FLASH: Fast Learning via GPU-Accelerated Simulation for High-Fidelity Deformable Manipulation in Minutes

Siyuan Luo[1*†], Bingyang Zhou[1*], Xin Liu[1], Zhenhao Huang[1], Gang Yang[1], Jason Pho[2], Ziqiu Zeng[1], and Fan Shi[1†]
[1]NUS Human-Centered Robotic Lab [2]Prana Labs
* Equal Contribution †Corresponding: sy.luo@nus.edu.sg, fan.shi@nus.edu.sg

Fig. 1: **Overview of the FLASH framework for GPU-parallel simulation and robot policy learning.** (a) GPU-native solvers enable scalable, massively parallel rollouts. (b–c) Sim-to-real results showing high-fidelity contact modeling across manipulation tasks. Our framework delivers 100–300× training speedup relative to real-time.

***Abstract*—Simulation frameworks such as Isaac Sim have enabled scalable robot learning for locomotion and rigid-body manipulation; however, contact-rich simulation remains a major bottleneck for deformable object manipulation. The continuously changing geometry of soft materials, together with large numbers of vertices and contact constraints, makes it difficult to achieve high accuracy, speed, and stability required for large-scale interactive learning. We present FLASH, a GPU-native simulation framework for contact-rich deformable manipulation, built on an accurate NCP-based solver that enforces strict contact and deformation constraints while being explicitly designed for fine-grained GPU parallelism. Rather than porting conventional single-instruction-multiple-data (SIMD) solvers to GPUs, FLASH redesigns the physics engine from the ground up to leverage modern GPU architectures, including optimized collision handling and memory layouts. As a result, FLASH scales to over 3 million degrees of freedom at 30 FPS on a single RTX 5090, while accurately simulating physical interactions. Policies trained solely on FLASH-generated synthetic data in minutes achieve robust zero-shot sim-to-real transfer, which we validate on physical robots performing challenging deformable manipulation tasks such as towel folding and garment folding, without any real-world demonstration, providing a practical alternative to labor-intensive real-world data collection.**

I. Introduction

Deformable object manipulation remains a bottleneck in robotics due to high degrees of freedom and complex dynamics. While learning-based methods are promising, their dependence on massive interaction data demands simulation platforms that offer both high physical fidelity and extreme computational throughput.

Existing GPU simulators often struggle to balance these needs due to architectural constraints in solver parallelism and memory access. We present **FLASH**, a GPU-native framework that unifies simulation, rendering, and learning. By co-optimizing these components, FLASH enables simulation-only training of robust, deployable policies. Our key contributions include:

- **High-Performance Solver:** A custom non-smooth Newton solver achieving real-time (30fps) simulation of 3M+ DoF scenes, a 100–300× speedup over traditional methods.
- **Efficiency:** A tightly coupled rendering and simulation pipeline that minimizes data latency for high-throughput vision-based learning.
- **Sim-to-Real:** Zero-shot transfer to real robots (e.g.,

TABLE I: Comparison of Robotics Simulation Platforms

Platform	Deformable	Dynamic Solver	Contact Model	Multi-Envs Training	GPU	Torch Support	Realistic Render
Isaac Sim[19]	✓	XPBD, FEM	Compliant	✓	✓	✓	✓
Newton[24]	✓	XPBD, VBD	Compliant	✓	✓	✗	✗
Physx[18]	✓	XPBD, FEM	Compliant	✗	✓	✗	✗
Genesis[1]	✓	PBD, MPM	Compliant	✓	✓	✓	✗
SOFA[34]	✓	FEM	LCP-PGS	✗	✓	✗	✗
Pinocchio[6]	✗	/	NCP (rigid)	✗	✗	✗	✗
Mujoco[31]	✗	/	CCP (rigid)	✓	✓	✓	✗
Flash (Ours)	✓	**FEM**	**NCP**	✓	✓	✓	✓

LCP/CCP/NCP: Linear/Cone/Nonlinear Complementarity Problem; PGS: Projected Gauss-Seidel method.

AdamU) enabled by high-fidelity contact modeling and systematic domain randomization.
- **Versatility:** A unified framework supporting diverse materials including cloth, foam, and volumetric objects.

II. Related Works

A. Deformable Objects Simulation

Simulators for deformable objects generally balance accuracy and speed. High-fidelity frameworks like SOFA [34] excel in medical modeling but lack the throughput for robot learning. Conversely, speed-oriented engines such as MuJoCo [31] initially prioritized rigid bodies. Recent GPU-accelerated platforms, including Isaac Sim [19] (PBD), Genesis [1] (MPM/PBD), and Newton [24] (PD/VBD [7]), have scaled soft-body simulation. However, these methods—ranging from high-accuracy FEM [29] to efficient PBD/XPBD [23, 17] and Local-Global PD [36]—still struggle to maintain physical stability under the massive parallelism required for contact-rich manipulation. **FLASH** addresses this by introducing a GPU-native non-smooth Newton solver.

B. Sim-to-Real Learning of Deformable Manipulation

Sim-to-real transfer mitigates the high cost of real-world data collection [5, 25, 37]. Early works utilized domain randomization [21, 10] or algorithmic supervision [28] to bridge the reality gap. To manage complexity, many approaches rely on motion primitives [35, 13, 3, 32, 16], dense flow prediction [33], generative dynamics [30], or language-guided planning [8, 2], often requiring expert demonstrations [27].

Despite these advances, gaps in dynamics and perception persist [26, 14]. Current solutions involve complex real-to-sim tuning [20], iterative residual learning [9], or abstracting perception via keypoints [15, 12] and neural fields [11]. Unlike these methods, **FLASH** provides high-throughput, contact-rich simulation that enables zero-shot transfer of policies trained entirely on diverse, synthetic interactions, enhancing real-world robustness without task-specific tuning.

III. Preliminary: Dynamics Modeling

FLASH builds upon a GPU-optimized local-global pipeline [36]. Given states at time t, implicit Euler integration computes updated positions $\mathbf{q}$ at $t+h$ by solving:

$$\mathbf{q} = \arg\min_{\mathbf{q}'} \left(\frac{1}{2h^2} \|\mathbf{M}^{\frac{1}{2}}(\mathbf{q}' - \tilde{\mathbf{q}})\|_F^2 + \sum_i \psi_i(\mathbf{q}') \right), \quad (1)$$

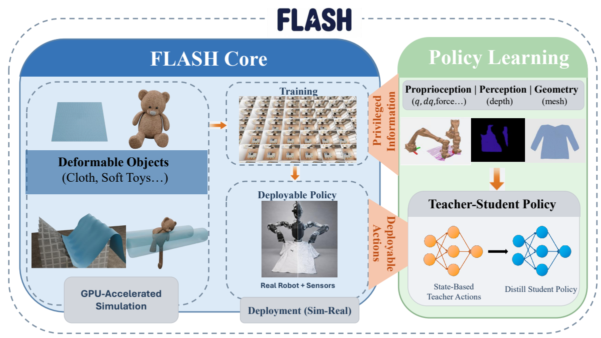

Fig. 2: **An illustration of the FLASH System.**

where $\mathbf{M}$ is the mass matrix, $\tilde{\mathbf{q}}$ is the inertial state, and ψ_i represents the elastic energy.

Following the Projective Dynamics (PD) framework [4], internal elastic forces are evaluated via auxiliary projections $\mathbf{p}_i$, linearizing the system into $\mathbf{A}\mathbf{q} = \mathbf{b}$, where $\mathbf{A} = \mathbf{M} + h^2 \sum_i w_i \mathbf{G}_i^\top \mathbf{G}_i$. To account for contact interactions, we augment the dynamics with Signorini–Coulomb laws $\boldsymbol{\phi}(\mathbf{q}, \boldsymbol{\lambda}) = \mathbf{0}$, leading to a non-smooth Newton system solved via each iteration:

$$\begin{bmatrix} \mathbf{A} & -\mathbf{J}^\top \\ \mathbf{J} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ h^2 \Delta\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{h} \end{bmatrix}. \quad (2)$$

Here, $\mathbf{J}$ is the contact Jacobian and $\mathbf{E}$ is the constraint compliance. The system is efficiently solved using the Schur complement $\mathbf{Z} = \mathbf{J}\mathbf{A}^{-1}\mathbf{J}^\top + \mathbf{E}$.

IV. FLASH System

FLASH is a unified framework designed for contact-rich deformable manipulation. By co-optimizing a GPU-native physics solver with a vision-based learning pipeline, it enables rapid policy acquisition and zero-shot sim-to-real transfer.

A. High-Performance Simulation Core

The computational bottleneck in contact-rich simulation is the resolution of the Schur complement $\mathbf{Z} = \mathbf{J}\mathbf{A}^{-1}\mathbf{J}^\top + \mathbf{E}$, which becomes fully dense when many constraints are active.

Lightweight Contact Solver: To maintain scalability, we adopt an inertia-dominated approximation:

$$\mathbf{Z} \approx \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\top + \mathbf{E}. \quad (3)$$

This relaxation preserves the sparsity of the system while retaining numerical robustness through the implicit treatment of elastic forces. This design allows the contact response to be governed by local geometry and inertia, which is ideal for soft materials like cloth or foam.

Multi-Env Parallelization: We exploit hardware parallelism by assembling multiple environments into a block-diagonal system $\bar{\mathbf{A}} = \mathrm{diag}(\mathbf{A}_1, \dots, \mathbf{A}_n)$. Since Eq. (3) eliminates dense coupling, all stages—including collision detection and Schur-complement solves—are executed as high-throughput sparse primitives using NVIDIA cuSPARSE and cuBLAS. This system-level parallelism achieves a 100×–300× speedup over real-time for scenes exceeding 3M DoF.

B. Policy Learning and Sim-to-Real

FLASH bridges the gap between synthetic data and physical deployment through a high-throughput distillation pipeline.

Learning Framework: We adopt a teacher-student distillation framework [22]. A state-based teacher policy, guided by heuristic primitives, supervises a student policy trained on a stacked history of proprioceptive and visual observations. To ensure robustness, we apply extensive domain randomization to both physical properties and visual depth streams.

Deployment: For zero-shot transfer, we employ a *segmented-depth-to-EE-pose* formulation. Perception is handled by segmenting target objects from raw depth via YOLOv8 and SAM, while the policy outputs end-effector (EEF) position deltas. This formulation decouples the manipulation strategy from specific robot kinematics, enabling robust deployment across hardware platforms via numerical Inverse Kinematics (IK).

V. Simulator & Learning Results

We evaluate FLASH's performance on a dual-sleeve T-shirt folding task against representative GPU simulators: Genesis (PBD), Isaac Sim (FEM), and Newton (VBD) on a single RTX 4090.

A. Cross-Simulator Comparison & Ablation

As shown in Fig. 3(a), FLASH most closely aligns with real-world folding behavior, achieving stable frictional sticking and symmetric folds without the numerical instabilities or excessive stiffness observed in baselines. Ablation studies (Fig. 3(b)) confirm that FLASH's parameters possess clear physical semantics: varying speed and bending stiffness produces predictable inertial and elastic responses. Even at ultra-low solver iterations (2 iterations), FLASH maintains numerical stability, showcasing its robustness for scalable robotics applications.

B. Throughput and Policy Acquisition

Throughput benchmarks (Table II) demonstrate FLASH's superior parallel scalability. While Isaac Sim is efficient, its physical artifacts (Fig. 3) limit its utility for generating high-fidelity training data. FLASH provides the optimal balance between physical realism and computational speed.

Using this high-throughput pipeline, we trained five diverse manipulation tasks (Fig. 4). On a single RTX 5090, obtaining deployable policies requires only 50 to 600 minutes of total wall-clock time. This efficiency enables rapid iteration from simulation to the real-world experiments discussed in Sec. VI.

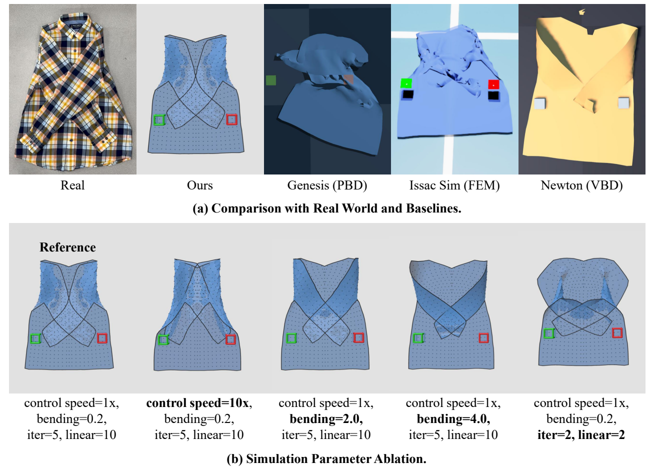

Fig. 3: **Folding Comparison & Ablation.** (a) FLASH aligns best with real-world results. (b) Predictable physical response under parameter variations.

TABLE II: **Throughput Comparison (ms/step).**

Simulator	#1	#32	#128	#256
FLASH (Ours)	**5.68**	24.79	90.41	185.10
Isaac Sim	18.02	**19.88**	**29.04**	**47.81**
Newton	9.57	24.47	143.76	480.64

VI. Real-Robot Experiments

Experimental Setup. We evaluate FLASH on two platforms: **Airbot Play** (dual desktop arms) and **AdamU** (humanoid). Both utilize a ZED Mini for depth perception. Policies predict incremental end-effector poses, executed via manufacturer SDKs or numerical IK.

A. Towel Folding: Efficiency and Robustness

Training on a single RTX 5090 yields a deployable policy in just 50 minutes. Despite the rapid training, the closed-loop policy exhibits:

- **Robust Initialization:** Tolerates ±8 cm spatial translation and arbitrary rotations.
- **Reactive Recovery:** As shown in Fig. 5, the system naturally re-attempts missed grasps and adapts to dynamic human interference (e.g., pulling the towel away) without resets.

Continuous Evaluation: We conducted 106 back-to-back trials on AdamU. The system achieved an 85.8% success rate (91/106), maintaining stable inference and convergence despite accumulated fabric deformations over one hour of operation.

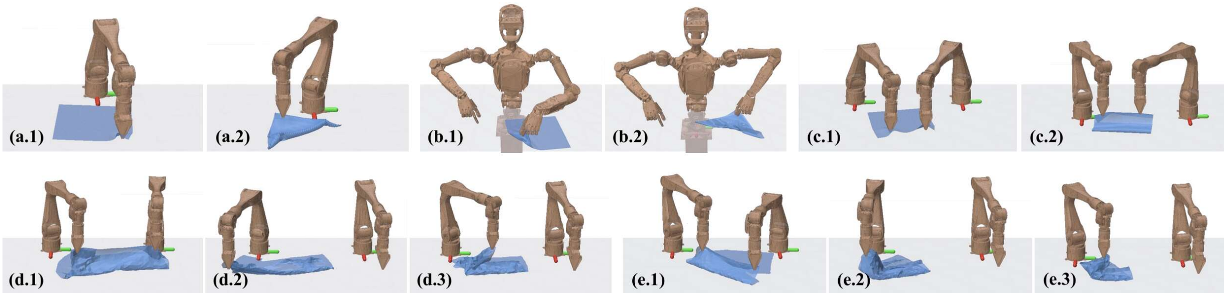

Fig. 4: **Learned policies.** Diverse folding tasks trained via FLASH in simulation.

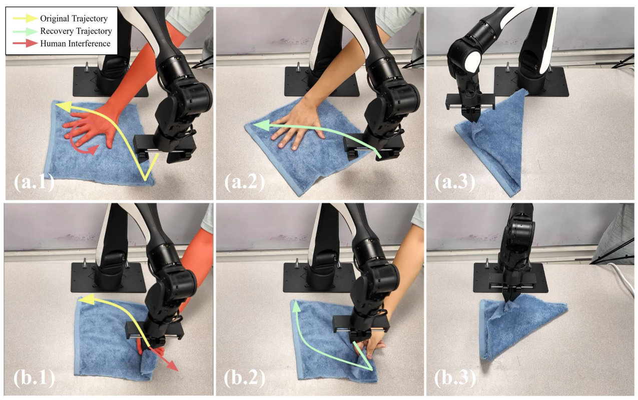

Fig. 5: **Recovery Behaviors.** FLASH adapts to human-induced displacements and perturbations in real-time.

B. Complex Garment Generalization

We extend FLASH to garments with diverse topologies (Fig. 6).

- **Shorts Folding:** Effectively handles dual-leg geometry with a **60%** success rate (12/20).
- **T-shirt Folding:** A long-horizon task requiring multi-step sequences. The policy preserves trajectory coherence, achieving a **70%** success rate (35/50) with zero-shot transfer.

C. Failure Analysis

Despite the robust sim-to-real transfer, we identified two primary failure modes:

- **Perception Bottleneck:** Depth sensor noise on thin fabrics and self-occlusions occasionally lead to grasp misalignment. This accounts for the majority of failures and the observed geometric imprecision in final folded states.
- **Hardware Abstraction Gap:** Our unified binary grasp model neglects motor-level dynamics (e.g., latency and backlash) and lacks tactile feedback. These factors introduce minor tracking deviations that can accumulate during long-horizon tasks.

VII. Conclusion and Discussion

In this paper, we presented FLASH, a GPU-native simulation platform for deformable simulation, rendering, and learning. By coupling GPU-parallel simulation with efficient depth rendering and a scalable training pipeline, FLASH supports high-throughput data generation for deformable manipulation learning. Through a suite of garment manipulation tasks, we demonstrate that FLASH reduces the sim-to-real gap and enables simulation-only training across multiple contact-rich scenarios. Notably, our vision-based policies can be trained in minutes and transferred zero-shot to real hardware, exhibiting robust execution and recovery behaviors. Our results strengthen the case for using large-scale synthetic interaction data to train deformable manipulation policies, substantially reducing reliance on costly real-world data collection. However, limitations remain. Our current implementation still incurs CPU–GPU transfer overhead in parts of the pipeline, leaving room for further optimization. In addition, handling more complex and longer-horizon garment manipulation will likely require improved learning formulations and supervision signals, including more effective reward or feedback design.

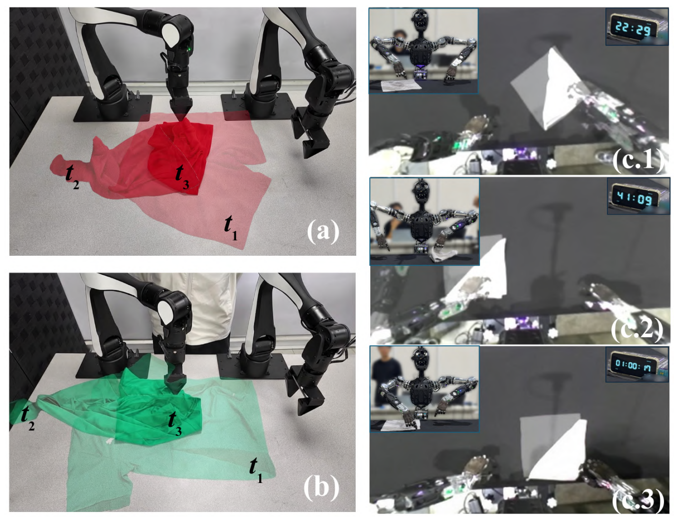

Fig. 6: **Real-World Results.** (Left) Bimanual folding of shorts and T-shirts. (Right) Snapshots from 60-min continuous evaluation.

References

[1] Genesis Authors. Genesis: A generative and universal physics engine for robotics and beyond, December 2024.

URL https://github.com/Genesis-Embodied-AI/Genesis.

[2] Oriol Barbany, Adrià Colomé, and Carme Torras. Bifold: Bimanual cloth folding with language guidance. *arXiv preprint arXiv:2501.16458*, 2025.

[3] David Blanco-Mulero, Gokhan Alcan, Fares J Abu-Dakka, and Ville Kyrki. Qdp: Learning to sequentially optimise quasi-static and dynamic manipulation primitives for robotic cloth manipulation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 984–991. IEEE, 2023.

[4] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. Projective dynamics: Fusing constraint projections for fast simulation. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 787–797. 2023.

[5] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[6] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE International Symposium on System Integrations (SII)*, 2019.

[7] Anka He Chen, Ziheng Liu, Yin Yang, and Cem Yuksel. Vertex block descent. *ACM Transactions on Graphics (TOG)*, 43(4):1–16, 2024.

[8] Haonan Chen, Junxiao Li, Ruihai Wu, Yiwei Liu, Yiwen Hou, Zhixuan Xu, Jingxiang Guo, Chongkai Gao, Zhenyu Wei, Shensi Xu, et al. Metafold: Language-guided multi-category garment folding framework via trajectory generation and foundation model. *arXiv preprint arXiv:2503.08372*, 2025.

[9] Cheng Chi, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects. *The International Journal of Robotics Research*, 43(4):389–404, 2024.

[10] Alexander Clegg, Zackory Erickson, Patrick Grady, Greg Turk, Charles C Kemp, and C Karen Liu. Learning to collaborate from simulation for robot-assisted dressing. *IEEE Robotics and Automation Letters*, 5(2):2746–2753, 2020.

[11] Donatien Delehelle, Darwin Caldwell, and Fei Chen. Garfield: Addressing the visual sim-to-real gap in garment manipulation with mesh-attached radiance fields. In *2024 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 77–84. IEEE, 2024.

[12] Yuhong Deng and David Hsu. General-purpose clothes manipulation with semantic keypoints. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13181–13187. IEEE, 2025.

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

[14] Wenkang Hu, Xincheng Tang, Yitong Li, Zhengjie Shu, Wei Li, Huamin Wang, Ruigang Yang, et al. Real garment benchmark (rgbench): A comprehensive benchmark for robotic garment manipulation featuring a high-fidelity scalable simulator. *arXiv preprint arXiv:2511.06434*, 2025.

[15] Thomas Lips, Victor-Louis De Gusseme, and Francis Wyffels. Learning keypoints for robotic cloth manipulation using synthetic data. *IEEE Robotics and Automation Letters*, 9(7):6528–6535, 2024.

[16] Haoran Lu, Ruihai Wu, Yitong Li, Sijie Li, Ziyu Zhu, Chuanruo Ning, Yan Zhao, Longzan Luo, Yuanpei Chen, and Hao Dong. Garmentlab: A unified simulation and benchmark for garment manipulation. *Advances in Neural Information Processing Systems*, 37:11866–11903, 2024.

[17] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pages 49–54, 2016.

[18] Miles Macklin, Kier Storey, Michelle Lu, Pierre Terdiman, Nuttapong Chentanez, Stefan Jeschke, and Matthias Müller. Small steps in physics simulation. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366779. doi: 10.1145/3309486.3340247. URL https://doi.org/10.1145/3309486.3340247.

[19] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[20] Omey M Manyar, Hantao Ye, Siddharth Mayya, Fan Wang, and Satyandra K Gupta. Real-to-sim parameter learning for deformable packages using high-fidelity simulators for robotic manipulation. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 89206, page V02AT02A004. American Society of Mechanical Engineers, 2025.

[21] Jan Matas, Stephen James, and Andrew J. Davison. Sim-to-real reinforcement learning for deformable object manipulation. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 734–743. PMLR, 29–31 Oct 2018.

[22] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning

robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022.

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

[24] Newton Project Contributors. Newton: An open-source, gpu-accelerated physics engine for robotics, 2025. URL https://github.com/newton-physics/newton. Initiated by Disney Research, Google DeepMind, and NVIDIA.

[25] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[26] Yingdong Ru, Lipeng Zhuang, Zhuo He, Florent P Audonnet, and Gerardo Aragon-Caramasa. Can real-to-sim approaches capture dynamic fabric behavior for robotic fabric manipulation? *arXiv preprint arXiv:2503.16310*, 2025.

[27] Gautam Salhotra, I-Chun Arthur Liu, Marcus Dominguez-Kuhne, and Gaurav S Sukhatme. Learning deformable object manipulation from expert demonstrations. *IEEE Robotics and Automation Letters*, 7(4):8775–8782, 2022.

[28] 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 *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9651–9658. IEEE, 2020.

[29] Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *Acm siggraph 2012 courses*, pages 1–50. 2012.

[30] Tongxuan Tian, Haoyang Li, Bo Ai, Xiaodi Yuan, Zhiao Huang, and Hao Su. Diffusion dynamics models with generative state estimation for cloth manipulation. *arXiv preprint arXiv:2503.11999*, 2025.

[31] 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.

[32] 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.

[33] Thomas Weng, Sujay Man Bajracharya, Yufei Wang, Khush Agrawal, and David Held. Fabricflownet: Bimanual cloth manipulation with a flow-based policy. In *Conference on Robot Learning*, pages 192–202. PMLR, 2022.

[34] J Westwood et al. *SOFA—an open source framework for medical simulation*, volume 125. IOS Press Amsterdam, The Netherlands, 2007.

[35] Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. Learning to manipulate deformable objects without demonstrations. In *Robotics: Science and Systems*, 2020.

[36] Ziqiu Zeng, Siyuan Luo, Fan Shi, and Zhongkai Zhang. Fast but accurate: A real-time hyperelastic simulator with robust frictional contact. 44(4), July 2025. ISSN 0730-0301. doi: 10.1145/3730834. URL https://doi.org/10.1145/3730834.

[37] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.