2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

SOFTMIMICGEN: A Data Generation System for Scalable Robot Learning in Deformable Object Manipulation

Masoud Moghani[1,2], Mahdi Azizian[1], Animesh Garg[3], Yuke Zhu[1], Sean Huver*,[1], Ajay Mandlekar*,[1]

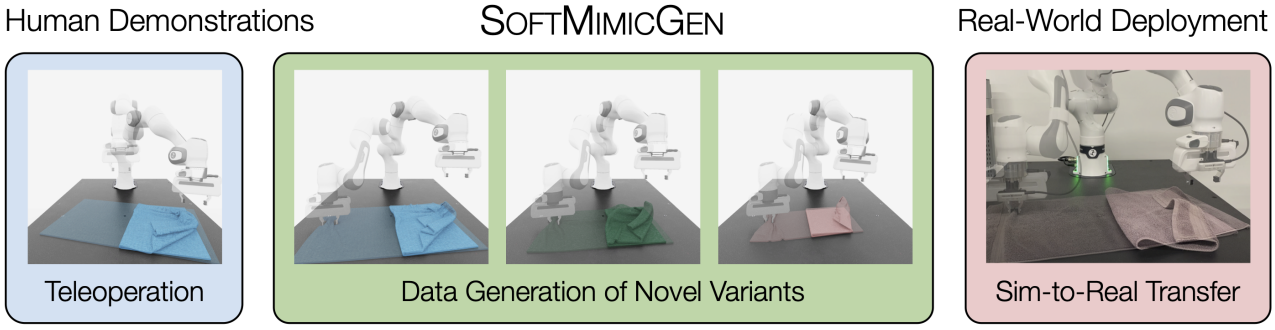

Fig. 1: SOFTMIMICGEN Overview. SOFTMIMICGEN provides an efficient pipeline for synthesizing robot trajectories for deformable object manipulation. (Left) A human teleoperator first collects a small set of fine-grained, dexterous robot trajectories. (Center) SOFTMIMICGEN then generates large-scale datasets for novel instances of deformable objects across new contexts. (Right) Generated demonstrations enable the training of high-performing policies in simulation, which can then be transferred to real-world platforms. SOFTMIMICGEN is compatible with diverse robot embodiments and tasks that require dynamic, contact-rich manipulation.

***Abstract*— Large-scale robot datasets have facilitated the learning of a wide range of robot manipulation skills, but these datasets remain difficult to collect and scale further, owing to the intractable amount of human time, effort, and cost required. Simulation and synthetic data generation have proven to be an effective alternative to fuel this need for data, especially with the advent of recent work showing that such synthetic datasets can dramatically reduce real-world data requirements and facilitate generalization to novel scenarios unseen in real-world demonstrations. However, this paradigm has been limited to rigid-body tasks, which are easy to simulate. Deformable object manipulation encompasses a large portion of real-world manipulation and remains a crucial gap to address towards increasing adoption of the synthetic simulation data paradigm. In this paper, we introduce SOFTMIMICGEN, an automated data generation pipeline for deformable object manipulation tasks. We introduce a suite of high-fidelity simulation environments that encompasses a wide range of deformable objects (stuffed animal, rope, tissue, towel) and manipulation behaviors (high-precision threading, dynamic whipping, folding, pick-and-place), across four robot embodiments: a single-arm manipulator, bimanual arms, a humanoid, and a surgical robot. We apply SOFTMIMICGEN to generate datasets across the task suite, train high-performing policies from the data, and systematically analyze the data generation system. Project website: softmimicgen.github.io.**

I. INTRODUCTION

Robot foundation models [1]–[4], trained on a combination of web-scale vision-language data and large-scale robot manipulation datasets, have shown an impressive capability to perform a wide range of complex manipulation tasks autonomously. However, these advances have been fueled chiefly by the availability of large robot manipulation datasets. These datasets are often collected via robot teleoperation by large teams of human operators over several time-consuming months [5]–[8]. While the robot teleoperation paradigm provides a simple means towards collecting robotics datasets, it remains a costly and labor-intensive endeavor, and hinders the broader development of robot foundation models.

Simulation is a promising alternative to fuel the need for large robot manipulation datasets, especially due to several recent developments. Designing high-quality simulation environments is becoming easier, due to the availability of high-fidelity physics simulators and photorealistic rendering [9], [10], and the advent of generative AI tools, which facilitate the automated generation of scenes, assets, and tasks [11], [12]. Recent automated data generation tools make it possible to synthesize large amounts of diverse, high-quality robot manipulation demonstrations with little human effort [13]–[16]. Furthermore, recent work highlights that large-scale synthetic simulation datasets can easily be used to train high-performance real-world manipulation policies by *co-training* on these synthetic simulation datasets and small amounts of real-world data [4], [17], [18]. This synthetic data generation paradigm can drastically reduce real-world data requirements and facilitate generalization to novel scenarios unseen in the real-world datasets. However, this paradigm has been useful only for tasks that can be easily simulated, which has limited its application to mostly rigid-body manipulation tasks.

Deformable object manipulation encompasses a significant portion of real-world manipulation and remains a crucial gap towards increasing the adoption of simulation tools and synthetic data generation. However, overcoming this gap is challenging for several reasons. First, simulating deformable objects is a difficult problem computationally, and getting

[1]NVIDIA, [2]University of Toronto, [3]Georgia Institute of Technology
*Equal Advising, Correspondence to: moghani@cs.toronto.edu

interactions with such objects to simulate in real-time (or faster) is even more challenging. Sourcing and annotating simulation assets for deformable manipulation tasks such that they behave as anticipated is also non-trivial. Second, common synthetic data generation solutions assume that objects can be assigned a rigid frame of reference, and exploit robot manipulation motion invariance with respect to this frame in order to generate new demonstrations [14]–[16]. However, this assumption breaks down quickly for deformable objects, since they do not remain rigid, requiring new algorithmic considerations for data generation [19]–[21].

Towards addressing these challenges, we create a simulation suite of deformable object manipulation tasks, and develop SOFTMIMICGEN, a synthetic data generation pipeline for deformable object manipulation. The simulation suite leverages recent advancements in deformable object simulation, ensuring all environments simulate in real-time (or faster). SOFTMIMICGEN builds upon MIMICGEN [14], which is an object-centric trajectory generation system for rigid object manipulation. Like MIMICGEN, SOFTMIMICGEN starts with a small source set of human demonstrations (1 to 10) and generates much larger datasets automatically. The key component of MIMICGEN is to extract and replay object-centric demonstrations from the source dataset by leveraging a static object reference frame that remains constant between the source dataset and the new task instance. However, no such static reference frame is guaranteed to exist for deformable objects – consequently, SOFTMIMICGEN instead leverages non-rigid registration techniques [19], [20] to adaptively transform source demonstrations while accounting for the changed state of deformable objects. This results in a more capable demonstration generation mechanism that can be applied to deformable and rigid object tasks alike. We train visuomotor policies via imitation learning on generated data, enabling direct manipulation of deformable objects from images without explicit registration at inference time.

Summary of contributions:

- SOFTMIMICGEN enables synthetic data generation for deformable object manipulation tasks.
- We release a suite of high-fidelity simulation environments that encompass a range of deformable objects (stuffed animal, rope, tissue, towel) and manipulation behaviors (high-precision threading, dynamic whipping, folding, pick-and-place) across four robot embodiments (Franka and YAM robot arms, humanoid, surgical robot).
- We apply SOFTMIMICGEN to generate thousands of demonstrations per task, train high-performing policies from the data, and provide a systematic analysis of the data generation system.
- We demonstrate real-world deployment across diverse deformable manipulation tasks, where policies trained on SOFTMIMICGEN-generated data achieve zero-shot sim-to-real transfer and are further improved via sim-real co-training.

II. RELATED WORK

Deformable Object Manipulation. Research in the robotic manipulation of deformable objects, including cloth, granular media, and soft materials, has pivoted from classical physics-based models to data-driven, learning-centric paradigms [22], [23]. Multiple methods construct explicit simulations using pre-scanned static objects and point cloud observations. Most recent approaches build upon SDFs [24], NeRF [25], or Gaussian Splatting [26] to support flexible physical digital twin creation. Furthermore, neural methods of dynamics learning using graph-based representations have been used to learn the dynamics of various types of deformable objects such as plasticine [27], cloth [28], and fluid [29]. Yet, studying deformable object manipulation in the context of large-scale imitation learning and foundation models remains a challenge due to the scarcity of open-source simulation environments and datasets. SOFTMIMICGEN is a first step towards enabling this kind of investigation.

Data Collection and Data Generation for Robotics. Robot teleoperation [30], [31] is a popular option for collecting demonstrations to train robots – humans use a teleoperation device (such as a smartphone or VR controller) to control a robot and perform different manipulation tasks. The robot sensor streams and controller actions are recorded into a dataset. This paradigm has been scaled up extensively in recent years through the use of teams of human operators and robots over extended periods of time [5]–[8]. Other works have used pre-programmed demonstrators [11], [13], [32], [33], but scaling these approaches to a larger variety of tasks can be difficult. The size of collected datasets can be increased using offline data augmentation [34]–[36]. These include the use of generative models to augment observations [37], [38], and leveraging counterfactual reasoning to augment observation-action pairs [39], [40].

MIMICGEN [14] is a data generation framework that exploits object-centric invariance to generate new trajectories. DexMimicGen [15] extended this approach to bimanual manipulation, and SkillMimicGen [16] extended the approach to incorporate motion planning (complementary to our approach). SOFTMIMICGEN, like MIMICGEN, generates new datasets online, but can be applied to a much broader set of tasks by using an improved trajectory transformation process based on non-rigid registration techniques.

Learning Manipulation from Human Demonstrations. Behavioral Cloning (BC) [41] is a widely used approach for learning robot manipulation policies from demonstrations [36], [42]–[44]. It consists of training the agent to produce actions that are consistent with observation-action pairs in the training dataset. This method has been shown to be extremely effective for robot manipulation [1], [3], [8], [33], [45], [46], but the quality of the results depends on the availability of large-scale, high-quality manipulation datasets. Some recent works [4], [17], [18] have shown that synthetic simulation data can supplement real-world datasets and reduce the amount of costly real-world data that is required.

III. PREREQUISITES

A. Behavioral Cloning

We model each manipulation task as a Partially Observable Markov Decision Process (POMDP). We are given a dataset of

N demonstrations $\mathcal{D} = \{(s_0^i, o_0^i, a_0^i, s_1^i, o_1^i, a_1^i, ..., s_{H_i}^i)\}_{i=1}^N$ with states $s \in \mathcal{S}$, observations $o \in \mathcal{O}$, and actions $a \in \mathcal{A}$. Each episode starts in an initial state $s_0^i \sim D$ sampled from the initial state distribution $D \subseteq \mathcal{S}$. The goal is to learn a policy $\pi : \mathcal{O} \rightarrow \mathcal{A}$ that takes observations as inputs and outputs a distribution over the action space. Policies are trained using Behavioral Cloning [41] via the maximum likelihood objective $\arg\max_\theta \mathbb{E}_{(s,o,a)\sim\mathcal{D}}[\log \pi_\theta(a \mid o)]$ using datasets generated by SOFTMIMICGEN.

B. Problem Statement

We are given a source dataset $\mathcal{D}_{\text{src}}$ consisting of a small (typically 1 to 10) number of human demonstrations, and our goal is to use it to generate a large dataset of demonstrations $\mathcal{D}$ on either the same task, or a task with a different initial state distribution $D' \subseteq \mathcal{S}$ (typically one with a larger set of possible placements for objects in the scene). To generate a new demonstration, (1) a start state is sampled from D', (2) one or more demonstrations $\tau \in \mathcal{D}_{\text{src}}$ are selected and adapted to produce and execute a new robot trajectory τ', and (3) if the task is completed successfully, the trajectory is added to the generated dataset. The core problem that must be addressed is the mechanism used to carry out step (2) – namely, the trajectory generation.

C. Assumptions

We make the following assumption on how deformable objects are represented: **(A0):** every deformable object is represented as a collection of 3-dimensional node positions, $O = \{\mathbf{n_i}\}_{i=1}^{N_O}$, where $\mathbf{n_i} \in \mathbb{R}^3$ are 3-dimensional positions (x_i, y_i, z_i), and N_O is the number of nodes for the object. These node positions are typically obtained through a simulator's soft object solver. Note that this is equivalent to a point cloud representation for objects, and rigid-body objects could also be represented in this manner.

We also make additional assumptions, similar to MIMICGEN [14], namely: **(A1):** The action space $\mathcal{A}$ consists of pose commands for an end-effector controller for each robot arm, and a gripper command for each arm. **(A2):** Each task can be divided into sequences of object-centric subtasks $(S_1(o_{S_1}), S_2(o_{S_2}), ..., S_M(o_{S_M}))$ for each arm, where the arm primarily interacts with one object. However, unlike MIMICGEN, the manipulation need not be relative to a specific object coordinate frame, which can be ill-posed for deformable objects. **(A3):** During data collection, we assume that object configurations can be observed or estimated prior to the start of each subtask. For rigid objects, this corresponds to the object pose; for deformable objects, it corresponds to the positions of all object nodes (see **A0**).

D. MIMICGEN

MIMICGEN first parses the source demonstrations in $\mathcal{D}_{\text{src}}$ into contiguous object-centric manipulation segments $\{\tau_i\}_{i=1}^M$, each of which corresponds to a subtask $S_i(o_i)$. Each of these segments is a sequence of end-effector control poses $\tau_i = (T_W^{C_0}, T_W^{C_1}, ..., T_W^{C_K})$ where W is the world reference frame. The segmentation can be done autonomously with heuristics or using human annotation. To generate a demonstration in a novel scene, it first observes the pose of the object for the current subtask $T_W^{o_i'}$. It then transforms the poses in a source human segment such that the relative poses between the end-effector frame and object frame are preserved in both the source segment and the new scene – this is the invariance that MIMICGEN exploits. MIMICGEN carries out this transformation using a constant SE(3) transform $T_W^{o_i'}(T_W^{o_i})^{-1}$. MIMICGEN then adds poses to the start of the transformed segment in order to linearly interpolate between the robot's current pose and the start of the transformed segment (which may start far from the current pose). It then executes the sequence of poses in the segment and repeats the process for the next subtask until all subtasks have been executed, and keeps the executed trajectory as a demonstration if it was successful. Unlike MIMICGEN, SOFTMIMICGEN must deal with deformable objects with nonlinear material properties and elasticity – consequently, it is not straightforward to use the same rigid SE(3) transformation strategy to exploit object-centric invariance. SOFTMIMICGEN overcomes this challenge by estimating non-rigid transformations between deformable object states (expressed as collections of node positions, see Assumption **A0** in Sec. III-C) and leveraging these transformations to perform non-rigid spatial adaptation of trajectory segments.

IV. SOFTMIMICGEN

SOFTMIMICGEN enables large-scale data generation for deformable object manipulation using only a small number of human teleoperated demonstrations. As described in Sec. III-B, to generate a new demonstration, reference source demonstrations must be selected, and adapted appropriately. As in MIMICGEN, we seek to exploit an object-centric invariance to re-purpose existing source human demonstrations and generate new trajectories, but doing this for deformable objects is not as straightforward. We first describe how deformable objects are represented and why the MIMICGEN strategy does not suffice (Sec. IV-A). Next, we describe how non-rigid registration can offer a solution to the problem of object-centric trajectory transfer for deformable objects (Sec. IV-B). Finally, we describe how we incorporate this mechanism in the data generation process (Sec. IV-C).

A. Deformable Object Representation

We assume that every deformable object is described by a collection of 3-dimensional node positions, $O = \{\mathbf{n_i}\}_{i=1}^{N_O}$, where $\mathbf{n_i} \in \mathbb{R}^3$ are 3-dimensional positions (x_i, y_i, z_i), and where N_O is the number of nodes for the object (Assumption **A0**, Sec. III-C). The state of the deformable object is fully described by the set of positions. By contrast, the trajectory transformation strategy employed by MIMICGEN assumes that there is a single canonical coordinate frame for each object, and each object is fully described by the pose (position and rotation) of that coordinate frame. Such a coordinate frame is ill-defined for deformable objects, and even if such a frame could be assigned (for example to a particular node), it would not fully describe the state of the deformable object, and

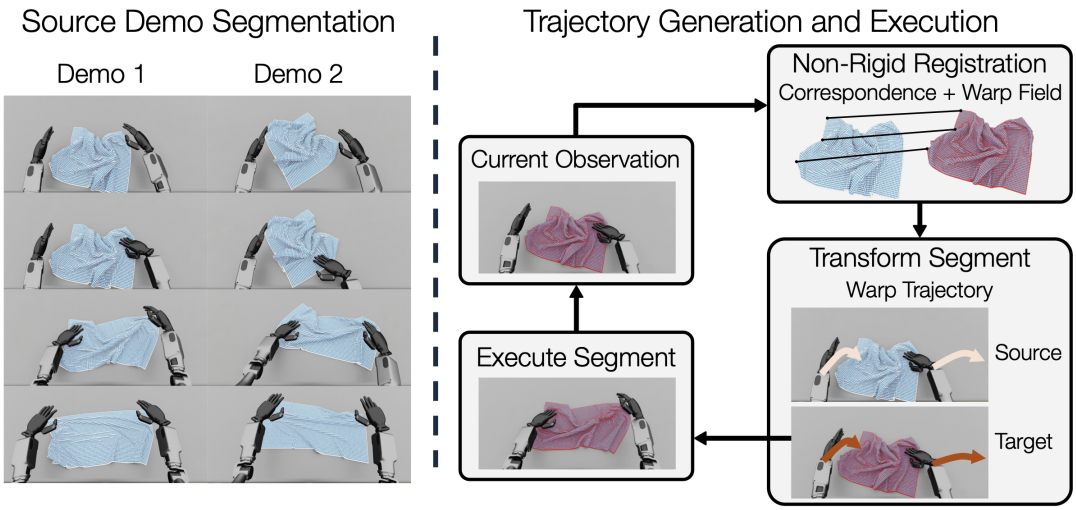

Fig. 2: SOFTMIMICGEN System Pipeline. *(Left)* Human-teleoperated demonstrations are segmented into object-centric subtasks using manual annotations or heuristic signals, forming a library of source segments. *(Right)* Given a new target scene, SOFTMIMICGEN *(i)* observes the current deformable state, *(ii)* performs non-rigid registration to establish correspondence and a warp field between the source and target geometries (*correspondence + warp field*), *(iii)* selects the source segment with the lowest registration cost and applies the resulting warp field to the end-effector trajectory (*warp trajectory; source end-effector trajectory in light arrows, warped trajectory in darker arrows*), and *(iv)* executes the warped trajectory in the target environment.

would violate the invariance assumption made by MIMICGEN. Unlike rigid bodies, deformable objects exhibit continuous and high-dimensional configurations characterized by local deformations and complex material properties. As a result, they require more expressive, nonlinear models that can capture dynamic shape changes. In the following section, we discuss non-rigid registration, which provides a means to compare the configurations of deformable objects, and forms the basis for SOFTMIMICGEN's data generation strategy.

B. Non-Rigid Registration

Consider a deformable object in two different configurations $O_1 = \{\mathbf{a_i}\}_{i=1}^{N}$ and $O_2 = \{\mathbf{b_i}\}_{i=1}^{N}$, where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$ are position vectors. For example, this could be a towel that is crumpled in two different ways. Non-rigid registration finds a smooth function $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^3$ that maps points from the first configuration to the second, see Fig. 2 (Non-Rigid Registration). It does this by solving an optimization problem that minimizes a cost consisting of the point-wise distances between $\mathbf{f}(\mathbf{a_i})$ and $\mathbf{b_i}$ and a regularization term to encourage smoothness (see [19] for more details). Note that in general, non-rigid registration does not require the number of points in each configuration to be equal, nor does it require point-wise correspondences to be known beforehand [19], [47]. In the next section, we describe how non-rigid registration can be used in the data generation process – namely how the costs of the non-rigid registration optimization can be used for source demonstration selection, and how the resultant continuous deformation field $\mathbf{f}(\cdot)$ can be used to warp source demonstrations for data generation.

C. Data Generation Mechanism

We now describe how we can leverage non-rigid registration during data generation. As described in Sec. III-B, the main step during data generation is to select source demonstrations $\tau \in \mathcal{D}_{\text{src}}$ and adapt them to produce and execute a new robot trajectory τ'.

Like MIMICGEN, we start with source demonstrations $\mathcal{D}_{\text{src}}$ that are parsed into contiguous object-centric manipulation segments $\{\tau_i\}_{i=1}^{M}$, each of which corresponds to a subtask $S_i(o_i)$. Recall that each segment is a sequence of end-effector control poses $\tau_i = (T_W^{C_0}, T_W^{C_1}, ..., T_W^{C_K})$ where W is the world reference frame, and that this segmentation can be done autonomously with heuristics or using human annotation [14]. To generate a demonstration in a novel scene, we proceed subtask by subtask.

Source Demonstration Selection. For each subtask, we must first select a source demonstration segment from the set of segments for the current subtask. To do so, we observe the object configuration $O_i' = \{\mathbf{v_j}\}_{j=1}^{N_{O_i}}$ corresponding to the object for the current subtask. Next, we compare it against the relevant object configurations at the start of each source segment by running non-rigid registration and comparing the cost achieved by each optimization problem. This is an analog to the nearest-neighbor source segment selection strategy from MIMICGEN [14], which selects source demos based on object pose distances between the new scene and the source demonstrations. Prior work [12], [14] has found that intelligent source demonstration selection can be crucial for improving data generation quality – this problem is exacerbated for deformable objects, since they have continuous, high-dimensional configuration spaces.

Trajectory Adaptation. Next, the selected source demonstration segment $\tau_i = (T_W^{C_0}, T_W^{C_1}, ..., T_W^{C_K})$ must be adapted to the current scene. To do so, we run non-rigid registration between the object configuration O_i at the start of the source demonstration segment and the one in the current scene O_i' and obtain a continuous deformation field $\mathbf{f}(\cdot)$. Next, each pose $T_t = (p_t, R_t)$ in the source segment can be transformed

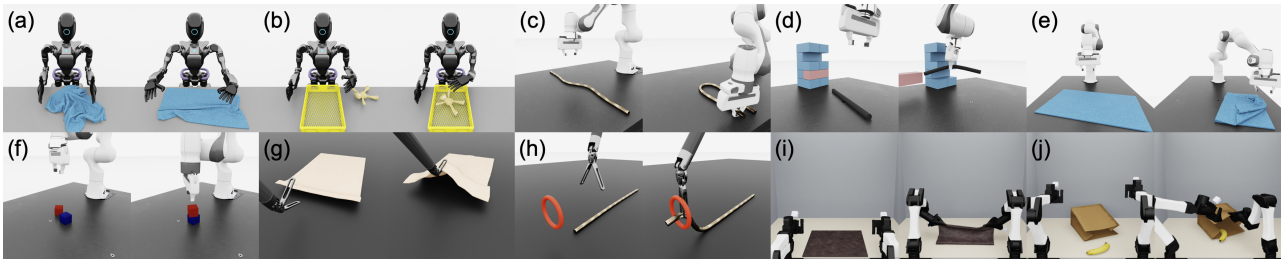

Fig. 3: Simulation Tasks. SOFTMIMICGEN is used to generate new demonstrations across 10 challenging tasks involving 4 distinct robot embodiments: (a-b) GR1 humanoid, (c-f) Franka arm, (g-h) dVRK surgical robot, and (i-j) bimanual YAM arms. These tasks demand high precision and fine-grained manipulation to be successfully executed.

using the field [19]:

$$p_t \to \mathbf{f}(p_t), \quad R_t \to \mathrm{orth}(\mathbf{J}_{\mathbf{f}}(p_t) R_t)$$

where $\mathbf{J}_{\mathbf{f}}(p_t)$ is the Jacobian of $\mathbf{f}$ evaluated at p_t, and $\mathrm{orth}(\cdot)$ orthonormalizes the resulting matrix to yield a valid rotation. This transformation preserves the local spatial relationship between the end-effector and the deformable object as it deforms, Fig. 2 (Transform Segment). As in MIMICGEN, a linear interpolation segment is added to the transformed trajectory segment τ' to ensure a smooth transition from the robot's current pose to the start of the warped trajectory. The transformed trajectory is then executed using the robot's controller. The process of source demonstration selection and trajectory adaptation and execution is repeated for every subtask, and if the demonstration achieves task success, it is added to the dataset.

By design, SOFTMIMICGEN enables data generation in manipulation settings that go beyond rigid-body assumptions, including tasks involving flexible materials and complex deformations. Furthermore, the deformable object representation is equivalent to a point cloud representation for objects, and rigid-body objects could also be represented in this manner, making SOFTMIMICGEN a strict generalization of MIMICGEN. In fact, SOFTMIMICGEN can be applied to rigid-body tasks and to rigid-body object geometries that are substantially different from the source demonstrations, unlike MIMICGEN.

V. EXPERIMENTAL RESULTS

In this section, we describe the experimental setup (Sec. V-A), introduce our suite of deformable manipulation tasks (Sec. V-B), present empirical evidence highlighting the capabilities of SOFTMIMICGEN (Sec. V-C), and conduct a systematic analysis of the system (Sec. V-D).

A. Experiment Setup

We use Apple Vision Pro to collect source human demonstrations in our task suite (described in Sec. V-B). Our teleoperation pipeline retargets human hand motions to either a parallel-jaw gripper or a dexterous robotic hand, depending on the target embodiment. For the Franka robot and the surgical robot, we collect relative end-effector poses and gripper actions. For the GR1 humanoid robot, we collect absolute wrist poses and finger joint positions. The bimanual YAM arms are controlled in joint space.

We collect one to three source human demonstrations per task and use SOFTMIMICGEN to generate 1,000 demonstrations per task, sampling from a broader initial state distribution. This setting reduces the burden on the human operator by limiting data collection to simpler task variations, while allowing the more challenging variations to be generated by SOFTMIMICGEN. Each resulting dataset is used to train visuomotor policies using two imitation learning approaches: BC-RNN-GMM [36] and Diffusion Policy [44]. For evaluation, we follow the protocol established in prior work [36]: each experiment is run with three different random seeds, and we report the maximum policy success rate across seeds, unless otherwise specified.

B. Simulation Tasks

We introduce a suite of high-fidelity deformable object manipulation tasks (Fig. 3) implemented in Isaac Lab [10]. They encompass a range of deformable objects (stuffed animal, rope, tissue, towel) and manipulation behaviors (high-precision threading, dynamic whipping, folding, pick-and-place) across four robot embodiments (Franka and YAM robot arms, humanoid, surgical robot).

Humanoid – Towel Unfold. A humanoid robot unfolds a crumpled towel by executing a sequence of movements to spread it flat. The task is considered successful when the towel is laid out smoothly and the humanoid arms are retracted above the table.

Humanoid – Teddy. A humanoid robot grasps the teddy plush toy and places it in a basket. The task is considered successful when the teddy bear is placed inside the basket, and the humanoid's left fingers are open and positioned above the basket.

Franka – Rope Manipulation. The Franka robot performs a rope manipulation task by shaping a rope into a "U" configuration, starting from a randomly initialized state. The task is considered successful when the two ends of the rope are positioned close together.

Franka – Jenga. The Franka robot performs dynamic whipping to remove a pink Jenga block from a Jenga tower. Both the tower configuration and the whip's initial state are randomized. The pink Jenga block is constrained to movement within the X-Y plane. The task is considered successful when the pink block is fully removed from the tower.

Franka – Towel. The Franka robot folds a flat towel in half. The task is considered successful when the towel is folded and the robot arm is retracted above the table.

Franka – Rigid Cube Stack. The Franka robot grasps a cube and places it on top of another cube. This task showcases the applicability of SOFTMIMICGEN to manipulation that does not involve deformable objects.

Surgical – Tissue Manipulation. A piece of soft tissue is fixed at two endpoints. The surgical robot uses forceps-style grippers to grasp the tissue and retract it upward.

Surgical – Threading. The surgical robot grasps a soft thread and passes it through a ring. The task is considered successful if the thread passes through the ring.

YAM – Towel. The bimanual YAM system folds a towel in half. The task is considered successful when the towel is folded and both arms have retracted.

YAM – Bag Loading. The right YAM arm first opens a shopping bag, after which the left arm grasps and places a banana inside. The task is considered successful when the banana is placed in the bag and satisfies the specified placement threshold.

C. SOFTMIMICGEN *Features*

SOFTMIMICGEN significantly reduces the burden of data collection for deformable object manipulation tasks. Collecting robot data for deformable object manipulation is particularly challenging, as it requires fine-grained coordination between robot arms and, in the case of dexterous hands, precise grasping of soft objects which significantly increases operator burden. To address this, we collect only a small number of high-quality teleoperated demonstrations using the Apple Vision Pro. SOFTMIMICGEN then leverages these seed demonstrations to generate large-scale synthetic datasets in simulation, covering a broader range of initial conditions and object states. The generation pipeline achieves success rates ranging from 70% to 100% across tasks. This approach enables efficient scaling of data collection while preserving task-relevant diversity, substantially reducing the human effort required to train performant visuomotor policies for deformable object manipulation tasks.

SOFTMIMICGEN improves policy performance relative to training only on source demonstrations. As shown in Table I, policies trained on SOFTMIMICGEN-generated data consistently outperform those trained solely on human-collected demonstrations – improvements range from 25% to 97%. By using a small set of human teleoperated trajectories, SOFTMIMICGEN enables more robust and effective policy learning, significantly reducing the reliance on costly and time-consuming human data collection.

SOFTMIMICGEN is applicable to a wide range of robots and object manipulation tasks. SOFTMIMICGEN successfully generates demonstrations across our entire task suite, which consists of four distinct platforms: GR1 humanoid, Franka Panda arm, surgical robot, and bimanual YAM arms, and includes diverse kinds of manipulation such as pick-and-place of a teddy bear, towel manipulation, dynamic whipping of a Jenga tower, surgical tissue manipulation, and high-precision threading. It even works for rigid object manipulation (Franka – Rigid Cube), showing that it is a strict generalization of MIMICGEN.

Task	Source Demo BC-RNN-GMM	Generated Demo BC-RNN-GMM	Generated Demo Diffusion Policy
Humanoid - Teddy	0.0 ± 0.0	32.0 ± 3.3	42.0 ± 2.0
Humanoid - Towel	1.3 ± 1.9	50.7 ± 1.9	56.0 ± 5.7
Franka - Rope	2.0 ± 2.0	99.3 ± 0.9	100.0 ± 0.0
Franka - Jenga	4.0 ± 3.3	89.3 ± 15.1	80.0 ± 3.3
Franka - Towel	0.0 ± 0.0	78.7 ± 6.8	70.7 ± 6.8
Franka - Rigid Cube	24.0 ± 4.0	90.7 ± 5.0	50.7 ± 6.8
Surgical - Tissue	56.0 ± 5.7	81.3 ± 12.4	94.7 ± 1.9
Surgical - Threading	5.3 ± 1.9	98.7 ± 1.9	58.7 ± 1.9
YAM - Towel	4.0 ± 3.3	13.3 ± 3.8	52.0 ± 18.2
YAM - Bag Loading	12.0 ± 6.5	14.7 ± 5.0	29.3 ± 5.0

TABLE I: Policy Performance on Source and Generated Datasets. Success rates of visuomotor policies trained on source demonstrations and on SOFTMIMICGEN-generated datasets. Success rates reported as the maximum over three training seeds.

D. SOFTMIMICGEN *Analysis*

How does SOFTMIMICGEN compare with MIMICGEN? We evaluate MIMICGEN on the task of **Franka – Rope Manipulation** and compare its data generation performance with that of SOFTMIMICGEN. In this experiment, a single demonstration is collected in which a straight rope is manipulated into a U-shape. For MIMICGEN, the object reference frame is centered at the midpoint of the rope, with the X-axis aligned along its length. During data generation, one half of the rope is kept stationary while the other half is randomized to produce a diverse distribution of initial rope configurations. MIMICGEN successfully generates only 4 out of 50 demonstrations, primarily in cases where the free end of the rope closely aligns with the configurations present in the source demonstration. In comparison, SOFTMIMICGEN achieves 49 out of 50 successful demonstrations, significantly outperforming MIMICGEN. Since MIMICGEN succeeds only on a narrow subset of configurations seen in the source data, policies trained on this data are unlikely to generalize to novel rope configurations.

How does dataset size affect policy learning? We train visuomotor policies on datasets of sizes 50, 250, 500, and 750, subsampled from the main generated dataset for each task. Table II illustrates the effect of dataset size on policy performance. We observe that success rates generally increase with dataset size. This highlights the importance of larger datasets for training performant policies and underscores the value of scalable data generation without human supervision for deformable object manipulation tasks that require high levels of dexterity and dynamic control.

Task	50 Demos	250 Demos	500 Demos	750 Demos
Humanoid - Teddy	24.0 ± 0.0	26.0 ± 10.0	44.0 ± 0.0	44.0 ± 0.0
Humanoid - Towel	61.3 ± 6.8	70.7 ± 6.8	64.0 ± 5.7	64.0 ± 9.8
Franka - Rope	82.7 ± 5.0	100.0 ± 0.0	98.7 ± 1.9	93.3 ± 6.8
Franka - Jenga	53.3 ± 8.2	77.3 ± 6.8	93.3 ± 5.0	84.0 ± 15.0
Franka - Towel	81.3 ± 5.0	76.0 ± 3.3	74.7 ± 5.0	84.0 ± 3.3
Franka - Rigid Cube	42.0 ± 2.0	68.0 ± 6.5	82.7 ± 3.8	84.0 ± 5.7
Surgical - Tissue	69.3 ± 15.1	56.0 ± 13.1	84.0 ± 5.7	82.7 ± 5.0
Surgical - Threading	56.0 ± 8.0	84.0 ± 3.3	98.7 ± 1.9	96.0 ± 3.3
YAM - Towel	8.0 ± 0.0	12.0 ± 3.3	9.3 ± 1.9	17.3 ± 10.5
YAM - Bag Loading	6.7 ± 3.8	20.0 ± 9.8	17.3 ± 1.9	17.3 ± 1.9

TABLE II: Dataset Size Comparison. Success rates of visuomotor policies trained on subsamples of each task's generated dataset. Success rates reported as the maximum over three training seeds.

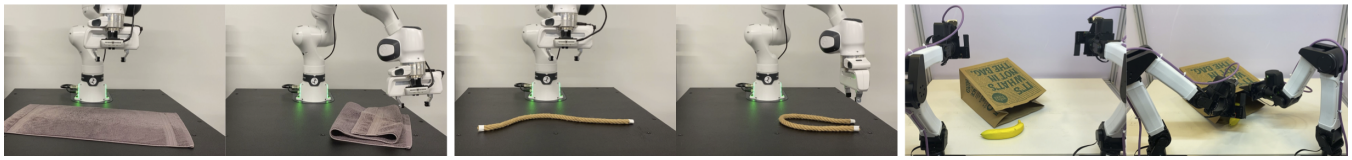

Fig. 4: Real-World Deployment. Example rollouts of real-world deformable manipulation tasks using policies trained on SOFTMIMICGEN-generated datasets.

Replay-based mechanisms vs. training visuomotor policies. Previous works have explored the use of scene registration and trajectory transfer as policies for deformable object manipulation [19], [20]. These approaches typically rely on point clouds from depth sensors to register objects from a source demonstration to a new context. However, point clouds can be noisy, which negatively impacts registration accuracy and the overall success rate of data generation.

In SOFTMIMICGEN, we instead leverage ground-truth nodal information provided by the soft-body simulator to perform precise scene registration, enabling large-scale dataset generation. While our generation pipeline achieves high success rates, we use the generated data to train visuomotor policies through imitation learning. For our simulation results in Tables I and II, these policies learn to manipulate deformable objects directly from raw image input, thereby bypassing explicit registration at inference time.

E. Real-World Evaluation

We evaluate policies trained on datasets generated by SOFTMIMICGEN in a real-world setup on three deformable manipulation tasks shown in Fig. 4. For each task, we generate 1,000 synthetic demonstrations using SOFTMIMICGEN from a single human teleoperated demonstration. We consider three training settings: real-only training, where policies are trained on 30 real-world demonstrations; zero-shot sim-to-real transfer, where policies are trained only on the 1,000 simulated demonstrations; and sim–real co-training [17], where policies are trained jointly on the 30 real demonstrations and the 1,000 SOFTMIMICGEN-generated simulated demonstrations.

For real-world evaluation, we leverage Point Bridge [48], which uses unified, domain-agnostic point-based representations to bridge the sim-to-real gap. During simulation data generation, we extract point clouds of deformable objects using ground-truth masks and depth maps. At deployment, Point Bridge's VLM-guided pipeline extracts task-relevant object points from RGB-D camera observations. Policies are trained and deployed on these point-based observations and output actions, without using non-rigid registration as an explicit online controller.

Policies trained on SOFTMIMICGEN-generated data achieve zero-shot sim-to-real transfer and are further improved with sim-real co-training. Table III reports results for three settings: real-only training, zero-shot sim-to-real transfer, and sim–real co-training. The results show that policies trained with large-scale simulation data can successfully transfer to real-world tasks. Furthermore, SOFTMIMICGEN-generated data improves the performance of policies trained with limited real-world demonstrations.

Task	Real 30 Demos	Zero-shot Sim 1,000 Demos	Sim-Real Co-Train 1,000 + 30 Demos
Franka - Towel	76.6	70.0	76.6
Franka - Rope	46.7	33.3	76.6
YAM - Bag Loading	33.3	63.3	93.3

TABLE III: Real-World Deployment Results. Performance comparison across three training settings: real-only (30 demonstrations), zero-shot sim-to-real (1,000 simulation demonstrations), and sim–real co-training (1,000 simulation + 30 real demonstrations). Results show that large-scale simulation enables effective real-world transfer, while SOFTMIMICGEN-generated data improves performance under limited real-world supervision.

VI. CONCLUSION

We introduce SOFTMIMICGEN, an automated data generation pipeline that synthesizes large-scale datasets for deformable object manipulation from a small number of human demonstrations, and a suite of high-fidelity simulation environments that encompasses a wide range of deformable objects and manipulation behaviors across four different robot embodiments. We apply SOFTMIMICGEN to generate datasets across the task suite and show that policies trained on data generated by SOFTMIMICGEN achieve strong performance across diverse task distributions, including tasks requiring precision and dynamic behavior. SOFTMIMICGEN assumes a fixed sequence of object-centric subtasks. Many real-world deformable manipulation tasks are less structured and may require multiple attempts or conditional transitions. Extending SOFTMIMICGEN to support flexible task structures is a promising direction for future work. We hope that our simulation environments, data generation pipeline, and datasets greatly reduce the barrier for practitioners to study deformable object manipulation, especially in the context of imitation learning and robot foundation models.

ACKNOWLEDGMENT

We thank Simon Schirm and Siddhant Haldar for their valuable discussions and insights.

REFERENCES

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[2] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "OpenVLA: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[3] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[4] NVIDIA, J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[5] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, "Bridge Data: Boosting Generalization of Robotic Skills with Cross-Domain Datasets," in *Robotics: Science and Systems*, 2022.

[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[7] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE Int'l Conf on Robotics and Automation (ICRA)*, 2024.

[8] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[9] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ Int'l Conf on Intelligent Robots and Systems*, 2012.

[10] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Munoz, X. Yao, R. Zurbrügg, N. Rudin *et al.*, "Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning," *arXiv preprint arXiv:2511.04831*, 2025.

[11] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan, "Robogen: Towards unleashing infinite data for automated robot learning via generative simulation," in *Forty-first Int'l Conf on Machine Learning*, 2023.

[12] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," in *Robotics: Science and Systems (RSS)*, 2024.

[13] M. Dalal, A. Mandlekar, C. R. Garrett, A. Handa, R. Salakhutdinov, and D. Fox, "Imitating task and motion planning with visuomotor transformers," in *Conf on Robot Learning*, 2023.

[14] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," *arXiv preprint arXiv:2310.17596*, 2023.

[15] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu, "Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning," *arXiv preprint arXiv:2410.24185*, 2024.

[16] C. Garrett, A. Mandlekar, B. Wen, and D. Fox, "Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment," *arXiv preprint arXiv:2410.18907*, 2024.

[17] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev *et al.*, "Sim-and-real co-training: A simple recipe for vision-based robotic manipulation," *arXiv preprint arXiv:2503.24361*, 2025.

[18] A. Wei, A. Agarwal, B. Chen, R. Bosworth, N. Pfaff, and R. Tedrake, "Empirical analysis of sim-and-real cotraining of diffusion policies for planar pushing from pixels," *arXiv preprint arXiv:2503.22634*, 2025.

[19] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration," in *Robotics Research: The 16th Int'l Symposium ISRR*, 2016.

[20] 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 Int'l Conf on Intelligent Robots and Systems*, 2013.

[21] 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*. IEEE, 2014, pp. 4402–4407.

[22] 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.

[23] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, 2021.

[24] Y.-L. Qiao, A. Gao, and M. Lin, "Neuphysics: Editable neural geometry and physics from monocular videos," *Advances in Neural Information Processing Systems*, 2022.

[25] Y. Feng, Y. Shang, X. Li, T. Shao, C. Jiang, and Y. Yang, "Pie-nerf: Physics-based interactive elastodynamics with nerf," in *IEEE/CVF Conf on Computer Vision and Pattern Recognition*, 2024.

[26] L. Zhong, H.-X. Yu, J. Wu, and Y. Li, "Reconstruction and simulation of elastic objects with spring-mass 3d gaussians," in *European Conf on Computer Vision*, 2024.

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

[28] Y. Li, A. Torralba, A. Anandkumar, D. Fox, and A. Garg, "Causal discovery in physical systems from videos," *Advances in Neural Information Processing Systems*, 2020.

[29] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, "Learning to simulate complex physics with graph networks," in *Int'l Conf on machine learning*, 2020.

[30] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay *et al.*, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Conf on Robot Learning*, 2018.

[31] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware," in *Robotics: Science and Systems*, Daegu, Republic of Korea, 2023.

[32] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, 2020.

[33] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conf on Robot Learning*, 2021.

[34] D. Yarats, I. Kostrikov, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," in *Int'l Conf on learning representations*, 2021.

[35] S. Young, D. Gandhi, S. Tulsiani, A. Gupta, P. Abbeel, and L. Pinto, "Visual imitation made easy," *arXiv e-prints*, 2020.

[36] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.

[37] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta, B. Ichter *et al.*, "Scaling robot learning with semantically imagined experience," *arXiv preprint arXiv:2302.11550*, 2023.

[38] Z. Chen, S. Kiami, A. Gupta, and V. Kumar, "Genaug: Retargeting behaviors to unseen situations via generative augmentation," *arXiv preprint arXiv:2302.06671*, 2023.

[39] S. Pitis, E. Creager, and A. Garg, "Counterfactual data augmentation using locally factored dynamics," *Advances in Neural Information Processing Systems*, 2020.

[40] S. Pitis, E. Creager, A. Mandlekar, and A. Garg, "Mocoda: Model-based counterfactual data augmentation," *Advances in Neural Information Processing Systems*, 2022.

[41] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in neural information processing systems*, 1989.

[42] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, 1999.

[43] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," *Proceedings 2002 IEEE Int'l Conf on Robotics and Automation*, vol. 2, 2002.

[44] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The Int'l Journal of Robotics Research*, 2023.

[45] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer Handbook of Robotics*, 2008.

[46] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics and Automation Magazine*, vol. 17, 2010.

[47] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114–141, 2003.

[48] S. Haldar, L. Johannsmeier, L. Pinto, A. Gupta, D. Fox, Y. Narang, and A. Mandlekar, "Point bridge: 3d representations for cross domain policy learning," *arXiv preprint arXiv:2601.16212*, 2026.