Diffusion Dynamics Models with Generative State Estimation for Cloth Manipulation

Anonymous Author(s)

***Abstract*—Manipulating deformable objects like cloth is challenging due to their complex dynamics, near-infinite degrees of freedom, and frequent self-occlusions, which complicate state estimation and dynamics modeling. Prior work has struggled with robust cloth state estimation, while dynamics models, primarily based on Graph Neural Networks (GNNs), are limited by their locality. Inspired by recent advances in generative models, we hypothesize that these expressive models can effectively capture intricate cloth configurations and deformation patterns from data. Building on this insight, we propose a diffusion-based generative approach for both perception and dynamics modeling. Specifically, we formulate state estimation as reconstructing the full cloth state from sparse RGB-D observations conditioned on a canonical cloth mesh and dynamics modeling as predicting future states given the current state and robot actions. Leveraging a transformer-based diffusion model, our method achieves high-fidelity state reconstruction while reducing long-horizon dynamics prediction errors by an order of magnitude compared to GNN-based approaches. Integrated with model-predictive control (MPC), our framework successfully executes cloth folding on a real robotic system, demonstrating the potential of generative models for manipulation tasks with partial observability and complex dynamics.**

I. Introduction

Textile deformable objects, such as clothing, are ubiquitous in daily life. Yet, manipulating these objects is a long-standing challenge in robotics [20, 42], due to their complex geometric structures and dynamics. Effective cloth manipulation requires accurately estimating the cloth's geometry despite severe self-occlusions, as well as reasoning over its complex, continuous dynamics to optimize actions. These difficulties highlight the need for advancements in both (i) state estimation and (ii) dynamics modeling to enable robust robotic cloth manipulation.

State estimation for cloth is particularly challenging due to frequent self-occlusions arising from its highly deformable structure. While humans intuitively infer full object shapes from partial observations using prior experience, most existing methods are unable to fully capture the complex mapping between highly partial observations and high-dimensional object states [7, 15, 1]. A promising direction is to develop perception models that can *"imagine"* full states from partial observations by leveraging extensive prior experience, akin to human reasoning.

Modeling cloth dynamics poses another significant challenge due to its highly nonlinear nature. Current approaches typically represent cloth using particle- or mesh-based structures and model their interactions with graph neural networks (GNNs)[44, 15, 18]. GNNs offer advantages in data-scarce domains through spatial equivariance and locality, but they scale inefficiently with the number of graph nodes[31]. Moreover, the locality inherent to graph structures often limits their ability to capture long-range dependencies, which is crucial for accurate dynamics modeling.

In this work, we formulate state estimation and dynamics prediction as conditional generation processes. State estimation reconstructs full states from partial observations, while dynamics prediction generates future states conditioned on the current state and robot actions. To model these complex high-dimensional mappings, we employ diffusion-based models, inspired by their recent successes in capturing complex data distributions in computer vision [22, 19], science [32], and robotics [8]. We hypothesize that diffusion models with scalable architecture (e.g., Transformer [35]) can enable accurate state reconstruction and dynamics modeling.

Building on these insights, we introduce UniClothDiff, a unified framework that integrates a Diffusion Perception Model (DPM), a Diffusion Dynamics Model (DDM), and model-predictive control for cloth manipulation. Conceptually, DPM leverages diffusion models and Transformers to reconstruct full cloth states from sparse and occluded RGB-D observations, while DDM predicts long-horizon dynamics conditioned on current states and actions. Trained on a large-scale cloth interaction dataset with 500K transitions in simulation and evaluated in both simulation and real-world, our models achieve substantial performance gains: DPM achieves superior performance compared to prior approaches in cloth state estimation, and DDM reduces long-horizon prediction error by an order of magnitude compared to GNN-based baselines. With an embodiment-agnostic action representation, our framework can be deployed on both parallel grippers and dexterous hands. Real-world experiments demonstrate superior manipulation performance over previous approaches, highlighting the potential of generative modeling in deformable object manipulation.

II. Method

A. Overview

We address the challenge of manipulating cloth with significant self-occlusions into target configurations. Our problem formulation comprises three key spaces: observation space $\mathcal{O}$, state space $\mathcal{S}$, and action space $\mathcal{A}$. The objective is to learn two essential components: a state estimator $g : \mathcal{O} \to \mathcal{S}$ and a transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ for model-based control.

At each timestep, the system processes multiview RGB-D observations $o_t \in \mathcal{O}$, represented as $o_t = \{I_t^0, I_t^1, \ldots, I_t^{l-1}\}$ with l camera views, to estimate the cloth's 3D state $s_t \in \mathcal{S}$

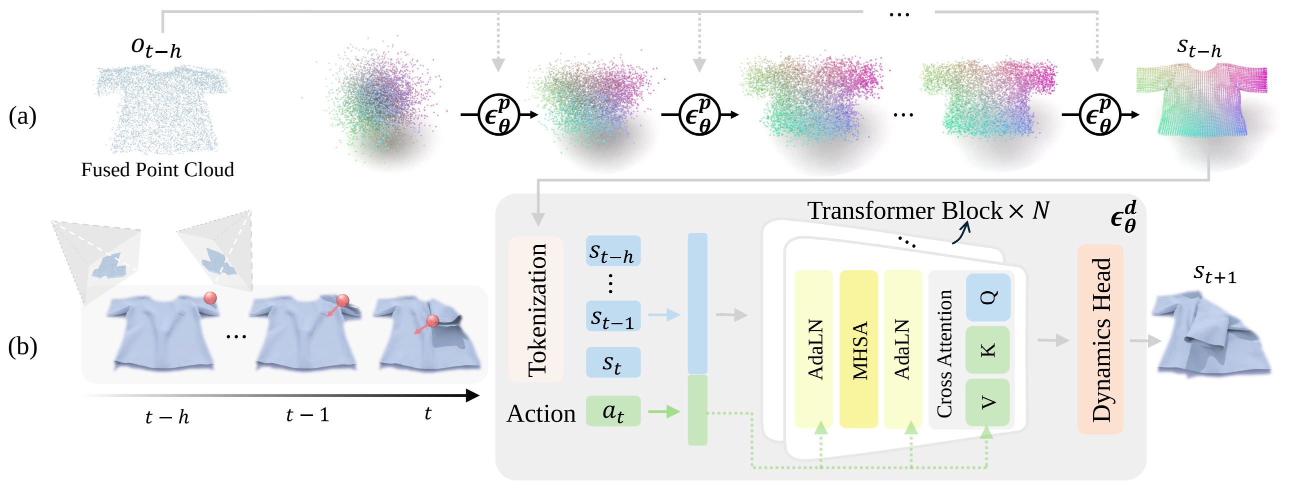


Fig. 1: **Overview. (a) Perception:** Our Diffusion Perception Model (DPM) reconstructs the full cloth state from a partial point cloud. Using a denoising process parameterized by $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^{\boldsymbol{p}}$, DPM refines the cloth state over K denoising steps, starting from random noise. **(b) Dynamics Prediction:** Our Diffusion Dynamics Model (DDM) generates future cloth states based on the current estimated state and robot actions, using a transformer-based architecture.

given canonical state of the template mesh s_c. The state of the cloth is defined by a mesh $s_t = \{V_t, E_t\}$, where E_t represents the invariant edge connectivity and $V_t \in \mathbb{R}^{N_v \times 3}$ denotes the positions of vertices in 3D space where N_v denotes the number of vertices. We propose that generative models can effectively infer unobserved patterns in partial RGB-D observations, enabling robust state estimation.

Given the estimated state, a learned dynamics model f predicts the future state $s_{t+1} \in \mathcal{S}$ based on state history $s_{t-i:t} \in \mathcal{S}$ and planned action $a_t \in \mathcal{A}$. This dynamics model is incorporated into a model-predictive control framework to optimize action sequences for achieving target state s_g:

$$(a_0, ..., a_{H-1}) = \underset{a_0,...,a_{H-1} \in \mathcal{A}}{\arg\min} \mathcal{J}(\mathcal{T}(s_0, (a_0, .., a_{H-1})), s_g)$$

B. State Estimation

We first address the challenging problem of inferring complete cloth configurations from partial observations.

a) Conditional Diffusion Process: We formulate cloth state estimation as a conditional denoising diffusion process. We use object point cloud as conditioning input to the model, as the sim-to-real gap is minimized in the particle space [5].

Specifically, we model the conditional distribution $p(s|s_c, \mathbf{e}_{\text{pc}})$ using standard denoising diffusion probabilistic model (DDPM) [11], where s_c represents the state of the canonical cloth mesh and $\mathbf{e}_{\text{pc}}$ denotes the embedding of the conditional point cloud. To get point cloud embedding, we partition the point cloud into patches by first sampling M center points using farthest point sampling (FPS) and performing K-Nearest Neighbors (KNN) clustering. Then each resulting patch is processed through a PointNet [29] to obtain its embedding representation $\mathbf{e}_{\text{pc}} \in R^{B \times M \times D_1}$. where B is the batch size and D_1 is the dimension of the point cloud embedding.

In the forward process, starting from the initial state s_0, gaussian noise is gradually added at levels $t \in \{1, ..., T\}$ to get noisy state as:$s_t = \sqrt{\bar{\alpha}_t} s_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, $\bar{\alpha}_t := \prod_{s=1}^{t} 1 - \beta_s$, and $\{\beta_1, \ldots, \beta_T\}$ is the variance schedule of a process with T steps. In the reverse process, starting from a noisy state s_t sampled from the normal distribution, the conditional denoising network ϵ_θ^p gradually denoising from s_t to s_{t-1} and finally construct s_0.

b) Model Architecture: We adopt vanilla Vision Transformer (ViT) architecture [9] as our backbone, which has been shown to be highly scalable in image and video generation [26, 22]. The model takes a point cloud and a canonical template mesh as input, in addition to the noisy mesh state that requires denoising. We detail our network architecture and training objective below.

Tokenization. We tokenize the input mesh as non-overlapping vertex patches. We first use farthest point sampling (FPS) to sample a fixed number of points as patch centers $C \in \mathbb{R}^{N \times 3}$. To patchify the mesh vertices, we use the N centers obtained from FPS to construct a Voronoi diagram in the 3D points space. This tessellation divides the point cloud into N distinct regions, where each region contains all points closer to its associated center than to any other center. Each Voronoi cell is treated as a distinct patch, encompassing a local neighborhood of points which will then go through a PointNet [29] layer for feature extraction.

Conditioning. Following the tokenization process, the input token is directly subjected to a sequence of transformer blocks for processing. To effectively condition the point cloud embedding, we adopt two approaches. First, the conventional layer normalization is replaced with an adaptive layer normalization (AdaLN) [39] to better incorporate conditional information. Then, we incorporate conditional information through a cross-attention layer positioned after the multi-head self-attention (MHSA). In this cross-attention operation, the hidden states x

serve as the query vector, while the conditional information acts as both the key and value vectors. The computation proceeds as $x = \text{CrossAttention}(W_Q^{(c)}x, W_K^{(c)}\mathbf{e}_{pc}, W_V^{(c)}\mathbf{e}_{pc})$ where $W^{(c)}$ are learnable parameters, enabling effective conditioning during the learning process.

Decoding. Finally, the decoding process transforms the hidden states x into 3D vertex coordinates through a two-stage process. First, we employ distance-weighted interpolation to upsample the hidden states, where interpolation weights are computed from canonical-space distances between vertices and their corresponding patch centers. This operation produces an intermediate representation $x \in \mathbb{R}^{B \times N_v \times D_2}$. A Multi-Layer Perceptron (MLP) then maps this representation to the final output $x_{out} \in \mathbb{R}^{B \times N_v \times 3}$, yielding the predicted noise added onto the 3D coordinates for each vertex during the diffusion forward process. Details of our model are presented in Appendix E-A.

c) Training: For state estimation, the denoising model $\epsilon_\theta^p(s^{(k)}|s_c, \mathbf{e}_{\text{pc}})$ for DPM is trained by minimizing the loss:

$$\mathcal{L}_{\text{MSE}} = \left\| \epsilon - \epsilon_\theta^p \left(\sqrt{1-\beta^{(k)}}s + \sqrt{\beta^{(k)}}\epsilon \middle| s_c, \mathbf{e}_{\text{pc}} \right) \right\|^2$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $\beta^{(k)} \in \mathbb{R}$ are K different noise levels for $k \in [1, K]$. Training details are presented in Appendix E-B.

C. Dynamics Prediction

Given the estimated state, the goal of dynamics prediction is to reason about future states of the cloth given robot actions. We extend our state estimation architecture to model dynamics by modifying the condition input to incorporate robot actions and enhancing the temporal modeling capability with additional temporal attention layers. The remaining components, including tokenization, training objective, and decoding of the model, are identical to those in the state estimation framework.

Conditional Diffusion Process. To learn the conditional posterior distribution $p(s_{t+1:t+j+1}|a_t, s_{t-i:t})$, we parameterize it using diffusion models. Here, a_t represents the robot action, $s_{t-i:t}$ denotes the historical states, and $s_{t+1:t+j+1}$ is the j frame future states to be predicted at timestep t. Following prior work [44, 41], we heuristically set $i = 3$ and $j = 5$. The diffusion reverse process construct s_t conditioned on history frames and action by gradually denoising from a normal distribution with the denoising network ϵ_θ^d. Since we use delta end-effector position as action representation, to effectively encode the action space, we employ a Fourier feature-based embedding following NeRF [24] to represent continuous spatial information, with detailed formulation in Appendix.

D. Model-Based Planning

We integrate our diffusion dynamics model with Model Predictive Control (MPC) for robotic cloth manipulation. Given a current cloth state sequence $s_{t-i:t} \in \mathcal{S}$ and target state s_g, we optimize an action sequence $\{a_t\}_{t=0}^{T-1}$ by minimizing:

$$\min_{\{a_t\}_{t=0}^{T-1}} \phi\left(s_T, s_g\right) + \sum_{t=0}^{T-1} \ell\left(s_t, a_t\right), \tag{1}$$

where ϕ combines weighted MSE and chamfer distance, and ℓ enforces action smoothness. Our planning framework utilizes Model Predictive Path Integral (MPPI) [37] for sampling-based optimization. Refer to Appendix E-C for details on the planning algorithm and hyperparameters.

Category	Method	Simulation		
		↓ MSE (10^{-1})	↓ CD (10^{-1})	↓ EMD (10^{-1})
Cloth	TRTM [1]	5.07 ± 0.22	2.67 ± 0.61	1.65 ± 0.71
	Transformer	5.44 ± 0.41	2.17 ± 0.19	1.61 ± 0.45
	DPM	**2.32** ± 0.21	**1.95** ± 0.25	**1.48** ± 0.47
T-shirt	GarmentNets [7]	18.6 ± 1.35	6.23 ± 0.79	2.79 ± 0.64
	MEDOR [15]	21.0 ± 1.54	6.87 ± 0.95	2.24 ± 0.29
	TRTM [1]	6.30 ± 0.45	5.15 ± 0.96	2.15 ± 0.29
	Transformer	9.12 ± 0.57	5.56 ± 0.63	1.99 ± 0.62
	DPM	**2.76** ± 0.19	**3.22** ± 0.41	**1.95** ± 0.56

TABLE I: **Quantitative results on state estimation.** Lower values indicate better performance. Errors represent a 95% confidence interval.

III. Experiments

In this section, we investigate three key research questions:

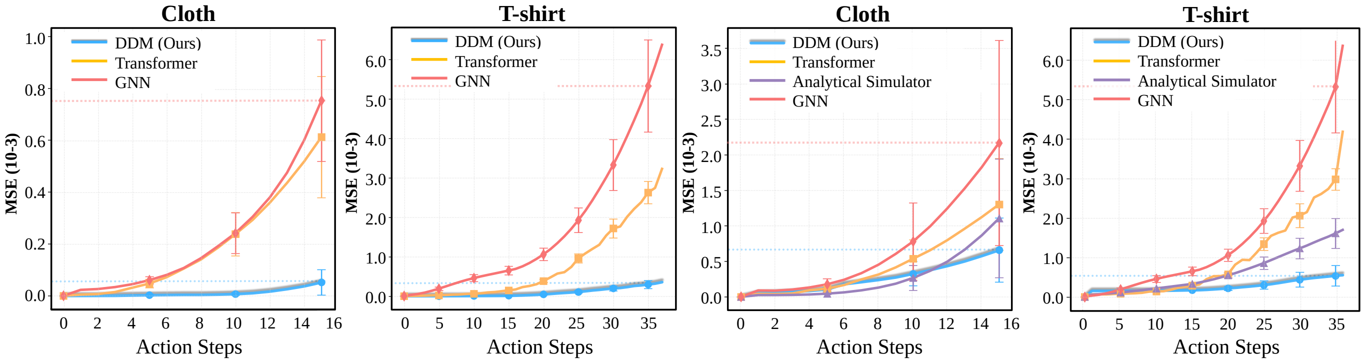
1) How effectively does the Diffusion Perception Model handle self-occlusions inherent in cloth manipulation?
2) How does the Diffusion Dynamics Model improve dynamics prediction compared to prior approaches?
3) How do these enhanced perception and dynamics models translate to overall system performance?

We address these questions in three stages: evaluating state estimation accuracy (Section III-A), assessing dynamics modeling performance (Section III-B), and validating our approach through real-world cloth manipulation experiments (Section III-C).

A. State Estimation

1) Baselines: We evaluate our perception module against four baseline approaches: **GarmentNets**[7], **MEDOR**[15], **TRTM**[1] and **Transformer**, an ablated version of our model.

2) Results: We evaluate our method against baselines in both simulation and real-world environments using Mean Squared Error (MSE), Chamfer Distance (CD), and Earth Mover's Distance (EMD). Quantitative results are presented in Table I. In the T-shirt object, TRTM [1] and Transformer greatly outperform GarmentNets and MEDOR [15], demonstrating that the topological information provided by the template cloth mesh significantly enhances the perception capabilities. Leveraging the cloth modeling prior during the learning process, TRTM [1] demonstrates better performance compared to Transformer. Our approach achieves further performance gains over both TRTM [1] and Transformer, highlighting the significant contributions of diffusion models to the task.



(a) Oracle Perception as Input

(b) DPM Perception as Input

Fig. 2: **Long-horizon dynamics prediction error over time.** Mean Squared Error (MSE) in dynamics prediction over time under two scenarios: (a) using oracle simulation states, and (b) using DPM perception estimates, evaluated on cloths and T-shirts. Our method significantly reduces error accumulation in both cases. Error bars represent 95% confidence intervals.

B. Dynamics Prediction Results

1) Baselines: We evaluated our diffusion dynamics models against three baseline approaches: a GNN-based method, an analytical simulator, and an ablated version of our model. For each baseline, we analyze the mean squared error (MSE) across different timesteps on clothes and T-shirts.

2) Results: Error analysis over time (Figure 2) shows that DDM consistently outperforms all baselines across both object types and input settings. When using ground-truth states, GNN performs the worst overall, except on cloth objects where its performance is comparable to Transformer due to simpler topology. On T-shirts with more complex dual-level topology, Transformer surpasses GNN, demonstrating the transformer architecture's advantage in modeling geometric and temporal dependencies. DDM achieves the lowest MSE at all timesteps, with minimal error accumulation, benefiting from the probabilistic nature of diffusion models that better capture cloth state dynamics compared to direct MSE supervision. When using estimated (noisy) states, we include Analytical Simulator as a baseline. While it initially performs well on cloth, its sensitivity to input noise leads to significant long-horizon degradation, especially on T-shirts. In contrast, DDM remains robust under perception noise and complex dynamics.

C. Real World Planning Results

a) Comparative Analysis: We demonstrate the seamless integration of DPM and DDM within a Model Predictive Control (MPC) framework for complex cloth manipulation tasks. Our approach is benchmarked against GNN as the dynamics model. We evaluate in three types of occlusion: self-occlusion, external occlusion by other objects (e.g., a robotic arm), and combined occlusion. Quantitative and qualitative results are presented in Table II and Figure 3, respectively. Our method consistently outperforms GNN across all occlusion scenarios. In simpler tasks like cloth folding, our model improves SR by 30%. For more complex cases, such as dual-level topology T-shirts where GNN struggles, it achieves up to a 30% gain.

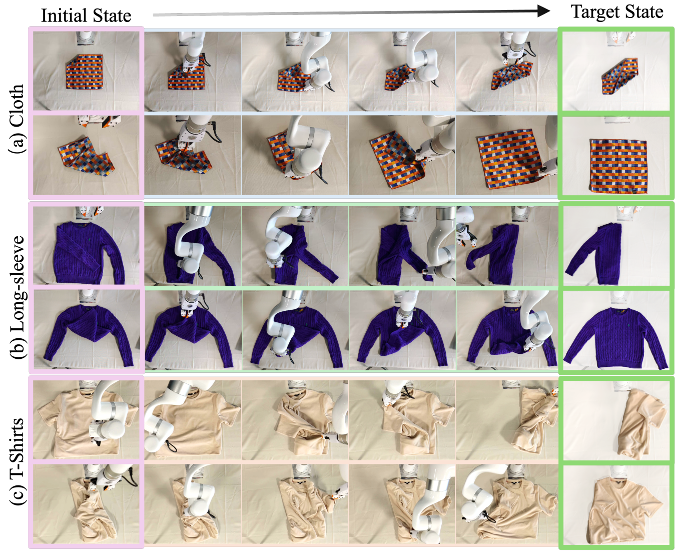


Fig. 3: **Qualitative results of real-world system deployment.**

Method	Cloth			T-shirt			Long-sleeve		
	Self	Ext.	Comb.	Self	Ext.	Comb.	Self	Ext.	Comb.
GNN	6/10	4/10	3/10	1/10	2/10	2/10	2/10	2/10	0/10
Ours	**9/10**	**8/10**	**6/10**	**9/10**	**7/10**	**6/10**	**7/10**	**6/10**	**4/10**

TABLE II: **Quantitative results of real-world manipulation.** Each scenario is repeated 10 times with randomized states.

IV. Conclusion

We present UniClothDiff, a unified framework for cloth manipulation that leverages Transformer-based diffusion models for both state estimation and long-horizon dynamics prediction. Our method reconstructs full cloth configurations from partial RGB-D inputs and significantly outperforms prior GNN-based approaches. When combined with model-based control, it enables precise and reliable manipulation. Moreover, our embodiment-agnostic action representation supports zero-shot transfer to novel grippers. Extensive experiments highlight the power of generative models in deformable object manipulation, advancing the robustness and versatility of robotic systems.

References

[1] Trtm: Template-based reconstruction and target-oriented manipulation of crumpled cloths. 2024.

[2] Bo Ai, Stephen Tian, Haochen Shi, Yixuan Wang, Cheston Tan, Yunzhu Li, and Jiajun Wu. Robopack: Learning tactile-informed dynamics models for dense packing. *Robotics: Science and Systems (RSS)*, 2024. URL https://arxiv.org/abs/2407.01418.

[3] Yahav Avigal, Lars Berscheid, Tamim Asfour, Torsten Kröger, and Ken Goldberg. Speedfolding: Learning efficient bimanual folding of garments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 2022. doi: 10.1109/IROS47612.2022.9981402.

[4] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. Projective dynamics: fusing constraint projections for fast simulation. *ACM Trans. Graph.*, 33(4), July 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601116. URL https://doi.org/10.1145/2601097.2601116.

[5] Bryan Chen, Alexander Sax, Francis Lewis, Iro Armeni, Silvio Savarese, Amir Zamir, Jitendra Malik, and Lerrel Pinto. Robust policies via mid-level visual representations: An experimental study in manipulation and navigation. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 2328–2346. PMLR, 16–18 Nov 2021. URL https://proceedings.mlr.press/v155/chen21f.html.

[6] Siwei Chen, Xiao Ma, Yunfan Lu, and David Hsu. Ab initio particle-based object manipulation. In Dylan A. Shell, Marc Toussaint, and M. Ani Hsieh, editors, *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021. doi: 10.15607/RSS.2021.XVII.071. URL https://doi.org/10.15607/RSS.2021.XVII.071.

[7] Cheng Chi and Shuran Song. Garmentnets: Category-level pose estimation for garments via canonical space shape completion. In *The IEEE International Conference on Computer Vision (ICCV)*, 2021.

[8] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[10] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning (CoRL)*, 2024.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[12] Ryan Hoque, Daniel Seita, Ashwin Balakrishna, Aditya Ganapathi, Ajay Kumar Tanwani, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. Visuospatial foresight for physical sequential fabric manipulation. *Auton. Robots*, 46(1):175–199, January 2022. ISSN 0929-5593. doi: 10.1007/s10514-021-10001-0. URL https://doi.org/10.1007/s10514-021-10001-0.

[13] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3d vitac:learning fine-grained manipulation with visuo-tactile sensing. In *Proceedings of Robotics: Conference on Robot Learning(CoRL)*, 2024.

[14] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024.

[15] Zixuan Huang, Xingyu Lin, and David Held. Mesh-based dynamics with occlusion reasoning for cloth manipulation. *ArXiv*, abs/2206.02881, 2022. URL https://api.semanticscholar.org/CorpusID:248942073.

[16] Rishabh Jangir, Guillem Alenyà, and Carme Torras. Dynamic cloth manipulation with deep reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4630–4636, 2020. doi: 10.1109/ICRA40945.2020.9196659.

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[18] Y. Li et al. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *arXiv preprint arXiv:1810.01566*, 2018.

[19] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.

[20] Alberta Longhini, Yufei Wang, Irene Garcia-Camacho, David Blanco-Mulero, Marco Moletta, Michael Welle, Guillem Alenyà, Hang Yin, Zackory Erickson, David Held, Júlia Borràs, and Danica Kragic. Unfolding the literature: A review of robotic cloth manipulation. *Annual Review of Control, Robotics, and Autonomous Systems*, 2024-12-02. ISSN 2573-5144.

[21] Mickaël Ly, Jean Jouve, Laurence Boissieux, and Florence Bertails-Descoubes. Projective dynamics with dry frictional contact. *ACM Trans. Graph.*, 39(4), August 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392396. URL https://doi.org/10.1145/3386569.3392396.

[22] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

[23] 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. URL https://proceedings.mlr.press/v87/matas18a.html.

[24] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[25] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36, 2024.

[26] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[27] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. 2025. URL https://doi.org/10.48550/arXiv.2501.09747.

[28] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022.

[29] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[31] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems*, 33:12559–12571, 2020.

[32] Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. *Advances in neural information processing systems*, 36:45259–45287, 2023.

[33] H. Shi et al. Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[34] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 642–660. PMLR, 2023. URL https://proceedings.mlr.press/v229/shi23a.html.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[36] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *IJCV*, 2024.

[37] Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. Aggressive driving with model predictive path integral control. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1433–1440, 2016. doi: 10.1109/ICRA.2016.7487277.

[38] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[39] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. *Advances in neural information processing systems*, 32, 2019.

[40] Wilson Yan, Ashwin Vangipuram, Pieter Abbeel, and Lerrel Pinto. Learning predictive representations for deformable objects using contrastive estimation. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 564–574. PMLR, 16–18 Nov 2021. URL https://proceedings.mlr.press/v155/yan21a.html.

[41] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

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

[43] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[44] Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[45] Xiaoshuai Zhang, Rui Chen, Ang Li, Fanbo Xiang,

Yuzhe Qin, Jiayuan Gu, Zhan Ling, Minghua Liu, Peiyu Zeng, Songfang Han, Zhiao Huang, Tongzhou Mu, Jing Xu, and Hao Su. Close the optical sensing domain gap by physics-grounded active stereo sensor simulation. *IEEE Transactions on Robotics*, pages 1–19, 2023. doi: 10.1109/TRO.2023.3235591.

Appendix A
Limitations

Our approach demonstrates strong performance in cloth manipulation but has certain limitations. While transformer-based diffusion models enable accurate state estimation and long-horizon dynamics prediction, they require substantial computational resources. Additionally, achieving the intended contact point can be challenging in real-world execution when the cloth is folded, creating overlapping layers. This issue can be mitigated by integrating tactile sensing to enhance grasp precision. As our method uses a particle-based representation, it is inherently compatible with tactile feedback for more robust manipulation [2, 13], which we leave as future work.

Moreover, our method does not explicitly perform online system identification, which could improve adaptability to different object materials. However, this aspect is orthogonal to our diffusion-based dynamics model architecture. Prior work has shown that historical information can be used to infer physical parameters [44, 2], and a similar system identification phase could be readily incorporated into our framework.

Appendix B
Related Work

A. Deformable Object Manipulation

The manipulation of deformable objects, particularly garments, remains a fundamental challenge in robotics due to high-dimensional state spaces and complex dynamics. Current approaches fall into two main categories: model-free and model-based methods. Model-free approaches, including reinforcement learning (RL) [23, 16] and imitation learning (IL) [3, 10, 8, 43, 27], learn direct observation-to-action mappings through end-to-end training. However, these methods struggle with precise shape control due to the lack of explicit object dynamics modeling. Model-based approaches require accurate state estimation [6, 33, 34, 2, 14], challenging when deformable objects like cloth exhibit severe self-occlusions. Furthermore, learning dynamics models demands extensive training data covering large state and action spaces. To overcome these challenges, we propose leveraging the expressive generative models, specifically diffusion models, for both full-state estimation from partial observations and dynamics modeling using large-scale simulation data.

B. Learning-Based Dynamics Models

Learning-based dynamics models aim to predict state transitions directly from interaction data, with the choice of state representation playing a critical role. Pixel-based representations frame the problem as action-conditioned video prediction [12, 40]. However, these models often require extensive training data, are sensitive to occlusions, and struggle to generate physically plausible predictions due to the lack of explicit 3D structure [41]. In contrast, particle-based representations, like point clouds and meshes, offer a more structured and physically grounded approach to modeling deformable objects. These representations are typically processed using graph neural networks (GNNs), which model state transitions through message passing [44, 15, 33, 34, 2]. While GNNs have demonstrated effectiveness, we find that diffusion models provide a more scalable and expressive alternative, enabling accurate state transition learning from large-scale datasets and advancing deformable object dynamics modeling.

C. Diffusion Models

Diffusion models [11] have emerged as a powerful paradigm in generative modeling, capable of capturing complex, high-dimensional data distributions precisely. They have demonstrated significant success across various domains, including image generation [30, 26], video generation [36, 22], 3D shape synthesis [28], and robotic policy learning [8, 43]. In this study, we extend the representational capabilities of diffusion models to tackle key challenges in deformable object manipulation: perception and dynamics modeling. Specifically, we employ diffusion models to recover full geometric details from partial observations, enabling accurate state estimation despite severe occlusions and complex object configurations. Additionally, we employ diffusion models to learn expressive dynamics models that capture highly non-linear state transitions, enabling robust model-based planning.

Appendix C
Experiment Setup

A. Task Description

We evaluate our method on challenging cloth manipulation tasks characterized by significant visual occlusion and complex physical dynamics, demonstrating the real-world performance of our diffusion-based perception and dynamics model.

Cloth folding. This task explores robotic cloth folding tasks across diverse fabrics. We employ prediction results from DPM to define target shapes, enabling accurate shape matching between the manipulated cloth and desired folding configurations. The system aims to robustly handle variations in fabric characteristics while maintaining folding accuracy. This task is more challenging than usual pushing or relocating tasks due to significant visual occlusions during the folding process, and the increased action complexity. Achieving precise folding to a specified target configuration requires both an accurate estimation and dynamic prediction of the cloth. We tested with square handkerchiefs made of three different materials. Each of these clothes has a different visual appearance and size.

T-shirt folding. This task focuses on folding a T-shirt into the target configuration. T-shirts present unique challenges due to their dual-layer structure and compliant dynamics. We evaluate our approach on four T-shirts of different sizes and physical properties. We set more challenging target states (such as diagonal fold and fold in half) that require higher motion accuracy. Incorrect actions will increase the recovery cost. Some target states also require changing the grasp contact points and performing multiple folds. Figure 4 shows all the test cloths and T-shirts with various materials and sizes used in our real-world experiments.

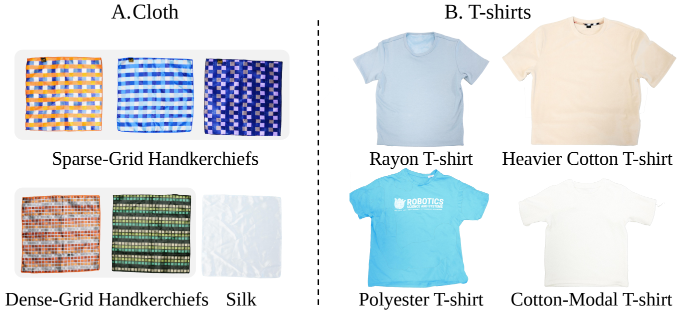


Fig. 4: **Cloth overview.** We evaluate our method on different square cloths and T-shirts with varying colors and materials.

B. Physical Setup

We validate our system on two robotic platforms: (1) a single UFactory xArm-6 robotic arm with Fin Ray Effect-based soft robotic fingers for gripping cloth, and (2) a stationary bimanual dexterous system consisting of two UFactory xArm-7 robotic arms, each equipped with a 6-DoF Ability hand. Both setups use a single RGB-D camera: the Intel RealSense D435 with 640 × 480 resolution for the xArm-6 and the L515 with 1024 × 768 resolution for the dual-arm system. Figure 5 illustrates our hardware setup.

C. Data Collection

We collect training data for learning state estimation and dynamics prediction in a simulation environment built on SAPIEN [38]. The rigid bodies, such as the robot arm, are simulated using the built-in PhysX-based simulator, while the cloth is simulated with the projective dynamics (PD) solver [4]. The two systems are coupled at the time step level by alternating updates: the PD system treats the positions and velocities of PhysX-managed objects as boundary conditions, and PhysX does the same for the PD-managed cloth. In the PD system, the cloth is modeled as a hyper-elastic thin shell. We follow Ly et al. [21] to simulate collision and friction in the PD system.

Our data collection pipeline consists of two main components: state estimation and state transitions. For state estimation, we use up to four calibrated depth cameras [45], which enable realistic stereo depth simulation, positioned at randomized angles to generate paired datasets of fused point clouds and ground-truth mesh states. The cloth is initialized by applying a random force. For state transition data, we simulate cloth trajectories by applying controlled manipulation actions. Each trajectory is generated by selecting a random vertex on the cloth and moving it to a target position (pick-and-place). We collected a comprehensive dataset of 500K examples, with action magnitudes carefully controlled between 0.02 and 0.05 units to ensure physical plausibility. Each example is an 8-step motion sequence, recording the cloth's deformation from the initial state through the entire manipulation process. Detailed data collection procedures are provided in Appendix D-B.

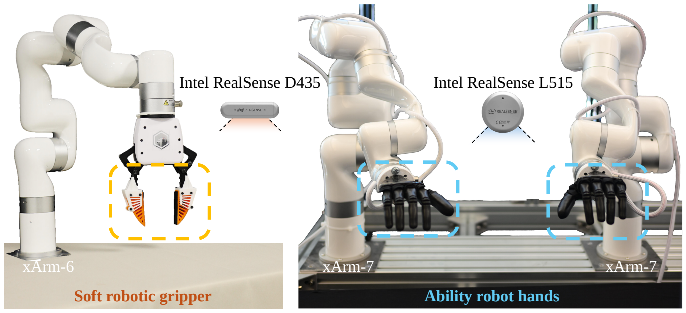


Fig. 5: **Hardware overview.** Our real-world platform includes a UFactory xArm-6 and a bimanual dexterous system consisting of two UFactory xArm-7 robots with Ability hands. Each robot is equipped with one RGB-D camera.

D. Implementation Procedure

Our system integrates OWLV2 [25] and Segment Anything [17] to detect and segment desktop objects from RGB-D input. A single-view partial point cloud of the target object serves as input, which is processed via DPM to infer the state of the cloth. To address the dimensional and positional discrepancies between predicted and observed point clouds, we implement a two-stage alignment process. First, we compute the spatial dimensions of the observed point cloud and apply appropriate scaling transformations to the predicted point cloud. Subsequently, we employ the Iterative Closest Point (ICP) algorithm for fine-grained alignment, ensuring that MPC-generated grasping positions and motion trajectories can be accurately mapped to the physical object. For manipulation, we model both soft robotic grippers and dexterous hands by representing their end effectors as particles that attach to mesh vertices during motion. To evaluate our system, we first collect realistic and challenging target states through teleoperation. We then conduct 10 experimental trials for the same target state, executing a delta action sequence through the MPC with the dynamics model. These actions are transformed into absolute positions in the base frame of the robotic arm, with smooth cartesian trajectories generated using joint online trajectory planning.

Appendix D
Simulation Details

A. Simulation Setup

Detailed physical parameters for cloth simulation is presented in Table III.

B. Data Collection

To collect state estimation data, we set up a comprehensive multi-view system that incorporates up to four calibrated stereo-depth sensors, strategically placed at randomized viewing angles within predefined ranges. Cloth is initialized by given an randomly pick-and-place action. This configuration enables the generation of paired datasets consisting of fused point clouds alongside their corresponding ground-truth mesh states across multiple viewpoints. The system leverages SAPIEN's advanced stereo depth simulation capabilities[45],

Phyical Parameter	Value
collision margin	1e-3
collision weight	5e3
collision sphere radius	8e-3
damping	1e-2
thickness	1e-3
density	1e3
stretch stiffness	1e3
bend stiffness	1e-3
friction	0.5
gravity	-9.81

TABLE III: Dataset statistics.

which significantly reduces the sim-to-real gap by faithfully reproducing point cloud characteristics observed in real-world scenarios. This high-fidelity simulation approach ensures robust and reliable state estimation performance when transferred to physical environments. In our point cloud fusion process, we augment camera extrinsic parameters to simulate real-world calibration errors. Specifically, we introduce rotational variations ranging from $-1.5°$ to $1.5°$ and translational variations from -0.5 to 0.5 cm. To better mimic real-world conditions, we also simulate depth sensor noise and occlusion effects by applying random point dropout with ratios between 0.1 and 0.2, and introducing noise to the fused point cloud.

To collect dynamics data, we employ diverse action sampling strategies to generate a comprehensive dataset of 500K examples. Our sampling approach encompasses two key methodologies designed to capture realistic cloth manipulation scenarios. The first method involves applying directionally-randomized displacements to selected mesh vertices, with particular emphasis on folding-oriented actions where the cloth is manipulated to create various folding patterns. We also simulate picking and relocation actions by applying upward and translational movements to randomly selected vertices. The second methodology focuses on pair-wise vertex manipulation, where vertex pairs are selected based on their spatial distances to simulate actions such as folding one point of the cloth onto another. Each incremental action is precisely controlled, with magnitudes ranging from 0.02 to 0.05 units. To evaluate the model's performance across different time horizons and assess the impact of auto-regressive inference error accumulation, we generate action sequences varying in length from 15 to 35 steps. All resultant mesh deformations throughout these sequences are meticulously recorded to capture the complete dynamics of the cloth's behavior.

Appendix E
Implementation Details

A. Model Details

a) Point Cloud Encoder: We employ a patch-based architecture for point cloud encoding that processes the input through local grouping and feature extraction. The encoder first groups points using a KNN-based strategy, then processes each local patch through a specialized patch encoder, and finally incorporates positional information through learnable embeddings. This design enables effective capture of both local geometric structures and global spatial relationships.

Hyperparameter	Value
Output dimension	1024
Number of groups	256
Group size	64
Group radius	0.15
Position embedding dimension	128
Patch encoder hidden dims	[128, 512]

TABLE IV: Point cloud encoder hyperparameters.

b) Model Architecture: We design a transformer-based architecture for state estimation, which consists of a point cloud encoder, positional embedding module, and a series of transformer blocks. The model takes both point cloud observations and mesh states as input. The point cloud is first processed through a patch-based encoder, while the mesh states are embedded using a patchified positional encoding scheme. These features are then processed through transformer blocks with cross-attention mechanisms to predict the mesh state.

Hyperparameter	Value
Number of attention heads	16
Attention head dimension	88
Number of transformer layers	4
Inner dimension	1408
Dropout	0.0
Cross attention dimension	1024
Point cloud embedding dimension	1024
Number of input frames	2
Number of output frames	1
Activation function	GEGLU
Output MLP dimensions	[512, 256]
Normalization type	AdaLayerNorm
Normalization epsilon	1e-5

TABLE V: Model hyperparameters.

c) Action Embedding: We employ a Fourier feature-based action encoding scheme to effectively represent mesh manipulation actions in a high-dimensional space. The action encoder consists of two main components: (1) a Fourier feature mapping that projects 3D action vectors into a higher-dimensional space using sinusoidal functions, and (2) a multi-layer perceptron that further transforms these features into the desired embedding dimension.

The Fourier feature mapping applies frequency-based encoding separately to each spatial dimension (A_x, A_y, A_z) of the action vectors using both sine and cosine functions, resulting in an intermediate representation of dimension $2 \times 3 \times F$, where F is the number of Fourier frequencies. This representation is then processed through an MLP to produce the final action embeddings.

B. Training Details

We train our model using distributed data parallel training on 4 H100 GPUs. The model is trained with a batch size of 128 per GPU and gradient accumulation steps of 4, resulting in an effective batch size of 2048. We use the AdamW optimizer

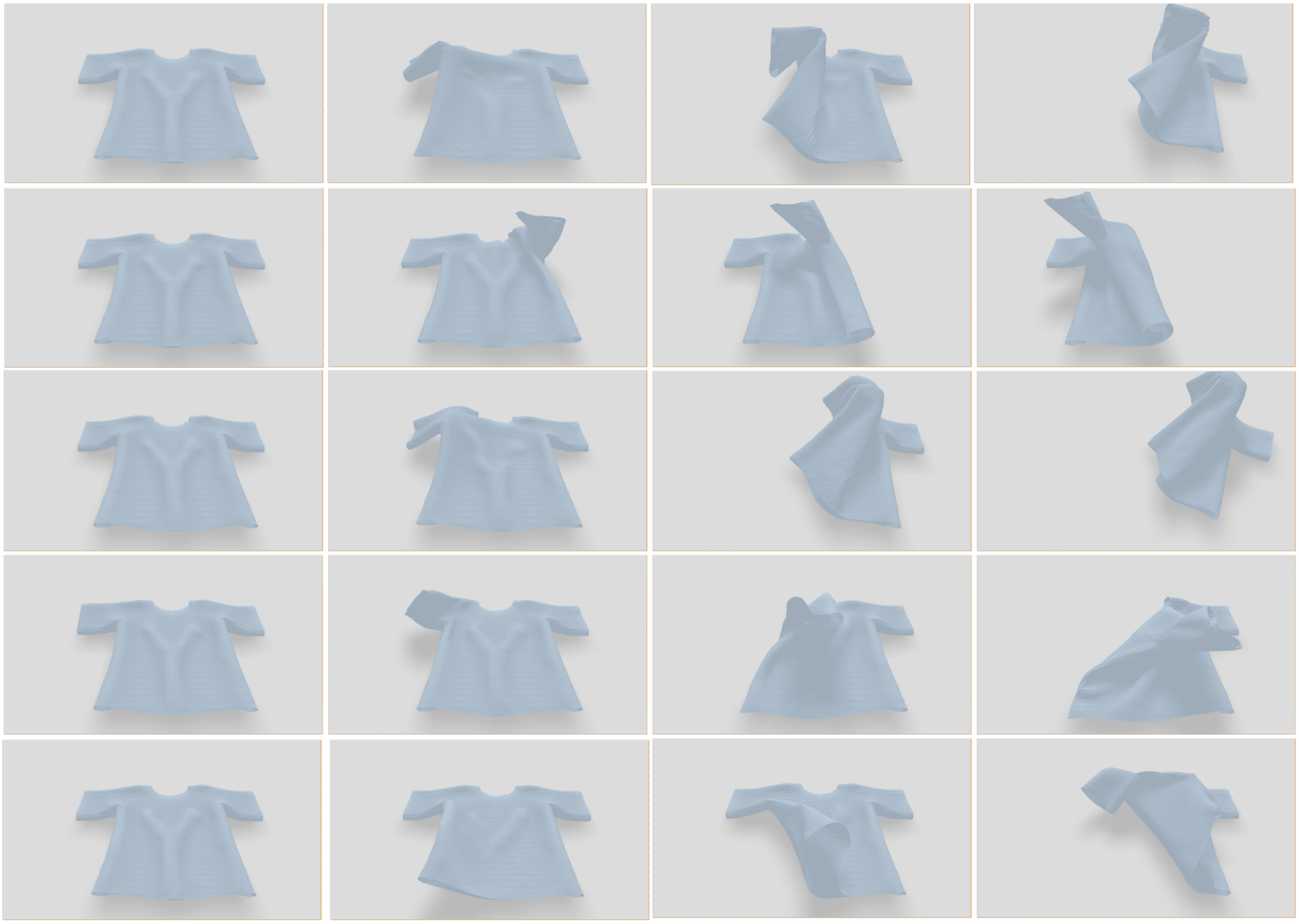


Fig. 6: **Example training data.**

Hyperparameter	Value
Fourier frequencies	8
Fourier feature dimension	48
MLP hidden dimensions	[512, 512]
Output dimension	output_dim
Activation function	SiLU
Position normalization	Center & Scale

TABLE VI: Action encoder hyperparameters.

Hyperparameter	Value
Number of GPUs	4
Batch size per GPU	128
Gradient accumulation steps	4
Effective batch size	1024
Learning rate	1e-5
Learning rate scheduler	Cosine
Warmup steps	1000
Mixed precision	bfloat16
Number of workers	16

TABLE VII: Training hyperparameters.

with a learning rate of 1e-5 and cosine learning rate scheduler with 1000 warmup steps. For numerical stability and training efficiency, we employ mixed-precision training with bfloat16 and enable TF32 on supported hardware.

C. Planning Details

For planning, we employ a hybrid approach combining Model Predictive Control (MPC) and Cross Entropy Method (CEM). Our planner optimizes action sequences by iteratively sampling actions, evaluating their outcomes using the learned dynamics model, and updating the sampling distribution based on the costs. To enhance planning efficiency, we introduce two key strategies: (1) an informed action sampling mechanism and (2) a grasp point selection method. For action sampling, we initialize the sampling distribution using a prior direction informed by the target state. Specifically, we identify the K vertices with highest mean squared error (MSE) between the current and target states, and compute a weighted average direction based on their distances to the grasp point:

$$d_{main} = \sum_{i=1}^{K} w_i(s_t^i - s_c^i), \quad w_i = \frac{1}{\|p_g - p_i\| + \epsilon} \qquad (2)$$

where s_t^i and s_c^i are target and current states of vertex i, p_g is the grasp point position, and p_i is the position of vertex i.

This informed direction guides the initial sampling distribution for more efficient exploration.

For grasp point selection, we employ a temperature-controlled softmax strategy based on vertex displacements. Given the current state S_c and target state S_t, we compute a probability distribution over all vertices:

$$p(i) = \frac{\exp(\|s_t^i - s_c^i\|_2/\tau)}{\sum_j \exp(\|s_t^j - s_c^j\|_2/\tau)} \tag{3}$$

where s_t^i and s_c^i represent the position of vertex i in target and current states respectively, and τ is a temperature parameter that controls the concentration of the probability distribution. A lower temperature leads to more deterministic selection focusing on maximum displacement vertices, while a higher temperature enables more exploratory behavior. The grasp point is then sampled from this distribution:

$$g \sim p(i) \tag{4}$$

This probabilistic selection mechanism provides several advantages over deterministic maximum displacement selection: (1) it allows for exploration of different grasp points, (2) it can adapt to different manipulation scenarios by adjusting the temperature parameter, and (3) it provides a smoother transition between different grasp point candidates. The planning algorithm is outlined in Algorithm 1. Hyperparameters for model-based planning is listed in Table IX.

D. Baseline Implementation

We introduce details of baseline implementation.

a) GNNs: We adopt the implementation from [44]. We construct a comprehensive graph representation for modeling cloth dynamics, incorporating object particles, end-effector interactions, and material properties. The graph structure consists of four main components: (1) state and action representations, (2) particle attributes and instance information, (3) relation matrices for particle interactions, and (4) material-specific physics parameters. The state representation captures both spatial positions and temporal dynamics through a history buffer of n_{his} frames and future predictions of n_{future} frames. Each state vector contains the 3D positions (x, y, z) of both cloth particles and the end-effector. We maintain a fixed-size particle set through Farthest Point Sampling (FPS) with an adaptive radius range of $[0.05, 0.1]$. We show detailed parameters for graph construction below.

Hyperparameter	Value
Maximum particles (N_{obj})	100
Maximum relations (N_R)	1000
History frames (n_{his})	3
Future frames (n_{future})	5
State dimension	3
Attribute dimension	2
FPS radius range	[0.05, 0.1]
Adjacency radius range	[0.74, 0.76]
Topk neighbors	5

TABLE VIII: GNN model hyperparameters.

Parameter	Value
Number of iterations	5
Samples per iteration	16
Sequence length	5
Action dimension	3
Initial std deviation	0.1
Temperature	1.0

TABLE IX: Planning hyperparameters.

Appendix F
Additional Results

a) Dynamics Prediction: Quantitative results on perception input scenario is presented in Table X.We present more qualitative results on forward dynamics prediction in Figure 8 and Figure 9.

Type	**Method**	↓ **MSE** (10^{-3})	↓ **CD** (10^{-2})	↓ **EMD** (10^{-2})
T-shirt	GNN	6.36 ± 1.30	8.88 ± 1.12	8.29 ± 1.94
	Transformer	4.18 ± 0.73	4.26 ± 0.51	7.93 ± 0.70
	DDM	**0.55** ± 0.27	**1.49** ± 0.13	**3.22** ± 0.47
Cloth	GNN	2.17 ± 1.44	5.02 ± 0.90	7.31 ± 4.65
	Transformer	1.30 ± 0.65	2.27 ± 0.46	7.06 ± 2.08
	DDM	**0.66** ± 0.45	**2.12** ± 0.54	**5.51** ± 1.03

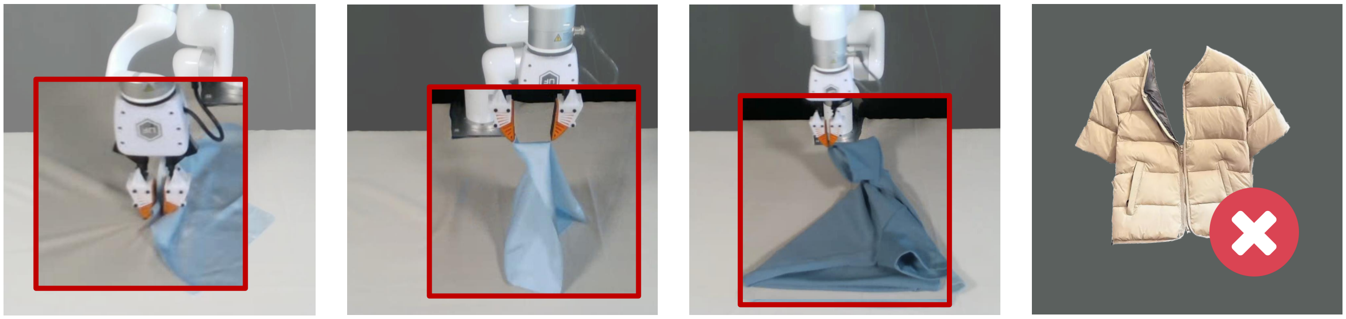
TABLE X: **Quantitative results of last frame dynamics prediction.** Long-horizon dynamics prediction results with DPM perception noisy input. Errors represent a 95% confidence interval.

b) Planning: We present more qualitative results in Figure 11 in the simulation environment on planning.

c) Failure Cases Analysis: We also present failure cases results in Figure 7. We observed four kinds of common cases that may cause failure of an action.

Algorithm 1 MPC Planning Algorithm

```
Require: Initial state $s_i$, target state $s_t$, dynamics model $f_\theta$, number of iterations $N$
Require: Number of samples $K$, sequence length $L$, action bounds $[a_{min}, a_{max}]$
Initialize $\mu \leftarrow \mathbf{0}$, $\sigma \leftarrow 0.1$
$a_{best} \leftarrow$ None, $c_{best} \leftarrow \infty$
for $i = 1$ to $N$ do
    $A_{mppi} \leftarrow \text{SampleGaussian}(K/2, L, \mu, \sigma, [a_{min}, a_{max}])$
    $A_{uniform} \leftarrow \text{SampleUniform}(K/2, L, [a_{min}, a_{max}])$
    $A \leftarrow \text{Concatenate}(A_{mppi}, A_{uniform})$
    $S_{pred} \leftarrow f_\theta(S, A)$                                   ▷ Predict trajectories
    $C \leftarrow \text{ComputeCost}(S_{pred}, A, T)$                      ▷ Evaluate costs
    if $\min(C) < c_{best}$ then
        $c_{best} \leftarrow \min(C)$
        $a_{best} \leftarrow A[\arg\min(C)]$
    end if
    $\mu, \sigma \leftarrow \text{UpdateDistribution}(A, C, \tau)$       ▷ Update using weighted averaging
    $\sigma \leftarrow \sigma \cdot (1 - i/N)$                               ▷ Anneal exploration
end for
return $a_{best}$
```



(a) Incorrect grasp point

(b) Gripper-induced cloth damage scenarios

(c) Wrong dynamics prediction

(d) Complex garments with specialized features (zippers, down-filled materials)

Fig. 7: **Analysis of failure modes.** Systematic categorization and visualization of common failure cases, highlighting key limitations: (a) grasp point selection errors, (b) gripper-induced cloth damage, (c) dynamics prediction inaccuracies, and (d) complex garment features (e.g., zippers, down-filled materials).

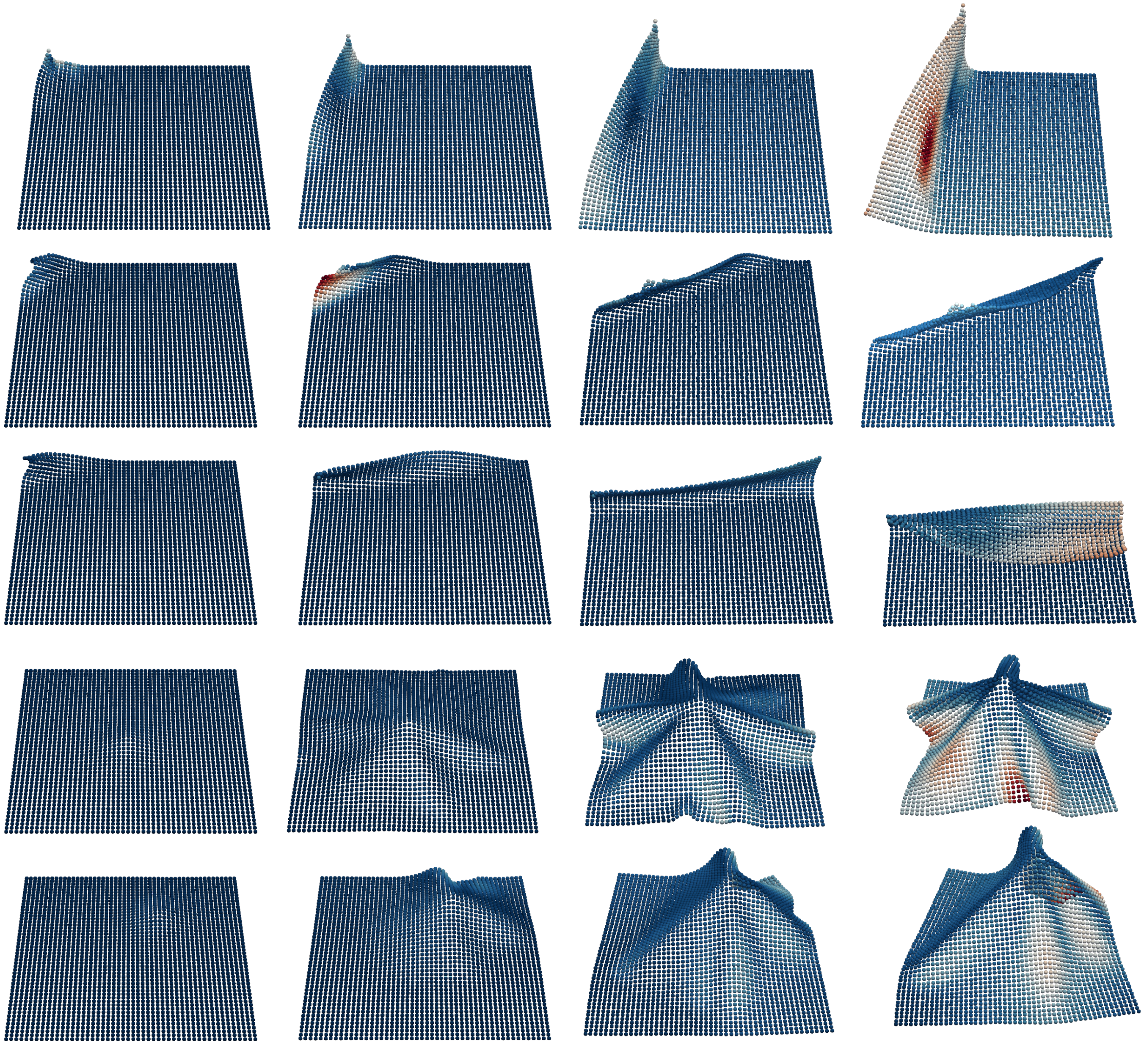


Fig. 8: **Qualitative results on cloth dynamics prediction using DDM.**

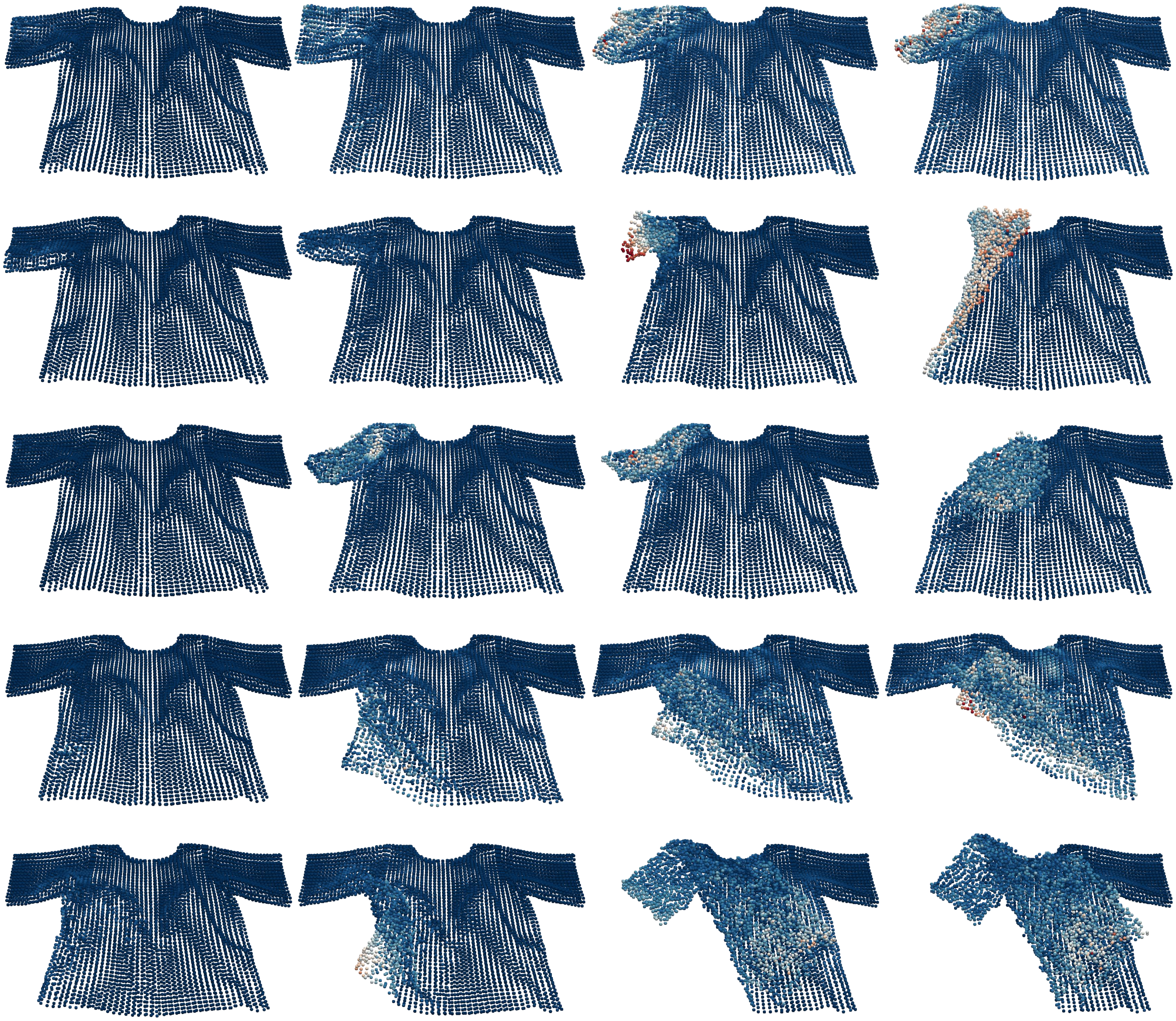
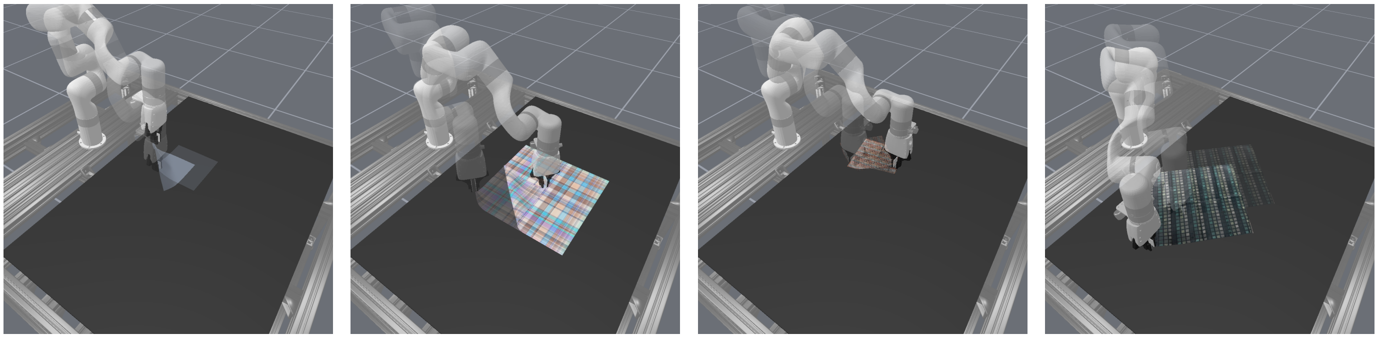


Fig. 9: **Qualitative results on t-shirt dynamics prediction using DDM.**



(a) Lift cloth (b) Fold cloth (c) Rotate cloth (d) Pull cloth

Fig. 10: **Simulated cloth manipulation environments.** Visualization of diverse manipulation scenarios in simulation: (a)-(d) demonstrate different cloth-robot interactions with varied object configurations and manipulation tasks.

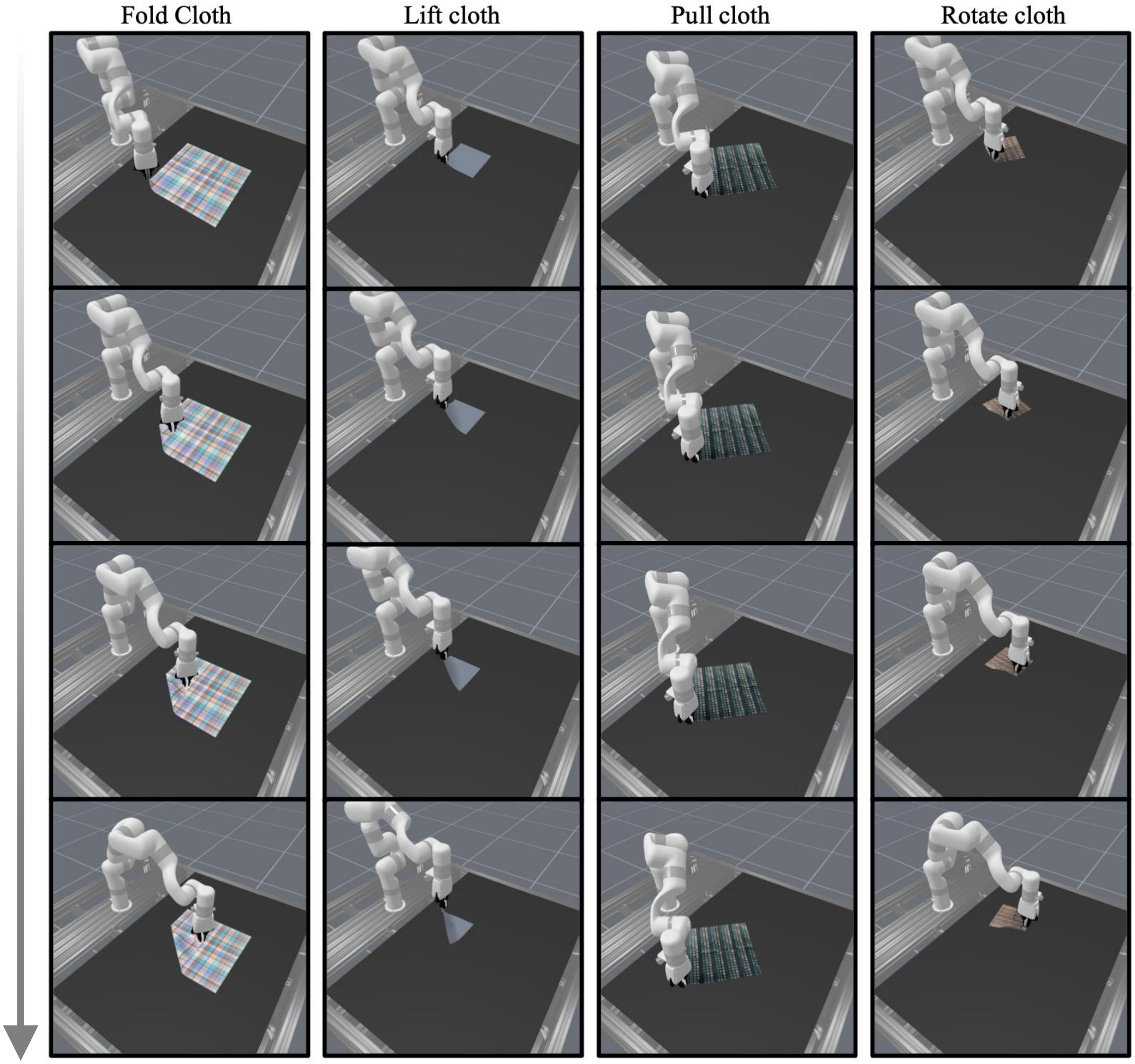


Fig. 11: **Model predictive control evaluation in simulation.** Demonstration of our diffusion-based dynamics model integrated with MPC across diverse manipulation tasks using xArm7, validated on various cloth types.