DeformGen: Dynamics-Based Topology Augmentation for Deformable Manipulation Policy Learning

Zili Lin[1,2] **Wenyao Zhang**[1,2†] **Yuyang Zhang**[1,2] **Zekun Qi**[3]
Junyan Lin[2,4] **Hanxin Zhu**[5,6] **Jiaolong Yang** **Zhibo Chen**[5,6]
Yao Mu[1] **Xiaokang Yang**[1] **Xin Jin**[2,6,7] **Wenjun (Kevin) Zeng**[2,7*]

[1]Shanghai Jiao Tong University, Shanghai, China
[2]Ningbo Key Laboratory of Spatial Intelligence and Digital Derivative,
Ningbo Institute of Digital Twin, Eastern Institute of Technology, Ningbo, China
[3]Tsinghua University, Beijing, China [4]The Hong Kong Polytechnic University, Hong Kong, China
[5]University of Science and Technology of China, Hefei, China
[6]Zhongguancun Academy, Beijing, China
[7]Zhejiang Key Laboratory of Industrial Intelligence and Digital Twin, Ningbo, China
[†]Project lead [*]Corresponding author

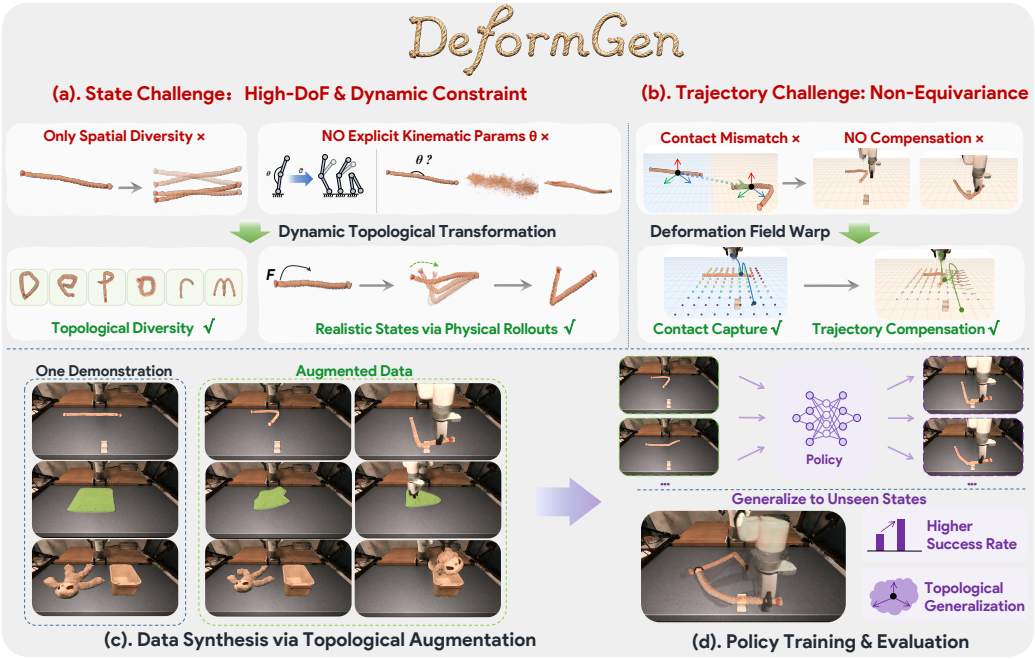


Figure 1: *Top:* We identify two core challenges – the state-space challenge and the trajectory-transfer challenge – that prevent rigid-style augmentation from extending to deformable manipulation. **DeformGen** addresses them through dynamics-based topology transformation and deformation-field warping. *Bottom:* Starting from a single demonstration, DeformGen synthesizes diverse demonstrations across deformable states, leading to improved policy generalization to unseen states.

Abstract

Demonstration augmentation is proposed for cost-efficient data acquisition, but existing methods are fundamentally limited in deformable manipulation due to two challenges: (1) the *state space* is high-dimensional with physics-induced constraints, making valid configurations impossible to reach via low-dimensional

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

pose perturbations; and (2) *trajectory transfer* is non-equivariant, as material points no longer move rigidly together under deformation. We present **DeformGen**, a dynamics-based augmentation framework that achieves *topological diversity* for deformable objects. For the state challenge, DeformGen expands the valid state distribution by applying localized physical disturbances and forward-simulating the dynamics to obtain topology-coherent, physically plausible deformable states. For the trajectory challenge, DeformGen transfers source manipulation trajectories via deformation-field warping, which lifts per-particle displacements into a continuous spatial function to adapt the end-effector trajectory consistently with the deformed geometry. In this way, our method jointly augments the state distribution and its associated manipulation behavior. Experiments on high-fidelity deformable manipulation benchmarks show that DeformGen generally improves policy learning compared with training on the original demonstrations alone and with rigid-style augmentation baselines. Project page: https://zili2002.github.io/DeformGen.

1 Introduction

Imitation learning and visuomotor policy learning have shown remarkable success in robot manipulation, enabling policies to conduct various tasks across diverse environments [1–9]. However, this progress has been driven in large part by access to large-scale, diverse demonstration data, whose collection remains expensive, time-consuming, and difficult to scale. To mitigate this bottleneck, a data augmentation paradigm has emerged: rather than collecting or synthesizing more demonstrations [10], these methods typically augment a single human demonstration to many training trajectories [11–14]. This idea is based on the simple but powerful *equivariant assumption*: rigid bodies satisfy the equivariant constraint that the Euclidean distance between any two material points is invariant under motion or contact forces. With the same rigid transformation applied to the end-effector, the relative pose between it and the object is preserved, leading to a valid trajectory.

This augmentation paradigm, however, is fundamentally mismatched to deformable object manipulation [15, 16]. As illustrated in Fig. 1, we identify two core challenges that break the rigid-object recipe. **State-Space Challenge (Fig. 1(a)):** (i) *High degrees of freedom.* For rigid bodies, a 6-DoF pose provides a sufficient state abstraction. Deformable objects, in contrast, exhibit rich, high-dimensional shape and topology variations [17, 18]. As a result, rigid transformations alone cannot meaningfully expand the valid state distribution required for deformable manipulation. (ii) *Dynamic constraint.* For rigid and articulated objects, valid perturbations can be constructed directly in pose or joint space via kinematic constraints. For deformable objects, however, internal constraints are dynamic, which means the deformation depends on the interaction between internal particles; therefore, naive geometric perturbations typically yield implausible shapes and discontinuous structural changes.

Trajectory-Transfer Challenge (Fig. 1(b)): The *equivariant assumption* no longer holds for deformable objects: material points are not equivariant [15]. Consequently, directly applying a global rigid isometry to augment the trajectories for topological variants of deformable objects introduces two problems: (i) the grasp pose becomes misaligned with the object's local geometry, so the end-effector can no longer grip the object correctly; and (ii) a rigid-style trajectory transfer can only translate and rotate the trajectory as a whole, and cannot capture or compensate for the object's local deformation. These challenges suggest that effective augmentation for deformable manipulation must jointly address two problems: synthesizing physically valid deformable states and transferring demonstrations in a deformation-aware manner.

To this end, we propose **DeformGen**, a dynamics-based topology data augmentation framework for deformable manipulation. Unlike prior rigid augmentation methods that are confined to SE(3) perturbations and thus only produce spatial diversity, DeformGen achieves effective *topological* diversity for deformable objects by jointly synthesizing physically valid deformed states and transferring demonstrations in a deformation-aware manner. Specifically, for the state-space challenge, the key insight is that physically plausible states form a constrained manifold within the high-dimensional particle state space, and naive geometric perturbations almost always fall off this manifold. Therefore, we propose *Dynamic Topological Transformation* to augment the state distribution by applying randomized, spatially localized forces to the object and forward-simulating the resulting dynamics to prevent leaving the valid manifold. These augmented assets can be used both to enrich the support of

training demonstrations and to broaden policy evaluation beyond the narrow state distribution covered by the original data. In response to the trajectory-transfer challenge, DeformGen transfers source demonstrations to each augmented state via *Deformation-Field Warping*. We compute per-particle displacements between the source and target object states and lift them, through K-nearest-neighbor inverse-distance interpolation, into a continuous deformation field $D(\mathbf{x})$ over the workspace. Applying D to the demonstration trajectory simultaneously re-orients the gripper pose according to local geometric changes near the grasp region and compensates the global trajectory to remain aligned with the deformed object as a whole. A single demonstration then can be reused across an entire family of deformable states without breaking contact or violating the object's physical structure (Fig. 1(c)).

As shown in Fig. 1(d), we train policies on the resulting augmented dataset to improve generalization and robustness. Experiments on the high-fidelity deformable manipulation benchmark Real2Sim-Eval [19] show that DeformGen generally improves policy learning: compared with training on the original demonstrations alone, or with rigid-style augmentation baselines, policies trained with DeformGen achieve higher success rates in most settings. These results suggest that effective augmentation for deformable manipulation requires dynamics-consistent state synthesis coupled with deformation-aware trajectory transfer, rather than rigid pose perturbation alone. The contributions of this work are three-fold:

- Formulation of demonstration augmentation for deformable manipulation that identifies physically valid deformable state synthesis as the key missing ingredient beyond rigid-style augmentation.
- A dynamics-consistent pipeline that generates topology-coherent deformable assets through localized perturbation, physics rollout, and stabilization, and synthesizes corresponding manipulation trajectories via deformation-field warping.
- Extensive empirical evidence that the resulting synthetic demonstrations improve policy learning over both no augmentation and rigid-style augmentation baselines in most settings, and that they benefit policies on real robots, both zero-shot and when co-trained with real demonstrations.

2 Related Works

2.1 Data augmentation for robot manipulation

Unlike pipelines that generate demonstrations from scratch using planners [20], generative models, or learned agents [21–23], data augmentation expands an existing dataset through task-agnostic transformations [11, 12]. Beyond appearance-only *visual* perturbations [24, 25], *behavioral* augmentation modifies object configurations and re-solves for task-successful trajectories, typically via physics-based planning [11, 26, 12–14], image/video generation models [27–31] and Real2Sim2Real [32, 33]. Within the physics-based line, DemoGen [12] edits 3D point clouds directly and its extension R2E2R [34] renders consistent videos via a depth-conditioned generator. Simulation-based variants [11, 35, 26, 33] augment via geometric transformations in a digital twin, extending to clutter, bimanual embodiments, and photorealistic 3DGS [36] rendering. However, they rely on rigid SE(3) transformations that break down on deformables. Unlike DeformGen, SoftMimicGen [15] has tried to mitigate the trajectory-transfer challenge, but it still faces the limitation of state-space.

2.2 Deformable object manipulation

Early works use model-based planning built on physical simulators or learned dynamics models, using mass–spring [37], FEM [38], or particle-based representations [39–41] to predict deformation and plan actions [42–47]. A second line of work removes the need for explicit dynamics by directly learning visuomotor policies [48, 49] from demonstrations or interaction, covering tasks such as knot tying [50], cable insertion [51], cloth folding [52–56], and dough shaping [43]. More recently, large-scale vision–language–action models have been extended to deformable settings, showing promising generalization but requiring more data than their rigid-object counterparts [4, 5, 7]. However, this progress has been driven in large part by access to large-scale, diverse demonstration data, whose collection remains expensive, time-consuming, and difficult to scale.

3 Method

In this section, we propose DeformGen, which aims to synthesize a large volume of valid manipulation data for the same task but with varying object initial states, starting from sparse demonstration data.

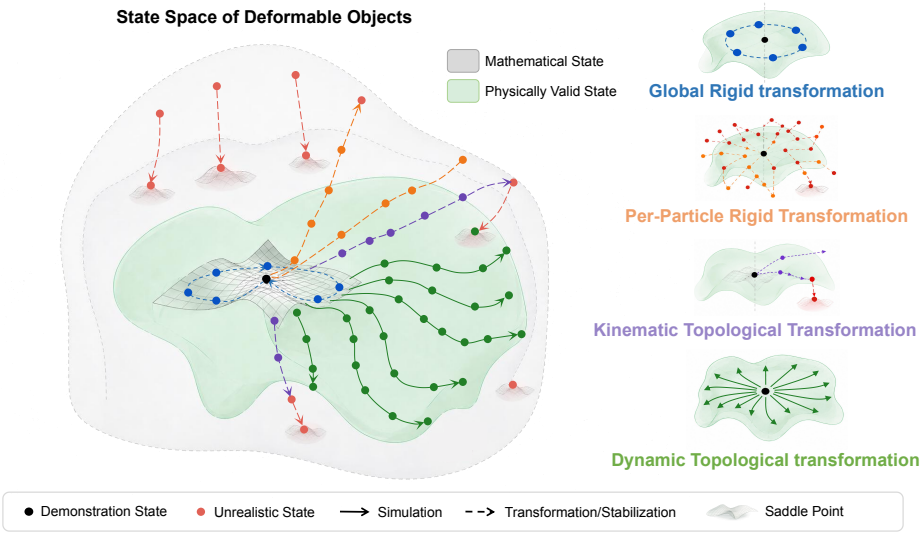


Figure 2: **Augmentation strategies in deformable state space.** Each strategy is visualized in the configuration space $\mathcal{S}$ with the physically plausible subspace $\mathcal{S}_{\text{real}}$ shaded. Dynamics-based augmentation is designed to keep all generated states within $\mathcal{S}_{\text{real}}$ while achieving broader coverage than alternatives.

First, we present a novel object initial state augmentation approach in Sec. 3.1. Based on the above deformable object representation, we propose a manipulation trajectory augmentation method in Sec. 3.2. Furthermore, we describe the policy training framework employed to verify the effectiveness and efficiency of the augmented data in Sec. 3.3. For soft object modeling and simulation, we leverage PhysTwin [57] and Real2Sim-Eval [19] due to their high fidelity in both visual rendering and physical dynamics, with detailed descriptions provided in Appendix C.1.

3.1 State Augmentation

The objective of this step is to generate diverse object configurations for the same task, serving both subsequent trajectory synthesis and policy evaluation. Fundamentally, synthesizing object states amounts to sampling the object's configuration space. A practical method should produce states that are physically plausible under the simulator's dynamics model and that are sufficiently diverse to improve downstream policy learning.

State space. Following standard practice in deformable object simulation [39, 40, 58, 59, 37, 60], we consider a deformable object discretized into N particles with configuration space $\mathcal{S} = \mathbb{R}^{3N}$, where each state $\mathbf{s} = (\mathbf{p}_1, \ldots, \mathbf{p}_N) \in \mathcal{S}$ specifies all particle positions. The *physically plausible subspace* $\mathcal{S}_{\text{real}} \subset \mathcal{S}$ contains all configurations consistent with real-world physical constraints. In general, $\mathcal{S}_{\text{real}} \subsetneq \mathcal{S}$: most points in $\mathbb{R}^{3N}$ do not correspond to any physically realizable configuration.

Working assumption. Our approach relies on the premise that a well-calibrated physics simulator $\Phi_{\text{sim}}(\mathbf{s}, \mathbf{f}, \Delta t)$ approximately preserves physical plausibility when evolving from a valid state, but cannot reliably restore it from an invalid one (Assumption 1 in Appendix A). This asymmetry implies that any method which first perturbs the state out of $\mathcal{S}_{\text{real}}$ and then relies on simulation to "fix" it has no reliable path back to plausibility.

Why existing strategies fall short. We identify three alternatives (details in Appendix A.2): (i) *Global rigid transformation* preserves plausibility but is confined to a 6-DoF subspace of $\mathbf{s}_0$, unable to capture shape or topological variation—confirmed by near-zero non-rigid residuals in Fig. 5. (ii) *Per-particle perturbation* faces a coverage–plausibility trade-off: large noise breaks connectivity; small noise yields only local wrinkles [61]. (iii) *Kinematic deformation fields* preserve topology but ignore material constraints, producing coherent yet dynamically inadmissible states. Both (ii) and (iii) leave $\mathcal{S}_{\text{real}}$ and rely on post-hoc repair that may fail.

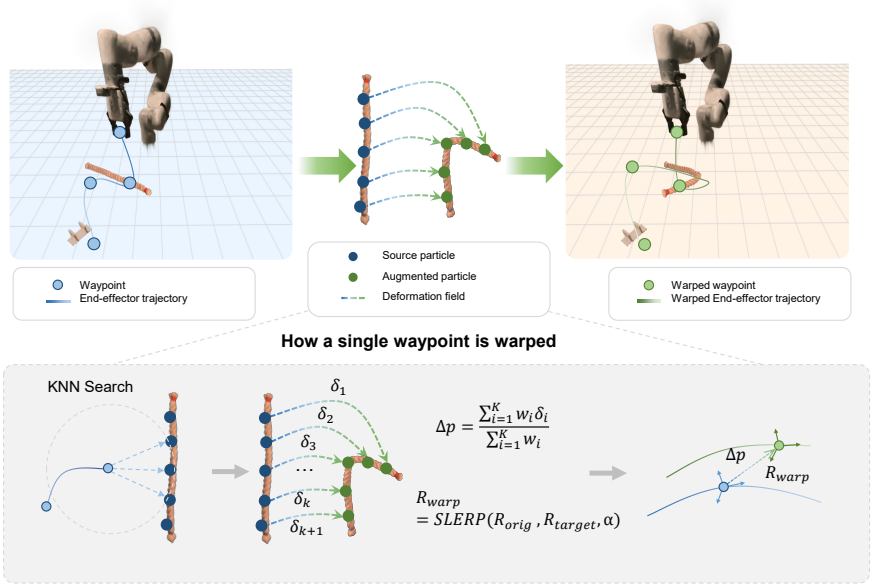


Figure 3: **Trajectory augmentation via deformation-field warping. Top:** The source trajectory (blue, left) is warped through the deformation field (dashed arrows, center) to produce an augmented trajectory (green, right) consistent with the deformed object. **Bottom:** For each waypoint, K-nearest-neighbor particle displacements are aggregated via inverse-distance weighting to obtain the position offset Δp, and a local Jacobian is estimated to derive the orientation update R_{warp} via SLERP.

Dynamics-based topological augmentation. We instead augment states by applying localized external forces and *forward-simulating* the dynamics from a known valid state:

$$\mathbf{s}_{\text{aug}} = \Phi_{\text{sim}}(\mathbf{s}_0, \mathbf{f}, \Delta t), \quad \mathbf{s}_0 \in \mathcal{S}_{\text{real}}, \tag{1}$$

where $\mathbf{f}$ is a localized force field. Because the method evolves the state through the simulator's own dynamics, it never explicitly leaves $\mathcal{S}_{\text{real}}$, requiring no post-hoc repair from invalid states. Since localized forces can induce diverse non-rigid deformations (bending, twisting, folding, draping), the reachable set is not restricted to a low-dimensional submanifold. We do not claim full coverage of $\mathcal{S}_{\text{real}}$, but treat this as a *practical sampling heuristic* that explores a substantially broader region than rigid transformations—verified empirically in Fig. 5. Table 1 summarizes the comparison.

Table 1: Comparison of augmentation strategies for deformable objects. *Coherence*: preserves topological coherence. $\subseteq \mathcal{S}_{\text{real}}$: reachable states remain plausible under Assumption 1.

Strategy	Coherence	Reachable set	$\subseteq \mathcal{S}_{\text{real}}$	$\mathcal{S}_{\text{real}}$-recoverable?
(i) Global rigid	✓	6-DoF subspace of $\mathbf{s}_0$	✓	N/A
(ii) Per-particle	✗	$\mathcal{S}$	✗	Unreliable
(iii) Kinematic topological	✓	$\mathcal{S}$	✗	Unreliable
(iv) Dynamics (Ours)	✓	$\mathcal{R}(\mathbf{s}_0) \subseteq \mathcal{S}_{\text{real}}$	✓†	**N/A**

†Under Assumption 1; in practice, subject to simulator fidelity.

3.2 Trajectory Augmentation

The objective of this step is to synthesize valid manipulation trajectories for unseen object configurations. Our synthesized trajectories consist of three phases: *approach*, *grasp*, and *manipulation*. The grasp poses and manipulation trajectories are synthesized using *Deformation Field Warping*, while the approach trajectory is generated via interpolation from the robot's reset pose to the grasp pose.

Rigid trajectory transfer methods [12, 33] assume a uniform transformation of the object, neglecting part-wise deformations. Inspired by [62], we construct a deformation field from per-particle displacements, yielding a closed-form spatial mapping without iterative optimization.

Position warping. Let $p_{\text{orig}}, p_{\text{def}} \in \mathbb{R}^{N\times 3}$ be the source and deformed point clouds. The per-point displacement is $\delta_i = p_{\text{def},i} - p_{\text{orig},i}$. For each end-effector position x_t at timestep t, we retrieve its

K nearest neighbors from p_{orig} and interpolate via inverse distance weighting:

$$w_{t,j} = \frac{1}{\left\| x_t - p_{\text{orig},\,\text{nn}_j(x_t)} \right\| + \varepsilon}, \quad \tilde{w}_{t,j} = \frac{w_{t,j}}{\sum_j w_{t,j}}, \quad d(x_t) = \sum_j \tilde{w}_{t,j}\, \delta_{\text{nn}_j(x_t)}, \tag{2}$$

where $\varepsilon > 0$ ensures numerical stability. The warped position incorporates a time-dependent decay:

$$x_t^{\text{warp}} = x_t + \alpha_t \cdot d(x_t), \tag{3}$$

where $\alpha_t = \text{decay}(t)$ allows the trajectory to follow local deformations initially while gradually reverting to the original path.

Orientation adaptation. For the end-effector orientation, we construct local relative coordinates within the KNN neighborhood of x_t:

$$\ell_{t,j}^{\text{orig}} = p_{\text{orig},\,\text{nn}_j(x_t)} - x_t, \quad \ell_{t,j}^{\text{def}} = \ell_{t,j}^{\text{orig}} + \delta_{\text{nn}_j(x_t)}. \tag{4}$$

A local Jacobian matrix J_t is estimated via least squares fitting to map the original local vectors to the deformed ones:

$$J_t = \arg\min_J \sum_j \left\| \ell_{t,j}^{\text{def}} - J\, \ell_{t,j}^{\text{orig}} \right\|^2. \tag{5}$$

Letting X_{orig} and X_{def} denote the matrices of stacked local vectors, the closed-form solution is:

$$J_t = X_{\text{def}} X_{\text{orig}}^\top \left(X_{\text{orig}} X_{\text{orig}}^\top \right)^+, \tag{6}$$

where $(\cdot)^+$ denotes the Moore-Penrose pseudoinverse. The induced rotation R_t' is obtained by projecting $J_t R_t$ onto the $SO(3)$ manifold via SVD. The final warped orientation is computed via:

$$R_t^{\text{warp}} = \text{SLERP}\left(R_t,\ R_t',\ \alpha_t\right). \tag{7}$$

In practice, the grasp pose correlates more strongly with nearby object points, so we use a small K for warping the grasp pose. The manipulation phase depends on the overall object state, so we set K to the total number of object points to capture global deformation. Given the tabletop scenario, we constrain rotations to the Z-axis perpendicular to the table surface. Details are in Appendix B.

3.3 Policy Training

To evaluate the effectiveness and efficiency of our augmentation approach, policies are trained via imitation learning and validated within a simulation environment. For three tasks—rope routing, toy packing, and cloth folding—we collect one teleoperation demonstration per task. Using the state augmentation method described in Sec. 3.1, we synthesize more than 1200 distinct object states for each task to facilitate trajectory synthesis and evaluation scenarios. Subsequently, we employ the *Deformation Field Warping* method and the local rigid-transfer ablation for trajectory augmentation.

The augmented trajectories are executed in simulation to verify their success, with task-specific success criteria detailed in Appendix C.2. We record third-person and wrist-mounted RGB images along with corresponding actions during execution. The successful episodes are split into training and held-out test sets (details in Sec. 4.1).

Following the protocol of Real2Sim-Eval [19], we train four policy architectures: ACT [49], Diffusion Policy [48], SmolVLA [63], and π_0 [1] (fine-tuned via LoRA). The trained policies are evaluated on held-out object states unseen during training, including configurations where the warping method failed to generate successful trajectories.

4 Experiments

4.1 Implementation Details

All experiments are conducted in Real2Sim-Eval [19] with PhysTwin [57] for soft-body dynamics and rendering. The robot is an xArm7 with two RGB cameras (third-person and wrist, 848×480, 30 Hz). We evaluate on three tasks: **rope routing** (thread a rope through a clip), **toy packing** (place a stuffed

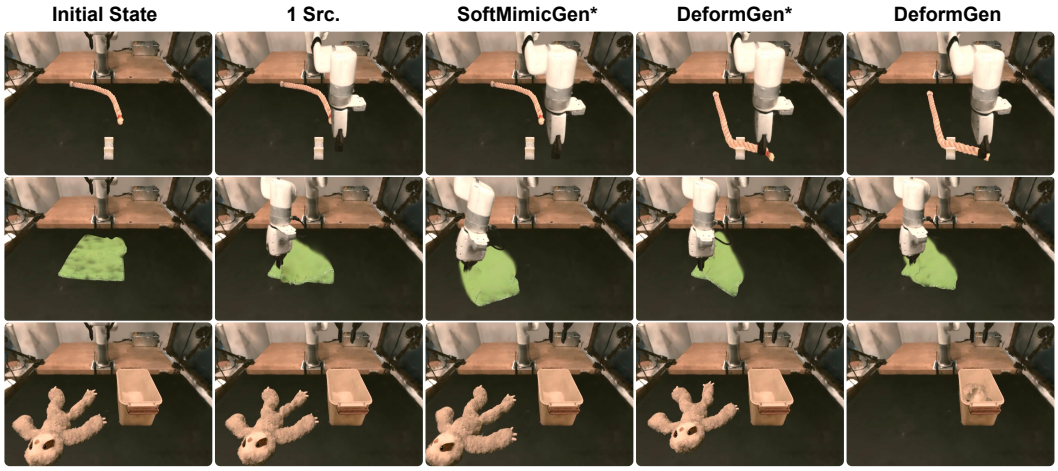


Figure 4: **Qualitative comparison of policy execution across methods.** Rows: rope routing, cloth folding, toy packing. Columns: initial state and final frame of policies trained under each regime; DeformGen completes all three tasks.

toy into a container), and **cloth folding** (fold cloth into a triangle). Success criteria, augmentation parameters, and training hyperparameters are detailed in the Appendix.

For state augmentation, the gripper executes randomized Cartesian perturbations while in contact with the object (180 steps for rope/toy, 260 for cloth), followed by stabilization. For each task, we generate augmented states and attempt trajectory synthesis to obtain 1,000 successful trajectories for training and 200 successful states for testing. ACT [49] and Diffusion Policy [48] are trained from scratch; SmolVLA [63] is fine-tuned from its public checkpoint with the vision encoder frozen, and π_0 [1] is fine-tuned with LoRA (Appendix C.6). Real2Sim-Eval has been validated as a predictor of real-world policy performance [19]; we further test on real robots (Sec. 4.5).

Compared methods. To disentangle the contributions of state augmentation and trajectory synthesis, we compare four training regimes:

- **1 Src.**: a single source demonstration without any augmentation.
- **SoftMimicGen* (SMG*)**: SoftMimicGen [15] shares a similar philosophy to ours in trajectory synthesis—adapting demonstrations to deformed object geometry—but its state augmentation remains rigid: its state distribution is "typically one with a larger set of possible placements for objects in the scene" [15], i.e., SE(3) perturbations of object pose. Since SoftMimicGen is not open source, we reimplement its core design following the descriptions in the original paper.
- **DeformGen* (DG*)**: topological state augmentation paired with local rigid trajectory transfer. This ablation uses the same distribution of augmented states as DG but replaces deformation-field warping with a rigid transform estimated from the K nearest material points around the grasp point and applied to the entire trajectory, isolating the effect of trajectory synthesis.
- **DeformGen (DG)**: our full method, which pairs dynamics-based topological state augmentation with deformation-field warping. Compared with DG*, DG adapts the trajectory through a continuous deformation field rather than a single local rigid transform, enabling better alignment with the deformed object throughout manipulation.

The comparison between SMG* and DG reveals the effect of *state augmentation* contrasting rigid with topological diversity, while the comparison between DG* and DG reveals the effect of *trajectory synthesis* contrasting local rigid transfer with deformation-field warping.

4.2 Experiment Results

Results are in Table 2, averaged over three evaluation seeds. DeformGen achieves the highest average success rate across three out of four policy architectures. The result reveals two insights:

(1) Topological state diversity contributes to generalization. Comparing SMG* with DG highlights the effect of state augmentation: both methods use deformation-field warping for trajectory transfer,

Table 2: **Policy evaluation success rate** (%), mean ± std over three evaluation seeds. **1 Src.**: 1 source demo. **SMG***: rigid state aug. + deformation-field warping. **DG***: topological state aug. + local rigid transfer. **DG**: full DeformGen.

	ACT [49]				DP [48]			
	1 Src.	SMG*	DG*	**DG**	1 Src.	SMG*	DG*	**DG**
Rope	0.00 ±0.00	68.50 ±0.50	89.50 ±0.87	**91.17** ±0.58	0.00 ±0.00	**66.67** ±2.75	63.50 ±2.78	55.33 ±10.91
Toy	0.00 ±0.00	75.00 ±2.00	50.17 ±1.26	**75.50** ±3.00	0.00 ±0.00	**44.67** ±8.81	30.83 ±22.37	31.00 ±19.92
Cloth	4.17 ±0.29	4.33 ±0.76	1.50 ±0.00	**11.67** ±1.15	**7.00** ±0.00	1.00 ±0.50	1.67 ±0.76	1.33 ±1.89
Average	1.39 ±0.10	49.28 ±1.08	47.06 ±0.54	**59.44** ±0.77	2.33 ±0.00	**37.44** ±3.20	32.00 ±8.01	29.22 ±8.17

	SmolVLA [63]				π_0 [1]			
	1 Src.	SMG*	DG*	**DG**	1 Src.	SMG*	DG*	**DG**
Rope	0.00 ±0.00	64.50 ±3.46	84.67 ±4.16	**88.50** ±3.77	0.00 ±0.00	54.83 ±1.26	96.67 ±1.76	**98.67** ±0.58
Toy	0.00 ±0.00	44.83 ±3.33	57.50 ±12.17	**63.00** ±12.76	0.00 ±0.00	12.00 ±1.80	31.00 ±1.32	**57.17** ±0.76
Cloth	7.50 ±0.00	10.33 ±7.15	**19.50** ±13.00	19.33 ±16.02	6.33 ±0.58	18.00 ±3.28	**33.83** ±12.21	14.00 ±3.12
Average	2.50 ±0.00	39.89 ±0.42	53.89 ±1.17	**56.94** ±1.23	2.11 ±0.19	28.28 ±1.55	53.83 ±3.33	**56.61** ±1.08

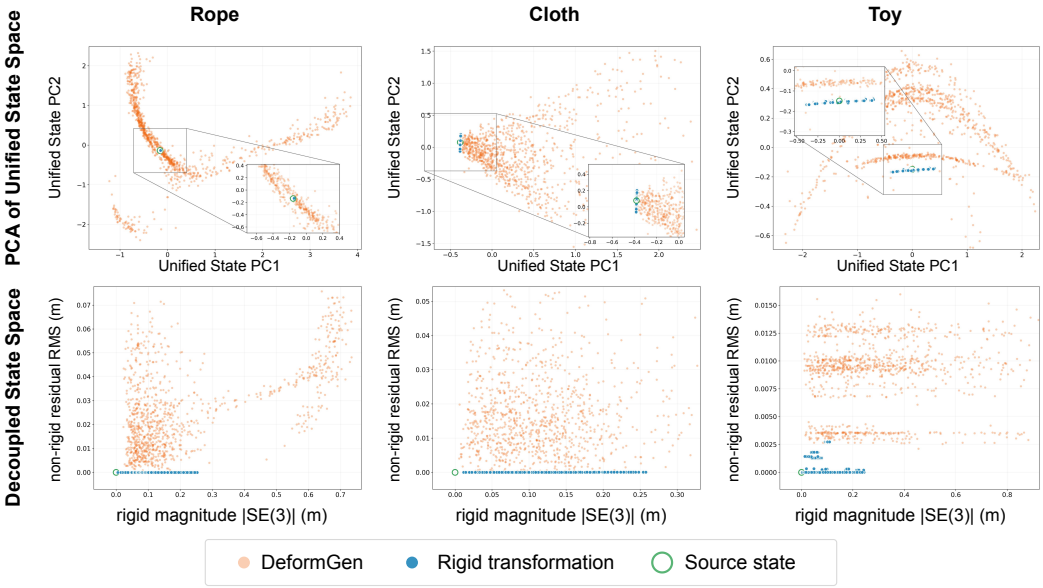


Figure 5: **State-space analysis across three tasks.** Each state is decomposed relative to the source (green circle) into a rigid $\mathrm{SE}(3)$ component and a non-rigid residual. **Top:** PCA of the unified state vector. **Bottom:** rigid magnitude vs. non-rigid residual RMS. Rigid samples (blue) stay near the source with near-zero residual; DeformGen (orange) spreads broadly with large residuals. Toy residuals are non-zero due to object interactions during stabilization.

but SMG* relies on rigid state perturbations whereas DG uses dynamics-based topological state augmentation. DG achieves higher average success rates in most architectures, suggesting that broader coverage of deformable-object configurations is important for policy generalization.

(2) Deformation-field warping provides complementary gains. Since DG* and DG use the same topologically augmented states, their comparison isolates trajectory transfer. DG often improves over DG*, suggesting that deformation-aware warping can further benefit policy learning.

Together, these findings suggest that both broader deformable-state coverage and deformation-aware trajectory transfer contribute to policy generalization, with their relative effects varying across architectures and tasks. Figure 4 illustrates this trend: policies trained with DeformGen complete the tasks, while comparison methods often exhibit grasp misalignment or incomplete manipulation.

4.3 State Coverage Analysis

We decompose each augmented state into a rigid SE(3) component and a non-rigid residual via Procrustes alignment (Figure 5). Rigid augmentation clusters near the source with negligible residual,

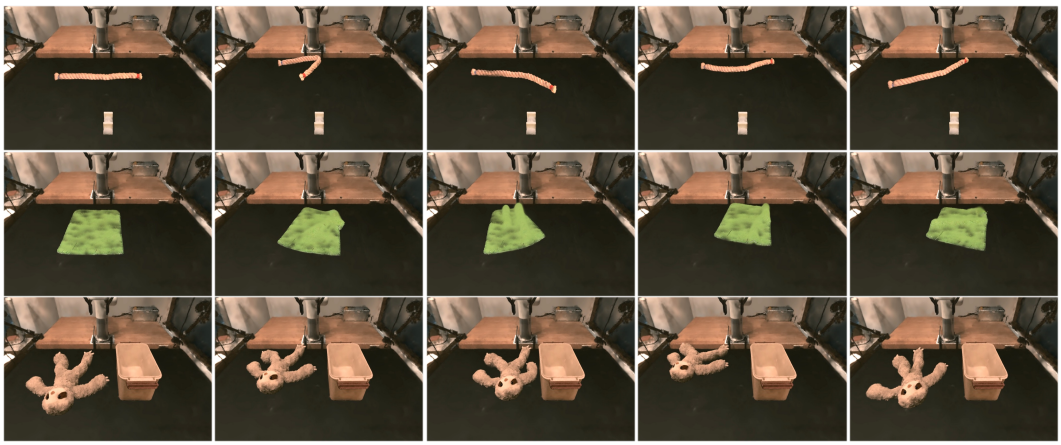


Figure 6: **Examples of augmented object states.** Each row shows one task: the source demonstration state (left) and states generated by DeformGen, all physically plausible and diverse.

Table 3: Success rate (%) on **rigid-only** test states.

Task	ACT			DP			SmolVLA			π_0		
	SMG*	DG*	DG	SMG*	DG*	DG	SMG*	DG*	DG	SMG*	DG*	DG
Rope	**94.00**	78.50	77.50	**92.50**	51.50	46.00	**91.00**	69.00	62.00	92.00	**95.50**	86.50
Toy	**90.00**	76.00	83.50	77.50	**84.50**	69.00	45.50	47.00	**57.00**	12.50	29.00	**39.50**
Cloth	6.50	10.00	**13.00**	2.00	**7.50**	7.00	27.00	20.50	**30.50**	8.50	20.50	**34.50**
Average	**63.50**	54.83	58.00	**57.33**	47.83	40.67	**54.50**	45.50	49.83	37.67	48.33	**53.50**

Table 4: Success rate (%) under different synthetic data quantity.

	ACT				SmolVLA			
	N = 100	N = 250	N = 500	N = 750	N = 100	N = 250	N = 500	N = 750
Rope	36.50	57.50	75.50	**88.50**	65.00	80.50	84.00	**91.00**
Toy	14.50	**82.50**	79.00	74.00	20.00	28.00	59.50	**66.00**
Cloth	7.50	14.50	21.50	**22.00**	25.50	5.50	24.00	**32.50**
Average	19.50	51.50	58.67	**61.50**	36.83	38.00	55.83	**63.17**

while DeformGen spreads broadly with large residuals, suggesting that the gains in Table 2 stem from broader coverage of non-rigid configurations rather than more data at similar configurations.

Figure 6 shows representative augmented object states generated by DeformGen for each task. Starting from a single source configuration, our dynamics-based augmentation produces a diverse set of topologically distinct states—including different rope curvatures, varied stuffed toy orientations and compressions, and diverse cloth folds and drapes—all of which are physically plausible under the simulator's dynamics model.

4.4 Ablation Studies

Q1: Can DeformGen generalize to rigid-only scenarios? Table 3 evaluates whether topological augmentation hurts performance when the test states only involve rigid transformations. SMG* achieves the highest average success rate for ACT, DP, and SmolVLA, which is expected because its rigid-state training distribution closely matches the rigid-only test set. In contrast, DG* and DG are trained on broader topological variations. Nevertheless, DG remains competitive and even performs best for π_0, suggesting that training on topologically diverse data does not substantially compromise performance on simpler rigid scenarios.

Q2: Impact of synthetic data quantity. Table 4 shows that average performance improves with scale. The average success rates increase monotonically from N=100 to N=750 for both ACT (19.50% $\rightarrow$ 61.50%) and SmolVLA (36.83% $\rightarrow$ 63.17%). This suggests that dynamics-based augmentation can benefit from increased data scale, with larger synthetic datasets providing average gains.

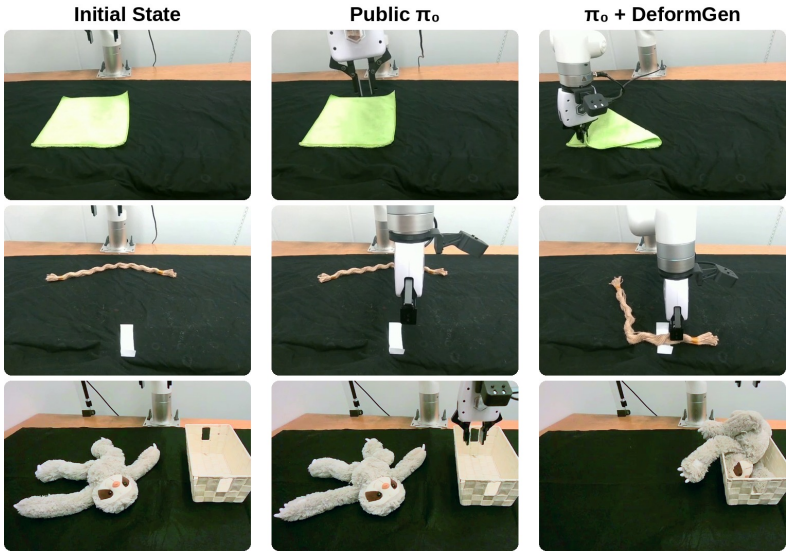


Figure 7: **Zero-shot sim-to-real (xArm7).** Public π_0 vs. π_0 fine-tuned only on DeformGen data.

Table 5: **Zero-shot cloth folding** results (%).

Policy	Trend	Contact	Success
Public π_0	100	10	0
π_0 + DeformGen	100	**70**	**40**

Table 6: **Co-training** success rate (%) of $\pi_{0.5}$.

Training data	Rope	Cloth	Toy	Avg.
Real only	40.0	50.0	43.3	44.4
Real + DeformGen	**80.0**	**60.0**	**73.3**	**71.1**

Q3: Can the policy generalize to synthesis failure cases? On *hard samples*, states for which our own synthesis failed, augmented policies still reach up to 55.5% success (Appendix G.4), i.e., they generalize beyond the reach of the data-generation pipeline.

4.5 Real-World Experiments

We evaluate DeformGen on real robots in two settings (details in Appendix H). **Zero-shot sim-to-real.** On an xArm7 with the same cameras as in simulation, we deploy π_0 fine-tuned only on DeformGen data, with no real demonstrations of these tasks, and compare it with the public π_0 checkpoint (Fig. 7). The public checkpoint misses the rope and does not reach the toy, whereas our policy inserts the rope into the clip and places the toy into the container. On cloth folding, where both show the intended task-level behavior (*Trend* in Table 5), the public checkpoint closes the gripper above the thin cloth, whereas ours folds it, raising contact from 10% to 70% and success from 0% to 40%. **Co-training.** On a Franka Research 3, we fine-tune $\pi_{0.5}$ [2] from the DROID checkpoint [64] with 50 real demonstrations per task, alone or with 1,000 DeformGen demonstrations retargeted from simulation. Adding DeformGen data improves every task, raising the average success rate from 44.4% to 71.1% (Table 6, Fig. 11).

5 Conclusion

In this work, we proposed DeformGen, a dynamics-based augmentation framework that expands the valid state distribution through localized physical disturbances, forward simulation, and stabilization, and transfers source manipulation trajectories via deformation-field warping. Experiments on high-fidelity deformable manipulation benchmarks showed that DeformGen generally improves policy learning over both training on the original demonstrations alone and rigid-style augmentation baselines, and real-robot experiments confirmed zero-shot transfer and co-training gains. More broadly, our results suggest that effective augmentation for deformable manipulation requires dynamics-consistent state synthesis and deformation-aware trajectory transfer.

Acknowledgments and Disclosure of Funding

This work was supported by Grants of NSFC 62672260, 62302246, ZJNSFC LQ23F010008, Ningbo 2023D-001-01 & 2023Z237 & 2024Z284 & 2024Z289 & 2023CX050011 & 2025Z038 & 2025Z028 & 2025Z059, and supported by High Performance Computing Center at Eastern Institute of Technology and Ningbo Institute of Digital Twin.

We thank Chenhao Hong (Columbia University), Hanxiao Jiang (Columbia University), Shaojie Ren (Zhongguancun Academy) and Chenyu Zhou (Zhongguancun Academy) for their generous support and help with the real-robot experiments. Special thanks go to Chenhao Hong for independently carrying out the costly real-robot experiments for zero-shot sim-to-real transfer under the tight schedule of the rebuttal period.

Competing interests. The authors declare no competing interests.

References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint*, 2024. 2, 6, 7, 8, 18, 19

[2] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint*, 2025. 10, 19, 22

[3] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint*, 2024.

[4] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint*, 2025. 3

[5] Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025. 3

[6] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint*, 2025.

[7] Wenyao Zhang, Bozhou Zhang, Zekun Qi, Wenjun Zeng, Xin Jin, and Li Zhang. Disentangled robot learning via separate forward and inverse dynamics pretraining. *arXiv preprint arXiv:2604.16391*, 2026. 3

[8] Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

[9] Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Liuao Pei, Xiaokang Yang, Jiangmiao Pang, Yao Mu, and Ping Luo. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint*, 2025. 2

[10] Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). In *ECCV*, 2025. 2

[11] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, pages 1820–1864. PMLR, 2023. 2, 3

[12] Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025. 3, 5

[13] Sizhe Yang, Wenye Yu, Jia Zeng, Jun Lv, Kerui Ren, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. *arXiv preprint arXiv:2504.13175*, 2025.

[14] Yuan Xu, Jiabing Yang, Xiaofeng Wang, Yixiang Chen, Zheng Zhu, Bowen Fang, Guan Huang, Xinze Chen, Yun Ye, Qiang Zhang, et al. Egodemogen: Novel egocentric demonstration generation enables viewpoint-robust manipulation. *arXiv preprint arXiv:2509.22578*, 2025. 2, 3

[15] Masoud Moghani, Mahdi Azizian, Animesh Garg, Yuke Zhu, Sean Huver, and Ajay Mandlekar. Softmimicgen: A data generation system for scalable robot learning in deformable object manipulation. *arXiv preprint arXiv:2603.25725*, 2026. 2, 3, 7

[16] Yunsong Zhou, Hangxu Liu, Xuekun Jiang, Xing Shen, Yuanzhen Zhou, Hui Wang, Baole Fang, Yang Tian, Mulin Yu, Qiaojun Yu, et al. Sim1: Physics-aligned simulator as zero-shot data scaler in deformable worlds. *arXiv preprint arXiv:2604.08544*, 2026. 2

[17] Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *The International Journal of Robotics Research*, 37(7):688–716, 2018. 2

[18] Hang Yin, Anastasia Varava, and Danica Kragic. Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics*, 6(54):eabd8803, 2021. 2

[19] Kaifeng Zhang, Shuo Sha, Hanxiao Jiang, Matthew Loper, Hyunjong Song, Guangyan Cai, Zhuo Xu, Xiaochen Hu, Changxi Zheng, and Yunzhu Li. Real-to-sim robot policy evaluation with gaussian splatting simulation of soft-body interactions. *arXiv preprint arXiv:2511.04665*, 2025. 3, 4, 6, 7, 17, 19

[20] Haoyu Zhao, Cheng Zeng, Linghao Zhuang, Yaxi Zhao, Shengke Xue, Hao Wang, Xingyue Zhao, Zhongyu Li, Kehan Li, Siteng Huang, Mingxiu Chen, Xin Li, Deli Zhao, and Hua Zou. High-fidelity simulated data generation for real-world zero-shot robotic manipulation learning with gaussian splatting. *IEEE Robotics and Automation Letters*, 11(5):5310–5317, 2026. doi: 10.1109/LRA.2026.3671535. 3

[21] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The Robot Learning Benchmark & Learning Environment. *arXiv preprint arXiv:1909.12271*, 2019. 3

[22] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. RoboGen: Towards Unleashing Infinite Data for Automated Robot Learning via Generative Simulation. In *International Conference on Machine Learning*, 2024.

[23] Atsushi Kanehira, Naoki Wake, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Rl-driven data generation for robust vision-based dexterous grasping. *arXiv preprint arXiv:2504.18084*, 2025. 3

[24] Zoey Chen, Zhao Mandi, Homanga Bharadhwaj, Mohit Sharma, Shuran Song, Abhishek Gupta, and Vikash Kumar. Semantically controllable augmentations for generalizable robot learning. *The International Journal of Robotics Research*, 44(10-11):1705–1726, 2025. 3

[25] GigaBrain Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jie Li, Jiagang Zhu, Lv Feng, et al. Gigabrain-0: A world model-powered vision-language-action model. *arXiv preprint arXiv:2510.19430*, 2025. 3

[26] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Jim Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16923–16930. IEEE, 2025. 3

[27] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through neural trajectories. *arXiv preprint*, 2025. 3

[28] Ying Li, Xiaobao Wei, Xiaowei Chi, Yuming Li, Zhongyu Zhao, Hao Wang, Ningning Ma, Ming Lu, and Sirui Han. Manipdreamer3d: Synthesizing plausible robotic manipulation video with occupancy-aware 3d trajectory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 6644–6652, 2026.

[29] Guanhua Ji, Harsha Polavaram, Lawrence Yunliang Chen, Sandeep Bajamahal, Zehan Ma, Simeon Adebola, Chenfeng Xu, and Ken Goldberg. Oxe-auge: A large-scale robot augmentation of oxe for scaling cross-embodiment policy learning. *arXiv preprint arXiv:2512.13100*, 2025.

[30] Boyang Wang, Haoran Zhang, Shujie Zhang, Jinkun Hao, Mingda Jia, Qi Lv, Yucheng Mao, Zhaoyang Lyu, Jia Zeng, Xudong Xu, et al. Robovip: Multi-view video generation with visual identity prompting augments robot manipulation. *arXiv preprint arXiv:2601.05241*, 2026.

[31] Fan Xie, Dan Zeng, Qiaomu Shen, and Bo Tang. A comprehensive survey on text-to-video generation. *Chinese Journal of Electronics*, 34(4):1009–1036, 2025. 3

[32] Chuer Pan, Litian Liang, Dominik Bauer, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, and Shuran Song. One demo is worth a thousand trajectories: Action-view augmentation for visuomotor policies. In *9th Annual Conference on Robot Learning*, 2025. 3

[33] Justin Yu, Letian Fu, Huang Huang, Karim El-Refai, Rares Andrei Ambrus, Richard Cheng, Muhammad Zubair Irshad, and Ken Goldberg. Real2render2real: Scaling robot data without dynamics simulation or robot hardware, 2025. URL https://arxiv.org/abs/2505.09601. 3, 5

[34] Yujie Zhao, Hongwei Fan, Di Chen, Shengcong Chen, Liliang Chen, Xiaoqi Li, Guanghui Ren, and Hao Dong. Real2edit2real: Generating robotic demonstrations via a 3d control interface. *arXiv preprint arXiv:2512.19402*, 2025. 3

[35] Caelan Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. *arXiv preprint arXiv:2410.18907*, 2024. 3

[36] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3

[37] Xavier Provot et al. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147. Canadian Information Processing Society, 1995. 3, 4

[38] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE transactions on Visualization and Computer Graphics*, 5(1):62–73, 1999. 3

[39] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 3, 4

[40] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007. 4

[41] Sergio Orozco, Brandon B. May, Tushar Kusnur, George Konidaris, and Laura Herlant. Learning equivariant neural-augmented object dynamics from few interactions. In *Beyond Rigid Worlds: Representing and Interacting with Non-Rigid Objects*, 2025. URL https://openreview.net/forum?id=JAiJpFozaD. 3

[42] Xingyu Lin, Zhiao Huang, Yunzhu Li, Joshua B. Tenenbaum, David Held, and Chuang Gan. Diffskill: Skill abstraction from differentiable physics for deformable object manipulations with tools. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[43] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. In *Conference on Robot Learning (CoRL)*, 2023. 3

[44] Haonan Chen, Yilong Niu, Kaiwen Hou, Shuijing Liu, Yixuan Wang, Yunzhu Li, and Katherine Driggs-Campbell. Predicting object interactions with behavior primitives: An application in stowing tasks. In *Conference on Robot Learning (CoRL)*, 2023.

[45] Isabella Huang, Yashraj Narang, Ruzena Bajcsy, Fabio Ramos, Tucker Hermans, and Dieter Fox. Defgraspsim: Physics-based simulation of grasp outcomes for 3d deformable objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[46] Lijun Han and Hesheng Wang. Robotic manipulation of deformable objects: a comprehensive review. *Robotic Intelligence and Automation*, pages 1–16, 2026.

[47] Ryan Paul McKennaa and John Oyekan. A perspective on open challenges in deformable object manipulation. *arXiv preprint arXiv:2602.22998*, 2026. 3

[48] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2023. 3, 6, 7, 8, 18, 19

[49] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint*, 2023. 3, 6, 7, 8, 18, 19

[50] Weikun Peng, Jun Lv, Yuwei Zeng, Haonan Chen, Siheng Zhao, Jichen Sun, Cewu Lu, and Lin Shao. Tiebot: Learning to knot a tie from visual demonstration through a real-to-sim-to-real approach. *arXiv preprint arXiv:2407.03245*, 2024. 3

[51] Kai Wu, Rongkang Chen, Qi Chen, and Weihua Li. Robotic assembly of deformable linear objects via curriculum reinforcement learning. *IEEE Robotics and Automation Letters*, 2025. 3

[52] Checheng Yu, Chonghao Sima, Gangcheng Jiang, Hai Zhang, Haoguang Mai, Hongyang Li, Huijie Wang, Jin Chen, Kaiyang Wu, Li Chen, Lirui Zhao, Modi Shi, Ping Luo, Qingwen Bu, Shijia Peng, Tianyu Li, and Yibo Yuan. χ_0: Resource-aware robust manipulation via taming distributional inconsistencies. *arXiv preprint arXiv:2602.09021*, 2026. 3

[53] Daniel Seita, Aditya Ganapathi, Ryan Hoque, Minho Hwang, Edward Cen, Ajay Kumar Tanwani, Ashwin Balakrishna, Brijen Thananjeyan, Jeffrey Ichnowski, Nawid Jamali, et al. Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[54] Thomas Weng, Sujay Bajracharya, Yufei Wang, Khush Agrawal, and David Held. Fabricflownet: Bimanual cloth manipulation with a flow-based policy. In *Conference on Robot Learning (CoRL)*, 2022.

[55] Yuran Wang, Ruihai Wu, Yue Chen, Jiarui Wang, Jiaqi Liang, Ziyu Zhu, Haoran Geng, Jitendra Malik, Pieter Abbeel, and Hao Dong. Dexgarmentlab: Dexterous garment manipulation environment with generalizable policy. *arXiv preprint arXiv:2505.11032*, 2025.

[56] Huy Ha and Shuran Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robot Learning (CoRL)*, pages 24–33. PMLR, 2022. 3

[57] Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. *ICCV*, 2025. 4, 6, 15, 17, 19

[58] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, 2020. 4

[59] Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)*, 38(6):1–16, 2019. 4

[60] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. In *NVIDIA GPU Technology Conference (GTC)*, volume 3, 2022. 4

[61] Yang Tian, Yuyin Yang, Yiman Xie, Zetao Cai, Xu Shi, Ning Gao, Hangxu Liu, Xuekun Jiang, Zherui Qiu, Feng Yuan, et al. Interndata-a1: Pioneering high-fidelity synthetic data for pre-training generalist policy. *arXiv preprint arXiv:2511.16651*, 2025. 4, 15

[62] John Schulman, Jonathan Ho, Cameron Lee, and Pieter Abbeel. Learning from demonstrations through the use of non-rigid registration. In *Robotics Research: The 16th International Symposium ISRR*, pages 339–354. Springer, 2016. 5, 19

[63] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint*, 2025. 6, 7, 8, 18, 19

[64] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint*, 2024. 10, 22

[65] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. 19, 22

Technical Appendices and Supplementary Material

A State Augmentation Details

A.1 Formal Assumption

Our approach relies on the premise that a well-calibrated physics simulator approximately preserves physical plausibility when evolving from a valid initial state. We formalize this as a working assumption on the simulator $\Phi_{\text{sim}}(\mathbf{s}, \mathbf{f}, \Delta t)$, which evolves state $\mathbf{s}$ under external forces $\mathbf{f}$ over time interval Δt:

Assumption 1 (Approximate conditional closure of $\mathcal{S}_{\text{real}}$)**.** *A sufficiently accurate physics simulator approximately preserves physical plausibility when starting from a valid state, whereas it cannot reliably restore plausibility from an out-of-distribution configuration:*

$$\mathbf{s} \in \mathcal{S}_{\text{real}} \implies \Phi_{\text{sim}}(\mathbf{s},\, \mathbf{f},\, \Delta t) \approx_{\mathcal{S}_{\text{real}}}, \quad \textit{for reasonable } \mathbf{f} \textit{ and } \Delta t; \tag{A1}$$

$$\mathbf{s} \notin \mathcal{S}_{\text{real}} \not\Longrightarrow \Phi_{\text{sim}}(\mathbf{s},\, \mathbf{f},\, \Delta t) \in \mathcal{S}_{\text{real}}. \tag{A2}$$

We note that this assumption is an idealization: real simulators introduce numerical integration errors and may not perfectly model all material properties, so the generated states are plausible *with respect to the simulator's dynamics model* rather than guaranteed to match real-world physics exactly. The use of high-fidelity simulators (PhysTwin [57]) narrows this gap in practice.

A.2 Detailed Analysis of Existing Strategies

We provide a detailed analysis of three representative augmentation strategies and their limitations when applied to deformable objects.

(i) Global rigid transformation. Applying a uniform $\mathbf{T} \in SE(3)$ to all particles preserves all inter-particle relations, so the augmented state remains in $\mathcal{S}_{\text{real}}$. However, the reachable set is confined to a 6-dimensional subspace spanned by rigid pose variations of $\mathbf{s}_0$, which cannot capture any shape or topological variation of deformable objects. This is confirmed empirically in Fig. 5: rigid augmentation produces near-zero non-rigid residuals across all three tasks.

(ii) Per-particle independent perturbation. Adding independent noise $\boldsymbol{\epsilon}_i \sim \mathcal{P}(\sigma)$ to each particle can in principle reach any $\mathbf{s} \in \mathcal{S}$, but faces a practical coverage–plausibility trade-off. Large σ produces disordered point clouds that break inter-particle connectivity and introduce topology artifacts (e.g., self-intersections, disconnected segments), pushing the state far outside $\mathcal{S}_{\text{real}}$ in ways that subsequent stabilization steps typically cannot recover. Small σ preserves local topology but induces only surface wrinkles [61], confining the resulting states to a local neighborhood of $\mathbf{s}_0$ with insufficient diversity for policy learning. Crucially, even when a stabilization step is applied, there is no mechanism to verify whether the result has returned to $\mathcal{S}_{\text{real}}$, making this approach unreliable in practice.

(iii) Kinematic topological transformation. Applying a continuous deformation field $\phi : \mathbb{R}^3 \to \mathbb{R}^3$ improves upon (ii) by preserving topological coherence—the connectivity structure is maintained by construction. However, ϕ is constructed without reference to the object's material model, so the deformed state may violate internal dynamic constraints (e.g., producing rest-shape configurations with unrealistic internal stress or interpenetration with the environment). These configurations are structurally coherent but dynamically inadmissible, and stabilization cannot reliably project them back onto $\mathcal{S}_{\text{real}}$ because the simulator's corrective dynamics may converge to a different basin or fail to converge at all.

A.3 Advantages of Dynamics-Based Augmentation over (ii) and (iii)

Beyond broader coverage relative to rigid transformations, our dynamics-based approach offers two practical advantages over per-particle perturbation and kinematic deformation:

- **No plausibility–diversity trade-off.** Both strategies (ii) and (iii) face a fundamental tension: increasing perturbation magnitude increases diversity but also the likelihood of producing implausible configurations. Our method sidesteps this trade-off because diversity is achieved through the simulator's own dynamics—larger or longer-duration forces naturally produce more diverse states, while the simulation's internal constraints (collision handling, material constitutive laws, boundary conditions) continuously enforce plausibility throughout the trajectory. There is no separate perturbation-then-repair pipeline that could fail.
- **Implicit enforcement of coupled constraints.** Deformable objects are subject to multiple interacting constraints simultaneously: material elasticity, self-collision avoidance, environmental contact, and gravitational settling. Strategies (ii) and (iii) perturb geometry without awareness of these coupled constraints, and a subsequent stabilization step can at best enforce them approximately and sequentially. In contrast, forward simulation enforces all constraints jointly at each time step through the simulator's integrated solver, producing states where internal stresses, contact forces, and boundary conditions are mutually consistent. This is particularly important for objects with complex rest-state interactions (e.g., a rope draped over a fixture, or cloth resting on a surface with folds), where violating one constraint easily cascades into violations of others.

A.4 Reachable Set Discussion

The set of states reachable via dynamics simulation from $\mathbf{s}_0$ is:

$$\mathcal{R}(\mathbf{s}_0) \;=\; \left\{ \Phi_{\text{sim}}(\mathbf{s}_0,\, \mathbf{f},\, \Delta t) \;\middle|\; \mathbf{f} \in \mathbb{R}^{3N},\; \Delta t > 0 \right\}. \tag{8}$$

In principle, $\mathcal{R}(\mathbf{s}_0)$ could be large, since any physically plausible configuration is connected to $\mathbf{s}_0$ through some physical process. However, we do not claim that our randomized sampling of $(\mathbf{f}, \Delta t)$ achieves full coverage of $\mathcal{S}_{\text{real}}$ or matches the true distribution of real-world configurations. We treat dynamics-based augmentation as a practical sampling heuristic that explores a substantially broader and more physically grounded region of the state space than rigid transformations. This is verified empirically in Sec. 4.3.

B Trajectory Augmentation Details

B.1 Decay Function

The decay function $\alpha_t = \text{decay}(t)$ in the position and orientation warping controls how strongly the deformation field influences the trajectory over time. We support three configurations:

- *None*: $\alpha_t = 1$ for all t. The deformation field is applied uniformly throughout the trajectory.
- *Linear*: $\alpha_t = \max(0,\; 1 - t/T)$, where T is the total trajectory length. The influence decreases linearly to zero.
- *Exponential*: $\alpha_t = e^{-\lambda t}$, where $\lambda > 0$ controls the decay rate. The influence decreases exponentially.

The decay allows the trajectory to closely follow local deformations near the grasp phase while gradually reverting to the original trajectory towards the end of the manipulation phase. The choice of decay function is task-dependent and specified in Appendix C.5.

B.2 KNN Scope for Grasp vs. Manipulation Phases

In practice, we observe that the grasp pose correlates more strongly with object points in the vicinity of the grasp point. Therefore, we employ a small K (e.g., $K = 5$–10) for warping the grasp pose, so that only nearby particle displacements influence the grasp alignment.

Conversely, the manipulation phase depends on the overall object state; the end-effector must compensate not only for local geometry changes but also for global shape shifts. We therefore set K equal to the total number of object points N for the manipulation trajectory, effectively using a globally weighted deformation field.

B.3 Orientation Constraints

Given the tabletop manipulation scenario in our experiments, significant orientation changes occur primarily around the Z-axis (perpendicular to the table surface). We therefore constrain the orientation warping to the Z-axis component: the original rotation matrix R_t and the induced rotation R'_t are first projected onto their Z-axis rotational components before SLERP interpolation is applied. This prevents spurious tilting or flipping of the end-effector that could arise from noisy Jacobian estimates in the other axes.

C Implementation Details

C.1 Simulation and Robot Setup

All experiments are conducted in Real2Sim-Eval [19], which provides physically accurate soft-body dynamics and photorealistic rendering via PhysTwin [57]. The robot is an xArm7 manipulator equipped with two RGB cameras: a fixed third-person camera and a wrist-mounted camera, both at 848×480 resolution and 30 Hz frame rate. The policy outputs 8-dimensional actions consisting of end-effector position (x, y, z), quaternion orientation (q_w, q_x, q_y, q_z), and gripper opening at 30 Hz control frequency. Internally, the simulation converts the policy output to a 13D command (xyz + 3×3 rotation matrix + gripper) before execution.

C.2 Task Descriptions and Success Criteria

Rope routing. The robot must thread a deformable rope through a clip. Success is evaluated over the final 100 frames of each episode. The episode is considered successful if at least 30 frames satisfy the condition that the rope forms sufficient intersections (at least 100 spring-segment crossings) with both the upper and lower planes of the clip, indicating that the rope has been threaded through.

Toy packing. The robot must place a stuffed toy into a container. Success is evaluated at the final frame. We construct a minimum oriented bounding box (OBB) from the initial reference mesh and scale it by a factor of 1.05. The episode succeeds if at least 3,050 object points fall within this scaled OBB.

Cloth folding. The robot must fold a cloth into a triangular shape. Success is evaluated at the final frame. The point cloud is projected onto the table plane to form a binary mask. We extract the largest connected component, fit a minimum bounding triangle, and verify three conditions: (i) the contour has 3–4 approximate vertices, (ii) the IoU between the mask and the fitted triangle ≥ 0.72, and (iii) the mask coverage of the triangle ≥ 0.80.

C.3 State Augmentation Parameters

Since the simulation environment does not expose a direct external-force API, we implement localized physical disturbances by commanding the gripper to execute randomized Cartesian perturbations while in contact with the object, transmitting forces through contact dynamics. Each augmentation episode consists of a sequence of random steps; each step applies either a planar translation (sampled from discrete $\pm x$, $\pm y$ directions) or a z-axis rotation (with probability p_{rot}). Task-specific configurations are as follows:

- **Rope / Toy**: 180 random steps, translation magnitudes sampled from $\{0.012, 0.006, 0.003\}$ m, rotation steps of $\pm 6^\circ$, rotation probability $p_{\text{rot}} = 0.45$.
- **Cloth**: 260 random steps, translation magnitudes sampled from $\{0.018, 0.009, 0.0045\}$ m, rotation steps of $\pm 8^\circ$, rotation probability $p_{\text{rot}} = 0.55$.

After perturbation, the object is stabilized for 30–40 simulation steps to reach quasi-static equilibrium.

Appearance and material variation. DeformGen augments the geometric configuration of deformable objects and is complementary to appearance and material augmentation, with which it can be combined. Our open-source code provides documented interfaces for varying object appearance (e.g., color) and material parameters (e.g., friction).

C.4 Data Splits

For each task, we generate augmented states via dynamics-based topological transformation and attempt trajectory synthesis until obtaining sufficient successful demonstrations. The generation statistics are:

- **Rope**: 1,294 successful trajectories out of 1,300 generated states. Trajectory synthesis success rate: 99.5%. Split into 1,000 training / 200 test. Remaining successful trajectories are unused.
- **Toy**: 1,327 successful trajectories out of 2,200 generated states. Trajectory synthesis success rate: 60.3%. Split into 1,000 training / 200 test, with 200 failed states sampled from 873 available. Remaining successful trajectories are unused.
- **Cloth**: 1,778 successful trajectories out of 4,500 generated states. Trajectory synthesis success rate: 39.5%. Split into 1,000 training / 200 test, with 200 failed states sampled from 2,722 available. Remaining successful trajectories are unused.

Failed states from toy and cloth tasks serve as *hard samples* for the generalization ablation in Sec. 4.4. Rope is excluded from this ablation due to insufficient failed samples.

Data-generation cost. Generating 500 augmented episodes for cloth folding, our most challenging task, including the trajectory syntheses rejected by execution-based verification, takes 1 h 57 min on five NVIDIA RTX 4090 GPUs. Since each state is generated and verified independently, the pipeline parallelizes naturally across multiple GPUs.

C.5 Trajectory Augmentation Hyperparameters

Task-specific trajectory warping configurations:

- **Rope**: Grasp KNN $K = 5$, manipulation KNN $K = N$ (all points), decay function: linear.
- **Toy**: Grasp KNN $K = 5$, manipulation KNN $K = N$, decay function: none.
- **Cloth**: Grasp KNN $K = 10$, manipulation KNN $K = N$, decay function: exponential ($\lambda = 0.02$).

C.6 Policy Training Hyperparameters

All policies are trained on a single NVIDIA A100 GPU. Hyperparameters are tuned per algorithm:

ACT [49]. Learning rate: 1×10^{-5}. Batch size: 512. Training epochs: 10.

Diffusion Policy (DP) [48]. Learning rate: 1×10^{-4}. Batch size: 512. Training epochs: 10. Scheduler: cosine with 500-step warmup.

SmolVLA [63]. Learning rate: 1×10^{-4}. Batch size: 128. Training epochs: 10. Scheduler: warmup 1000.

π_0 **[1].** Fine-tuned via LoRA using the OpenPI framework. Peak learning rate: 2.5×10^{-5}, decay learning rate: 2.5×10^{-6}. Batch size: 8. Training epochs: 10. Optimizer: AdamW (β_1=0.9, β_2=0.95, weight decay 1×10^{-10}). Scheduler: cosine decay.

Training regimes and evaluation. ACT and DP are trained from scratch. SmolVLA is fine-tuned from its publicly released checkpoint with the vision encoder frozen and all remaining parameters updated. π_0 is fine-tuned from its public base checkpoint with LoRA. Table 2 reports the mean and standard deviation over three evaluation seeds (42, 123 and 456).

D Limitations

We acknowledge several limitations of the current work:

- **Single-arm manipulation only.** All experiments use a single arm (xArm7 in simulation and zero-shot deployment, Franka Research 3 for co-training). Extending DeformGen to bimanual or multi-robot settings—where coordination between arms introduces additional constraints on trajectory synthesis—remains future work.
- **Limited task diversity.** We validate on three deformable manipulation tasks (rope, stuffed toy, cloth), which cover a range of material properties (1D, quasi-rigid 3D, 2D sheet). However, other important categories such as dough/clay shaping, surgical tissue manipulation, or cable routing in cluttered environments have not been evaluated. The generality of our dynamics-based augmentation to these domains remains to be demonstrated.
- **Privileged state at data-generation time.** Deformation-field warping uses per-particle object states, which are available in simulation; the trained policies take only RGB images and proprioception as input. Generating data directly from real-world scenes would require source-to-target correspondences, which could be obtained with off-the-shelf non-rigid registration on observed point clouds, whose per-point displacements are exactly the input of our warping.
- **Source demonstration and warping scheme.** Each task uses a single teleoperated demonstration, and we use one deformation-field formulation. Studying the effect of demonstration quality and comparing alternative warping schemes as interchangeable modules are left to future work.
- **Trajectory synthesis is not universally successful.** Guaranteeing successful trajectory transfer for arbitrary initial states is inherently difficult. Following the core philosophy of [62], our warping assumes that geometric correspondence preserves task semantics, but this only holds approximately—complex contact dynamics, large topological changes, and kinematic constraints can cause warped trajectories to fail. This is reflected in our varying success rates. Encouragingly, policies trained on successful trajectories still generalize to some failure states (Appendix G.4). Future work could mitigate this via iterative human-in-the-loop demonstration collection, closed-loop trajectory refinement, or multi-source warping that selects the most compatible demonstration for each target state.

E Broader Impacts

This work aims to reduce the cost of collecting manipulation demonstrations for deformable objects by providing an automated data augmentation pipeline. The positive societal impact includes enabling more accessible and scalable robot learning for tasks involving soft materials (e.g., household assistance, garment handling, food preparation), potentially benefiting applications in elder care and manufacturing.

Data synthesis runs entirely in simulation, and our real-robot experiments use teleoperated demonstrations collected by the authors; the work involves no human subjects or generation of potentially harmful content. We do not foresee direct negative societal impacts beyond standard safety considerations for robotic manipulation systems, such as workspace monitoring and emergency stops when deploying learned policies on physical robots. The trained policies are task-specific manipulation controllers without broader capabilities that could be misused.

F Licenses

We list the licenses of all external assets used in this work:

- **Real2Sim-Eval** [19]: MIT License
- **PhysTwin** [57]: MIT License
- **ACT** [49]: MIT License
- **Diffusion Policy** [48]: MIT License
- **SmolVLA** [63]: Apache License 2.0
- π_0 [1]: Apache License 2.0
- $\pi_{0.5}$ [2]: Apache License 2.0
- **MuJoCo** [65]: Apache License 2.0

Our use of these assets complies with their respective license terms.

G Additional Results

We provide additional qualitative visualizations and the full results of the hard-sample study (Q3) to complement the main paper.

G.1 Trajectory Warping Examples

Figure 8 visualizes the trajectory warping process for representative examples. The source trajectory (blue) is adapted to different object configurations via deformation-field warping, producing target trajectories (orange) that maintain correct grasp alignment and compensate for global shape changes.

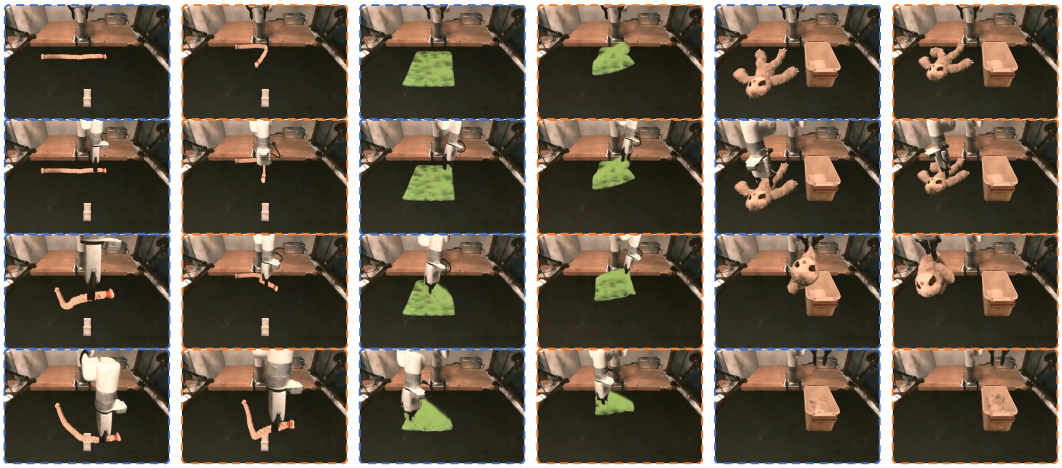


Figure 8: **Trajectory warping examples.** For each task, we show the source trajectory (blue) on the original object state and the warped trajectory (orange) on the augmented state. The deformation field successfully adapts both the grasp pose and the manipulation path to the new geometry.

G.2 Policy Execution Rollouts

Figure 9 shows successful policy execution rollouts on held-out test states for each task, demonstrating that policies trained on DeformGen-augmented data can handle diverse unseen configurations.

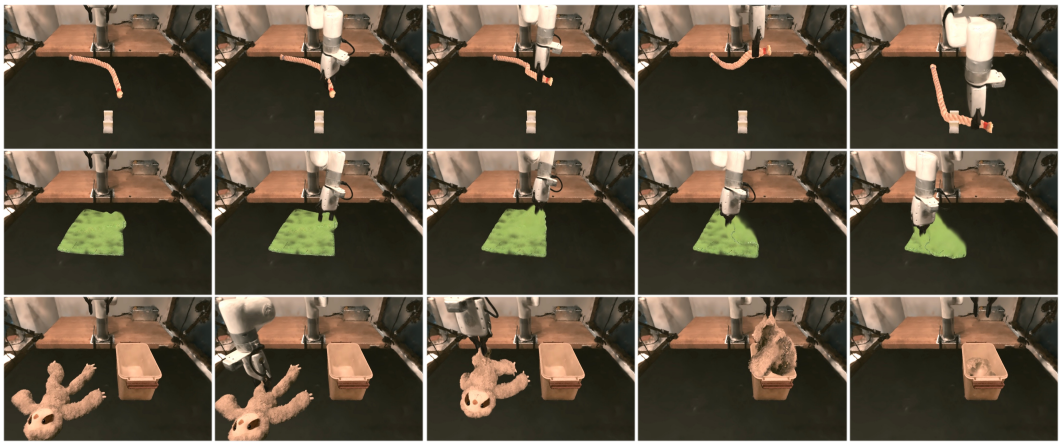


Figure 9: **Policy execution rollouts on unseen test states.** Each row shows key frames from a successful episode. The policies are trained on DeformGen-augmented data and evaluated on held-out object configurations not seen during training.

G.3 Failure Cases

Figure 10 shows representative failure cases to illustrate remaining challenges. Common failure modes include: (i) imprecise grasp on highly deformed configurations where the visual appearance deviates significantly from training data, (ii) premature release due to unstable contact under large deformations.

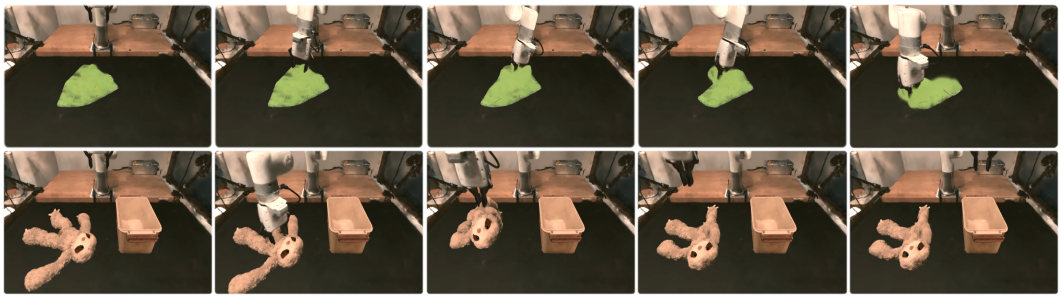


Figure 10: **Representative failure cases.** Common failure modes include grasp misalignment on extreme deformations (Top), and premature release due to contact instability (Bottom).

G.4 Generalization to Synthesis Failure Cases

Q3 (Sec. 4.4): Can the policy generalize to synthesis failure cases? This ablation tests whether the policy merely memorizes the augmented trajectories or learns transferable manipulation strategies. We evaluate on *hard samples*—states where trajectory synthesis itself failed to produce a valid demonstration, meaning the policy has never seen a successful trajectory for these configurations. Since rope achieves nearly 100% synthesis success, this study is conducted on toy and cloth only.

As shown in Table 7, policies trained with augmentation achieve non-trivial success on these out-of-distribution states, though performance varies across tasks and architectures. Augmentation helps policies learn generalizable manipulation strategies rather than memorizing individual demonstrations, enabling some degree of extrapolation to unseen configurations. However, the inconsistent trends suggest that generalization to synthesis failure cases remains architecture-dependent and challenging.

Table 7: Success rate (%) on **hard samples** (synthesis failed). Rope excluded (near-100% synthesis success). **SMG***: rigid state aug. **DG***: topological state aug. + rigid transfer. **DG**: full DeformGen.

Task	ACT			DP			SmolVLA			π_0		
	SMG*	DG*	DG	SMG*	DG*	DG	SMG*	DG*	DG	SMG*	DG*	DG
Toy	45.50	37.50	**55.50**	37.50	**55.00**	47.00	15.00	15.00	**18.50**	7.50	18.50	**36.50**
Cloth	**11.00**	6.00	5.50	**6.00**	5.00	5.00	**17.50**	16.50	10.00	11.50	**19.00**	4.00
Average	28.25	21.75	**30.50**	21.75	**30.00**	26.00	**16.25**	15.75	14.25	9.50	18.75	**20.25**

H Real-Robot Experiments

Zero-shot sim-to-real. The robot is an xArm7 with a wrist-mounted Intel RealSense D405 and a fixed D455 camera, matching the simulated setup. We deploy the π_0 policies fine-tuned only on DeformGen-synthesized simulation data (Sec. 4.1) for cloth folding, rope routing and toy packing, without collecting any real-world demonstrations of these tasks, i.e., zero-shot sim-to-real transfer with no real data from the deployment scene, and the public π_0 checkpoint as a reference. Both policies start from the same initial configurations. Since the public checkpoint rarely reaches the rope or the toy, we quantify the comparison on cloth folding, where both policies show the intended task-level behavior: we run 10 trials per policy and record whether the trajectory shows the intended task-level behavior (trend), whether the gripper makes contact with the cloth, and task success under the same criterion as in simulation (Table 5). On cloth folding, the public checkpoint typically closes the gripper above the thin cloth, whereas the DeformGen-trained policy grasps a corner of the cloth and folds it onto the opposite corner. On rope routing, the public checkpoint usually misses the rope, whereas the DeformGen-trained policy grasps, routes and inserts the rope into the clip. On toy

packing, the public checkpoint moves toward the container without reaching the toy, whereas the DeformGen-trained policy grasps the toy, lifts it and places it into the container.

Co-training. The co-training experiments use a Franka Research 3 arm with a wrist-mounted ZED Mini and a front-facing Intel RealSense D435i as policy inputs. We fine-tune $\pi_{0.5}$ [2] from the `pi05_droid` checkpoint, trained on the DROID dataset [64], as a single multi-task policy on rope insertion, cloth folding and toy packing, with 50 teleoperated real demonstrations and 1,000 DeformGen demonstrations per task. The baseline uses the same recipe with the real demonstrations only. DeformGen demonstrations are converted from the simulated xArm7 to the Franka as follows. (i) We extract the end-effector trajectory (position, quaternion and gripper opening) of each synthesized episode. (ii) We anchor the trajectory at the initial tool-center-point pose of the real robot for the same task and solve inverse kinematics frame by frame in MuJoCo [65] to obtain the seven Franka joint angles q_t. (iii) Real and simulated data share one representation: the state consists of the seven joint angles and the gripper opening, and the action consists of the seven joint velocities and the absolute gripper opening, where simulated trajectories are downsampled from 30 Hz to 15 Hz to match DROID and joint velocities are computed as $\dot{q}_t = 15\,(q_{t+1} - q_t)$. (iv) The simulated episodes keep their rendered xArm7 images. During training, simulated and real episodes are sampled at a ratio of about 80:20, and normalization statistics are computed for this mixture. Each policy is evaluated for 30 trials per task (results in Table 6). Figure 11 shows key frames from successful rollouts of the co-trained policy.

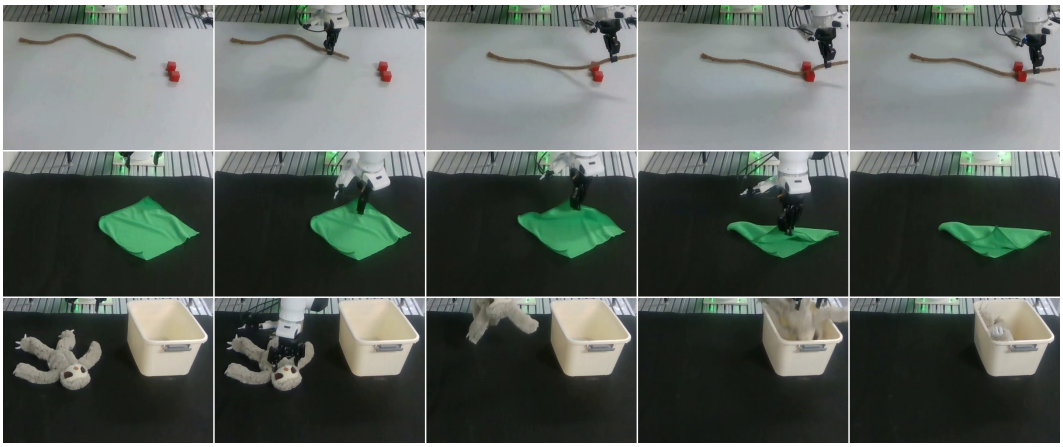


Figure 11: **Co-training rollouts (Franka Research 3).** Each row shows key frames from a successful episode of $\pi_{0.5}$ co-trained with DeformGen data (rope insertion, cloth folding, toy packing).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state three contributions (formulation, pipeline, empirical evidence) that are directly supported by the method description in Sec. 3 and experimental results in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in the Appendix, including single-arm manipulation, the set of evaluated tasks, the reliance on privileged simulator state for data generation, and the varying trajectory synthesis success rates across tasks.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include formal theorems or proofs. The method section provides mathematical formulations (e.g., deformation field construction) but these are definitions rather than theoretical claims requiring proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sec. 4.1 and the Appendix provide the simulation environment, robot setup, observation/action spaces, augmentation parameters, data splits, success criteria, and per-policy hyperparameters. Code is available at `https://github.com/Zili2002/DeformGen`.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The complete code, with instructions for data generation, training and evaluation, is publicly available at https://github.com/Zili2002/DeformGen.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Key details are in Sec. 4.1; full hyperparameters (learning rates, batch sizes, schedulers, training epochs) and data splits per task are provided in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We run the main experiments with multiple seeds (42, 123, 456) and report the mean and standard deviation in Table 2 to reflect statistical significance.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources in the supplementary.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work involves robot manipulation in simulation and on real robots, with teleoperated demonstrations collected by the authors and no human subjects, private data, or dual-use concerns. It conforms fully with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss broader impacts in the Appendix. The positive impact is reducing the cost of robot learning for deformable manipulation. We do not foresee direct negative societal impacts beyond standard safety considerations for deploying learned policies on physical robots.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released assets (augmentation code and task-specific manipulation pipelines) pose no high risk for misuse and contain no private data.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite all used frameworks (Real2Sim-Eval, PhysTwin, ACT, Diffusion Policy, SmolVLA, π_0, $\pi_{0.5}$, MuJoCo) and their original papers. License information is provided in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The DeformGen codebase is released with documentation at `https://github.com/Zili2002/DeformGen`, and the paper and appendix describe the pipeline in sufficient detail for independent reimplementation.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This work does not involve human subjects research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a component of the core methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.