,R}$. The action space is all the tie mesh's vertices for both gripper. The action for each gripper is a one dimension one-hot vector with a dimension of $(M+1)$, where M is the number of tie mesh's vertices. After selecting the grasping vertex for each gripper $\{\mathcal{V}_{t,i}\}_{i=L,R}$, we call inverse kinematic solvers to calculate the feasible and collision-free grasping poses $\{\mathcal{T}_{t,i}\}_{i=L,R}$. Note that we does not formulate the action space to be 6D pose for each gripper and there is feasible collision-free path to move grippers from $\{\mathcal{T}^B_{t-1,i}\}_{i=L,R}$ to $\{\mathcal{T}^A_{t,i}\}_{i=L,R}$. The reward function $\mathcal{R}$ as shown in equation 4.

$$\mathcal{R}(s,a) = \begin{cases} C_1, & \text{if knotting-tie succeeds} \\ -C_2, & \text{if fail to reach subgoal} \\ C_3 - \|\mathcal{X}^S_t - \mathcal{X}^{\mathcal{G}}_t\|, & \text{Otherwise} \end{cases} \tag{4}$$

Here C_1, C_2, C_3 are constant positive values. $\mathcal{X}^S_t$ is the result tie mesh of the pull function defined in Eqn. 3. The failure to reach subgoal is due to 1) the grasping point on the vertex, there is no collision-free grasping approach direction. 2) the distance $\|\mathcal{X}^S_t - \mathcal{X}^{\mathcal{G}}_t\|$ is larger than a given threshold, the tie could not be pulled close to the subgoal by grasping on the selected index.

We apply popular model-free reinforcement algorithms to learn where to grasp by taking the whole sequences into consideration. We put the detailed explanation, including multiple feasible 6D poses associated with one grasping index, on the website due to page limit.

2) Imitation Learning: After returning the action sequences to accomplish the knotting-tie task, we denote the associated tie shape sequences $\{\mathcal{X}^{S*}_t\}$, the selected grasping vertices $\{\{\mathcal{V}^*_{t,i}\}_{i=L,R}\}$, the robot motion trajectories $\{\{\mathcal{T}^{A*}_{t,i}, \mathcal{T}^{B*}_{t,i}\}_{i=L,R}\}$.

We train an actor π^{sim} that takes as inputs point clouds of the tie's mesh $\mathcal{X}_t$ to predict the grasping vertices and the associated robot motion trajectories.

$$a^{sim}_t = \{\mathcal{T}^{A,sim}_{t,i}, \mathcal{T}^{B,sim}_{t,i}\}_{i=L,R} = \pi^{sim}(\mathcal{X}_t) \tag{5}$$

We add perturbations to the tie's shape and update the associated robot motion trajectories to generate large-scale ground-truth annotations in the simulation. We also slightly adjust the tie's mesh size and mesh shape. So that the actor network can generalize to point clouds from similar but different tie's shapes. Due to page limit, we put detailed description on the website.

D. Real2Sim

When the robot knots the tie in the real world, the robot receives a segmented point cloud of the tie denoted as $\mathcal{X}^{real}_t$ at each time step. Instead of directly applying the action $\pi^{sim}(\mathcal{X}^{real}_t)$ in the real world, we train a residual policy that takes in the real point cloud $\mathcal{X}^{real}_t$ and the $\pi^{sim}(\mathcal{X}^{real}_t)$, outputs the residual action δa^{real}. Here δa^{real} represents 6D offset to the $\{\mathcal{T}^A_{t,i}, \mathcal{T}^B_{t,i}\}_{i=L,R}$.

$$\delta a^{real} = \pi^{res}(\mathcal{X}^{real}, a^{sim}) \tag{6}$$

Then actor model in the real world gets the action as follows.

$$a^{real} = \pi^{real}(\mathcal{I}^{real}, a^{sim}) = a^{sim} + \delta a^{real} \tag{7}$$

We then run the motion planner to generate the motion trajectories from the starting poses to the ending poses. We follow the training process of residual policy described in [60]. Details are reported on the website due to page limit.

IV. Experiments

This section focuses on evaluating the following questions: 1) How effective are our proposed iterative learning keypoints and hierarchical matching in Sec.III-B? 2) How does our action generation approach in Sec. III-C compare to model-free and model-based deep reinforcement learning algorithms? (See. website) 3) How effective is our proposed model-based LfD compared to other LfD approaches? (See. website) 4) Can our approach generalize to other deformable manipulation tasks? 5) How effective is the residual policy on real robot executions?

A. Experimental Setup

We set up the real world experiment with a dual-arm robot as shown in Fig. 1. The MOVO robot [61] has two 7 DoF arms and a Kinect RGB-D camera over head. We perform position controls and use rangeIK [62] for solving inverse kinematics. We record two types of knotting tie videos and one folding task video as shown in Fig. 6 and Fig. 7.

B. Iteratively Learning Keypoints

To demonstrate the effectiveness of our proposed iteratively learning keypoint, we collect data in the simulation by applying random action to the tie starting from the initial shape of the tie. We automatically annotate the keypoint position and train the same keypoint neural network described in Sec. III-B.2. We report our proposed result **Ours** and compare it with random sample approach **RS** in Fig. 5. More results are on the website. **Ours** can produce effective keypoints while **RS** fails due to out-of-distribution issue. We also test the **Ours** and **RS** on 90 point clouds produced from the simulation with ground-truth annotation. The average keypoint prediction errors of **Ours** and **RS** are 0.028 and 0.183 meter, respectively.

C. Hierarchical Matching

We conduct ablation experiments to illustrate the importance of each component in our hierarchical matching with results shown in Fig. 9. Given two frames, we show the feature matching and keypoint prediction results. Due to the large deformation and occlusion, the quality of feature

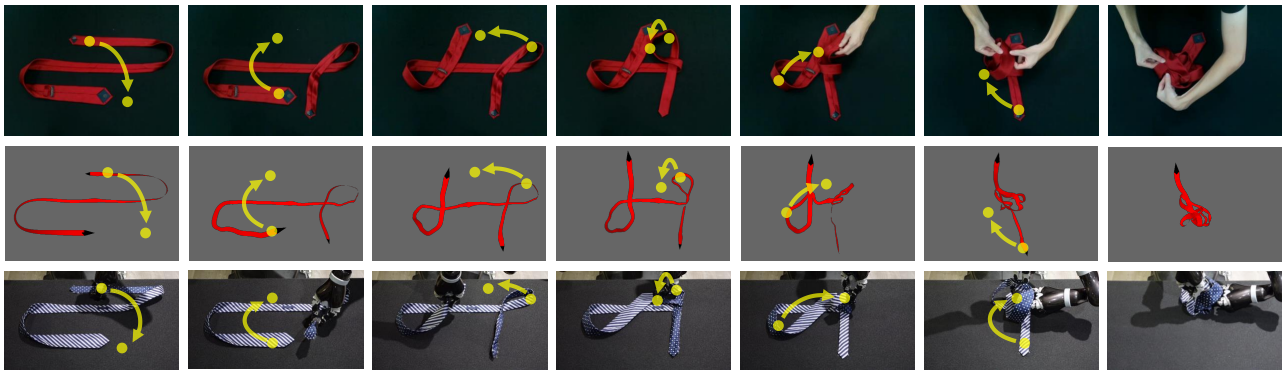

Fig. 6. The visualization of the tie-knoting process. The fist column is the human demonstration, the second column is the estimated states in simulation, and the third column is the robot manipulation process. To better visualize, we show the manipulation action in yellow dots and arrows.

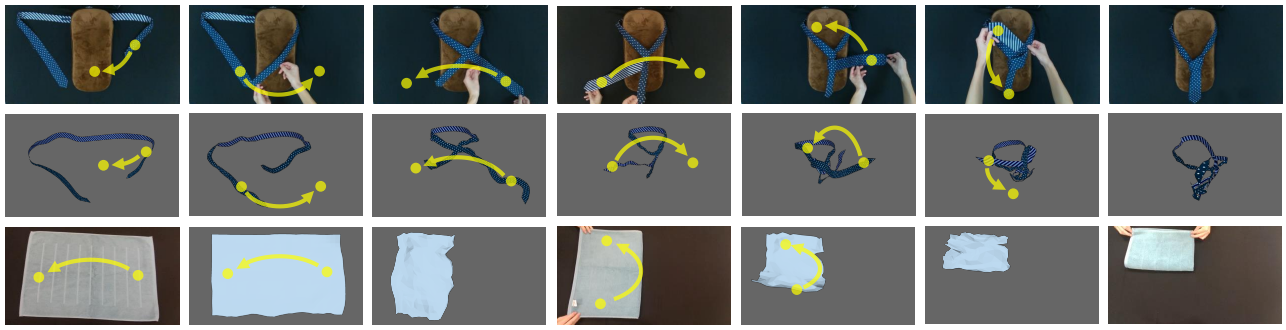

Fig. 7. The visualization of a differentway to knot a tie and the towel folding. The fist column is the human demonstration of tie-knotting, the second column is the estimated states in simulation. While the third column is showing the towel folding. To better visualize, we show the manipulation action in yellow dots and arrows.

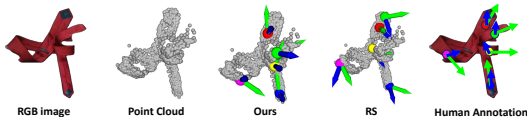

Fig. 8. Comparison between the keypoint network trained with random sample approach **RS** and our iteratively sample approach **Ours**. The blue arrow represents z direction and the green arrow represents x direction.

matching is affected but the keypoint predictions are informative since the global structure contributes the keypoint predictions. We illustrate the fitting results using our proposed approach denoted as **Ours** and perform ablation experiments by removing the keypoint (**w/o KP**), the local frame of the keypoint(**w/o LF**), and the feature matching (**w/o FM**). All the ablation experiments degenerate indicating the importance of these components. We also compare the matching method including using the Chamfer Distance or non-rigid registration methods [5]. More results are available on the website.

Fig. 9. Visualization of ablation experiments. **Ours** achieves the best fitting result. More examples are put on on the website, especially for more comparison between **Ours** and **Ours w/o FM**

D. Residual Policy

The robot fails to accomplish the tie-knotting task if no residual policy is adopted. As shown in Fig. 10, compared to the leftmost image, the real action (middle image) adapts to generate large circle (see red circle) for the later step (rightmost image).

Fig. 10. Comparison between with and without residual policy

E. Generalization to other tasks

We apply our proposed framework to other tasks. One is a different way to knot a tie. The other one is to learn to fold a towel. We visualize the process in Fig. 7. The whole video sequence is on the website.

V. Conclusion

This work introduces a model-based framework called *TieBot* for the robots to learn to knot a tie from visual demonstration. *TieBot* introduces the iterative keypoint learning and the hierarchical matching approach to estimate the tie's shape from the demonstrated video. *TieBot* combines model-free RL and differentiable simulation to generate goal-conditioned action sequences. Then, an imitation learning approach is used to learn the action sequences from raw point clouds in the simulation. Lastly, our pipeline learns a residual policy when the imitated policy is applied to real-world execution, mitigating the sim2real gap. We demonstrate that a dual-arm robot successfully knots the tie with our framework.

REFERENCES

[1] X. Yu, S. Zhao, S. Luo, G. Yang, and L. Shao, "Diffclothai: Differentiable cloth simulation with intersection-free frictional contact and differentiable two-way coupling with articulated rigid bodies," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023.

[2] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4568–4575.

[3] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018. [Online]. Available: https://doi.org/10.1177/0278364918779698

[4] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li *et al.*, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022.

[5] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3426–3433, 2018.

[6] T. Morita, J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi, "Knot planning from observation," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 3, 2003, pp. 3887–3892 vol.3.

[7] M. Yan, G. Li, Y. Zhu, and J. Bohg, "Learning topological motion primitives for knot planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9457–9464.

[8] T. Tang, Y. Fan, H.-C. Lin, and M. Tomizuka, "State estimation for deformable objects by point registration and dynamic simulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2427–2433.

[9] H. Ha and S. Song, "Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *Conference on Robotic Learning (CoRL)*, 2021.

[10] C. Bersch, B. Pitzer, and S. Kammel, "Bimanual robotic cloth manipulation for laundry folding," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1413–1419.

[11] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 249–267, 2012.

[12] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, "Folding deformable objects using predictive simulation and trajectory optimization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 6000–6006.

[13] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrik, A. Kargakos, L. Wagner, V. Hlaváč, T.-K. Kim, and S. Malassiotis, "Folding clothes autonomously: A complete pipeline," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1461–1478, 2016.

[14] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, "Folding deformable objects using predictive simulation and trajectory optimization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 6000–6006.

[15] T. Weng, S. M. Bajracharya, Y. Wang, K. Agrawal, and D. Held, "Fabricflownet: Bimanual cloth manipulation with a flow-based policy," in *Conference on Robot Learning*. PMLR, 2022, pp. 192–202.

[16] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for multi-step, multi-task fabric manipulation," *arXiv preprint arXiv:2003.09044*, 2020.

[17] D. Tanaka, S. Arnold, and K. Yamazaki, "Emd net: An encode–manipulate–decode network for cloth manipulation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1771–1778, 2018.

[18] D. Tanaka, S. Tsuda, and K. Yamazaki, "A learning method of dual-arm manipulation for cloth folding using physics simulator," in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2019, pp. 756–762.

[19] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, "Speedfolding: Learning efficient bimanual folding of garments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1–8.

[20] H. chaandar Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annu. Rev. Control. Robotics Auton. Syst.*, vol. 3, pp. 297–330, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:208958394

[21] J. Schulman, J. Ho, C. Lee, and P. Abbeel, *Learning from Demonstrations Through the Use of Non-rigid Registration*. Cham: Springer International Publishing, 2016, pp. 339–354. [Online]. Available: https://doi.org/10.1007/978-3-319-28872-7_20

[22] J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick, and P. Abbeel, "A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4111–4117.

[23] A. X. Lee, S. H. Huang, D. Hadfield-Menell, E. Tzeng, and P. Abbeel, "Unifying scene registration and trajectory optimization for learning from demonstrations with application to manipulation of deformable objects," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4402–4407.

[24] A. X. Lee, M. A. Goldstein, S. T. Barratt, and P. Abbeel, "A non-rigid point and normal registration algorithm with applications to learning from demonstrations," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 935–942.

[25] A. X. Lee, A. Gupta, H. Lu, S. Levine, and P. Abbeel, "Learning from multiple demonstrations using trajectory-aware non-rigid registration with applications to deformable object manipulation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5265–5272.

[26] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 177–184.

[27] D. Seita, N. Jamali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, "Deep transfer learning of pick points on fabric for robot bed-making," in *Robotics Research: The 19th International Symposium ISRR*. Springer, 2022, pp. 275–290.

[28] R. P. Joshi, N. Koganti, and T. Shibata, "Robotic cloth manipulation for clothing assistance task using dynamic movement primitives," in *Proceedings of the Advances in Robotics*, ser. AIR '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3132446.3134878

[29] G. Salhotra, I.-C. A. Liu, M. Dominguez-Kuhne, and G. S. Sukhatme, "Learning deformable object manipulation from expert demonstrations," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8775–8782, 2022.

[30] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen,

A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9651–9658.

[31] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[32] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[33] 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 *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/842424a1d0595b76ec4fa03c46e8d755-Paper.pdf

[34] J. Degrave, M. Hermans, J. Dambre *et al.*, "A differentiable physics engine for deep learning in robotics," *Frontiers in neurorobotics*, p. 6, 2019.

[35] G. Yang, S. Luo, and L. Shao, "Jade: A differentiable physics engine for articulated rigid bodies with intersection-free frictional contact," *arXiv preprint arXiv:2309.04710*, 2023.

[36] 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)*, vol. 38, no. 6, p. 201, 2019.

[37] Y. Hu, J. Liu, A. Spielberg, J. B. Tenenbaum, W. T. Freeman, J. Wu, D. Rus, and W. Matusik, "Chainqueen: A real-time differentiable physical simulator for soft robotics," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6265–6271.

[38] 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.

[39] 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)*, vol. 39, no. 6, pp. 1–15, 2020.

[40] T. Du, K. Wu, P. Ma, S. Wah, A. Spielberg, D. Rus, and W. Matusik, "Diffpd: Differentiable projective dynamics," *ACM Trans. Graph.*, vol. 41, no. 2, nov 2021. [Online]. Available: https://doi.org/10.1145/3490168

[41] J. Liang, M. Lin, and V. Koltun, "Differentiable cloth simulation for inverse problems," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/28f0b864598a1291557bed248a998d4e-Paper.pdf

[42] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin, "Scalable differentiable physics for learning and control," *arXiv preprint arXiv:2007.02168*, 2020.

[43] Y. Li, T. Du, K. Wu, J. Xu, and W. Matusik, "Diffcloth: Differentiable cloth simulation with dry frictional contact," *ACM Transactions on Graphics (TOG)*, 2022.

[44] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu, "Fast and feature-complete differentiable physics for articulated rigid bodies with contact," *arXiv preprint arXiv:2103.16021*, 2021.

[45] 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 *Robotics*, ser. Robotics: Science and Systems, S. Srinivasa, N. Ayanian, N. Amato, and S. Kuindersma, Eds. United States: MIT Press Journals, 2017, 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.

[46] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin, "Efficient differentiable simulation of articulated bodies," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8661–8671.

[47] 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*, vol. 33, pp. 6111–6122, 2020.

[48] 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.

[49] P. Holl, V. Koltun, and N. Thuerey, "Learning to control pdes with differentiable physics," *arXiv preprint arXiv:2001.07457*, 2020.

[50] 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*, vol. 35, no. 7, pp. 6138–6146, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16764

[51] 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," 2023.

[52] 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.

[53] K. M. Jatavallabhula, M. Macklin, F. Golemo, V. Voleti, L. Petrini, M. Weiss, B. Considine, 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. [Online]. Available: https://openreview.net/forum?id=c_E8kFWfhp0

[54] P. Ma, T. Du, J. B. Tenenbaum, W. Matusik, and C. Gan, "Risp: Rendering-invariant state predictor with differentiable simulation and rendering for cross-domain parameter estimation," in *International Conference on Learning Representations*, 2021.

[55] X. Li, Y.-L. Qiao, P. Y. Chen, K. M. Jatavallabhula, M. Lin, C. Jiang, and C. Gan, "PAC-neRF: Physics augmented continuum neural radiance fields for geometry-agnostic system identification," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=tVkrbkz42vc

[56] X. Zhu, J. Ke, Z. Xu, Z. Sun, B. Bai, J. Lv, Q. Liu, Y. Zeng, Q. Ye, C. Lu, M. Tomizuka, and L. Shao, "Diff-lfd: Contact-aware model-based learning from visual demonstration for robotic manipulation via differentiable physics-based simulation and rendering," in *Conference on Robot Learning (CoRL)*, 2023.

[57] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng, "Track anything: Segment anything meets videos," 2023.

[58] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision*

and pattern recognition, 2017, pp. 605–613.
[59] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," *CVPR*, 2021.
[60] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling, "Residual policy learning," *arXiv preprint arXiv:1812.06298*, 2018.
[61] Kinova, "Kinova-movo." [Online]. Available: https://docs.kinovarobotics.com/kinova-movo/Concepts/c_movo_hardware_overview.html
[62] Y. Wang, P. Praveena, D. Rakita, and M. Gleicher, "Rangedik: An optimization-based robot motion generation method for ranged-goal tasks," *arXiv preprint arXiv:2302.13935*, 2023.