Diffusion Dynamics Models with Generative State Estimation for Cloth Manipulation

Tongxuan Tian[1*] **Haoyang Li**[1,2*]
Bo Ai[1] **Xiaodi Yuan**[1] **Zhiao Huang**[1,2] **Hao Su**[1,2]
[*]Equal contribution
[1]University of California San Diego, USA [2]Hillbot Inc, USA
https://uniclothdiff.github.io/

Abstract: Cloth manipulation is challenging due to its highly complex dynamics, near-infinite degrees of freedom, and frequent self-occlusions, which complicate both state estimation and dynamics modeling. Inspired by recent advances in generative models, we hypothesize that these expressive models can effectively capture intricate cloth configurations and deformation patterns from data. Therefore, we propose a diffusion-based generative approach for both perception and dynamics modeling. Specifically, we formulate state estimation as reconstructing full cloth states from partial observations and dynamics modeling as predicting future states given the current state and robot actions. Leveraging a transformer-based diffusion model, our method achieves accurate state reconstruction and reduces long-horizon dynamics prediction errors by an order of magnitude compared to prior approaches. We integrate our dynamics models with model predictive control and show that our framework enables effective cloth folding on real robotic systems, demonstrating the potential of generative models for deformable object manipulation under partial observability and complex dynamics.

Keywords: Deformable Object Manipulation, Dynamics Model Learning, State Estimation, Generative Models, Cross-Embodiment Generalization

1 Introduction

Textile deformable objects, such as clothing, are ubiquitous in daily life. Yet, manipulating these objects is a long-standing challenge in robotics [1, 2], due to their complex geometric structures and dynamics. Effective cloth manipulation requires accurately estimating the state of cloth 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 [3–5]. 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) [4, 6–8]. GNNs offer advantages in data-scarce domains through spatial equivariance and locality, but they scale inefficiently with the number of graph nodes [9]. 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

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

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 [10, 11], science [12], and robotics [13]. We hypothesize that diffusion models with scalable architecture (e.g., Transformer [14]) 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.

2 Related Work

Deformable Object Manipulation. Manipulating deformable objects such as garments remains a long-standing challenge in robotics, due to their high-dimensional state space and complex, nonlinear dynamics. Model-free approaches, including reinforcement learning (RL) [15, 16] and imitation learning (IL) [13, 17–20], 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 dynamics reasoning. Model-based approaches require accurate state estimation [21–26], which is highly challenging with partial observations. Further, learning dynamics models demands extensive training data to cover large state and action spaces. Thus, we propose to learn expressive generative models for state estimation and dynamics modeling using large-scale simulation data.

Learning-Based Dynamics Models. Learning-based dynamics models [27] predict state transitions from interaction data, where the choice of state representation is crucial. Pixel-based models view the problem as action-conditioned video prediction [28, 29], but they are often sample-inefficient, vulnerable to occlusions, and lack physical realism for contact-rich scenarios [30]. Structured representations, such as particles or meshes, provide stronger physical priors and are typically coupled with graph neural networks (GNNs) that perform inference via message passing [4, 6, 8, 22–24]. While sample-efficient, GNNs often struggle with scalability and long-range interactions. In contrast, we find diffusion models offer greater expressiveness and scalability, enabling accurate dynamics prediction from large-scale data and improving modeling of deformable object behavior.

Diffusion Models. Diffusion models [31], a class of generative models with expressive capability of capturing complex, high-dimensional data distributions precisely, have emerged as a powerful paradigm and been applied across diverse domains, including generation of images [32, 33], videos [10, 34], 3D shapes [35], as well as robot policy learning [13, 19] and world modeling [36, 37]. In this work, we adapt diffusion models for deformable objects manipulation, leveraging their superior data distribution modeling capability for (i) estimating full cloth configurations from partial point cloud observations, and (ii) modeling state transitions to enable accurate future prediction and model-based planning for cloth manipulation.

3 Method

3.1 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 $\mathcal{S}$,

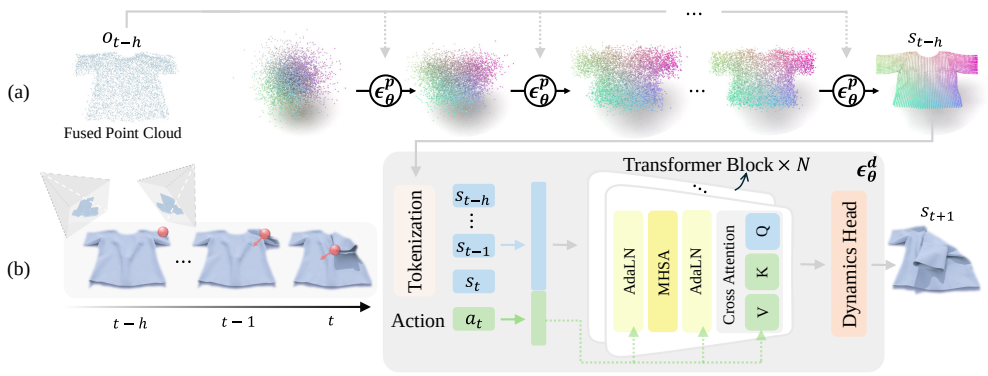


Figure 1: **Overview. (a) Perception:** Our Diffusion Perception Model (DPM) reconstructs the full cloth state from a partial point cloud. Using a denoising process parameterized by ϵ_θ^p, DPM refines the cloth state over K denoising steps, starting from random noise. **(b) Dynamics Prediction:** Our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

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 $s_t \in \mathcal{S}$ 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 integrated with model-predictive control to optimize action sequences for achieving the target state s_g:

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

3.2 State Estimation

We first address the problem of inferring cloth configurations from partial observations. Despite using four multi-view RGB-D cameras, severe self-occlusions make accurate state estimation infeasible. Inspired by the human ability to infer hidden object states from partial views, we propose using diffusion models to generate full cloth configurations from limited observations.

Conditional Diffusion Process. We formulate cloth state estimation as a conditional denoising diffusion process, using the object point cloud as the conditioning input. Conditioning on point clouds helps minimize the sim-to-real gap due to their nature as a mid-level visual representation and maintains geometric invariance [38, 39].

Specifically, we model the conditional distribution $p(s|s_c, \mathbf{e}_{\text{pc}})$ using standard denoising diffusion probabilistic model (DDPM) [31], 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 [40] 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 denoises from s_t to s_{t-1} and finally constructs s_0.

Model Architecture. We adopt vanilla Vision Transformer (ViT) architecture [41] as our backbone, which has been shown to be highly scalable in image and video generation [10, 33]. 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 in canonical space. 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 [40] 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) [42] to better incorporate conditional information, which modulates the normalization parameters based on the point cloud condition embedding for effective feature modulation. 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 C.2.

Training. We train the denoising model $\epsilon_\theta^p(s^{(k)}|s_c, \mathbf{e}_{\text{pc}})$ to minimize the loss:

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

where $\epsilon \sim \mathcal{N}(0, I)$ and $\beta^{(k)} \in \mathbb{R}$ are K different noise levels for $k \in [1, K]$. Details of the training process are available in Appendix C.3.

3.3 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 [6, 30], 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

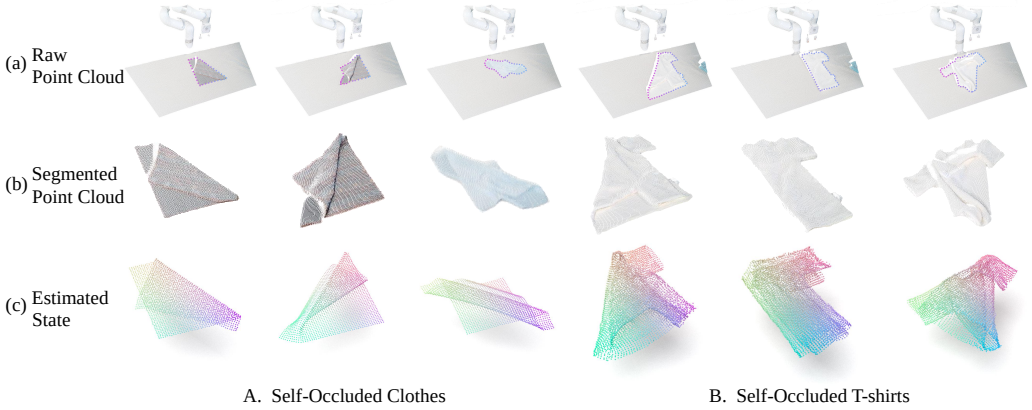


Figure 2: **Qualitative results on state estimation.** Row (a) shows raw point cloud of the work space. Row (b) shows the segmented point clouds of **real-world** clothes, all of which are highly crumpled. Row (c) shows the predicted cloth states.

feature-based embedding following NeRF [43] to represent continuous spatial information, with detailed formulation in Appendix C.2.

3.4 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 $\{a_t\}_{t=0}^{T-1}$ over horizon T 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 mean squared error (MSE) and chamfer distance (CD), and ℓ enforces action smoothness. We use Model Predictive Path Integral (MPPI) [44] for sampling-based optimization. Actions are defined as relative end-effector displacements applied to a selected cloth grasp point. To improve planning efficiency, we introduce an informed action sampling strategy and a probabilistic grasp point selection mechanism. Specifically, the grasp point is selected using a temperature-controlled softmax distribution based on vertex displacements between the current and target states, while action sampling is guided by a weighted direction computed from high-displacement vertices. After each action, the robot updates its state estimate using DPM before replanning. Refer to Appendix C.4 for details on the planning algorithm and hyperparameters.

4 Experiments

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 study these questions by evaluating state estimation accuracy (Section 4.2), assessing dynamics modeling performance (Section 4.3), and real-world deployment across two system setups(Section 4.4).

4.1 Experiments Setup

We evaluate our method in both simulation and real-world environments. Specifically, we use SAPIEN [45] as the simulation platform for data collection and training, and demonstrate effective sim-to-real transfer in the real-world setting. Additional details of the pipeline implementation and experimental setup are provided in Appendix B.

4.2 State Estimation Results

Baselines. We compare our perception module against four baselines: **GarmentNets** [3] which formulates cloth pose estimation problem as a shape completion task in the canonical space; **MEDOR** [4] which improves GarmentNets by introducing test-time fine-tuning for mesh refinement; **TRTM** [5] which employs a template-based approach for explicit mesh reconstruction; and **Transformer**, an ablated version of our model that retains the original architecture but without diffusion training. These baselines systematically comprise both optimization-based and non-optimization-based prior works on cloth pose estimation, along with ablation studies for our model..

Results. We evaluate our method and baselines in both simulation and the real world using MSE, CD, and Earth Mover's Distance (EMD). The results are presented in Table 1. In the T-shirt setting, TRTM and Transformer greatly outperform GarmentNets and MEDOR, 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 demonstrates better performance compared to Transformer. Our approach achieves further performance gains over both TRTM and Transformer, highlighting the significant contributions of diffusion models to the task. We provide qualitative results in Figure 2.

Category	Method	Simulation			Real World	
		↓ MSE (10^{-1})	↓ CD (10^{-1})	↓ EMD (10^{-1})	↓ CD (10^{-1})	↓ EMD (10^{-1})
Cloth	TRTM [5]	5.07 ± 0.22	2.67 ± 0.61	1.65 ± 0.71	1.85 ± 0.15	0.86 ± 0.23
	Transformer	5.44 ± 0.41	2.17 ± 0.19	1.61 ± 0.45	1.72 ± 0.22	0.78 ± 0.33
	DPM	**2.32** ± 0.21	**1.95** ± 0.25	**1.48** ± 0.47	**1.13** ± 0.25	**0.54** ± 0.49
T-shirt	GarmentNets [3]	18.6 ± 1.35	6.23 ± 0.79	2.79 ± 0.64	7.18 ± 0.51	2.86 ± 0.46
	MEDOR [4]	21.0 ± 1.54	6.87 ± 0.95	2.24 ± 0.29	5.01 ± 0.48	2.49 ± 0.32
	TRTM [5]	6.30 ± 0.45	5.15 ± 0.96	2.15 ± 0.29	3.18 ± 0.44	1.99 ± 0.29
	Transformer	9.12 ± 0.57	5.56 ± 0.63	1.99 ± 0.62	2.34 ± 0.37	1.91 ± 0.33
	DPM	**2.76** ± 0.19	**3.22** ± 0.41	**1.95** ± 0.56	**2.17** ± 0.28	**1.88** ± 0.61

Table 1: **Quantitative results on state estimation.** We report estimation errors in both simulated and real-world scenarios, with 95% confidence intervals. Lower values indicate better performance.

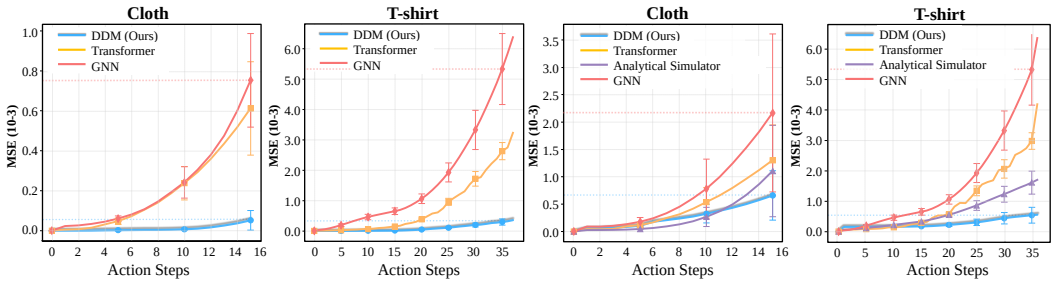
4.3 Dynamics Prediction Results

Baselines. We evaluated our diffusion dynamics models against three baseline approaches: a **GNN**-based method [6] which is the most widely adopted approach for modeling dynamics; an **Analytical Simulator** specifically for configurations using the DPM's output; and an ablated version of our model with dynamics module trained directly with MSE loss supervision termed **Transformer**. For each baseline model, we analyze the MSE across different timesteps on clothes and T-shirts.

Results. Our evaluation compares the proposed approach against three baselines using MSE across two experimental scenarios: (1) using ground truth states from the simulator and (2) using perception states estimated by DPM. The second scenario, which includes a direct comparison with Analytical Simulator, demonstrates the robustness of our method to noisy states that typically degrade the performance of the analytical simulator.

Error analysis over time in Figure 3 shows that DDM consistently outperforms all baselines. GNN exhibits the weakest performance, particularly for complex objects like T-shirts. Transformer improves over GNN by leveraging transformer architectures, but still suffers from error accumulation. In contrast, DDM achieves the lowest MSE across all timesteps with minimal temporal error accumulation, benefiting from the diffusion model's expressive distribution modeling. Qualitative results in Figure 4 further highlight the physical plausibility of DDM's predictions.

When using estimated states with perception noise, we introduce Analytical Simulator as an additional baseline. Although Analytical Simulator initially achieves low error on cloth objects, it is highly sensitive to inconsistent inputs, leading to rapid error accumulation and worse long-horizon performance than DDM. This degradation is even more pronounced for T-shirt objects due to their



(a) Oracle Perception as Input (b) DPM Perception as Input

Figure 3: **Long-horizon dynamics prediction error over time.** MSE in dynamics prediction over time under two scenarios: (a) using oracle simulation states, and (b) using DPM perception estimates, evaluated on clothes and T-shirts. Error bars represent 95% confidence intervals.

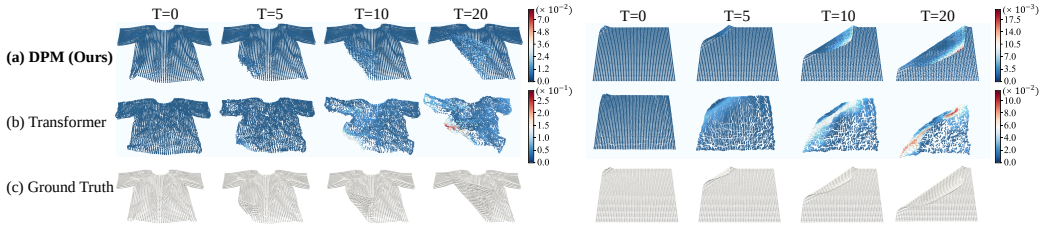


Figure 4: **Qualitative results on dynamics prediction.** Visualization of predicted clothes configurations with each vertex color-coded by its point-wise L2 error from ground truth.

complex topology. Under these realistic conditions, both GNN and Transformer exhibit even larger gaps, demonstrating that DDM provides the most robust and accurate dynamics predictions for planning with noisy observations. We provide additional quantitative results in Appendix A.1.

4.4 Real World Planning Results

Comparative Analysis. We demonstrate the seamless integration of DPM and DDM within an MPC framework for complex cloth folding tasks. Our approach is benchmarked against GNN as a dynamics module. We categorize our tasks into folding and unfolding scenarios, all of which involve long-horizon challenges and require multi-step prediction, with three distinct occlusion types: self-occlusion, external occlusion by other objects (e.g., a robotic arm), and combined occlusion, which poses challenges for accurate perception in cloth manipulation. We report the success rates (SR) of the quantitative results in Table 2. A trial is deemed successful if the geometric metric (EMD) is below a certain threshold (Appendix A.2). Our method consistently outperforms GNN across all occlusion scenarios. In simpler tasks, such as folding clothes, our model achieves an improvement of approximately 30% points in SR. When manipulating more challenging objects, such as a dual-level topology T-shirt where GNN struggles to accurately model dynamics, our approach achieves up to a 50% increase in SR. Qualitative results are shown in Figure 5, illustrating challenging initial and target configurations with severe self-occlusion, representing substantially more difficult setups than those considered in many prior works [4, 6, 17, 25]. By leveraging an embodiment-agnostic action space design, our approach enables effective cross-embodiment transfer. We demonstrate a successful transfer of the embodiment from a parallel gripper to a dexterous hand in Figure 6.

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 2: **Quantitative results of real-world manipulation.** We repeat each scenario for 10 trials, with randomized initial and target states.

Method (Dynamics + Perception)	Cloth SR↑	T-shirt SR↑
DDM + DPM (Ours)	**9/10**	**8/10**
Transformer + DPM	3/10	5/10
GNN + DPM	6/10	1/10
DDM + Transformer	5/10	3/10
Transformer + Transformer	2/10	1/10
GNN + Transformer	5/10	0/10

Table 3: Success rates of system variants with different combinations of dynamics and perception modules.

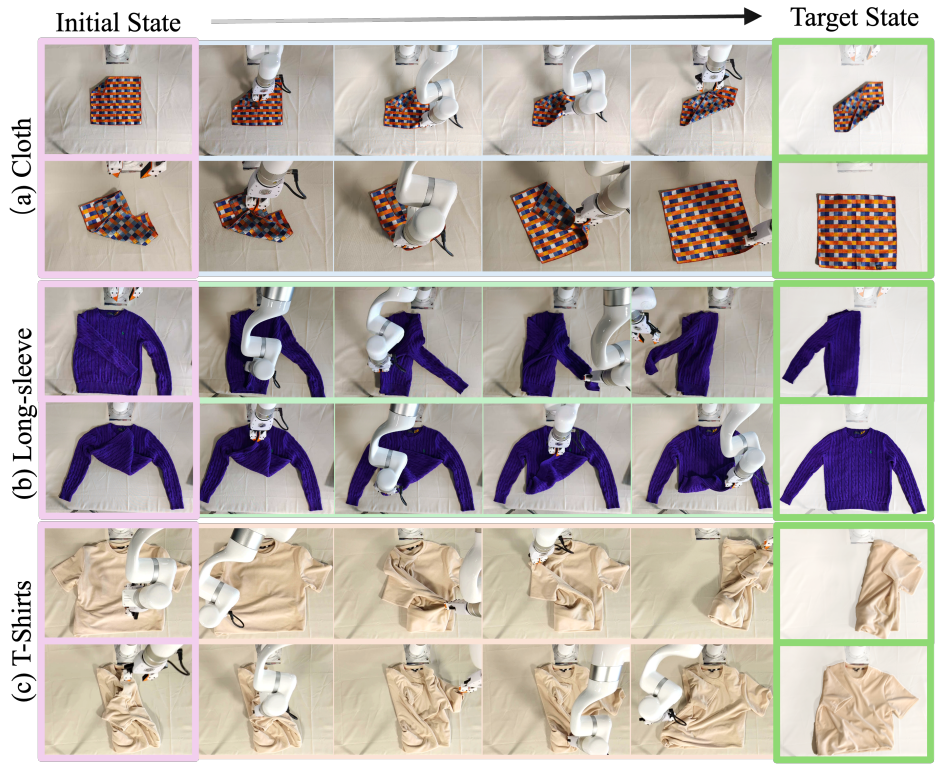


Figure 5: **Qualitative results on real-world cloth manipulation.** We evaluate our system on three types of garment: square clothes, long-sleeve shirts and T-shirts. For each garment, the first row corresponds to folding and the second row to unfolding.

Ablation. The ablation study results presented in Table 3 demonstrate the critical contributions of both DPM and DDM to the system's overall performance, with their combination yielding significantly higher success rates than either component alone. For objects with simpler topology, such as cloth, accurate perception is most critical since the dynamics are relatively straightforward and easy to model. However, for objects with more complex topologies, having an accurate dynamics model becomes equally important for effective planning.

5 Conclusion

We introduce UniClothDiff, a unified framework that tackles key challenges in state estimation and dynamics prediction in cloth manipulation with Transformer-based diffusion models. Our approach reconstructs full cloth configurations from partial RGB-D observations and predicts long-horizon dynamics with significantly lower error than prior GNN-based methods. Integrated with model-based control, it enables cloth manipulation in various scenarios, significantly outperforming existing approaches. Through extensive experiments, we demonstrate the potential of generative models for deformable object manipulation, paving the way for more robust and versatile robotic systems.

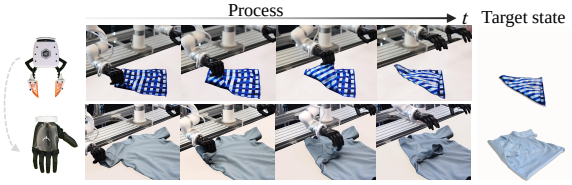


Figure 6: **Cross-embodiment generalization results.**

6 Limitations

One limitation of our method is the substantial computational cost associated with training large transformer-based diffusion models. Second, our experiments primarily focus on cloth manipulation; extending the framework to contact-rich rigid body tasks is a promising future direction. Given

the generality of the model design, we expect feasible adaptation with suitable training data. Additionally, our model currently lacks explicit uncertainty estimates. Incorporating uncertainty quantification into perception and dynamics models, and combining them with control methods with theoretical guarantees, could improve robustness in safety-critical scenarios.

7 Acknowledgments

This research was funded by Hillbot Inc. Hao Su is the CTO for Hillbot and receives income. The terms of this arrangement have been reviewed and approved by the University of California, San Diego in accordance with its conflict of interest policies.We thank Xuhui Kang for his help with video production and editing and Zhan Ling for his helpful discussions.

References

[1] A. Longhini, Y. Wang, I. Garcia-Camacho, D. Blanco-Mulero, M. Moletta, M. Welle, G. Alenyà, H. Yin, Z. Erickson, D. Held, J. Borràs, and D. Kragic. Unfolding the literature: A review of robotic cloth manipulation. *Annual Review of Control, Robotics, and Autonomous Systems*, 2024-12-02. ISSN 2573-5144.

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

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

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

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

[6] K. Zhang, B. Li, K. Hauser, and Y. Li. Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

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

[8] Z. He, B. Ai, Y. Liu, W. Wan, H. I. Christensen, and H. Su. Learning dexterous deformable object manipulation through cross-embodiment dynamics learning. In *RSS Workshop on Dexterous Manipulation: Learning and Control with Diverse Data*, 2025.

[9] Y. Rong, Y. Bian, T. Xu, W. Xie, Y. Wei, W. Huang, and J. Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems*, 33:12559–12571, 2020.

[10] X. Ma, Y. Wang, G. Jia, X. Chen, Z. Liu, Y.-F. Li, C. Chen, and Y. Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

[11] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. 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.

[12] S. Rühling Cachay, B. Zhao, H. Joren, and R. Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. *Advances in neural information processing systems*, 36: 45259–45287, 2023.

[13] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. 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.

[15] J. Matas, S. James, and A. J. Davison. Sim-to-real reinforcement learning for deformable object manipulation. In A. Billard, A. Dragan, J. Peters, and J. 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.

[16] R. Jangir, G. Alenyà, and C. 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] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. 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.

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

[19] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[20] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. Fast: Efficient action tokenization for vision-language-action models. 2025. URL https://doi.org/10.48550/arXiv.2501.09747.

[21] S. Chen, X. Ma, Y. Lu, and D. Hsu. Ab initio particle-based object manipulation. In D. A. Shell, M. Toussaint, and M. A. 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.

[22] 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.

[23] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. In J. Tan, M. Toussaint, and K. 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.

[24] B. Ai, S. Tian, H. Shi, Y. Wang, C. Tan, Y. Li, and J. Wu. Robopack: Learning tactile-informed dynamics models for dense packing. *Robotics: Science and Systems (RSS)*, 2024. URL https://arxiv.org/abs/2407.01418.

[25] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024.

[26] A. Longhini, M. C. Welle, Z. Erickson, and D. Kragic. Adafold: Adapting folding trajectories of cloths via feedback-loop manipulation. *IEEE Robotics and Automation Letters*, 2024.

[27] B. Ai, S. Tian, H. Shi, Y. Wang, T. Pfaff, C. Tan, H. I. Christensen, H. Su, J. Wu, and Y. Li. A review of learning-based dynamics models for robotic manipulation. *Science Robotics*, 2025.

[28] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg. Visuospatial foresight for physical sequential fabric manipulation. *Auton. Robots*, 46(1):175–199, Jan. 2022. ISSN 0929-5593. doi:10.1007/s10514-021-10001-0. URL https://doi.org/10.1007/s10514-021-10001-0.

[29] W. Yan, A. Vangipuram, P. Abbeel, and L. Pinto. Learning predictive representations for deformable objects using contrastive estimation. In J. Kober, F. Ramos, and C. 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.

[30] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

[31] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[32] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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.

[33] W. Peebles and S. Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[34] Y. Wang, X. Chen, X. Ma, S. Zhou, Z. Huang, Y. Wang, C. Yang, Y. He, J. Yu, P. Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *IJCV*, 2024.

[35] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022.

[36] E. Alonso, A. Jelley, V. Micheli, A. Kanervisto, A. Storkey, T. Pearce, and F. Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Thirty-eighth Conference on Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.12399.

[37] Z. Ding, A. Zhang, Y. Tian, and Q. Zheng. Diffusion world model: Future modeling beyond step-by-step rollout for offline reinforcement learning. *arXiv preprint arXiv:2402.03570*, 2024.

[38] B. Chen, A. Sax, F. Lewis, I. Armeni, S. Savarese, A. Zamir, J. Malik, and L. Pinto. Robust policies via mid-level visual representations: An experimental study in manipulation and navigation. In J. Kober, F. Ramos, and C. 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.

[39] B. Ai, Z. Wu, and D. Hsu. Invariance is key to generalization: Examining the role of representation in sim-to-real transfer for visual navigation. In *International Symposium on Experimental Robotics*, pages 69–80. Springer, 2023.

[40] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

[41] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[42] J. Xu, X. Sun, Z. Zhang, G. Zhao, and J. Lin. Understanding and improving layer normalization. *Advances in neural information processing systems*, 32, 2019.

[43] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[44] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. 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.

[45] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[46] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. 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.

[47] M. Ly, J. Jouve, L. Boissieux, and F. Bertails-Descoubes. Projective dynamics with dry frictional contact. *ACM Trans. Graph.*, 39(4), Aug. 2020. ISSN 0730-0301. doi:10.1145/3386569.3392396. URL https://doi.org/10.1145/3386569.3392396.

[48] X. Zhang, R. Chen, A. Li, F. Xiang, Y. Qin, J. Gu, Z. Ling, M. Liu, P. Zeng, S. Han, Z. Huang, T. Mu, J. Xu, and H. 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.

[49] H. Bertiche, M. Madadi, and S. Escalera. Cloth3d: clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020.

[50] M. Minderer, A. Gritsenko, and N. Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36, 2024.

[51] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Appendices

A Additional Results

A.1 Dynamics Prediction

Quantitative Results We provide additional quantitative results of forward dynamics prediction for both ground truth state and perception noisy input in Table 4 and Table 5. Our dynamics model consistently outperforms all baselines across both evaluation scenarios. When using ground-truth states from the simulator as input, DDM achieves approximately 10× lower error than the best-performing baseline, highlighting the superiority of diffusion models in capturing long-horizon dynamics. When using estimated states from DPM, which introduces additional noise, DDM still achieves 2× lower error than all baselines, demonstrating that the diffusion-based training paradigm significantly enhances noise tolerance through its expressive data distribution modeling capacity.

Type	Method	↓ MSE (10^{-3})	↓ CD (10^{-2})	↓ EMD (10^{-2})
Cloth	GNN	0.75 ± 0.23	3.89 ± 0.80	6.13 ± 2.96
	Transformer	0.61 ± 0.23	1.85 ± 0.33	5.47 ± 1.50
	DDM	**0.05** ± 0.04	**0.63** ± 0.28	**3.47** ± 0.60
T-shirt	GNN	6.36 ± 1.45	8.57 ± 1.06	7.89 ± 1.79
	Transformer	3.22 ± 0.30	2.80 ± 0.23	7.57 ± 0.52
	DDM	**0.35** ± 0.13	**0.73** ± 0.07	**2.84** ± 0.47

Table 4: **Quantitative results on dynamics prediction with ground truth input.** Errors represent a 95% confidence interval.

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 5: **Quantitative results on dynamics prediction with perception input.** Errors represent a 95% confidence interval.

Qualitative Results We present additional qualitative results for dynamics prediction in Figure 7 and Figure 8. Each row represents a predicted dynamics sequence. The results demonstrate the physical plausibility of the generated outputs.

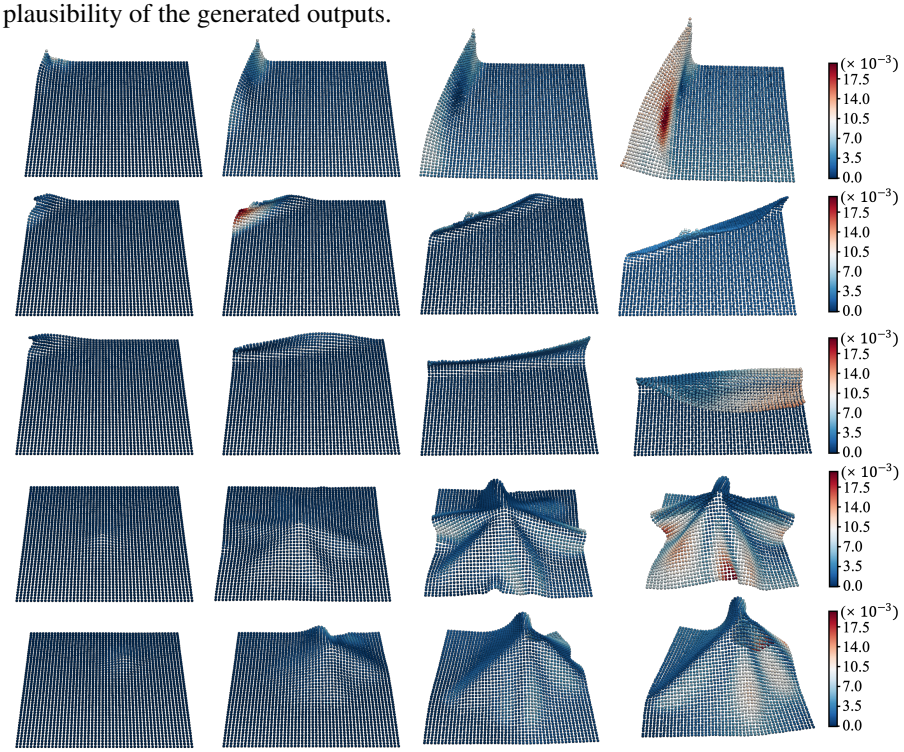


Figure 7: **Qualitative results on cloth dynamics prediction using DDM.**

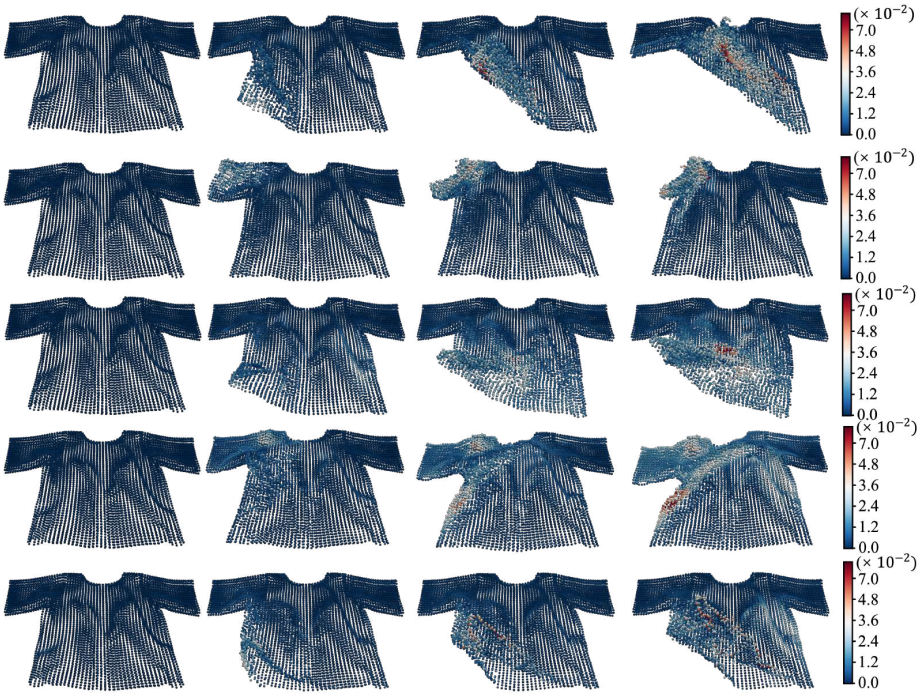
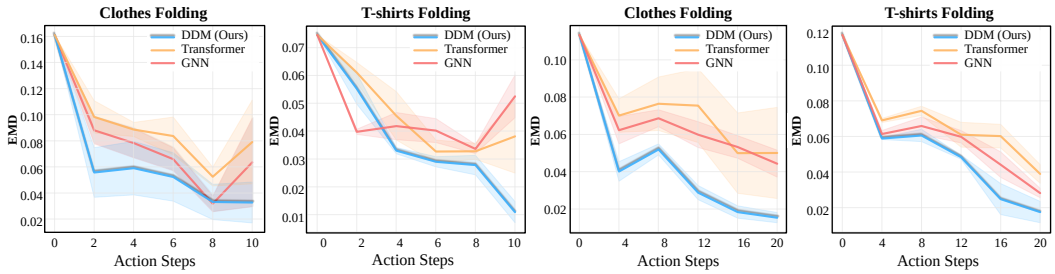


Figure 8: **Qualitative results on t-shirt dynamics prediction using DDM.**

A.2 Real-world Planning

Additional Quantitative Results We present quantitative results using the EMD metric, which measures the distance from the initial state to the target state in real-world planning scenarios in Figure 9.



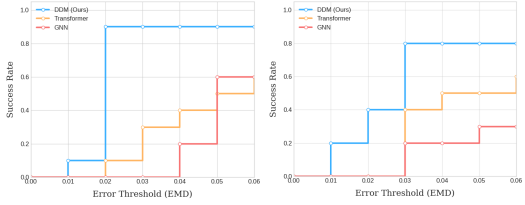
(a) Single-Step Manipulation (b) Long-Horizon Manipulation

Figure 9: **Quantitative evaluation of planning performance.** Earth Mover's Distance (EMD) convergence during the planning stage, measured over 10 repeated trials with identical initial and target configurations. Errors represent 95% confidence intervals. Our method outperforms baselines by achieving lower EMD values and faster convergence to goal states.

In single-step manipulation scenarios, our dynamics model exhibits superior performance across both object types. For cloth folding, our method consistently achieves lower EMD values with reduced confidence intervals, indicating enhanced prediction reliability compared to baseline approaches. This performance advantage is particularly evident in T-shirt folding, where topological complexity presents heightened challenges. While baseline methods, especially GNN, exhibit increased variance and elevated EMD values, our approach demonstrates consistent performance improvements throughout the planning horizon, suggesting enhanced handling of complex geometric relationships.

The multi-step scenarios, extending to 20 steps, further highlight our method's efficacy in long-horizon predictions. Our approach maintains significantly reduced EMD values with a consistent downward trajectory for both cloth and T-shirt manipulation tasks. The performance gap between our method and baselines becomes increasingly pronounced over extended horizons, particularly in T-shirt manipulation, where dual-layer structures introduce additional complexity. This sustained performance advantage in multi-step scenarios underscores our model's robust capability in mitigating error accumulation while maintaining prediction accuracy across extended planning sequences.

We present a comprehensive breakdown of the success rates in real-world long-horizon setting (Figure 10) under different thresholds in Figure 1, where our method consistently outperforms all baselines.



(a) Cloth folding (b) T-shirt folding

Figure 10: **Success rate under different threshold.**

Additional Qualitative Results Accordingly, we also provide additional qualitative results for single step and multi step scenarios on clothes and T-shirts in Figure 11. The results validate our system's capability to accurately manipulate diverse fabric items from arbitrary initial configurations to challenging target folding states.

Additional Intra-class Generalization Results We provide additional qualitative results for intra-class generalization on square cloth object in Figure 12 with sizes ranging from 20 cm to 40 cm. Our model successfully executes precise folding trajectories across these variations, consistently achieving target configurations and confirming robust intra-class generalization.

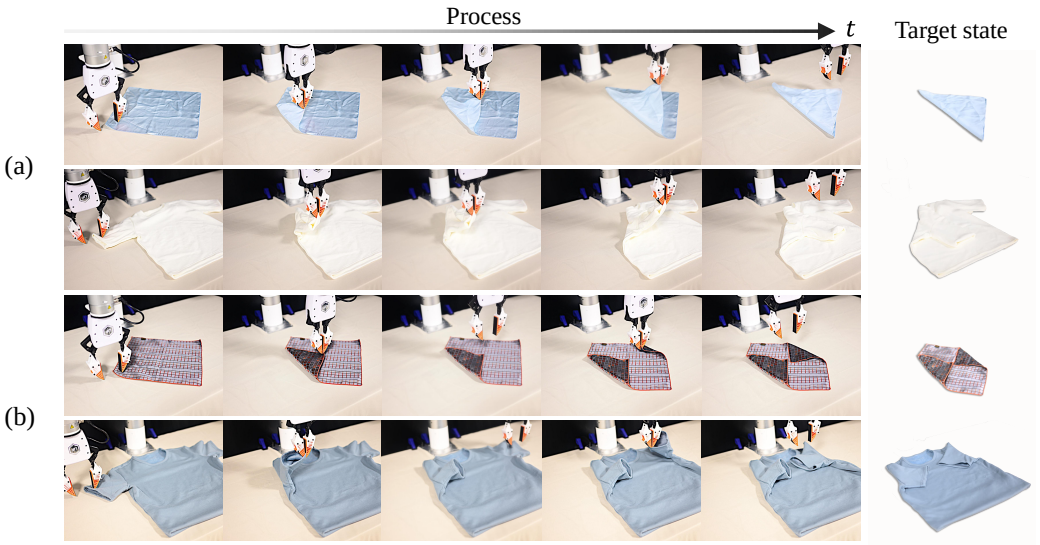


Figure 11: **Qualitative results of real-world system deployment.** The target state is represented in the last column of each row. Each experimental sequence illustrates the progressive deformation states during folding tasks. The first two rows correspond to single-step scenarios, while the last two represent multi-step scenarios.

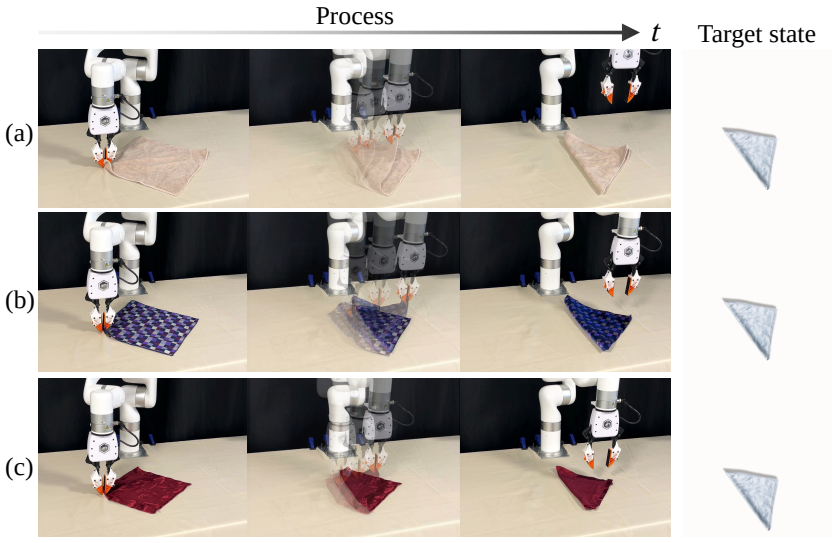
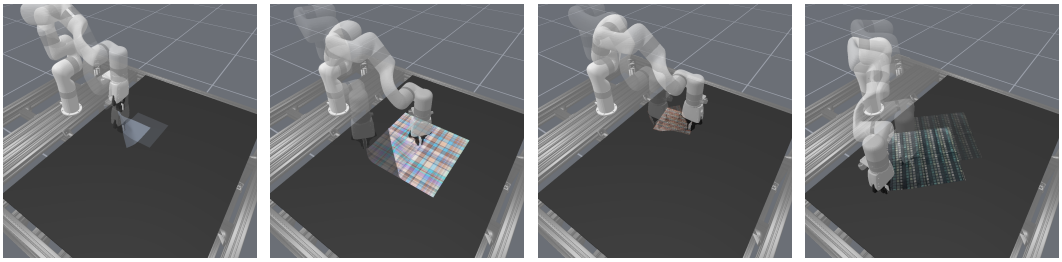


Figure 12: **Intra-class generalization evaluation.** We demonstrate that our method can generalize across garments with varying physical attributes (size, material, and color). The garment size progressively decreases from (a) to (c).

A.3 Simulation Planning Results

We present more qualitative results in Figure 14 in the simulation environment on planning. We design four simple tasks in simulation for system validation as visualized in Figure 13.



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

Figure 13: **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.

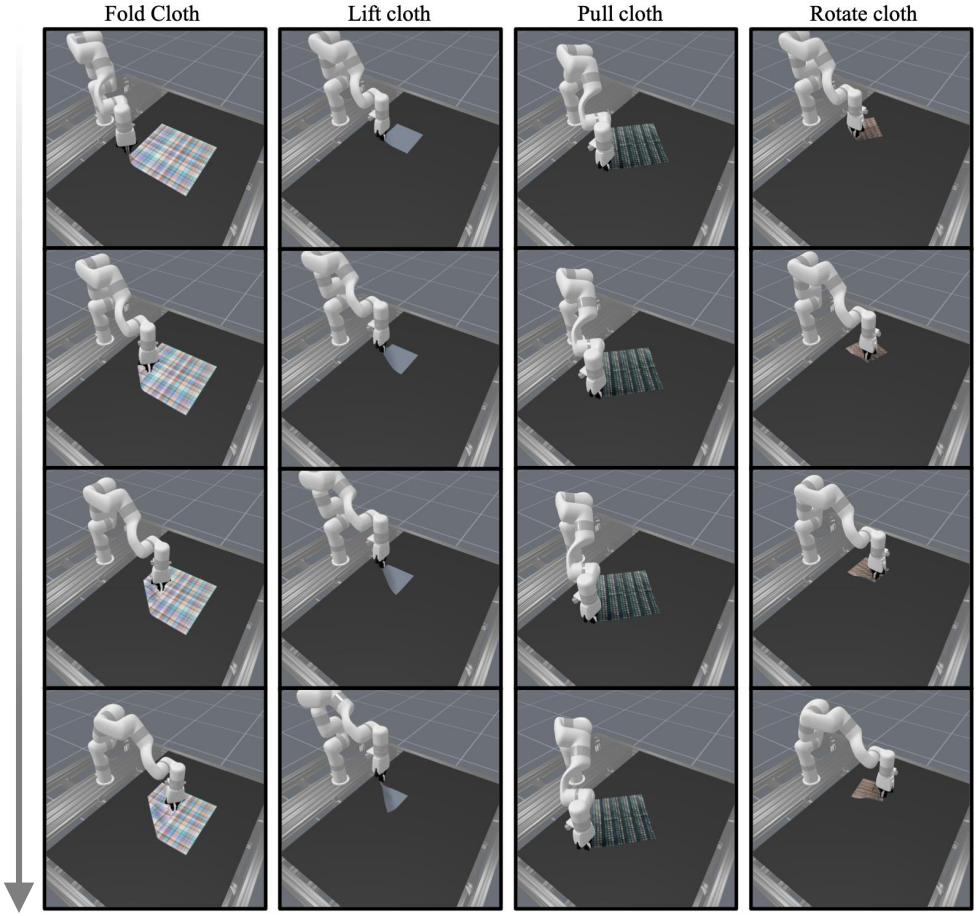


Figure 14: **Model predictive control evaluation in simulation.** Demonstration of our diffusion-based dynamics model integrated with MPC across diverse manipulation tasks using xArm7, validated on various cloth types.

B Experiment Setup

B.1 Task Description

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

Square Cloth folding and unfolding. This task explores robotic cloth folding and unfolding 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 the 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.

Garment folding and unfolding. This task focuses on folding or unfolding a T-shirt or a long-sleeve top into the target configuration. Such garments present unique challenges due to their dual-layer structure and compliant dynamics. We evaluate our approach on garments 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 17 shows all the test cloths and garments with various materials and sizes used in our real-world experiments.

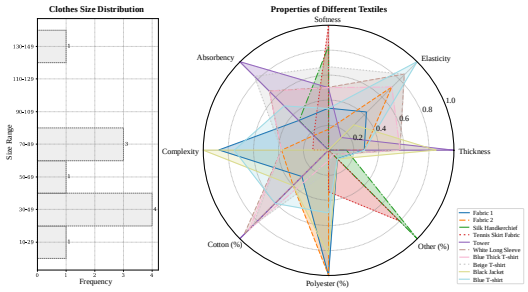


Figure 15: **Coverage of different clothes in our experiments.**

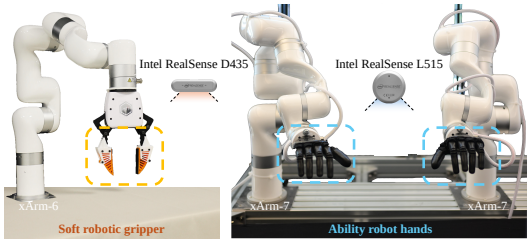


Figure 16: **Hardware overview.** Our real-world platform includes a UFactory xArm-6 and a bimanual dexterous system consisting of two UFactory xArm-7 robots with Ability hands. Each robot is equipped with one RGB-D camera.

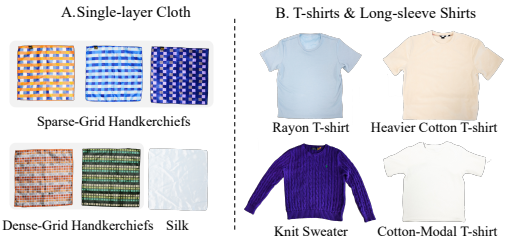


Figure 17: **Cloth overview.** We evaluate our method on different single-layer clothes as well as dual-layer T-shirts and long-sleeve shirts with varying colors and materials.

We visualize the distribution of all the clothes and garments in our expriments in Figure 15

B.2 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 $\times$ 480 resolution for the xArm-6 and the L515 with 1024 $\times$ 768 resolution for the dual-arm system. Figure 16 illustrates our hardware setup.

C Implementation Details

C.1 Data Collection

We collect training data for learning state estimation and dynamics prediction in a simulation environment built on SAPIEN [45]. 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 [46]. 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 hyperelastic thin shell. We follow Ly et al. [47] to simulate collision and friction in the PD system. We provide detailed physical parameters for cloth simulation in Table 6.

Physical 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 6: Simulation physical parameters.

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 given a 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[48], 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 dynamic 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.

C.2 Model Details

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.

Model Architecture We design a transformer-based architecture for state estimation, which consists of a point cloud encoder, a positional embedding module, and a series of transformer blocks.

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 7: Point cloud encoder hyperparameters.

The model takes both point cloud observations and mesh states as input. In dynamics model, the model takes an additional input channel containing a binary mask, indicating grasped mesh vertices. 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	GELU
Output MLP dimensions	[512, 256]
Normalization type	AdaLayerNorm
Normalization epsilon	1e-5

Table 8: Model hyperparameters.

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 $2\times3\times F$, where F is the number of Fourier frequencies. Given the input action $a \in \mathbb{R}^{B\times N\times 3}$, where N is the number of actions and 3 represents the dimension of (x, y, z) coordinates, we compute the embedding $\mathbf{e} \in \mathbb{R}^{B\times N\times D_3}$ as:

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 9: Action encoder hyperparameters.

$$\mathbf{e}_{b,n,d} = \begin{bmatrix} \sin(2\pi f_d a_{b,n,x}) & \cos(2\pi f_d a_{b,n,x}) \\ \sin(2\pi f_d a_{b,n,y}) & \cos(2\pi f_d a_{b,n,y}) \\ \sin(2\pi f_d a_{b,n,z}) & \cos(2\pi f_d a_{b,n,z}) \end{bmatrix} \tag{2}$$

where D_3 is the action embedding dimension, $d \in \{0, \ldots, D_3/6 - 1\}$, and $f_d = 100^{d/(D_3/6)}$ are the Fourier feature frequencies. The resulting embedding e provides a rich, high-dimensional representation of the action space. This representation is then processed through an MLP to produce the final action embeddings, which is later injected as the condition into our model through cross-attention layers.

Figure 18: **Example training data.**

C.3 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 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.4 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 the 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:

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 10: Training hyperparameters.

$$d_{main} = \sum_{i=1}^{K} w_i (s_t^i - s_c^i), \quad w_i = \frac{1}{\|p_g - p_i\| + \epsilon} \tag{3}$$

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:

$$p(i) = \frac{\exp(\|s_t^i - s_c^i\|_2/\tau)}{\sum_j \exp(\|s_t^j - s_c^j\|_2/\tau)} \tag{4}$$

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:

$$g \sim p(i) \tag{5}$$

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 are listed in Table 11.

C.5 State Estimation Baseline Implementation

To create the fairest possible comparison, we provided the GT canonical mesh to both GarmentNets and MEDOR. This isolates the evaluation to their performance for mapping a known shape to a deformed configuration in the observation space. We retrained the TRTM baseline from scratch on our data, and evaluated GarmentNets and MEDOR using their official pretrained checkpoints. The model input domain gap is minimal, as both our work and these baselines use the CLOTH3D dataset [49] with the same crumpled-state generation procedures. The evaluation is particularly fair for MEDOR for its test-time adaptation mechanism.

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 11: Planning hyperparameters.

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 12: GNN model hyperparameters.

C.6 Dynamics Baseline Implementation

We introduce details of the dynamics baseline implementation.

GNNs We adopt the implementation from [6]. 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 with an adaptive radius range of $[0.05, 0.1]$. We show detailed parameters for graph construction in Table 12.

C.7 Manipulation Pipeline Details

Our system integrates OWLV2 [50] and Segment Anything [51] 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.

Algorithm 1 MPC Planning Algorithm

Require: Initial state s_i, target state s_t, dynamics model f_θ, 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**
    $A_{mppi} \leftarrow$ SampleGaussian$(K/2, L, \mu, \sigma, [a_{min}, a_{max}])$
    $A_{uniform} \leftarrow$ SampleUniform$(K/2, L, [a_{min}, a_{max}])$
    $A \leftarrow$ Concatenate$(A_{mppi}, A_{uniform})$
    $S_{pred} \leftarrow f_\theta(S, A)$ ▷ Predict trajectories
    $C \leftarrow$ 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
    $\mu, \sigma \leftarrow$ UpdateDistribution(A, C, τ) ▷ Update using weighted averaging
    $\sigma \leftarrow \sigma \cdot (1 - i/N)$ ▷ Anneal exploration
end for
return a_{best}