

SIM1: Physics-Aligned Simulator as Zero-Shot Data Scaler in Deformable Worlds

Yunsong Zhou^{*,†,1}, Hangxu Liu^{*,1,2}, Xuekun Jiang^{*,1}, Xing Shen^{*,1},
Yuanzhen Zhou¹, Hui Wang^{1,3}, Baole Fang¹, Yang Tian^{1,4}, Mulin Yu¹, Qiaojun Yu¹,
Li Ma¹, Hengjie Li¹, Hanqing Wang¹, Jia Zeng¹ and Jiangmiao Pang^{†,1}
¹Shanghai AI Lab, ²Fudan University, ³Shanghai Jiao Tong University,
⁴Peking University

Robotic manipulation with deformable objects represents a data-intensive regime in embodied learning, where shape, contact, and topology co-evolve in ways that far exceed the variability of rigids. Although simulation promises relief from the cost of real-world data acquisition, prevailing sim-to-real pipelines remain rooted in rigid-body abstractions, producing mismatched geometry, fragile soft dynamics, and motion primitives poorly suited for cloth interaction. We posit that simulation fails not for being synthetic, but for being ungrounded. To address this, we introduce **SIM1**, a physics-aligned real-to-sim-to-real data engine that grounds simulation in the physical world. Given limited demonstrations, the system digitizes scenes into metric-consistent twins, calibrates deformable dynamics through elastic modeling, and expands behaviors via diffusion-based trajectory generation with quality filtering. This pipeline transforms sparse observations into scaled synthetic supervision with near-demonstration fidelity. Experiments show that policies trained on purely synthetic data achieve parity with real-data baselines at a 1:15 equivalence ratio, while delivering 90% zero-shot success and 50% generalization gains in real-world deployment. These results validate physics-aligned simulation as scalable supervision for deformable manipulation and a practical pathway for data-efficient policy learning.

[Code](#) | [Data](#) | [Homepage](#) | [Demo](#)

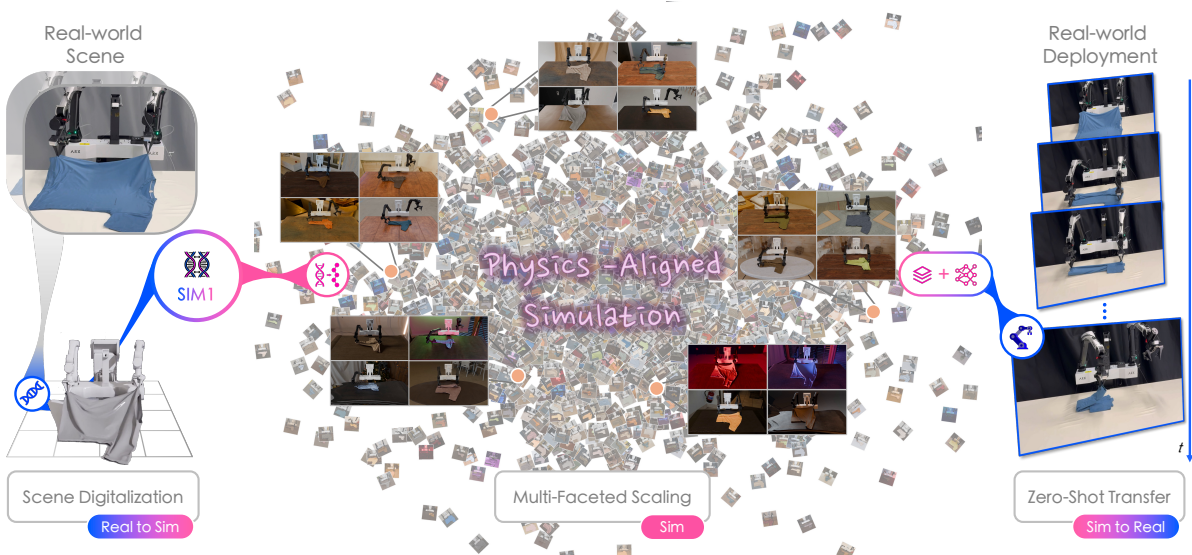


Figure 1. **SIM1** pioneers *real-to-sim-to-real* data generation for deformable manipulation. It constructs simulation data whose deployment behavior is the **same one** as reality, enabling zero-shot transfer and scalable performance on physical robots.

1. Introduction

Dexterous manipulation of deformable objects, notably garments, underpins a wide spectrum of daily human activities and remains a frontier for robotics. Recent advances (Bjorck et al., 2025; Black et al., 2024; Cai et al., 2026; Intelligence et al., 2025) reveal a clear scaling trend whereby expanding real-robot datasets consistently improves performance, reinforcing the growing consensus on data scaling as a primary driver of embodied learning. While this paradigm shows promise in rigid-object settings, deformable manipulation intensifies the hunger for data, as its evolving geometry and contact-rich dynamics demand substantially broader state and visual coverage. Yet, acquiring real-world manipulation data at scale remains prohibitively expensive. Such efforts demands skilled operators, specialized hardware, and extensive human labor, placing practical limits on dataset scale. Consequently, vision-language-action models trained on limited deformable-interaction data exhibit constrained generalization.

In the field, sim-to-real (S2R) synthetic data generation has emerged as a compelling strategy for scaling manipulation data (Deng et al., 2025; Gu et al., 2023; He et al., 2025; Xue et al., 2025; Ye et al.). In rigid-object domains, increasing synthetic diversity often translates into measurable real-world gains, supported by rich asset libraries and automated generation pipelines (Chen et al., 2025a; Nasiriany et al.; Tian et al., 2025). However, this paradigm breaks down in deformable manipulation. *Simulated scenes* are often weakly aligned, or not aligned at all, with real environments, relying on coarse calibration or manually constructed assets without metric fidelity (Gao et al.; Li et al.; Lu et al.; Wang et al., 2024). Such geometric inconsistencies are amplified under soft-body deformation. *Physics engines* are predominantly optimized for rigid-body dynamics, producing unstable or inaccurate responses for cloth and other soft materials (Chen et al., 2025a; Deng et al., 2025; Zhang et al.). *Behavior generation* follows rigid-object paradigms based on grasp-point and simple pick-and-place primitives, rendering these strategies inadequate for garment manipulation (Yang et al., 2025). As a result, synthetic data serves primarily as pre-training signals, while reliable performance still depends on real-world post-training.

We contend that the first principle of simulation is grounding; scaling becomes valuable only once simulated physics correspond to reality. Real-to-sim (R2S) alignment is thus foundational for deformable manipulation, prioritizing correspondence between simulated and physical dynamics over superficial realism or asset import (Tian et al., 2025; Yin et al.). This motivates a shift from post-hoc adaptation to alignment-first simulation, raising a central question: *how can simulation be grounded so that synthetic data is a reliable substrate for real-robot deployment?*

To address this, we advocate that a real-to-sim-to-real (R2S2R) paradigm offers a principled path toward real-equivalent simulation by aligning each stage with the physical world, thereby enabling direct transfer to real robots (Torre et al., 2024). As depicted in Figure 1, we present **SIM1**, the *first* physics-aligned simulation pipeline for deformable manipulation that digitalizes real scenes (R2S) and generates scalable, unbiased synthetic data consistent with real-world settings (S2R). By expanding limited demonstrations into diverse training data, **SIM1** enables policies that transfer directly to real-robot deployment without additional tuning.

Specific designs operationalize the R2S2R paradigm across three complementary alignment stages. For *geometric alignment*, high-precision 3D scans are reconstructed into metric-accurate, textured meshes, producing simulation-ready digital representations of real-world scenes. For *dynamical alignment*, a stabilized soft-body solver (Giles et al., b) enforces physically consistent elastic and bending responses while suppressing excessive deformation, thereby enabling realistic interaction modeling. A coupled simulation infrastructure maps robot operations to simulation, supporting parameter calibration and stable soft-body manipulation. For *movement alignment*, deformable

manipulation trajectories are synthesized through structured two-stage planning that decouples interaction from motion. Diffusion-based trajectory generation models human-like behavior (Chen et al., b), while automatic filtering and appearance randomization (Tobin et al.) enhance diversity and robustness. These components transform simulation into a real-equivalent data source for scalable generation to real robots.

We summarize our contributions as follows: 1) We introduce **SIM1**, which minimizes the sim-to-real gap through a physics-aligned R2S2R paradigm, enabling synthetic data to serve as high-fidelity training data for direct deployment in deformable manipulation. 2) We enhance simulation fidelity and data utility through metric-accurate scene digitization, a deformation-stabilized solver with physics-based calibration, and a diffusion-based motion framework coupled with filtering to generate high-quality manipulation data. 3) Experiments on $\pi_{0.5}$ and π_0 achieve zero-shot success rates of 90% and 76%, with generalization gains of +50% and +56% over real-data baselines. Furthermore, 15 synthetic samples from **SIM1** provide training value comparable to a real demonstration, validating its effectiveness as a data scaler for deformable manipulation.

2. Related Work

Data scaling and manipulation datasets. Scaling real-robot data improves vision-language-action models (VLAs) (Bjorck et al., 2025; Black et al., 2024; Cai et al., 2026; Cheang et al., 2024; Chen et al., 2025b; Intelligence et al., 2025; Kim et al., 2024; Li et al., 2026; Yang et al., 2026; Yu et al., 2026). Large-scale datasets such as Open X-Embodiment (Collaboration et al., 2023), DROID (Khazatsky et al., 2024), and BridgeData v2 (Walke et al.) advance rigid and articulated manipulation, while simulation datasets (e.g., ManiSkill2 (Gu et al., 2023), RoboCasa (Nasiriany et al.), Genie Sim 3.0 (Yin et al.)) extend data coverage for rigid scenarios. However, deformable manipulation remains underrepresented: soft-body dynamics introduce continuous shape variation and contact-rich behaviors that are more data-hungry than rigid-object tasks. Our work addresses this gap by using simulation to synthesize scalable deformable demonstrations, expanding training data specifically for soft-object manipulation.

Simulation-to-real synthetic data generation. Recent methods generate large-scale demonstrations via transformation (Jiang et al., b; Mandlekar et al., 2023), R2S alignment (Torne et al., 2024), and physical twin reconstruction (Jiang et al., a), achieving strong results in rigid manipulation and emerging capabilities for humanoid (He et al., 2025; Xue et al., 2025) and deformable tasks (Yu et al., 2025). Nevertheless, S2R degradation persists because simulated dynamics are not fully aligned with real-world physics. Our R2S2R framework addresses this by grounding simulation in physical reality through mesh-level alignment, physics-faithful dynamics solving and human-like trajectory generation, producing synthetic data that supports zero-shot transfer to real robots.

Real-to-simulation asset digitization. The digitization of deformable objects employs MPM (Chen et al., c), spring-mass models (Jiang et al., a), and platforms such as GarmentLab (Lu et al.), which integrate multiple physics engines. However, solver accuracy remains limited: VBD suffers from unrealistic stretching (Chen et al., a), while PBD and FEM involve trade-offs between accuracy and efficiency (Bridson et al.; Müller et al.). Recent solvers (Giles et al., b) improve strain limiting but remain isolated from broader pipelines. While existing solvers achieve accurate offline deformation through particle-state optimization, they are not designed for the real-time requirements of embodied manipulation where rigid-soft interaction must be updated dynamically during control. Our solver and calibration infrastructure support online rigid-soft coupling with stable deformation dynamics, enabling high-fidelity simulation and data generation for deformable manipulation.

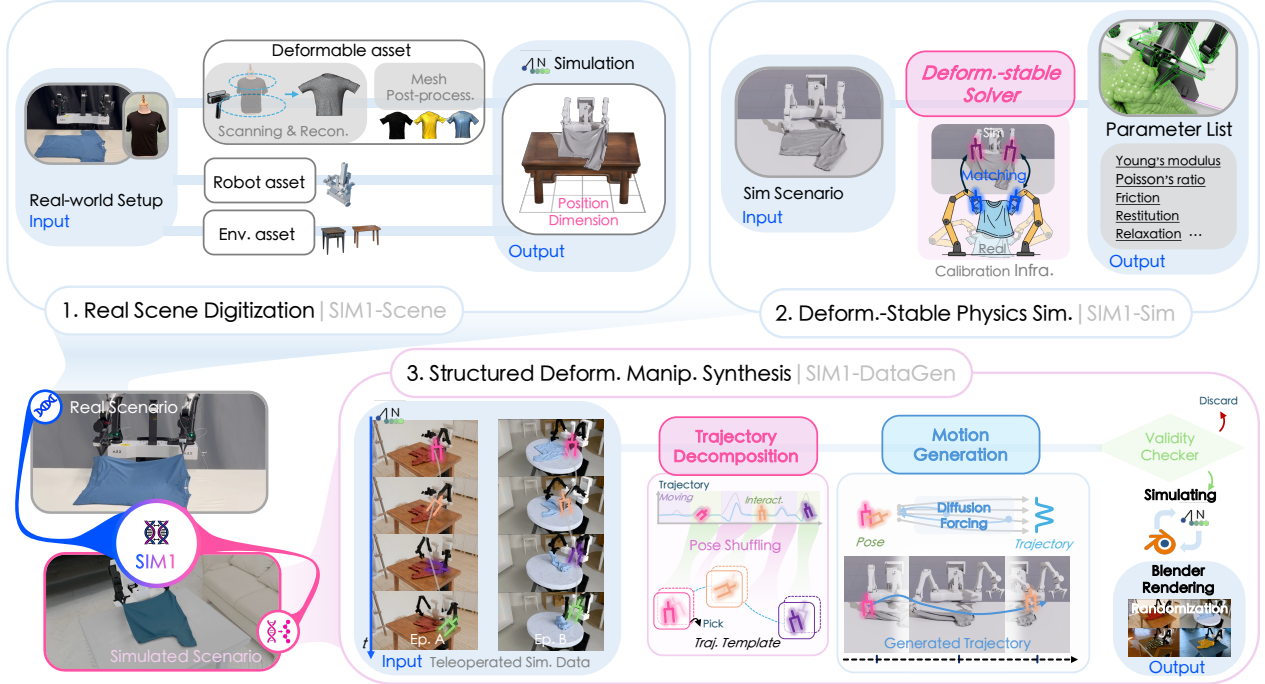


Figure 2. **Framework of SIM1.** (1) Real-world objects are reconstructed into metric-accurate, textured simulation assets; (2) They are then executed within a deformation-stable simulation framework calibrated through real-to-sim behavior matching. (3) Upon physical alignment, diverse manipulation trajectories are synthesized via structured subtask decomposition and diffusion-based motion generation, and rendered with appearance randomization to produce real-equivalent synthetic training data.

3. SIM1 Framework

SIM1 adopts the real-to-sim-to-real (R2S2R) paradigm to bridge geometry, dynamics, and motion across stages. This approach addresses the asymmetry of one-way simulation or reconstruction methods and enables synthetic data to function as a real-equivalent substrate for robot learning (Figure 2). High-precision scans (e.g., garment meshes) and object imports (e.g., URDFs, environment assets) are converted into metric-accurate digital scenes, providing simulation-ready geometric configurations (Section 3.1). Within the aligned simulator, a deformation-stabilized solver and parameter calibration infrastructure reproduce realistic dynamics, enabling interactive soft-body manipulation with calibrated physics (Section 3.2). Demonstration data from teleoperated simulation are first decomposed into motion segments and subsequently synthesized via diffusion, with visual randomization used to generate scaled training data that enhances generalization (Section 3.3).

3.1. SIM1-Scene: Real Scene Digitization

The first stage establishes metric-accurate geometric alignment as a prerequisite for closing the sim-real gap. Since even minor discrepancies in shape, scale, or spatial configuration can propagate into dynamic and contact errors, we construct static simulation assets that faithfully reproduce their real-world counterparts. In particular, recovering high-quality meshes for deformable clothing is complicated by fine wrinkles and intricate topology. We address this challenge by combining high-precision 3D scanning with dedicated mesh post-processing to obtain textured and dimensionally accurate models. Robots and static environments are incorporated using calibrated URDF imports

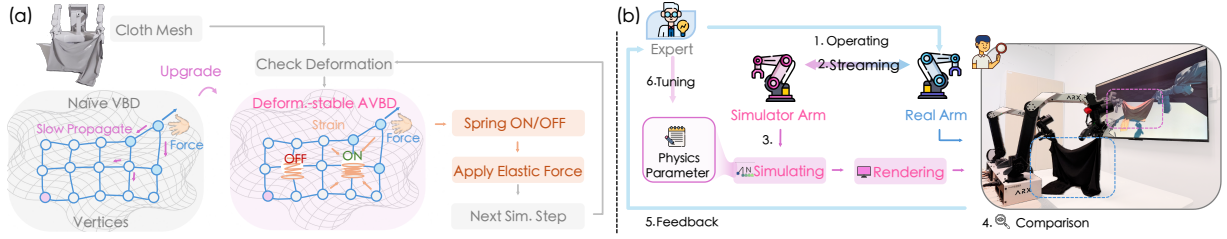


Figure 3. **Paradigm of deformation-stable physics simulation.** (a) After naive VBD (Chen et al., a) updates under external forces, edge deformation is monitored and virtual elastic constraints are activated when stretch exceeds a threshold, injecting strain forces that accelerate convergence toward physically plausible cloth configurations. (b) A bidirectionally synchronized simulation infrastructure replaces identical dual-arm executions in simulation and aligns deformation behaviors through visual calibration.

and asset libraries. The following sections detail the digitization for each asset type.

Deformable assets. For garments, we employ a professional 3D scanner (EinScan Rigil Pro) to capture high-fidelity meshes and textures. The garment is mounted on a mannequin to maintain its natural shape during scanning. Multi-view RGB images and LiDAR scans are captured and fused to generate a dense point cloud. As the scan includes the mannequin, we perform a manual segmentation step to remove mannequin points, retaining only the garment. The resulting point cloud is then processed through surface refinement (e.g., Poisson reconstruction (Kazhdan and Hoppe)) followed by mesh post-processing, including hole filling, smoothing, and remeshing to obtain a clean, watertight mesh suitable for simulation. Texture is mapped from the RGB images onto the static mesh, resulting in a textured OBJ asset. This process yields a geometric replica of the real garment with submillimeter precision.

Robot assets. The robot used in this study is the ARX ACONE robot, a bimanual platform designed for dexterous manipulation tasks. Its kinematic structure, joint limits, collision geometries, and visual meshes are defined in a URDF file generated from CAD models (e.g., SolidWorks) provided by the manufacturer. We directly import this URDF into our simulation environment, ensuring that each arm’s degrees of freedom and workspace exactly match the real hardware. No additional scanning is required; however, we verify dimensional accuracy and adjust the root transform to align with the world coordinate frame. The two arms are calibrated relative to each other to preserve correct bimanual coordination.

Environment assets. Static objects in the environment (i.e., diverse tables) are obtained from publicly available 3D model repositories or created manually. These assets are imported as mesh files and placed in the scene at positions and orientations that replicate the real-world setup. Dimensions are scaled according to real-world measurements to maintain physical consistency.

3.2. SIM1-Sim: Deformation-Stable Physics Simulation

Achieving reliable S2R transfer for deformable manipulation requires physically consistent rigid–soft coupling, a setting that remains poorly supported by existing simulation engines. Modern physics solvers are typically optimized for either rigid-body (widely used in robotics (Chen et al., 2025a; Li et al.; Wang et al., 2024)) or soft-body (largely studied in graphics (Bridson et al.; Han et al.; Huang et al.; Li et al., 2020)), but not their interaction. In soft-body solvers, deformation is resolved through iterative updates of particle vertex states, where contact forces propagate gradually across the mesh (Figure 3 (a)). While accurate deformation can emerge with sufficient solver iterations in offline

simulation, embodied manipulation requires cloth states to update in lockstep with robot motion during online teleoperation. Limited force propagation and delayed strain updates, therefore, lead to excessive stretching, unstable contact, and misalignment with real worlds (Section B). To close this gap, we design a deformation-stable solver that constrains strain evolution during contact, enabling stable soft-body dynamics under real-time interaction, and establish a calibration infrastructure that aligns simulated and real behaviors under identical control inputs.

Stabilized solver via Augmented Vertex Block Descent. We develop a deformation-stable solver inspired by the Augmented Vertex Block Descent (AVBD) formulation (Giles et al., b), extending the Newton–VBD solver (Chen et al., a). Instead of relying solely on energy minimization to update vertex positions, our solver introduces explicit strain constraints between vertices during optimization. These constraints act as adaptive elastic links that rapidly propagate forces across the cloth mesh, allowing vertex motion to remain consistent with gripper dynamics. This design stabilizes rigid–soft interaction while preserving the efficiency required for large-scale simulation data generation.

Solver workflow. As illustrated in Figure 3 (a), the solver operates on a cloth mesh whose vertices represent particles connected by edges. When external forces are applied (e.g., robot contact), vertex positions are first updated through the standard Newton–VBD iteration. After each update, we examine the deformation of every mesh edge. If the distance between two connected vertices exceeds a predefined stretch threshold, a virtual elastic constraint is activated. This constraint introduces an additional tensile force between the vertices, which is injected into the VBD optimization and accelerates convergence toward physically plausible configurations. The procedure iterates until vertex positions stabilize.

Strain constraint. Consider an edge $\mathbf{e}$ connecting vertices i and j with rest length l_0 . Let $\mathbf{e}_i, \mathbf{e}_j \in \mathbb{R}^3$ denote their current positions. We define a maximum stretch ratio $\xi > 0$ such that the edge length should not exceed $(1 + \xi)l_0$. The strain constraint is therefore:

$$C(\mathbf{e}) = \|\mathbf{e}_i - \mathbf{e}_j\| - (1 + \xi)l_0 \leq 0. \quad (1)$$

The constraint becomes active when $C(\mathbf{e}) > 0$, indicating that the local deformation exceeds the allowable limit.

Constraint energy formulation. When activated, we apply an additional energy term that penalizes excessive stretch:

$$E_{\text{strain}}^{(n)}(\mathbf{e}) = \begin{cases} \frac{1}{2}k^{(n)}C(\mathbf{e})^2 + \lambda^{(n)}C(\mathbf{e}), & C(\mathbf{e}) > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where $k^{(n)}$ is the penalty stiffness parameter at Newton iteration n , and $\lambda^{(n)}$ is the Lagrange multiplier accumulating constraint forces. The gradient of $E_{\text{strain}}^{(n)}$ are added to the Newton system and solved within the VBD vertex updates. Intuitively, this term acts as a virtual spring that activates only when excessive stretching occurs, injecting corrective forces that guide vertices toward physically consistent configurations.

Parameter update. After each Newton iteration, the penalty stiffness and variables are updated to progressively enforce the strain constraint:

$$k^{(n+1)} = \min(k_{\text{max}}, k^{(n)} + \beta|C(\mathbf{e})|), \quad \lambda^{(n+1)} = \lambda^{(n)} + k^{(n)}C(\mathbf{e}), \quad (3)$$

where k_{max} is the maximum stiffness and β controls the ramping rate. This update integrates correction forces over iterations, preventing explosive stretching and improving deformation stability during contact-rich manipulation.

Simulation infrastructure for parameter calibration. Physical parameters list $\Theta = \{\rho, E, \nu, \mu, \eta, \zeta\}$ (density, Young’s modulus, Poisson’s ratio, friction, restitution, relaxation) cannot be recovered to their true physical values through direct optimization. Instead, we calibrate simulation by aligning its behavior with real observations. Figure 3 (b) shows the calibration pipeline following a bidirectional workflow. An expert operates the dual-arm robot to execute representative manipulation sequences. The robot’s joint states are streamed to the simulator so that the simulated twin reproduces identical motions. The renderings are visually compared with real executions, allowing experts to assess discrepancies in draping, folding, and contact behavior. Based on this visual feedback, parameters Θ are iteratively adjusted until simulated interactions exhibit behavior that matches the real system at an operational level. This process does not guarantee recovery of true physical parameters, but it establishes behavioral consistency: simulated scenes respond to manipulation in ways that closely resemble reality.

3.3. SIM1-DataGen: Structured Deformable Manipulation Synthesis

Rigid-object manipulation can often rely on trajectory slicing and recombination of motion primitives (Jiang et al., b; Mandlekar et al., 2023). For deformable objects, however, contact dynamics are state-dependent and highly non-deterministic: valid grasp locations cannot be reliably detected, and naive slicing breaks interaction fidelity. To address this, we generalize trajectory decomposition by decoupling interaction from motion (Figure 2 right-bottom). Grasp configurations are directly reused from expert demonstrations; selection is randomized to increase diversity while preserving the original ordering. Motion between interactions is synthesized via diffusion-based generation, learning human-like transitions from demonstration data. Visual randomization further expands appearance diversity, producing large-scale training data suitable for policy learning (Section C).

Trajectory decomposition. Given demonstrations $\mathcal{D} = \{\tau_i\}$ collected via teleoperation, each trajectory is segmented into *interacting* and *moving* regions. *Interacting* segments encode stable grasp configurations and contact states; these are preserved and pooled without modification. To generate new demonstrations, we randomly pick segments from this pool, reusing templated grasp poses while varying sequence order and task context. This shuffling operation expands behavioral diversity while maintaining physically valid interactions, addressing the lack of reliable grasp detection under deformation.

Diffusion-based motion generation. Between consecutive *interacting* segments, we generate smooth transitions by treating trajectory synthesis as sequence completion. Given picked boundary poses $(\mathbf{p}_s, \mathbf{p}_e)$ and robot history $\mathbf{h}$, the model predicts intermediate motions that satisfy physical and kinematic consistency. We employ conditional diffusion forcing (Chen et al., b), where a transformer sequence model reconstructs trajectories from partially corrupted tokens. The diffusion process applies stochastic masking with noise level $\mathbf{k}$, and learning proceeds by recovering the original sequence from soft-corrupted inputs. Formally,

$$\forall \mathbf{k} \in (0, 1]^T, \min_{\theta} \mathbb{E} \|\epsilon - \epsilon_{\theta}(g(\mathbf{x}^0, \mathbf{k}); \mathbf{h}, \mathbf{p}_s, \mathbf{p}_e, \mathbf{k})\|_2^2, \quad (4)$$

where g denotes the corruption function that applies noise to the clean sequence $\mathbf{x}^0$ according to noise $\mathbf{k}$, and ϵ_{θ} is a transformer predicting the noise residual conditioned on history and keyposes. This formulation enables physically consistent motion synthesis that bridges interaction segments while preserving realism.

Validity checking. Even with optimization, rigid–soft interaction is inherently uncertain: gripper contacts with deformable objects can produce penetration or adhesion, yielding physically implausible trajectories.

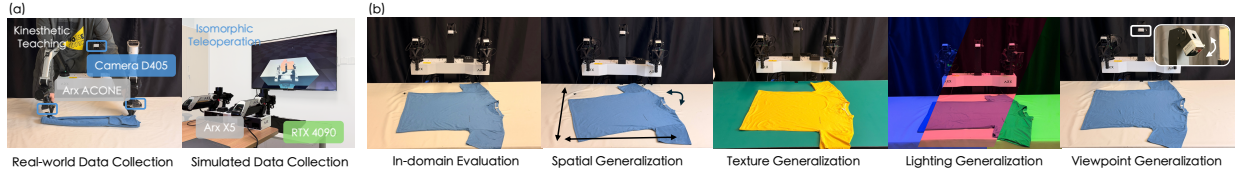


Figure 4. **Illustration of data collection and evaluation.** (a) Real-world and simulated data collection via kinesthetic teaching and isomorphic teleoperation on Arx Acone and Arx X5. (b) Domain settings for in-domain and out-of-domain evaluation in real-world experiments. Representative long-horizon T-shirt folding task (over 20 seconds) illustrating complex sequential manipulation capabilities.

To filter such failures, we first perform lightweight state-based filtering using garment particle states. From simulation-derived positive and negative trajectories, we leverage vibe coding to synthesize threshold rules over particle statistics, defining admissible regions that favor positive states and exclude negative ones. This step efficiently prunes invalid samples early. We then train a binary video discriminator D on head-view RGB observations $\mathbf{V}$ to distinguish valid demonstrations from low-quality samples. Negative cases arise naturally during simulation, avoiding manual annotation. A ResNet-18 feature extractor (He et al.) and Transformer encoder (Vaswani et al.) aggregate temporal information and output a validity score $s = D(\mathbf{V})$. Trajectories with $s > \tau_{\text{disc}}$ are retained; others are discarded. This lightweight filtering removes implausible demonstrations without costly physics-based validation.

Visual randomization. Valid trajectories are rendered in Blender (ble, 2026) with appearance randomization of materials, lighting, and camera parameters. Multiple variations are generated per trajectory using cycle path tracing to produce photorealistic RGB images synchronized with trajectory timestamps. The final dataset combines rendered observations with robot states and actions in the LeRobot format (Cadene et al., 2024) for imitation learning.

4. Experiments

We evaluate whether a physics-grounded simulation pipeline can serve as a scalable and reliable substitute for real-world data in structured deformable manipulation. Our experiments focus on S2R transfer, cross-domain generalization, and data scaling efficiency under purely simulation-trained policies. Specifically, we investigate the following questions. **Q1:** Can models trained solely in simulation achieve performance comparable to real-data-trained counterparts? **Q2:** Does simulation-induced diversity improve out-of-domain robustness beyond real-world data? **Q3:** Does synthetic data enable more efficient performance gains through scaling than real-world collection?

4.1. Protocol and Setup

Data collection. Data are collected in both real and simulated environments. In Figure 4 (a), we use an ARX Acone dual-arm platform equipped with parallel-jaw grippers. In the real world, we adopt kinesthetic teaching, in which the operator directly guides the robot’s end-effectors by hand; in total, 1,000 real trajectories are recorded. In simulation, we deploy our Newton-based simulation infrastructure with the deformation-stable AVBD solver on an RTX 4090 workstation, aligning garment geometry, robot kinematics, and camera parameters with the physical setup. A teleoperator controls two ARX X5 arms, whose motions are mapped to the simulated robot in real time. The operator observes rendered simulator views and performs folding through visual teleoperation. This process yields 200 source simulation demonstrations, which serve as seeds for subsequent synthetic data

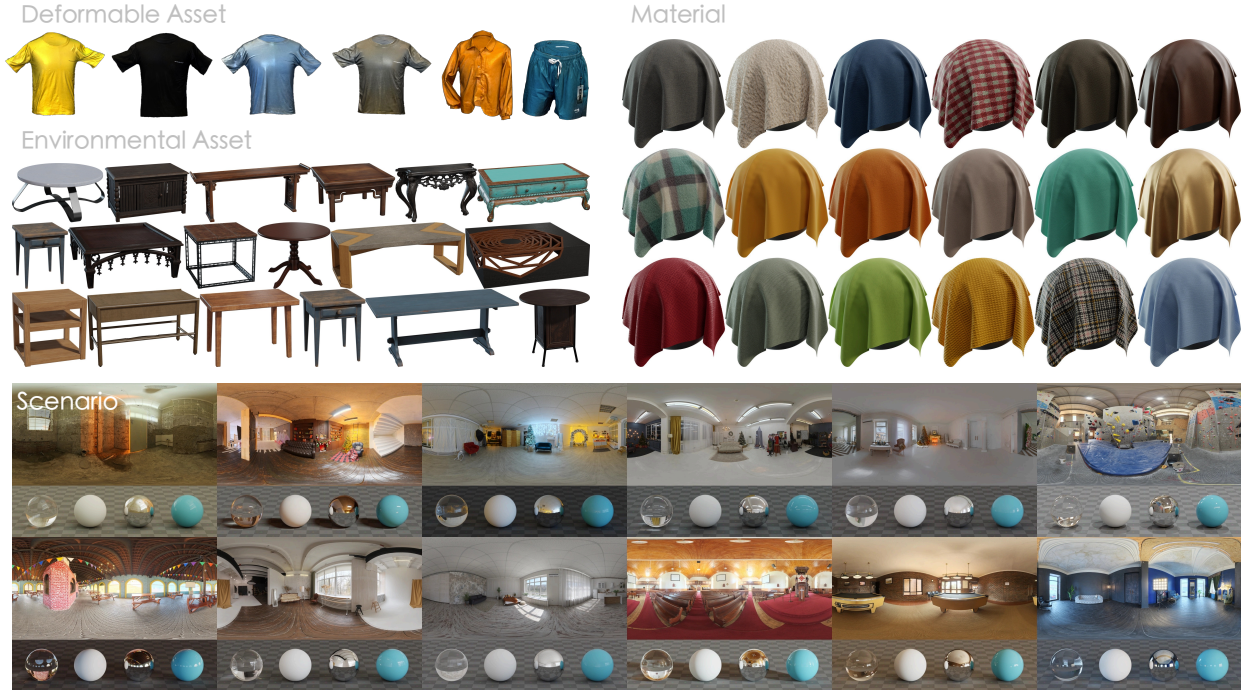


Figure 5. **Illustration of assets used in data generation.** Scanned deformable assets and open-source environmental assets used in simulation (top-left). Diverse garment textures for appearance variation (top-right). Room-scale environments with randomized layouts and lighting for scene-level randomization (bottom).

generation.

Tasks and baselines. T-shirt folding is used as a representative benchmark for real-world validation and does *not* constrain the scope of the proposed simulation pipeline, which targets a broader class of deformable manipulation tasks. We introduce additional manipulation tasks in the following section to further demonstrate the general applicability of our framework. Policies are trained using either real demonstrations or our synthetic data exclusively, and evaluated over 30 trials per configuration. A trial is considered successful if the garment reaches the target folded configuration without dropping or unfolding. We conduct both in-domain and generalization evaluations (Figure 4 (b)). In-domain experiments use the same spatial layout, garment, table surface, lighting, and camera configuration as training. For generalization, we introduce controlled distribution shifts along multiple factors: the garment is randomly translated up to 8 cm and rotated within $\pm 15^\circ$ (spatial); unseen table and garment textures are substituted with corresponding changes in friction properties (texture); illumination direction and intensity are randomized (lighting); and the camera elevation is perturbed within $\pm 5^\circ$ (viewpoint).

Synthetic data generation. For synthetic data generation, we follow the pipeline in Section C. The framework supports diverse asset combinations and environment variations; in this work, we use a representative configuration for real-world validation while demonstrating the general capability of the approach. To enhance diversity during synthesis, the garment is translated within 5 cm and rotated within $\pm 15^\circ$, table and cloth materials are sampled from 17 and 28 types, respectively, and 90 randomized environment combinations are applied with head-camera elevation perturbations of $\pm 5^\circ$ (Figure 5). Generated samples are visualized in Figure 6.

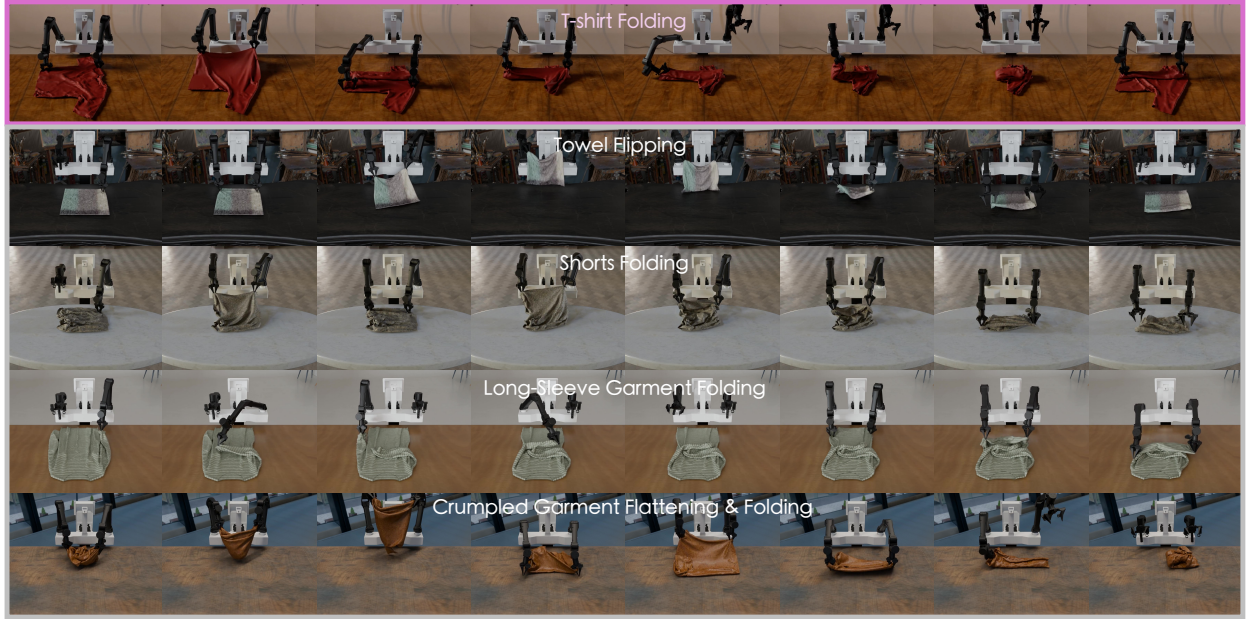


Figure 6. **Visualization of generated data across garments and tasks.** The pink box highlights the representative T-shirt folding task used as the real-world benchmark in this work. The gray box shows additional garments and manipulation tasks generated in simulation, including folding, flipping, and flattening, illustrating the broader task coverage enabled by our framework beyond the single benchmark task.

4.2. Main Experiment Results

We evaluate policies trained in simulation and real data under both in-domain and out-of-domain settings (Figure 7). Real demonstrations and teleoperated simulation data use 200 samples, while simulation-generated data are scaled to 10k samples for in-domain training and 2k samples for out-of-domain evaluation.

Simulation versus real-data performance (A1). Simulation-trained policies achieve performance comparable to real-data-trained counterparts under equal data budgets. For the representative $\pi_{0.5}$ setting, real data reach average success **97%**, while policies trained on sim-teleoperated data achieve **87%**, a marginal gap of **10%**. This suggests that physics-aligned simulation provides supervision of considerable fidelity to match real-world training with controlled data volumes.

Out-of-domain robustness (A2). Simulation-induced diversity yields substantial generalization gains under domain shifts. In spatial shifts, texture variation, and lighting perturbations, simulation-trained policies outperform real-data-trained baselines by **50%**, **13%**, and **47%**, respectively. These improvements indicate that simulation enables broader coverage of variations than limited real-world demonstrations, enhancing robustness beyond the training distribution.

Pretraining confound analysis. To isolate the effect of pretrained priors, we evaluate task learning under de novo initialization with no reliance on preexisting manipulation knowledge. The real-data baseline ($\pi_{0.5}$ trained from scratch) fails completely (**0%** success), indicating that limited real demonstrations alone do not enable deformable manipulation. In contrast, the synthetic training pipeline achieves strong task acquisition (**76%**) under the same from-scratch condition, demonstrating that performance gains originate from generated data rather than pretrained skills. These results dispel the possibility that task success is driven by prior knowledge and confirm the effectiveness of synthetic supervision.

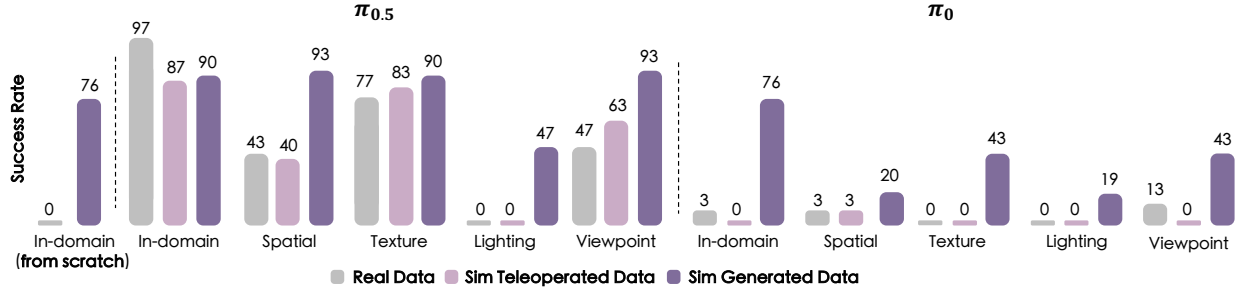


Figure 7. **In-domain and out-of-domain evaluation.** Policies are trained with real data, collected simulation data, or simulator-generated data. Groups: $\pi_{0.5}$ trained from scratch, $\pi_{0.5}$ post-trained, and π_0 post-trained. Simulated data matches real episodes under equal budgets and surpasses them when scaled, especially under domain shifts.

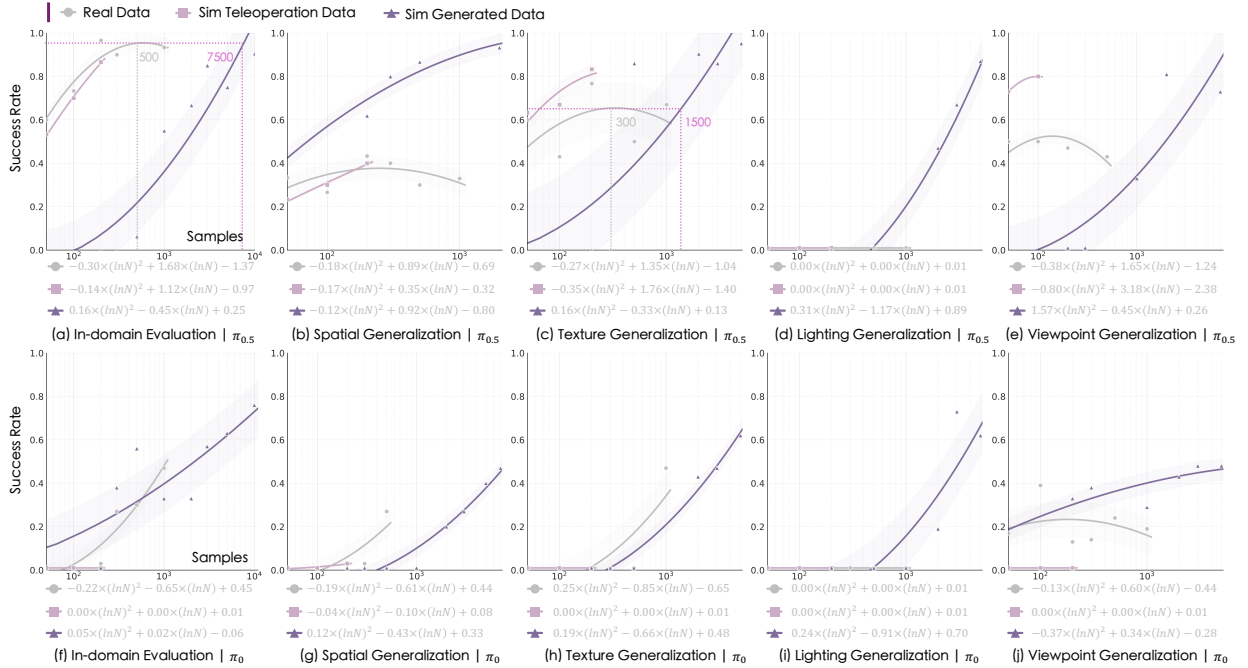


Figure 8. **Curves of performance versus data scale.** Synthetic data scaling improves performance and can surpass real-data-only training. Dashed lines denote the equivalence points where M synthetic samples match one real sample at saturation.

Scaling analysis (A3). Figure 8 examines whether synthetic data enables more efficient performance gains through scaling than real-world collection. As data volume increases, the success rates of $\pi_{0.5}$ and π_0 improve steadily, following the fitted scaling curves. Simulation demonstrates favorable scaling behavior compared to real-world data. For the representative $\pi_{0.5}$ model under in-domain evaluation, one real demonstration provides comparable benefit to approximately 15 synthetic samples. In a representative out-of-domain setting (texture generalization), the equivalence shifts to roughly 5 synthetic samples per real sample.

Findings. (1) Synthetic data is weak in extremely low-data regimes but scales more effectively than real data as volume increases. Performance grows rapidly with additional simulation samples and eventually surpasses real-data training, while real-data gains saturate due to limited diversity. (2) $\pi_{0.5}$ outperforms π_0 under fixed budgets because of richer pretraining or a more expressive state

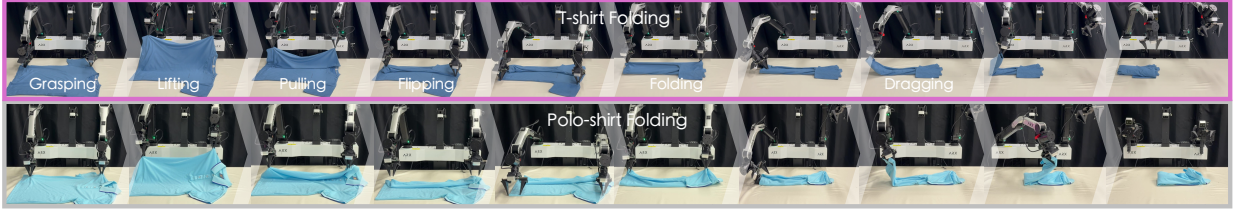


Figure 9. **Qualitative results of zero-shot sim-to-real transfer.** Top: Real-robot deployment of our representative T-shirt folding task using policies trained on synthetic data, illustrating successful folding from grasping to completion. Bottom: Deployment on highly dissimilar garments with material, shape, and size configurations absent from training and simulation, demonstrating generalization across extreme domain shifts.

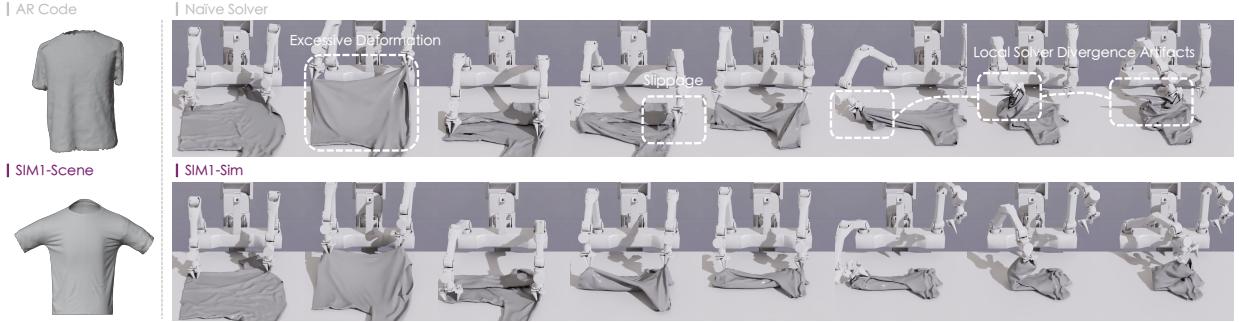


Figure 10. **Scene digitization and solver stability.** Our pipeline produces detailed geometry suitable for simulation, whereas marker-based methods (e.g., AR Code) yield coarser reconstructions. Conventional solvers exhibit artifacts in rigid-soft interaction (dashed regions); our solver remains stable and closely matches real-world behavior.

representation. π_0 requires greater data volume to achieve similar performance, consistent with lower data efficiency rather than fundamental task limitations.

Qualitative Results. Figure 9 shows real-world deployments of policies trained purely on synthetic data. The policy successfully performs garment folding on a physical robot without any real demonstrations. It further generalizes to an unseen polo shirt with different material, texture, and geometry, where the real-data baseline achieves **20%** success while ours reaches **70%** on the real robot.

4.3. Ablation Study

Scene reconstruction and physics solver. Figure 10 presents qualitative comparisons of scene reconstruction and physics simulation. Marker-assisted Reconstruction (e.g., AR Code (AR Code, 2026)) produces centimeter-level meshes with noticeable artifacts. In contrast, our pipeline reconstructs assets with sub-millimeter level accuracy, enabling precise geometry alignment for simulation. Generic deformable solvers (e.g., FEM (Zienkiewicz and Taylor, 2005), VBD (Chen et al., a), etc.) are not designed for rigid-soft interaction and exhibit unrealistic dynamics due to particle motion lag. This leads to excessive stretching during pulling, particle gaps that cause slippage, and local delays that produce spiky deformations. Our solver eliminates these artifacts and yields stable, physically consistent garment behavior, demonstrating the necessity of both *geometric alignment* via accurate scene reconstruction and *dynamical alignment* via stabilized physics simulation.

Quantitative evaluation. Table 1 evaluates module contributions. The baseline trajectory strategy, adapted from rigid-body manipulation MimicGen (Mandlekar et al., 2023), fails to generate valid

Table 1. **Ablation on designs in SIM1.** Pass rate denotes the proportion of generated samples accepted by the discriminator. All designs contribute to the final performance.

Method	Pass Rate (%)	Success Rate (%)					Average (%)
		In-domain	Spatial	Texture	Lighting	Viewpoint	
Baseline	0	-	-	-	-	-	-
+ Traj. decomposition	65	0	0	0	0	0	0
+ Diff.-based generation	38	47	33	60	20	7	33
+ Deform.-stable solver	40	67 (+20)	93 (+60)	90 (+30)	82 (+62)	47 (+40)	76 (+43)

training data (pass rate **0%**), confirming that naive segmentation is insufficient for deformable tasks. Trajectory decomposition enables data synthesis but produces discontinuous segments and no task success, indicating limited utility without physical consistency. Diffusion-based generation improves data realism (in-domain **47%**) and captures more human-like trajectories, yet generalization remains weak. Incorporating the deformation-stable solver yields substantial gains (in-domain **67%**, average **76%**), demonstrating that solver stability is essential for translating scalable simulation data into policies that generalize to the physical world, underscoring the role of *movement alignment* in synthesizing physically consistent manipulation trajectories.

5. Conclusion

We present a physics-aligned real-to-sim-to-real pipeline that transforms limited real demonstrations into scalable synthetic data for deformable manipulation. By jointly addressing geometric accuracy, dynamic fidelity, and motion synthesis, our approach achieves zero-shot sim-to-real transfer in garment folding. Experiments demonstrate that policies trained purely on synthetic data achieve comparable performance to real-data baselines, with consistent improvement as data scales. These results show that high-fidelity simulation is a viable and scalable source of supervision for deformable manipulation, complementing real-data collection and enabling broader policy learning at reduced cost.

Limitations and broader impact. A current limitation is that material calibration requires expert-guided parameter tuning for each asset, which constrains full automation across arbitrary cloth types. By demonstrating that high-fidelity simulation can complement real-robot data and enable scalable policy learning, this work provides a practical foundation for data-efficient robotic development.

Acknowledgement

We sincerely thank Jiafei Cao, Yang Li, and Junjie Xia for their invaluable support in building and maintaining the data pipeline. We are also grateful to all data collection contributors for their dedicated efforts in large-scale data acquisition. We appreciate Chaoyang Lv for his assistance with the simulation system and related technical support. We thank Haochen Tian for developing the visualization and plotting scripts used in this work. We also acknowledge the support and guidance from Prof. Minyi Guo and Prof. Hongzi Zhu.

References

Blender: a free and open-source 3d computer graphics software tool. <https://www.blender.org/>, 2026.

- AR Code. Ar genai: Turn a single photo into an ar-ready 3d model. <https://ar-code.com/blog/ar-genai-turn-a-single-photo-into-an-ar-ready-3d-model>, 2026.
- J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.
- K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics (TOG 2002)*.
- R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, page 594–603, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581135211. doi: 10.1145/566570.566623. URL <https://doi.org/10.1145/566570.566623>.
- R. Cadene, S. Aliberts, F. Capuano, M. Aractingi, A. Zouitine, P. Kooijmans, J. Choghari, M. Russi, C. Pascal, S. Palma, M. Shukor, J. Moss, A. Soare, D. Aubakirova, Q. Lhoest, Q. Gallouédec, and T. Wolf. Lerobot: An open-source library for end-to-end robot learning. *arXiv preprint arXiv:2602.22818*, 2024.
- J. Cai, Z. Cai, J. Cao, Y. Chen, Z. He, L. Jiang, H. Li, H. Li, Y. Li, Y. Liu, et al. Internvla-a1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026.
- C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, H. Zhang, and M. Zhu. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.
- A. H. Chen, Z. Liu, Y. Yang, and C. Yuksel. Vertex block descent. *ACM Transactions on Graphics (TOG 2024)*, a.
- B. Chen, D. Martí Monsó, Y. Du, M. Simchowitz, R. Tedrake, and V. Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems (NeurIPS 2024)*, b.
- T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li, Q. Liang, X. Lin, Y. Ge, Z. Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025a.
- X. Chen, Y. Chen, Y. Fu, N. Gao, J. Jia, W. Jin, H. Li, Y. Mu, J. Pang, Y. Qiao, et al. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025b.
- Y. Chen, Y. Hu, L. Sun, T. Kusnur, L. Herlant, and C. Jiang. Empm: Embodied mpm for modeling and simulation of deformable objects. *IEEE Robotics and Automation Letters (RA-L 2026)*, c.
- O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky,

- A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Bharadhwaj, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Vakil, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, M. Z. Irshad, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Bajjal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Tulsiani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Kumar, V. Vanhoucke, V. Guizilini, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023.
- S. Deng, M. Yan, S. Wei, H. Ma, Y. Yang, J. Chen, Z. Zhang, T. Yang, X. Zhang, W. Zhang, H. Cui, Z. Zhang, and H. Wang. Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data. *arXiv preprint arXiv:2505.03233*, 2025.
- N. Gao, Y. Chen, S. Yang, X. Chen, Y. Tian, H. Li, H. Huang, H. Wang, T. Wang, and J. Pang. Genmanip: Llm-driven simulation for generalizable instruction-following manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2025)*.
- C. Giles, E. Diaz, and C. Yuksel. Augmented vertex block descent. *ACM Transactions on Graphics (TOG 2025)*, a.
- C. Giles, E. Diaz, and C. Yuksel. Augmented vertex block descent. *ACM Transactions on Graphics (SIGGRAPH 2025)*, b. ISSN 0730-0301.
- J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.
- X. Han, T. F. Gast, Q. Guo, S. Wang, C. Jiang, and J. Teran. A hybrid material point method for frictional contact with diverse materials. *Proceedings of the ACM on Computer Graphics and Interactive Techniques (PACMCGIT 2019)*.

- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*.
- T. He, Z. Wang, H. Xue, Q. Ben, Z. Luo, W. Xiao, Y. Yuan, X. Da, F. Castañeda, S. Sastry, C. Liu, G. Shi, L. Fan, and Y. Zhu. Viral: Visual sim-to-real at scale for humanoid loco-manipulation. *arXiv preprint arXiv:2511.15200*, 2025.
- K. Huang, X. Lu, H. Lin, T. Komura, and M. Li. Stiffgipc: Advancing gpu ipc for stiff affine-deformable simulation. *ACM Transactions on Graphics (TOG 2025)*. ISSN 0730-0301.
- P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.
- H. Jiang, H.-Y. Hsu, K. Zhang, H.-N. Yu, S. Wang, and Y. Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. *IEEE/CVF International Conference on Computer Vision (ICCV 2025)*, a.
- Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. J. Fan, and Y. Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA 2025)*, b.
- M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG 2013)*.
- A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.
- R. Kikuuwe, H. Tabuchi, and M. Yamamoto. An edge-based computationally efficient formulation of saint venant-kirchhoff tetrahedral finite elements. *ACM Trans. Graph.*, 28(1), Feb. 2009. ISSN 0730-0301. doi: 10.1145/1477926.1477934. URL <https://doi.org/10.1145/1477926.1477934>.
- M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, M. Anvari, M. Hwang, M. Sharma, A. Aydin, D. Bansal, S. Hunter, K.-Y. Kim, A. Lou, C. R. Matthews, I. Villa-Renteria, J. H. Tang, C. Tang, F. Xia, S. Savarese, H. Gweon, K. Liu, J. Wu, and L. Fei-Fei. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning (CoRL 2022)*.
- J. Li, G. Daviet, R. Narain, F. Bertails-Descoubes, M. Overby, G. E. Brown, and L. Boissieux. An implicit frictional contact solver for adaptive cloth simulation. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201308. URL <https://doi.org/10.1145/3197517.3201308>.
- M. Li, Z. Ferguson, T. Schneider, T. Langlois, D. Zorin, D. Panozzo, C. Jiang, and D. M. Kaufman. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM transactions on graphics*, 2020.

- Y. Li, H. Jiang, J. Xia, H. Zhang, J. Du, Y. Zhou, J. Zeng, C. Hao, J. Ren, Q. Yu, et al. Forcevla2: Unleashing hybrid force-position control with force awareness for contact-rich manipulation. *arXiv preprint arXiv:2603.15169*, 2026.
- H. Lu, R. Wu, Y. Li, S. Li, Z. Zhu, C. Ning, Y. Shen, L. Luo, Y. Chen, and H. Dong. Garmentlab: A unified simulation and benchmark for garment manipulation. In *Advances in Neural Information Processing Systems (NeurIPS 2024)*.
- A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.
- M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation (JVCI 2007)*.
- R. Narain, A. Samii, and J. F. O’Brien. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph.*, 31(6), Nov. 2012. ISSN 0730-0301. doi: 10.1145/2366145.2366171. URL <https://doi.org/10.1145/2366145.2366171>.
- S. Nasiriany, S. Nasiriany, A. Maddukuri, and Y. Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations (ICLR 2026)*.
- Y. Tian, Y. Yang, Y. Xie, Z. Cai, X. Shi, N. Gao, H. Liu, X. Jiang, Z. Qiu, F. Yuan, et al. Interndata-1: Pioneering high-fidelity synthetic data for pre-training generalist policy. *arXiv preprint arXiv:2511.16651*, 2025.
- J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017)*.
- M. Torne, A. Simeonov, Z. Li, A. Chan, T. Chen, A. Gupta, and P. Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS 2017)*.
- H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, A. Lee, K. Fang, C. Finn, and S. Levine. Bridgedata v2: A dataset for robot learning at scale. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning (CoRL 2023)*.
- H. Wang, J. Chen, W. Huang, Q. Ben, T. Wang, B. Mi, T. Huang, S. Zhao, Y. Chen, S. Yang, et al. Grutopia: Dream general robots in a city at scale. *arXiv preprint arXiv:2407.10943*, 2024.
- H. Xue, T. He, Z. Wang, Q. Ben, W. Xiao, Z. Luo, X. Da, F. Castañeda, G. Shi, S. Sastry, L. J. Fan, and Y. Zhu. Opening the sim-to-real door for humanoid pixel-to-action policy transfer. *arXiv preprint arXiv:2512.01061*, 2025.
- J. Yang, K. Lin, J. Li, W. Zhang, T. Lin, L. Wu, Z. Su, H. Zhao, Y.-Q. Zhang, L. Chen, et al. Rise: Self-improving robot policy with compositional world model. *arXiv preprint arXiv:2602.11075*, 2026.

- S. Yang, W. Yu, J. Zeng, J. Lv, K. Ren, C. Lu, D. Lin, and J. Pang. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. *arXiv preprint arXiv:2504.13175*, 2025.
- J. Ye, K. Wang, C. Yuan, R. Yang, Y. Li, J. Zhu, Y. Qin, X. Zou, and X. Wang. Dex1b: Learning with 1b demonstrations for dexterous manipulation. In *Robotics: Science and Systems (RSS 2025)*.
- C. Yin, D. Huang, D. Yang, J. Wang, N. Zhao, C. Xu, W. Sun, L. Hou, Z. Li, J. Wu, Z. Liu, Z. Xiao, S. Zhang, L. Bao, R. Feng, Z. Pang, J. Li, Q. Wang, and M. Yao. Genie sim 3.0 : A high-fidelity comprehensive simulation platform for humanoid robot.
- C. Yu, S. Ma, W. Du, Z. Zong, H. Xue, W. Chen, C. Lu, Y. Yang, X. Han, J. Masterjohn, et al. Right-side-out: Learning zero-shot sim-to-real garment reversal. *arXiv preprint arXiv:2509.15953*, 2025.
- C. Yu, C. Sima, G. Jiang, H. Zhang, H. Mai, H. Li, H. Wang, J. Chen, K. Wu, L. Chen, L. Zhao, M. Shi, P. Luo, Q. Bu, S. Peng, T. Li, and Y. Yuan. χ_0 : Resource-aware robust manipulation via taming distributional inconsistencies. *arXiv preprint arXiv:2602.09021*, 2026.
- J. Zhang, H. Liu, D. Li, X. Yu, H. Geng, Y. Ding, J. Chen, and H. Wang. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *Proceedings of The 8th Annual Conference on Robot Learning (CoRL 2024)*.
- C. Zhou, X. Jin, C. C. Wang, and J. Feng. Plausible cloth animation using dynamic bending model. *Progress in Natural Science*, 18(7):879–885, 2008.
- O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, 2005.

A. Author contributions

All authors contributed to writing.

- Yunsong Zhou (zhouyunsong@pjlab.org.cn): Proposed and led project.
- Hangxu Liu: Contributed to simulation infrastructure, simulated data generation, as well as open-sourcing.
- Xuekun Jiang: Led simulation infrastructure and open-sourcing. Contributed to the validity checker and experiments.
- Xing Shen: Led simulation solver development.
- Yuanzhen Zhou: Contributed to scene randomization and rendering.
- Hui Wang: Contributed to early versions of simulators.
- Baole Fang: Contributed to scene digitalization.
- Yang Tian: Provided suggestions and contributed to outreach efforts.
- Mulin Yu: Advised asset scanning.
- Qiaojun Yu: Advised project. Contributed to experiments.
- Li Ma: Advised infrastructures.
- Hengjie Li: Advised infrastructures.
- Hanqing Wang: Advised project. Supported computational resources.
- Jia Zeng: Advised project. Contributed to data collection pipelines.
- Jiangmiao Pang (pangjiangmiao@gmail.com): Supervised project direction with critical feedback.

B. SIM1 Implementation Details

In our framework, we use the Augmented Vertex Block Descent (AVBD) (Giles et al., a) solver to simulate cloth dynamics.

B.1. Solver Overview

Our cloth solver operates on a triangular mesh where edges $\mathbf{e} = (\mathbf{e}_i, \mathbf{e}_j)$ represent connections between vertices. The goal of each Newton-type iteration is to compute a displacement update $\Delta \mathbf{e}_i$ for each vertex such that the mesh configuration minimizes the total energy while satisfying constraints (stretch limits, bending, collision avoidance). The overall workflow is summarized as follows:

1. Compute internal forces derived from elastic and bending energies.
2. Evaluate constraint violations (stretch and strain) and compute corrective forces (**new**).
3. Assemble total forces acting on each vertex.
4. Solve for vertex updates $\Delta \mathbf{e}_i$ using the Newton system.
5. Apply penetration avoidance to truncate updates that would violate collision safety (**new**).
6. Iterate until convergence, producing the new vertex positions $\mathbf{e}_i^{\text{new}}$.

B.2. Internal Force Computation

StVK elasticity model. To simulate how cloth resists stretching and shearing, we employ the *St. Venant–Kirchhoff* (StVK) hyperelastic model (Kikuuwe et al., 2009). Intuitively, this model assigns an energy to how much each triangle in the cloth mesh is deformed compared to its rest shape: the more a triangle stretches or shears, the higher the energy. Minimizing this energy naturally generates forces that restore the cloth toward its undeformed configuration.

Formally, let $\mathbf{e}$ denote the positions of the mesh vertices, and let $\mathbf{F}$ be the deformation gradient

mapping rest-state coordinates to the current positions. The Green-Lagrange strain tensor $\mathbf{G} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I})$ measures the nonlinear strain, where $\mathbf{I}$ is the identity matrix. The elastic energy density of a triangle is then:

$$\Psi(\mathbf{e}) = \mu \|\mathbf{G}\|_F^2 + \frac{1}{2} \hat{\lambda} (\text{tr}(\mathbf{G}))^2, \quad (5)$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\text{tr}(\cdot)$ is the trace operator, and μ and $\hat{\lambda}$ are the Lamé parameters that control the material's resistance to shear and area change, respectively.

The elastic (stretching) force acting on vertex i is obtained as the negative gradient of the energy with respect to its position $\mathbf{f}_i^{\text{stretch}} = -\frac{\partial \Psi}{\partial \mathbf{e}_i}$. Intuitively, this force acts like a spring that resists local stretching and shearing, driving the cloth mesh toward physically plausible configurations.

Dihedral-angle bending model. To capture out-of-plane deformations such as folding or wrinkling, we adopt a *dihedral-angle* bending model (Zhou et al., 2008). Conceptually, each edge shared by two adjacent triangles has a preferred rest angle; deviations from this angle induce a restoring force that resists bending.

Let $\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k, \mathbf{e}_l$ denote the four vertices forming the two triangles sharing an edge. The current dihedral angle θ is a function of these vertex positions, and θ_0 is the rest angle. Denoting the edge length by l_{edge} and bending stiffness by k_{bend} , the bending energy is:

$$E_{\text{bend}}(\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k, \mathbf{e}_l) = k_{\text{bend}} l_{\text{edge}} (\theta(\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k, \mathbf{e}_l) - \theta_0)^2. \quad (6)$$

The bending force on vertex $\mathbf{e}_i$ is $\mathbf{f}_i^{\text{bend}} = -\frac{\partial E_{\text{bend}}}{\partial \mathbf{e}_i}$, and similarly for the other vertices of the edge. Each vertex accumulates contributions from all adjacent edges, which combined with the stretching and constraint forces to form the total force used by the solver to compute the vertex displacement update $\Delta \mathbf{e}_i$.

B.3. Constraint Force Computation

Spring strain constraint. To suppress unphysical stretching, we enforce a maximum stretch ratio ξ on all mesh edges, as formulated in Section 3.2 (Equation (1)-Equation (2)) in the main paper. The resulting force on vertex $\mathbf{e}_i$ is computed as $\mathbf{f}_i^{\text{strain}} = -\frac{\partial E_{\text{strain}}}{\partial \mathbf{e}_i}$, and similarly for $\mathbf{e}_j$. These constraint forces are then integrated with stretching, bending, and external forces to determine the total vertex displacement $\Delta \mathbf{e}_i$ during each AVBD iteration.

B.4. Total Force Assembly

The vertex update $\Delta \mathbf{e}_i$ is computed by solving a local 3×3 linear system derived from the Newton optimization (Li et al., 2018). Specifically, we define the system matrix $\mathbf{A}_i$ and the residual vector $\mathbf{b}_i$ as:

$$\mathbf{A}_i = \frac{M_i}{\Delta t^2} \mathbf{I} + \sum \mathbf{H}_i, \quad \mathbf{b}_i = \mathbf{f}_i^{\text{total}} - \frac{M_i}{\Delta t^2} (\mathbf{e}_i^{(n)} - \hat{\mathbf{e}}_i), \quad (7)$$

where M_i is the lumped mass of vertex i , Δt is the time step, and $\mathbf{H}_i$ is the sum of Hessians of the energy terms associated with vertex i (Narain et al., 2012), $\mathbf{e}_i^{(n)}$ is the vertex position at the current Newton-type iteration n , $\hat{\mathbf{e}}_i$ is the inertial predictive position, a constant reference for the current time step derived from the vertex's velocity and acceleration in the preceding time frame.

The total force $\mathbf{f}_i^{\text{total}}$ is assembled from multiple contributions:

$$\mathbf{f}_i^{\text{total}} = \mathbf{f}_i^{\text{stretch}} + \mathbf{f}_i^{\text{bend}} + \mathbf{f}_i^{\text{strain}} + \mathbf{f}_i^{\text{ext}}, \quad (8)$$

where $\mathbf{f}_i^{\text{stretch}}$ comes from StVK elasticity, $\mathbf{f}_i^{\text{bend}}$ from dihedral-angle bending, $\mathbf{f}_i^{\text{strain}}$ from the spring strain constraint, and $\mathbf{f}_i^{\text{ext}}$ includes any external forces applied by robots or environment interactions. The displacement is obtained by solving the linear system, $\Delta \mathbf{e}_i = \mathbf{A}_i^{-1} \mathbf{b}_i$.

B.5. Penetration Avoidance

To prevent interpenetration with obstacles or self-collision, we introduce a geometric safety filter. A safe displacement bound d_{safe} is defined as $d_{\text{safe}} = \beta \cdot \min(r_{\text{collision}}, d_{\text{tri}}^{\min}, d_{\text{edge}}^{\min})$ (Bridson et al., 2002), where $\beta = 0.42$ is a relaxation factor, $r_{\text{collision}}$ is the collision radius, and $d_{\text{tri}}^{\min}, d_{\text{edge}}^{\min}$ represent the minimum distances to the nearest triangle and edge primitives, respectively. Based on this bound, a scalar clipping factor s is computed to truncate the raw Newton update $\Delta \mathbf{e}_i$:

$$s = \min \left(1, \frac{d_{\text{safe}}}{\|\Delta \mathbf{e}_i\|} \right). \quad (9)$$

This mechanism ensures that the vertex does not overshoot the safety margin during a single iteration, effectively preventing tunneling and numerical instability.

B.6. Position Updation

The sequence concludes with the synthesis of the final vertex position $\mathbf{e}_i^{\text{new}}$ for the current iteration. By applying the clipped displacement to the iteration’s starting point $\mathbf{e}_i^{(n)}$, we obtain:

$$\mathbf{e}_i^{\text{new}} = \mathbf{e}_i^{(n)} + s \cdot \Delta \mathbf{e}_i. \quad (10)$$

This integration ensures that the updated mesh state is not only physically optimal according to the AVBD energy gradients but also strictly compliant with geometric safety constraints, leading to a robust and collision-free simulation.

B.7. Simulation Infrastructure

To support high-fidelity data collection for deformable manipulation, we develop a simulation infrastructure. This pipeline enables real-time synchronization between physical teleoperation and GPU-accelerated simulation, featuring a tight integration of rigid-body and cloth dynamics.

Bidirectional teleoperation and joint mapping. During teleoperation, we establish a direct correspondence between the physical robot and the simulated actuators. Specifically, the simulation joint state $\mathbf{q}_{\text{sim}}$ is updated as:

$$\mathbf{q}_{\text{sim}}(t)[\mathcal{I}_{\text{sim}}] = \mathbf{q}_{\text{real}}(t)[\mathcal{I}_{\text{arm}}], \quad (11)$$

where $\mathcal{I}_{\text{sim}}$ and $\mathcal{I}_{\text{arm}}$ represent the index sets of controllable joints in the simulator and the physical robot arm, respectively. For the end-effectors, the left and right gripper openness ($o_L, o_R \in [0, 1]$) are decoupled and mapped independently from their respective physical finger joints to ensure asymmetric grasping fidelity:

$$o_L = -\text{clip} \left(\frac{q_{\text{finger},L}}{2 \cdot q_{\text{max}}}, -1, 0 \right), \quad o_R = -\text{clip} \left(\frac{q_{\text{finger},R}}{2 \cdot q_{\text{max}}}, -1, 0 \right), \quad (12)$$

where $q_{\text{finger},L}$ and $q_{\text{finger},R}$ are the joint angles of the left and right physical fingers, and $q_{\text{max}} = 1.62$ rad is the mechanical limit.

Physical parameter calibration. We calibrate the cloth’s physical properties within the AVBD framework to ensure realism and stability, as summarized in Table 2. This includes StVK moduli for

Table 2. Calibrated simulation parameters for cloth physics in SIM1.

Category	Parameter	Value
Particle	Radius	0.008 m
	Density	2.0 kg/m ²
StVK Elasticity	Shear modulus μ	1.0×10^2
	Area modulus λ	1.0×10^2
	Damping	1.5×10^{-6}
Bending	Stiffness	8.0×10^{-4}
	Damping	1.0×10^{-3}
Strain Limit	Maximum stretch ratio	0.05 (5%)
	Initial stiffness	1.0×10^4
	Maximum stiffness	1.0×10^6
	Growth rate	1.0×10^3
Contact	Soft contact stiffness	5.0×10^2
	Soft contact damping	5.0×10^{-3}
	Robot friction	1.5
	Table friction	0.0
	Self-contact friction	0.25
	Self-contact radius	0.002 m
	Body-cloth margin	0.01 m
	Friction smoothing	1.0×10^{-2}
AVBD Temporal	Lambda decay	0.94
	Stiffness decay	0.95
	Regularization α	0.99
	Bound relaxation β	0.42

elasticity, a 5% strain limit to prevent over-stretching, and contact parameters such as robot-specific friction and the bound relaxation factor $\beta = 0.42$. These parameters are specifically tuned to maintain numerical robustness during the high-speed, contact-rich interactions characteristic of SIM1 tasks.

GPU-accelerated simulation. The simulation is powered by NVIDIA Warp, a high-performance framework that compiles Python code into native CUDA kernels for GPU execution. This allows our rigid-body dynamics and AVBD cloth solver to run entirely on the GPU within a unified memory space (~ 15 fps).

Data recording. The system employs asynchronous logging to maintain simulation throughput. Per-frame robot states and gripper openness are stored in NPZ format for policy learning, while full session trajectories and contact manifolds are serialized to USD for post-hoc diagnostic analysis and visual inspection.

C. Synthetic Data Generation Algorithm

Algorithm 1 presents the pseudo-code of our synthetic data generation pipeline, which transforms teleoperated demonstrations into large-scale synthetic trajectories through structured decomposition and diffusion-based trajectory synthesis.

The process begins by extracting reusable interaction segments from expert demonstrations (line 1), forming a library of grasp-to-release primitives that capture meaningful manipulation phases. During data generation, sequences of these segments are sampled to construct a high-level task skeleton (lines 2–3). For each adjacent segment pair, a diffusion model synthesizes feasible transition

Algorithm 1 Synthetic data generation via trajectory decomposition and diffusion-based motion generation

Require: Expert demonstrations $\mathcal{D}$, simulator $\mathcal{E}$, diffusion model f_θ , discriminator D , renderer $\mathcal{R}$, appearance count K

Ensure: Synthetic dataset $\mathcal{D}_{\text{synth}}$

```

1: Extract interaction segments  $\mathcal{P} = \{(\mathbf{p}_s^i, \mathbf{p}_t^i)\}_{i=1}^N$  from  $\mathcal{D}$  ▷ Structured decomposition
2: while  $|\mathcal{D}_{\text{synth}}| < N_{\text{target}}$  do
3:   Sample segment sequence  $\{p_1, \dots, p_L\} \sim \mathcal{P}$  ▷ Decomposed task skeleton
4:    $\tau_{\text{gen}} \leftarrow \emptyset$ 
5:   for  $k = 1$  to  $L - 1$  do
6:      $\mathbf{p}_s \leftarrow \text{end}(p_k), \mathbf{p}_t \leftarrow \text{start}(p_{k+1})$ 
7:      $\mathbf{h} \sim \mathcal{D}$  ▷ Sample demonstration history
8:      $\mathbf{x}_{1:M} \leftarrow f_\theta(\mathbf{h}, \mathbf{p}_s, \mathbf{p}_t)$  ▷ Diffusion-based trajectory synthesis
9:      $\tau_{\text{gen}} \leftarrow \tau_{\text{gen}} \cup \mathbf{x}_{1:M}$ 
10:  end for
11:  Execute  $\tau_{\text{gen}}$  in  $\mathcal{E}$ , obtain video  $\mathbf{V}$ 
12:  if  $\text{success}(\tau_{\text{gen}})$  and  $D(\mathbf{V}) > \tau$  then
13:    for  $v = 1$  to  $K$  do
14:       $\mathbf{V}_v \leftarrow \mathcal{R}(\tau_{\text{gen}})$  with randomized appearance
15:      Store  $(\tau_{\text{gen}}, \mathbf{V}_v)$  in  $\mathcal{D}_{\text{synth}}$ 
16:    end for
17:  end if
18: end while
19: return  $\mathcal{D}_{\text{synth}}$ 

```

Table 3. **Generalization performance on additional tasks.** †: Out-of-domain experiments where policies are evaluated on entirely unseen real-world tasks.

Task	Towel Flipping	Shorts Folding	Polo-shirt Folding [†]
Success rate (%)	80	93	93

trajectories that connect the segment endpoints (lines 5–9), producing a complete manipulation trajectory. The generated trajectory is then executed in the simulator to obtain a corresponding video sequence (line 10). To ensure data quality, trajectories are filtered using both task success checks and a discriminator that evaluates visual realism (line 11). Finally, valid trajectories are rendered multiple times with randomized appearances (lines 12–14), producing diverse observations while sharing the same underlying physical interaction. This procedure is repeated until the desired dataset size is reached.

D. Experiments

D.1. Additional Tasks

To demonstrate generalizability beyond T-shirts, we apply our pipeline to towels and shorts. For each category, interaction primitives are manually designed. Starting with 100 teleoperated demonstrations per category, we generate 1,000 synthetic trajectories with texture randomization. We evaluate policies trained on the synthetic datasets over 30 real-world trials per task, achieving 80% success on towel folding and 93% on shorts and polo-shirt folding (Table 3). While towels and shorts require

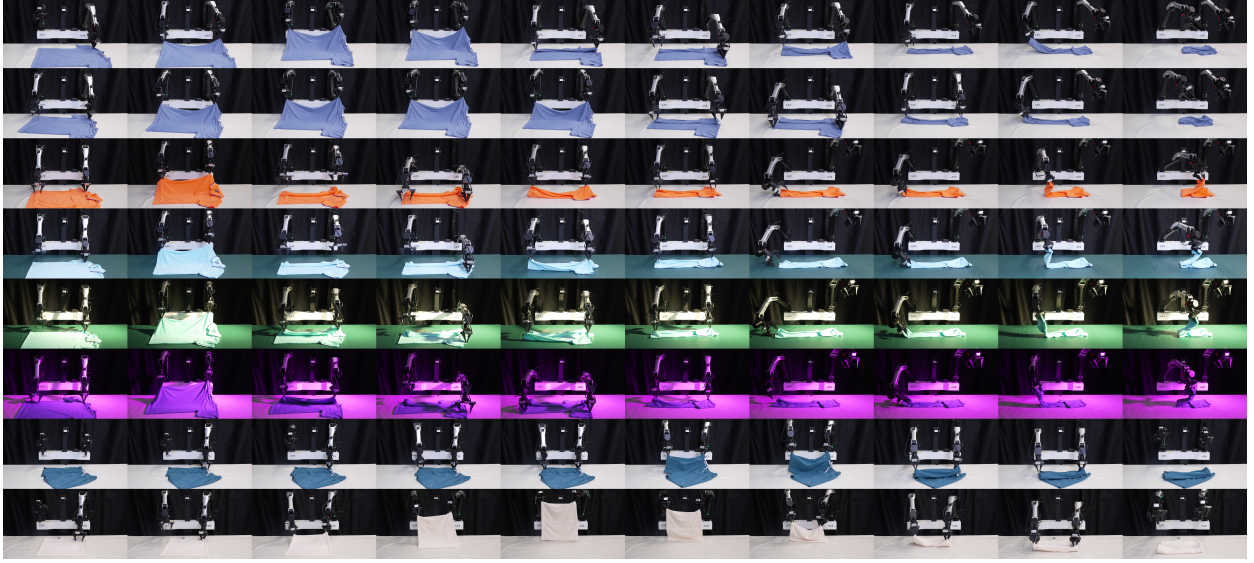


Figure 11. **Real-world deployment experiments.** Representative results include T-shirt folding under both in-domain and out-of-domain settings. We further demonstrate generalization to additional garments, including polo-shirts, shorts, and towels.

teleoperated demonstrations for data generation, the polo-shirt result is achieved in a **zero-shot** manner without any task-specific demonstrations.

Notably, the polo-shirt exhibits substantially different geometry, size, material, and frictional properties compared to the garments used during training. Moreover, no similar garment instance appears in the training dataset. Despite this significant distribution shift, the learned policy transfers successfully to the real-world task, highlighting strong cross-garment generalization.

D.2. Real-world Deployment

Figure 11 shows representative real-world deployment results. Each row corresponds to a different experiment. From top to bottom, we present in-domain T-shirt folding, T-shirt folding with randomized object positions, polo-shirt folding under varying textures and lighting conditions, and experiments on shorts and towels. The viewpoint randomization setting is not shown, as it produces minimal visual differences.

D.3. More Visualizations

We visualize synthetic demonstrations generated by our pipeline to highlight both diversity and realism. Figure 12 shows representative T-shirt folding scenes with randomized garments, tables, and lighting conditions. Beyond T-shirts, we also illustrate towel, shorts, and polo-shirt folding to demonstrate generality. Figure 13 and Figure 14 present temporally sampled sequences captured from head-mounted cameras, aligned with real-robot data, showcasing the diversity achieved through appearance and scene randomization.

D.4. Cost-efficiency Analysis

We conduct a comparative analysis to quantify the economic and throughput advantages of our simulation pipeline against traditional real-world data acquisition in Table 4. Real-world data collection



Figure 12. **Simulated T-shirt folding scenarios.** Each row shows a distinct configuration generated by our pipeline.



Figure 13. **Synthetic generated data of T-shirt folding.** Each row shows a temporally sampled trajectory with randomized textures, illustrating the visual diversity generated by our pipeline.

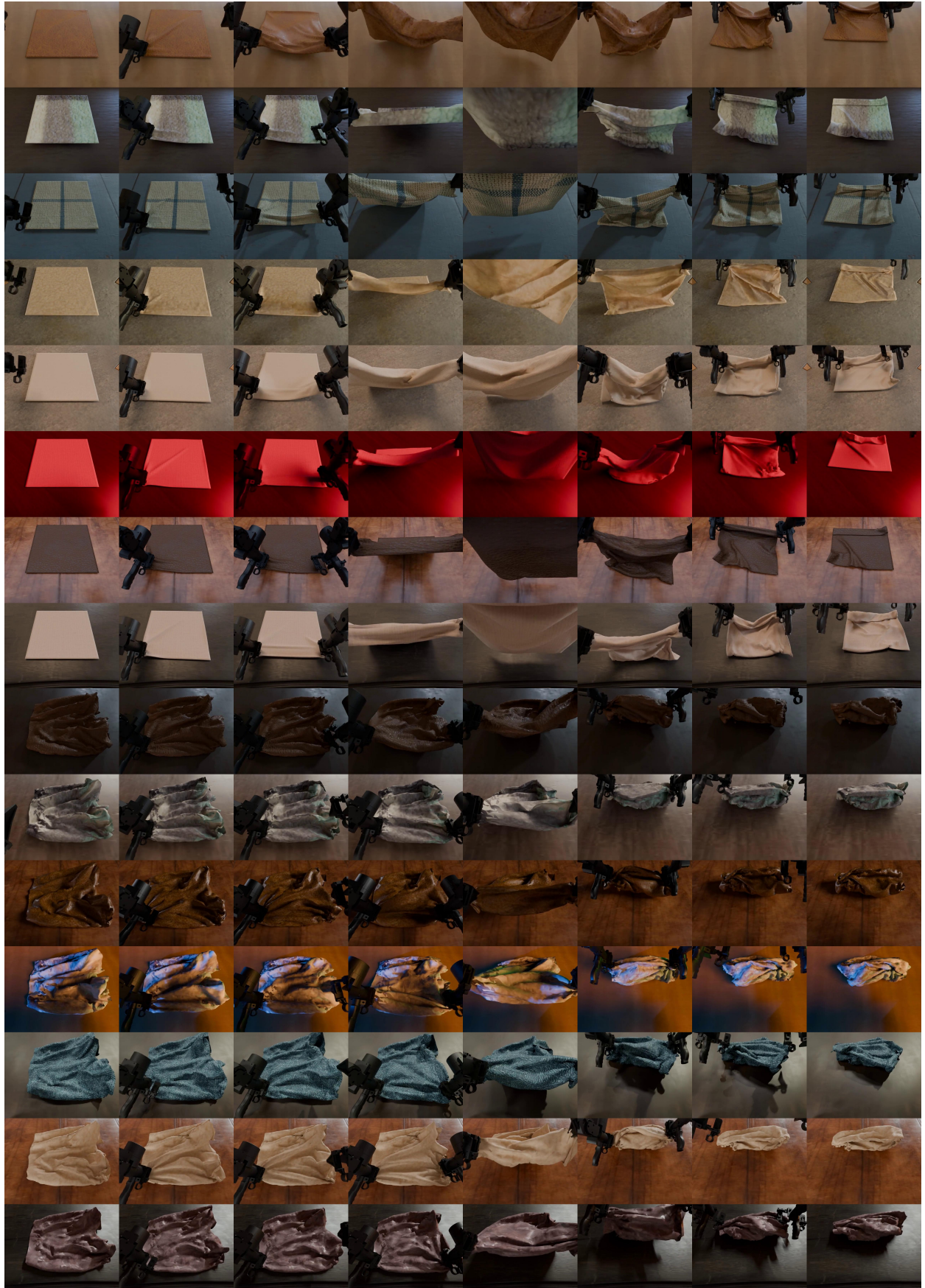


Figure 14. **Synthetic generated data of towels and shorts.** Each row shows a temporally sampled trajectory with randomized textures, demonstrating the framework’s applicability to multiple garment types.

Table 4. **Comparison of data acquisition efficiency.** All costs are estimated based on the daily expense of employing two data operators (\$282).

Method	Throughput (traj./day)↑	Unit Cost (\$/traj.)↓	Relative Cost ↓
Real-world Collection	104	2.71	1.0×
Ours (Simulation)	710	0.10	0.037×

incurs a daily cost of approximately \$282, comprising \$200 for manual labor (8 hours at \$25/h) and \$82 for hardware depreciation (calculated for a \$30,000 platform over a one-year lifecycle). Given an average yield of 104 trajectories per day, the unit cost amounts to approximately \$2.71 per trajectory.

In contrast, our simulation framework, deployed on a server equipped with 8× NVIDIA RTX 4090 GPUs, operates at a daily cost of roughly \$71 (amortized to \$0.37 per hour). With an average rendering time of 16.2 minutes per GPU, the system enables the parallel generation of approximately 710 trajectories daily, reducing the unit cost to \$0.10 per trajectory. This represents a 27× reduction in cost and a 6.8× increase in throughput compared to physical data collection. These results demonstrate that physics-aligned simulation provides a highly scalable and cost-efficient alternative for generating large-scale training data.

D.5. Failure Cases

Our experiments indicate that policy performance is highly sensitive to the quality of generated data. When low-quality or invalid samples enter the training set, due to discriminator errors, they can poison the model, leading to systematic failures. Specifically, corrupted samples often result in unreasonable behaviors such as overreaching, premature gripper closure before contact, or misaligned grasps. In contrast, genuine out-of-distribution scenarios primarily produce failures due to limited generalization, such as missing the garment edge or mispositioning relative to the object. In our pipeline, the discriminator achieves over 99% success in filtering invalid trajectories, and its performance can be further improved with simple rule-based checks, ensuring that the synthetic dataset remains highly reliable and minimally corrupted.