MESHPRIORDIT: HIERARCHICAL MODELING FOR ACTION-CONDITIONED CLOTH DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned cloth prediction must preserve material structure while limiting error accumulation during autoregressive rollouts. Mesh graph networks encode connectivity, but local propagation can miss distant interactions; global models capture such interactions without guaranteeing locally consistent motion. We introduce MeshPriorDiT, which treats a constrained mesh rollout as an explicit, correctable reference. A mesh GNN is autoregressively rolled out for five steps to form a motion prior. A deterministic spatiotemporal DiT regresses the corresponding displacement residuals in one forward pass per five-step window; a parameter-free topology-aware decoder coordinates neighboring corrections, and hard grasp projection enforces commanded motion. We construct action-conditioned cloth-dynamics datasets in SoftGym and Genesis and train and evaluate the model separately in each. In a nine-setting comparison on the full SoftGym TEST set under the same 15-step protocol, MeshPriorDiT outperforms all external baselines: its Global MSE is 60.56% lower than PGND-Point, the strongest external baseline, and 88.66% lower than our UniClothDiff-Dynamics reproduction. It is also 44.52% lower than the best Flow Matching-based internal alternative.

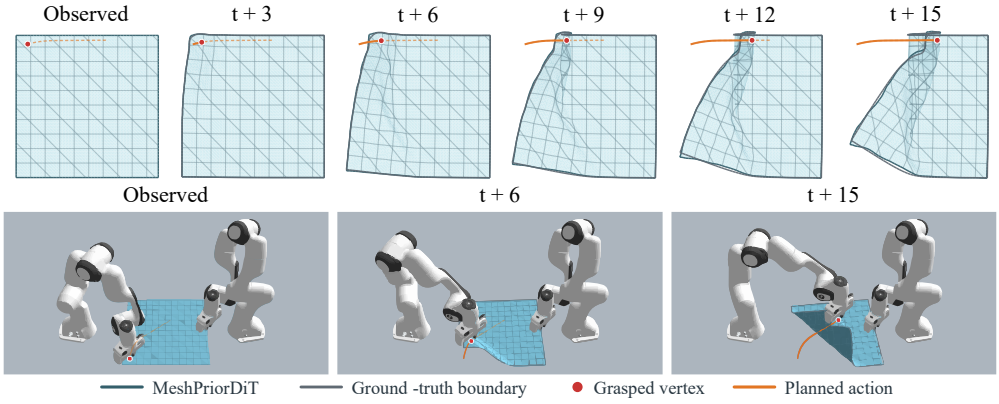

Figure 1: **MeshPriorDiT cloth rollouts.** MeshPriorDiT uses a GNN rollout as the reference trajectory for DiT-based residual prediction. SoftGym (top) and Genesis (bottom) show rollouts through $t + 15$, with ground-truth boundaries overlaid.

1 INTRODUCTION

Cloth manipulation is an important problem in robotic handling of deformable objects, with folding as a representative task (Lin et al., 2021). Learned cloth dynamics can also support model-based action planning (Tian et al., 2025). In mesh-based representations, cloth is discretized into interconnected vertices; its dynamics are shaped by material connectivity and grasp constraints and complicated by folding and self-contact (Tian et al., 2025; Longhini et al., 2025; Liao et al., 2024). Dynamics models must therefore maintain both locally plausible deformations and coordinated motion across distant regions. Otherwise, errors accumulated over multi-step rollouts may alter fold

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

locations and contact configurations, potentially affecting subsequent action selection (Pfaff et al., 2021; Sanchez-Gonzalez et al., 2020; Tian et al., 2025; Zhang et al., 2024).

Existing learned approaches largely follow two paradigms. Mesh-based graph neural networks (GNNs) represent vertices and material connections as graphs and learn deformation propagation through local message passing (Pfaff et al., 2021; Longhini et al., 2025; Zhang et al., 2024). Although explicit topology provides a useful inductive bias for local physics, finite message-passing depth restricts communication between distant mesh regions (Fortunato et al., 2022; Grigorev et al., 2023). Moreover, single-step graph models recursively consume their own predictions, which can amplify errors over long-horizon rollouts (Sanchez-Gonzalez et al., 2020; Würth et al., 2025). Multiscale and hierarchical graphs extend the propagation range but do not by themselves guarantee stable autoregressive rollouts.

Transformer-based and generative models instead capture cross-region motion through global interactions (Tian et al., 2025; Shao et al., 2022; Zhang et al., 2026). However, when material connectivity is not explicitly encoded, global attention does not guarantee consistent responses between neighboring material vertices. Hybrid methods also incorporate implicit interactions, hierarchical graphs, or mesh–Transformer representations (Shao et al., 2022; Würth et al., 2025; Iparraguirre et al., 2026; Zhang et al., 2026). Yet when local and global modeling are integrated within one predictor, aggregate accuracy alone cannot identify the mesh contribution or show whether global predictions preserve local consistency.

We propose **MeshPriorDiT**, which decomposes future cloth motion into a grasp-constrained mesh prior and a learned residual. A mesh GNN rolls out a reference trajectory along material connections. Conditioned on this trajectory, historical states, and planned actions, a deterministic spatiotemporal Diffusion Transformer (DiT)-style predictor (Peebles & Xie, 2023) regresses five-step displacement residuals in one forward pass. These residuals are scaled, accumulated into position corrections, smoothed across material neighbors, and composed with the prior under hard grasp constraints. This explicit interface makes the prior and correction separately measurable. Prior–residual modeling has also been explored in other domains (Kutsuna, 2025); here, the reference is an action-conditioned mesh rollout and the correction is deterministic.

Our contributions are:

- We introduce **MeshPriorDiT**, an action-conditioned cloth dynamics model with an explicitly correctable mesh-prior–residual interface.
- We compare nine settings on the full SoftGym TEST set under a shared 15-step protocol. Compared with PGND-Point (the strongest external baseline in Global MSE), our UniClothDiff-Dynamics reproduction, and the best Flow Matching-based internal alternative, MeshPriorDiT reduces rollout-averaged Global MSE by 60.56%, 88.66%, and 44.52%, respectively.
- We evaluate the mesh prior alone and analyze the composed predictions across tasks. On the full SoftGym TEST set, the full model reduces 15-step rollout-averaged Global MSE by 78.23% relative to GNN-Only.
- We construct action-conditioned cloth-dynamics datasets in SoftGym and Genesis. On the full Genesis TEST set, a separately trained MeshPriorDiT reduces 15-step rollout-averaged Global MSE by 82.13% relative to GNN-Only.

2 Related Work

2.1 Mesh-Based Learned Dynamics Models

Graph-based simulators represent physical elements as nodes and learn relational dynamics through message passing (Battaglia et al., 2016; Sanchez-Gonzalez et al., 2018; 2020). MeshGraphNets adapts this formulation to mesh vertices and mesh edges, predicting node-level dynamics through an encode–process–decode architecture (Pfaff et al., 2021). Multiscale extensions shorten propagation paths using coarse graphs, hierarchical representations, or bi-stride pooling (Fortunato et al., 2022; Grigorev et al., 2023; Cao et al., 2023). Mesh-based models encode mesh connectivity, yet

finite receptive fields may still limit long-range coordination; autoregressive prediction can also accumulate errors over time.

Cloth-oriented models further incorporate observations, material variation, and contact. ClothSplatting combines action-conditioned dynamics prediction with RGB-supervised state refinement, while AdaptiGraph adapts graph dynamics to material properties (Longhini et al., 2025; Zhang et al., 2024). Physics-guided garment models learn deformation from unsupervised or self-supervised objectives (Bertiche et al., 2021; Santesteban et al., 2022; Bertiche et al., 2022); SENC and ContourCraft target self-collision and multi-garment intersections (Liao et al., 2024; Grigorev et al., 2024). These works address complementary aspects of cloth modeling—state estimation, material adaptation, garment deformation, and collision handling—but do not explicitly use a grasp-constrained mesh rollout as a separately evaluated reference for residual dynamics prediction.

2.2 Generative and Hybrid Dynamics Models

Transformer-based models capture distant interactions beyond local graph propagation. Diffusion models denoise perturbed states (Ho et al., 2020), with DiTs using Transformer backbones (Peebles & Xie, 2023); Flow Matching and rectified flow learn continuous velocity fields along probability paths (Lipman et al., 2023; Liu et al., 2023). UniClothDiff applies Transformer diffusion to cloth state estimation and action-conditioned prediction, while TIE models particle interactions through implicit-edge attention (Tian et al., 2025; Shao et al., 2022). Attention alone, however, does not enforce material-edge consistency or grasp constraints.

Hybrid approaches combine geometry with broader spatial or temporal modeling. Mesh-reduced methods combine learned compression with temporal attention or conditional normalizing flows (Han et al., 2022; Sun et al., 2023), and Manifold-Aware Transformers inject geodesic surface relations into attention (Li et al., 2024). ROBIN combines hierarchical graphs with rolling diffusion for solid mechanics, whereas MeshGraphNet-Transformer and ClothTransformer use global and latent-space Transformers, respectively (Würth et al., 2025; Iparraguirre et al., 2026; Zhang et al., 2026). These methods do not explicitly separate a grasp-constrained mesh rollout from a learned displacement residual.

Iterative and residual methods offer another route to correction: PDE-Refiner improves long neural-solver rollouts, learned contact corrections augment reduced deformable dynamics, and Residual Prior Diffusion models residual variation relative to a coarse latent prior (Lippe et al., 2023; Romero et al., 2021; Kutsuna, 2025). These works do not study action-conditioned cloth-mesh rollouts under explicit grasp constraints. MeshPriorDiT instead uses the graph rollout as a structured local-physics prior, predicts residual motion with Residual DiT, and coordinates corrections over material neighbors before grasp-constrained composition, making the two components separately measurable.

3 Method

3.1 Problem Setup and Overview

3.1.1 Problem Setup.

We predict cloth motion from full mesh states. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{U})$ be a fixed template mesh with $N = |\mathcal{V}|$ vertices, material edges $\mathcal{E}$, and two-dimensional material coordinates $\mathbf{U} \in \mathbb{R}^{N\times 2}$. The vertex positions at time t are $\mathbf{q}_t \in \mathbb{R}^{N\times 3}$. Given K historical states and H planned end-effector displacements,

$$\mathbf{Q}_t^- = [\mathbf{q}_{t-K+1}, \ldots, \mathbf{q}_t], \qquad \mathbf{A}_t = [\mathbf{a}_{t+1}, \ldots, \mathbf{a}_{t+H}],$$

where $\mathbf{a}_{t+h} \in \mathbb{R}^3$, the model predicts future displacements. A grasp mask $\mathbf{M} \in \{0,1\}^{N\times 1}$ identifies controlled vertices ($M_i = 1$). We use $K = 3$ and $H = 5$. The ground-truth displacement sequence is

$$\Delta\mathbf{Q}_t^* = [\mathbf{q}_{t+h}^* - \mathbf{q}_{t+h-1}^*]_{h=1}^{H}, \qquad \mathbf{q}_t^* = \mathbf{q}_t.$$

3.1.2 Method Overview.

Figure 2 outlines the prior–residual decomposition. The Mesh Prior autoregressively predicts H-step displacements over the material topology under grasp constraints, forming a reference trajec-

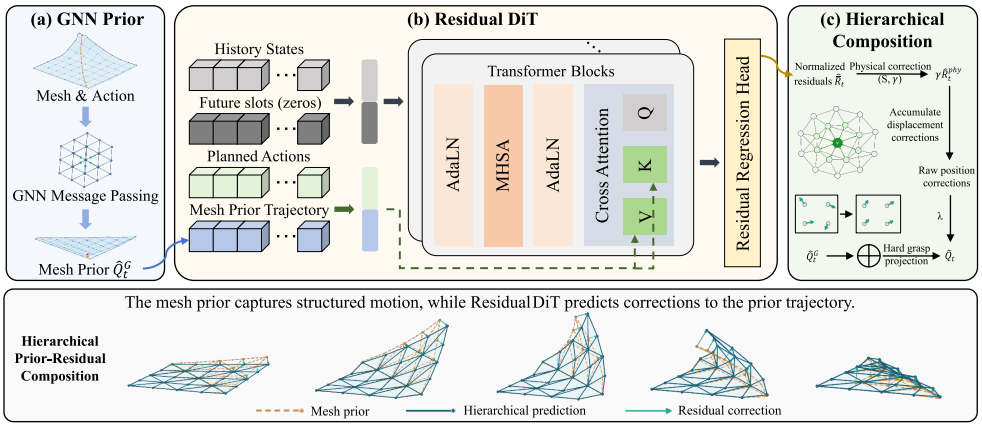

Figure 2: **Overview of MeshPriorDiT.** (a) A GNN rolls out an action-conditioned mesh prior along material connections. (b) Conditioned on the history, actions, and prior, Residual DiT predicts five-step displacement corrections in one forward pass. (c) The corrections are scaled by γ, accumulated into position corrections, smoothed over material neighbors, and combined with the prior under hard grasp constraints. In the bottom row, orange dashed lines, blue solid lines, and green arrows denote the prior, final prediction, and residual correction, respectively.

tory. Conditioned on the history, planned actions, and this prior, Residual DiT predicts displacement residuals in one forward pass. The residuals are scaled by γ and accumulated into position corrections; one row-normalized Jacobi update with weight λ coordinates these corrections over material neighbors before they are added to the prior. The final positions enforce the commanded grasp motion and are fed back during long-horizon rollout.

3.2 Action-Conditioned Mesh Dynamics Prior

3.2.1 Graph Encoding and Constrained Prediction.

The Mesh Prior predicts action-conditioned motion over the fixed material topology (Longhini et al., 2025; Pfaff et al., 2021). Each material edge is represented in both directions. Node features combine the previous displacement at free vertices or the planned displacement at grasped vertices with a vertex-type encoding. Edge features encode current three-dimensional relative positions and fixed two-dimensional material offsets, together with their norms. Full feature definitions are given in the appendix.

The normalized features pass through separate MLP encoders, $L = 15$ residual MeshGraphNets message-passing blocks, and a node decoder. Let $\mathbf{h}_i^L$ be the final latent state of vertex i. At inference, the decoded normalized discrete acceleration gives

$$\mathbf{z}_{k,i}^{\text{pred}} = \text{Dec}(\mathbf{h}_i^L), \qquad \mathbf{q}_{k+1}^{\text{cand}} = 2\mathbf{q}_k - \mathbf{q}_{k-1} + \mathcal{N}_\alpha^{-1}(\mathbf{z}_k^{\text{pred}}),$$

where $\mathcal{N}_\alpha^{-1}$ undoes discrete-acceleration normalization. The candidate state is then overwritten at grasped vertices:

$$\widehat{\mathbf{q}}_{k+1}^{\text{G}} = \mathbf{F} \odot \mathbf{q}_{k+1}^{\text{cand}} + \mathbf{M} \odot (\mathbf{q}_k + \mathbf{a}_{k+1}), \qquad \mathbf{F} = \mathbf{1} - \mathbf{M}.$$

The action is broadcast to grasped vertices, so their commanded motion is enforced exactly.

3.2.2 Prior Training and Multi-Step Rollout.

The Mesh Prior is trained independently using normalized single-step discrete acceleration. For an unperturbed transition, the target is $\mathbf{z}_k^{\text{tar}} = \mathcal{N}_\alpha(\mathbf{q}_{k+1}^* - 2\mathbf{q}_k + \mathbf{q}_{k-1})$. During training, the current positions of free vertices are perturbed and the target is adjusted accordingly, as detailed in the appendix. With $N_{\text{F}} = \sum_i F_i$, the loss for one example is

$$\mathcal{L}_{\text{prior}} = \frac{1}{N_{\text{F}}} \sum_{i=1}^{N} F_i \left\| \mathbf{z}_{k,i}^{\text{pred}} - \mathbf{z}_{k,i}^{\text{tar}} \right\|_2^2,$$

where $\mathbf{z}_k^{\mathrm{tar}}$ denotes the adjusted target when training perturbation is applied.

Although the full model uses $K = 3$ historical states, the second-order Mesh Prior uses only the two most recent states. From the two most recent history states, we apply the constrained update for H steps to obtain

$$\widehat{\mathbf{Q}}_t^{\mathrm{G}} = \left[\widehat{\mathbf{q}}_{t+1}^{\mathrm{G}}, \ldots, \widehat{\mathbf{q}}_{t+H}^{\mathrm{G}}\right],$$
$$\widehat{\Delta\mathbf{Q}}_t^{\mathrm{G}} = \left[\widehat{\mathbf{q}}_{t+h}^{\mathrm{G}} - \widehat{\mathbf{q}}_{t+h-1}^{\mathrm{G}}\right]_{h=1}^{H}, \qquad \widehat{\mathbf{q}}_t^{\mathrm{G}} = \mathbf{q}_t.$$

The position trajectory is the reference for topology-aware decoding; the displacement sequence defines the residual target and conditions Residual DiT.

3.3 Residual DiT

3.3.1 Prior-Conditioned Spatiotemporal Modeling.

Residual DiT uses the mesh rollout as a reference and jointly models the difference between the ground-truth future displacements and the H-step mesh prior. The physical residual and its normalized target are

$$\mathbf{R}_t^* = \Delta\mathbf{Q}_t^* - \widehat{\Delta\mathbf{Q}}_t^{\mathrm{G}},$$
$$\overline{\mathbf{R}}_t^* = \mathbf{F}_H \odot \mathrm{clip}(\mathbf{R}_t^* \oslash \mathbf{S}, -1, 1).$$

Here $\mathbf{S} \in \mathbb{R}^3$ is a fixed coordinate-wise scale computed from training-set residual statistics and broadcast over time and vertices; $\oslash$ and clip denote element-wise division and clipping. The mask $\mathbf{F}_H$ broadcasts $\mathbf{F}$ over time and coordinates, zeroing grasped-vertex residuals.

We use a patch-token-based DiT-style spatiotemporal Transformer (Peebles & Xie, 2023). Its input contains the K historical states followed by H zero-filled future slots, with the grasp mask appended to every frame. Neither future ground truth nor noisy residuals are supplied to the model. Planned actions are encoded as action tokens, while mesh-prior tokens contain step-wise prior displacements, their cumulative displacements, and grasp indicators. Conditioned on these inputs, the model predicts the normalized residuals for all H future steps in one forward pass. The mesh prior thus defines the residual target and provides an explicit condition for modeling corrections across vertices and time.

3.3.2 Direct Residual Regression.

With the Mesh Prior frozen, Residual DiT directly regresses the normalized target $\overline{\mathbf{R}}_t^*$ defined above. Let d_ϕ denote the DiT predictor with parameters ϕ. Its free-vertex prediction is

$$\widehat{\overline{\mathbf{R}}}_t = \mathbf{F}_H \odot d_\phi\left(\mathbf{Q}_t^-, \mathbf{A}_t, \widehat{\Delta\mathbf{Q}}_t^{\mathrm{G}}, \mathbf{M}\right).$$

The DiT backbone receives a fixed zero timestep input. For one training window, the loss is

$$\mathcal{L}_{\mathrm{res}}(\phi) = \frac{\left\|\widehat{\overline{\mathbf{R}}}_t - \overline{\mathbf{R}}_t^*\right\|_F^2}{3HN_{\mathrm{F}}}.$$

The loss is averaged over training windows; both residual tensors are zero at grasped vertices. At inference, the prediction is converted to physical displacement units by

$$\widehat{\mathbf{R}}_t^{\mathrm{phy}} = \mathbf{S} \odot \widehat{\overline{\mathbf{R}}}_t.$$

Inference requires one deterministic DiT evaluation per prediction window, without noise sampling or numerical integration. Residual gain and topology-aware decoding are applied in the next stage.

3.4 Topology-Aware Residual Decoding and Composition

The free-vertex residual loss does not explicitly couple corrections at neighboring material vertices. We therefore apply one non-learned Jacobi update to the position residual relative to the online Mesh Prior.

Let $\widehat{\mathbf{r}}^{\mathrm{phy}}_{t+h}$ and $\widehat{\Delta\mathbf{q}}^{\mathrm{G}}_{t+h}$ denote the physical residual and Mesh Prior displacement at step h. The raw prediction is

$$\widehat{\Delta\mathbf{q}}^{\mathrm{raw}}_{t+h} = \mathbf{F} \odot \left(\widehat{\Delta\mathbf{q}}^{\mathrm{G}}_{t+h} + \gamma \widehat{\mathbf{r}}^{\mathrm{phy}}_{t+h} \right) + \mathbf{M} \odot \mathbf{a}_{t+h},$$

$$\widehat{\mathbf{q}}^{\mathrm{raw}}_{t+h} = \mathbf{q}_t + \sum_{j=1}^{h} \widehat{\Delta\mathbf{q}}^{\mathrm{raw}}_{t+j},$$

where γ controls correction magnitude and the action is broadcast to grasped vertices.

Let $\widetilde{\mathbf{W}}_{\mathcal{G}} = \mathbf{D}^{-1}\mathbf{W}_{\mathcal{G}}$ be the row-normalized binary adjacency of the material mesh, with degree matrix $\mathbf{D}$. The position correction and final prediction are

$$\begin{aligned}
\widehat{\mathbf{e}}^{\mathrm{raw}}_{t+h} &= \widehat{\mathbf{q}}^{\mathrm{raw}}_{t+h} - \widehat{\mathbf{q}}^{\mathrm{G}}_{t+h}, \\
\widehat{\mathbf{e}}^{\mathrm{topo}}_{t+h} &= (1-\lambda)\widehat{\mathbf{e}}^{\mathrm{raw}}_{t+h} + \lambda \widetilde{\mathbf{W}}_{\mathcal{G}} \widehat{\mathbf{e}}^{\mathrm{raw}}_{t+h}, \\
\widehat{\mathbf{q}}_{t+h} &= \mathbf{F} \odot \left(\widehat{\mathbf{q}}^{\mathrm{G}}_{t+h} + \widehat{\mathbf{e}}^{\mathrm{topo}}_{t+h} \right) + \mathbf{M} \odot \widehat{\mathbf{q}}^{\mathrm{raw}}_{t+h}.
\end{aligned}$$

Thus, λ controls neighborhood smoothing while the final projection leaves commanded grasp positions unchanged; $\lambda = 0$ recovers the raw prediction.

For longer rollouts, only the first H-step block uses observed history. Subsequent blocks use the most recent K final predicted states and recompute the Mesh Prior and residual without future ground truth or teacher forcing.

4 Experiments

We organize our experiments around four research questions:
Q1: Does MeshPriorDiT outperform competing methods over multi-window autoregressive cloth rollouts?
Q2: Does correcting the mesh prior help, and how do the tested full-state and residual designs compare?
Q3: Within the prior–residual design, how do residual modeling and topology-aware decoding affect accuracy, mesh quality, and inference cost?
Q4: Does the framework remain effective when trained separately on a second cloth dynamics dataset?

4.1 Experimental Setup

Dataset and Splits. We construct 500,000 action-conditioned cloth dynamics windows in SoftGym across Diagonal Fold, Corner-to-Side Fold, and Vertical Lift. Complete trajectories are split before window extraction into 24,822/3,106/3,106 train/validation/test trajectories (400,000/50,000/50,000 windows); each window contains three observed and five target frames. The nine-setting 15-step (H15) comparison evaluates all 3,106 SoftGym TEST trajectories. We evaluate a separately generated Genesis dataset in Q4.

Compared Methods. MeshPriorDiT uses deterministic residual prediction. External comparisons include UniClothDiff-Dynamics (our DiT–DDPM reproduction) and our state-level adaptations of Particle-Grid Neural Dynamics (PGND) (Zhang et al., 2025) with point- and radius-grasp forcing. GNN-Only is the same trained mesh prior without residual correction; it serves as both a structured reference and a component control. We additionally compare three Flow Matching designs: DiT-Only predicts full motion without a prior, Prior-conditioned DiT predicts full motion conditioned on the prior, and Prior-residual DiT predicts a correction to it. Balanced and MinMSE are validation-selected inference settings of the same Prior-residual DiT checkpoint.

Training and Evaluation Protocol. The mesh prior is trained independently and frozen during residual-model training. MeshPriorDiT predicts five frames with one deterministic DiT evaluation per window; each Flow Matching design uses 100 velocity-network evaluations per window. We report Global, Active, Fold, and Edge-strain MSE after restoring predictions to physical space. H15

comprises three consecutive five-frame windows: only the first uses observed history, and subsequent windows use model predictions without teacher forcing. Relative changes are computed from unrounded records.

Table 1: **SoftGym H15 Global MSE on the full TEST set.** Rollout-averaged values are in units of $10^{-5}\,\mathrm{m}^2$ (lower is better). **Bold** and underlined values indicate the best and second-best results, respectively, across all nine evaluation settings. UniClothDiff-Dynamics is our reproduction. †GNN-Only is the mesh-prior branch without residual correction and also serves as a component ablation. The two Prior-residual DiT operating points share one checkpoint.

Method	**Overall**	Corner-to-Side	Diagonal Fold	Vertical Lift
External baselines				
UniClothDiff-Dynamics (repr.)	14.898	10.946	13.133	17.896
PGND (point grasp)	4.283	2.990	5.018	4.577
PGND (radius grasp)	5.867	4.612	8.817	4.978
Component baseline				
GNN-Only†	7.758	6.086	4.307	10.445
Internal alternatives				
DiT-Only	4.762	3.886	7.341	3.870
Prior-conditioned DiT	6.680	5.089	13.462	3.958
Prior-residual DiT (Balanced)	3.756	2.751	<u>4.133</u>	4.084
Prior-residual DiT (MinMSE)	<u>3.045</u>	<u>2.271</u>	4.836	<u>2.511</u>
Primary method				
MeshPriorDiT (Ours)	**1.689**	**0.986**	**3.609**	**1.051**

4.2 Q1: Comparison with Competing Methods

Table 1 compares nine settings on all 3,106 SoftGym TEST trajectories at H15. MeshPriorDiT achieves the lowest overall Global MSE of $1.689 \times 10^{-5}\,\mathrm{m}^2$. This is 60.56% lower than PGND-Point, the strongest external baseline, and 88.66% lower than our UniClothDiff-Dynamics reproduction (paired 95% CIs: 55.66–64.90% and 86.98–90.12%, respectively).

The advantage extends across all three action categories. MeshPriorDiT obtains Global MSEs of 0.986, 3.609, and 1.051 in units of $10^{-5}\,\mathrm{m}^2$ for Corner-to-Side Fold, Diagonal Fold, and Vertical Lift, respectively. These are 67.01%, 28.09%, and 77.03% lower than the corresponding PGND-Point results. Figure 3 shows that its running-average Global MSE remains the lowest throughout H15, including the later five-frame blocks that use predicted rather than observed history.

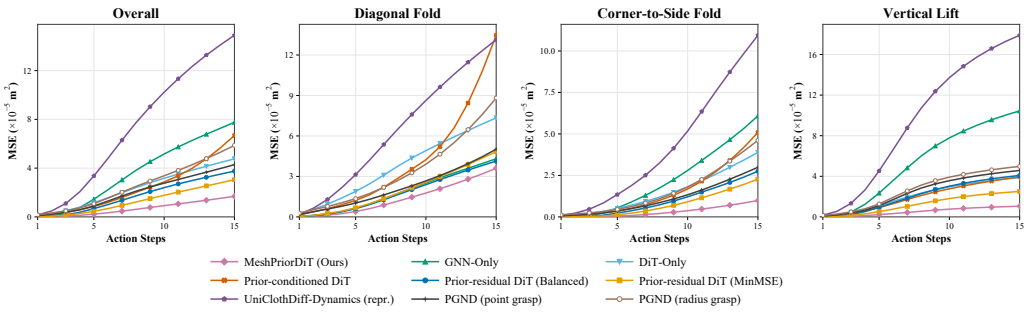

Figure 3: **Running-average Global MSE over SoftGym H15 rollouts.** Nine settings are compared over H15 autoregressive rollouts, overall and by action category; lower is better.

Answer to Q1. MeshPriorDiT achieves the lowest H15 Global MSE overall, in every action category, and throughout the autoregressive rollout among the nine evaluated settings.

4.3 Q2: Mesh Prior and Prediction Target

We first assess the value of correcting the Mesh Prior. GNN-Only uses the same trained prior without the residual correction stage. On all 3,106 SoftGym TEST trajectories at H15, MeshPriorDiT reduces rollout-averaged Global MSE from 7.758×10^{-5} to $1.689 \times 10^{-5}\,\mathrm{m}^2$, a 78.23% reduction. Figure 4 illustrates the refinement in one folding example.

We then compare three Flow Matching-based designs under the same H15 evaluation protocol in Table 1. DiT-Only predicts full displacements without a prior; Prior-conditioned DiT predicts full displacements with the mesh rollout as a condition; Prior-residual DiT predicts corrections added to that rollout. Their overall Global MSEs are 4.762, 6.680, and 3.045, respectively, in units of $10^{-5}\,\mathrm{m}^2$, with the last value obtained at the MinMSE operating point. The residual variant has 36.07% lower Global MSE than DiT-Only and 54.42% lower Global MSE than Prior-conditioned DiT. This within-Flow comparison provides complementary evidence for the prior–residual design used by MeshPriorDiT: under an alternative learning formulation, treating the mesh rollout as an explicit, correctable reference outperforms both omitting the prior and using it only as a condition in the tested settings.

Answer to Q2. The learned correction substantially improves the Mesh Prior, while the tested Flow-based designs favor an explicit residual target over full-displacement prediction. Together, these results support the prior–residual interface used by MeshPriorDiT.

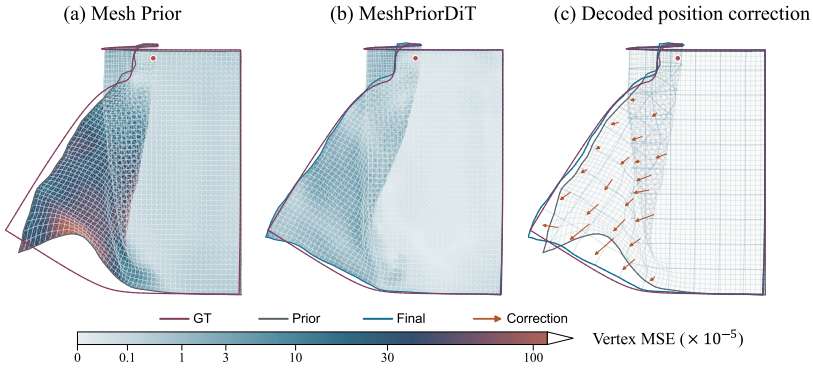

Figure 4: **Illustrative Mesh Prior refinement.** MeshPriorDiT reduces the prior's deformation error and brings the predicted cloth geometry closer to ground truth, particularly around the fold.

4.4 Q3: Residual Modeling, Topology-Aware Decoding, and Inference Cost

Having compared prediction targets, we examine two implementations of the prior–residual design. MeshPriorDiT directly regresses the residual, whereas the Prior-residual DiT (MinMSE) from Q2 uses Flow Matching. Both use $(\gamma, \lambda) = (1.0, 0.8)$. On the full SoftGym TEST set, MeshPriorDiT has 44.52% lower H15 Global MSE (1.689 versus 3.045, in units of $10^{-5}\,\mathrm{m}^2$) and 35.80% lower Edge-strain MSE (4.951 versus 7.712, in units of 10^{-3}). The Balanced setting of the same Flow checkpoint obtains 5.637×10^{-3} Edge-strain MSE, also above MeshPriorDiT.

We examine residual gain γ and topology-aware smoothing λ with a post-hoc 5×7 sweep of the frozen deterministic checkpoint. Each setting evaluates all 3,106 SoftGym TEST trajectories at H15. Unlike the trajectory-weighted results in Table 1, Figure 5 reports equal-action macro averages. At the validation-selected point $(1, 0.8)$, Global and Edge-strain MSEs are 1.882×10^{-5} and 4.257×10^{-3}, compared with 6.946×10^{-5} and 4.846×10^{-3} for GNN-Only. Removing smoothing, $(1, 0)$, lowers Global MSE to 1.819×10^{-5} but raises Edge-strain MSE to 4.719×10^{-3}. The setting $(0.75, 1)$ lowers Edge-strain MSE to 4.058×10^{-3} at a Global MSE of 2.132×10^{-5}. The validation-selected operating point remains fixed in the main comparisons.

Direct regression requires one DiT evaluation per five-frame window; Heun-50 requires 100 velocity-network evaluations. In a batch-one H15 benchmark on one preselected VALID trajectory, median inference latency is 0.196 s for MeshPriorDiT and 7.380 s for Prior-residual DiT (MinMSE),

including online mesh-prior rollout. Both use approximately 2.25 GiB of peak Torch-allocated memory.

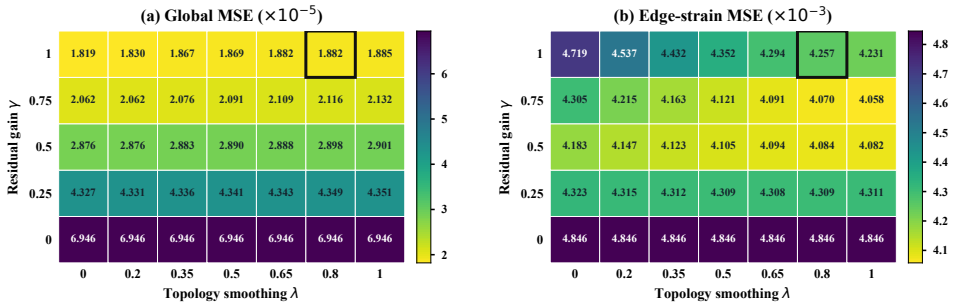

Figure 5: **Sensitivity to residual gain and topology-aware smoothing.** Each H15 setting evaluates all SoftGym TEST trajectories; results are macro-averaged equally across action categories. Black outlines mark the validation-selected operating point $(\gamma, \lambda) = (1, 0.8)$.

Answer to Q3. Among the evaluated prior–residual implementations, MeshPriorDiT achieves lower H15 Global and Edge-strain MSE with fewer network evaluations and lower measured latency. Residual gain and topology-aware smoothing allow coordinate accuracy and edge-strain error to be balanced.

4.5 Q4: Evaluation on a Second Cloth Dataset

We evaluate the same prior–residual framework in a separately generated Genesis dataset with a 50×50 cloth mesh, two active robot grasps, and Diagonal Fold, Corner-to-Side Fold, and Vertical Lift actions. The mesh prior and deterministic Residual DiT are trained on the Genesis training split, and the checkpoint is selected using validation data. The H15 evaluation follows the same three-block autoregressive protocol and compares MeshPriorDiT with its GNN-Only prior on all 3,106 Genesis TEST trajectories.

MeshPriorDiT reduces rollout-averaged Global MSE from 2.496×10^{-4} to $4.461 \times 10^{-5}\,\mathrm{m}^2$, an 82.13% reduction (paired trajectory-bootstrap 95% CI: 80.60–83.68%). The reductions are 94.70% for Corner-to-Side Fold, 62.41% for Diagonal Fold, and 90.94% for Vertical Lift, showing improvements across all three actions. Overall Active and Fold MSE decrease by 86.13% and 79.14%, respectively; aggregate Edge-strain MSE decreases by 2.84%.

Answer to Q4. When trained independently in the bimanual Genesis setting, MeshPriorDiT improves substantially over its mesh-prior component across all three actions, especially in Global, Active, and Fold MSE.

5 Conclusion

We presented MeshPriorDiT, which combines an action-conditioned mesh prior with deterministic DiT-based residual prediction for cloth dynamics. On the full SoftGym TEST set, it reduces H15 Global MSE by 60.56% relative to PGND-Point, the strongest external baseline, and by 44.52% relative to the best Flow Matching-based internal alternative. Its 78.23% reduction relative to GNN-Only shows the benefit of correcting the mesh prior. With separate training in Genesis, it achieves an 82.13% reduction relative to GNN-Only. A fixed-checkpoint sensitivity analysis reveals a trade-off between coordinate accuracy and edge-strain error. These results support explicit prior–residual modeling in the evaluated simulation settings, but do not establish transfer to unseen mesh topologies or real cloth. Future work will examine varying material properties and real-robot manipulation.

AI USE STATEMENT

In this work, we used generative AI tools solely to improve the clarity, grammar, and style of author-written manuscript text. We did not use these tools for research ideation or execution, data generation, mathematical proofs, or other research tasks. The authors reviewed all AI-assisted edits to ensure that they did not change the scientific claims, citations, or reported results. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The Method and Experimental Setup sections describe the primary model and evaluation protocol. Appendix A details dataset construction, trajectory-level splits, autoregressive evaluation, and metric aggregation. Appendix B provides model inputs and training procedures, while Appendix C documents comparison methods and protocol differences. The appendix also specifies the alternative residual models and the separately trained Genesis setting. We will publicly release the code, datasets, and trained checkpoints after the review process to support reproduction of the reported experiments.

REFERENCES

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/3147da8ab4a0437c15ef51a5cc7f2dc4-Abstract.html.

Hugo Bertiche, Meysam Madadi, and Sergio Escalera. PBNS: Physically based neural simulation for unsupervised garment pose space deformation. *ACM Transactions on Graphics*, 40(6), 2021. doi: 10.1145/3478513.3480479.

Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Neural cloth simulation. *ACM Transactions on Graphics*, 41(6), 2022. doi: 10.1145/3550454.3555491.

Yadi Cao, Menglei Chai, Minchen Li, and Chenfanfu Jiang. Efficient Learning of Mesh-Based Physical Simulation with Bi-Stride Multi-Scale Graph Neural Network. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 3541–3558. PMLR, 2023. URL https://proceedings.mlr.press/v202/cao23a.html.

Meire Fortunato, Tobias Pfaff, Peter Wirnsberger, Alexander Pritzel, and Peter Battaglia. MultiScale MeshGraphNets, 2022. URL https://arxiv.org/abs/2210.00612.

Artur Grigorev, Michael J. Black, and Otmar Hilliges. HOOD: Hierarchical Graphs for Generalized Modelling of Clothing Dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16965–16974, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Grigorev_HOOD_Hierarchical_Graphs_for_Generalized_Modelling_of_Clothing_Dynamics_CVPR_2023_paper.html.

Artur Grigorev, Giorgio Becherini, Michael J. Black, Otmar Hilliges, and Bernhard Thomaszewski. ContourCraft: Learning to Resolve Intersections in Neural Multi-Garment Simulations. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–10, New York, NY, USA, 2024. Association for Computing Machinery. doi: 10.1145/3641519.3657408. URL https://doi.org/10.1145/3641519.3657408.

Xu Han, Han Gao, Tobias Pfaff, Jian-Xun Wang, and Li-Ping Liu. Predicting Physics in Mesh-reduced Space with Temporal Attention. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=XctLdNfCmP.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://papers.nips.cc/paper_files/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Mikel M. Iparraguirre, Iciar Alfaro, David Gonzalez, and Elias Cueto. MeshGraphNet-Transformer: Scalable Mesh-based Learned Simulation for Solid Mechanics, 2026. URL https://arxiv.org/abs/2601.23177.

Takuro Kutsuna. Residual Prior Diffusion: A Probabilistic Framework Integrating Coarse Latent Priors with Diffusion Models, 2025. URL https://arxiv.org/abs/2512.21593.

Peizhuo Li, Tuanfeng Y. Wang, Timur Levent Kesdogan, Duygu Ceylan, and Olga Sorkine-Hornung. Neural Garment Dynamics via Manifold-Aware Transformers. *Computer Graphics Forum*, 43(2): e15028, 2024. doi: 10.1111/cgf.15028. URL https://doi.org/10.1111/cgf.15028.

Zhouyingcheng Liao, Sinan Wang, and Taku Komura. SENC: Handling Self-collision in Neural Cloth Simulation. In *Computer Vision – ECCV 2024*, volume 15067 of *Lecture Notes in Computer Science*, pp. 385–402. Springer, 2024. doi: 10.1007/978-3-031-72673-6_21. URL https://doi.org/10.1007/978-3-031-72673-6_21.

Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. SoftGym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 432–448. PMLR, 2021. URL https://proceedings.mlr.press/v155/lin21a.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2946. URL https://papers.nips.cc/paper_files/paper/2023/hash/d529b943af3dba734f8a7d49efcb6d09-Abstract-Conference.html.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Alberta Longhini, Marcel Büsching, Bardienus Pieter Duisterhof, Jens Lundell, Jeffrey Ichnowski, Mårten Björkman, and Danica Kragic. Cloth-Splatting: 3D Cloth State Estimation from RGB Supervision. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2845–2865. PMLR, 2025. URL https://proceedings.mlr.press/v270/longhini25a.html.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning Mesh-Based Simulation with Graph Networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=roNqYL0_XP.

Cristian Romero, Dan Casas, Jesús Pérez, and Miguel A. Otaduy. Learning Contact Corrections for Handle-Based Subspace Dynamics. *ACM Transactions on Graphics*, 40(4):1–12, 2021. doi: 10.1145/3450626.3459875. URL https://doi.org/10.1145/3450626.3459875.

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4470–4479. PMLR, 2018. URL https://proceedings.mlr.press/v80/sanchez-gonzalez18a.html.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to Simulate Complex Physics with Graph Networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 2020. URL https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html.

Igor Santesteban, Miguel A. Otaduy, and Dan Casas. SNUG: Self-Supervised Neural Dynamic Garments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8140–8150, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Santesteban_SNUG_Self-Supervised_Neural_Dynamic_Garments_CVPR_2022_paper.html.

Yidi Shao, Chen Change Loy, and Bo Dai. Transformer with Implicit Edges for Particle-Based Physics Simulation. In *Computer Vision – ECCV 2022*, volume 13679 of *Lecture Notes in Computer Science*, pp. 549–564. Springer, 2022. doi: 10.1007/978-3-031-19800-7_32. URL https://doi.org/10.1007/978-3-031-19800-7_32.

Luning Sun, Xu Han, Han Gao, Jian-Xun Wang, and Liping Liu. Unifying predictions of deterministic and stochastic physics in mesh-reduced space with sequential flow generative model. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/bea78e2bb0abccc14404b24b90d9299f-Abstract-Conference.html.

Tongxuan Tian, Haoyang Li, Bo Ai, Xiaodi Yuan, Zhiao Huang, and Hao Su. Diffusion Dynamics Models with Generative State Estimation for Cloth Manipulation. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1703–1725. PMLR, 2025. URL https://proceedings.mlr.press/v305/tian25c.html.

Tobias Würth, Niklas Freymuth, Gerhard Neumann, and Luise Kärger. Diffusion-Based Hierarchical Graph Neural Networks for Simulating Nonlinear Solid Mechanics. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/bcd3c132bb19544815c5e660a2b044bd-Abstract-Conference.html.

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*, Delft, Netherlands, 2024. doi: 10.15607/RSS.2024.XX.010. URL https://www.roboticsproceedings.org/rss20/p010.html.

Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Particle-grid neural dynamics for learning deformable object models from RGB-D videos. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.036.

Yu Zhang, Yidi Shao, Wenqi Ouyang, Yushi Lan, Zhexin Liang, Chengrui Wu, Xudong Xu, and Xingang Pan. ClothTransformer: Unified Latent-Space Transformers for Scalable Cloth Simulation, 2026. URL https://arxiv.org/abs/2605.27852.

A Datasets and Evaluation Protocol

A.1 Dataset Construction and Splits

SoftGym. We generate action-conditioned cloth trajectories for Diagonal Fold, Corner-to-Side Fold, and Vertical Lift on a fixed 50×50 mesh with $N = 2{,}500$ vertices. Each model step contains 16 simulator substeps. The recorded state comprises full vertex positions, fixed material connectivity, a grasp mask, and the commanded gripper displacement. The gripper remains attached throughout each trajectory; release actions are not included.

Complete trajectories are assigned to training, validation, and TEST before overlapping prediction windows are extracted. The splits contain 24,822, 3,106, and 3,106 trajectories and yield 400,000, 50,000, and 50,000 windows, respectively, with $K = 3$ observed and $H = 5$ target frames. No

trajectory, and hence no overlapping window from that trajectory, crosses a split. The three action families are approximately balanced by window count; their complete-trajectory counts need not be equal.

Diagonal Fold moves a corner region toward the opposite corner, Corner-to-Side Fold moves a corner toward an opposing side, and Vertical Lift raises a selected grasp point. Future gripper paths are represented as step-wise displacement commands. All methods evaluated on a particular trajectory receive the same observed states and planned commands.

Genesis. We construct a separate dataset with a native 50×50 Genesis cloth mesh ($2{,}500$ vertices, $4{,}802$ faces, and $7{,}301$ material edges). Its action families are diagonal folding, Corner-to-Side Fold (toward the opposite edge), and Vertical Lift. The experiment uses 300,000 training windows from 18,616 trajectories, 50,000 validation windows, and 50,000 TEST windows. The frozen H15 TEST evaluation contains 3,106 trajectories, whose identifiers are disjoint from the training and validation trajectory identifiers. The Genesis prior and residual model are trained separately from the SoftGym models.

Genesis uses a bimanual control interface. Each step records commands for two arms, separate moving- and holding-arm grasp masks, and a commanded position for each controlled cloth vertex. Both grasp clusters remain constrained during a trajectory. Accordingly, the Genesis implementation uses per-vertex commanded boundaries rather than broadcasting the single three-dimensional SoftGym action to every grasped vertex.

A.2 Autoregressive Rollout

For a rollout of $L = WH$ steps, the model processes W successive, non-overlapping five-step action blocks. The first block uses observed history. Each later block receives the most recent $K = 3$ *final predicted* states as history. The Mesh Prior, residual prediction, topology-aware decoding, and grasp constraints are recomputed in every block. Neither future ground truth nor teacher forcing is used during evaluation.

The principal SoftGym H15 evaluation uses all 3,106 eligible TEST trajectories: 795 Corner-to-Side Fold, 795 Diagonal Fold, and 1,516 Vertical Lift trajectories. A cross-method comparison additionally requires matching trajectory identifiers, commands, rollout length, metric implementation, and aggregation rule.

A.3 Metrics and Aggregation

The definitions below specify the SoftGym metrics used in the full H15 evaluation. Let s index a TEST trajectory, $h = 1, \ldots, L$ a rollout step, and $i = 1, \ldots, N$ a vertex. Predictions and targets are restored to simulator-coordinate space before computing

$$d_{s,h,i} = \frac{1}{3} \left\| \widehat{\mathbf{q}}_{s,h,i} - \mathbf{q}^*_{s,h,i} \right\|_2^2 .$$

Thus, $d_{s,h,i}$ is the mean squared error over the three spatial coordinates of one vertex at one step.

For Active MSE, eligible free-vertex–step pairs within each trajectory are ranked by ground-truth displacement magnitude $\|\mathbf{q}^*_{s,h,i} - \mathbf{q}_{s,0,i}\|_2$ relative to the initial rollout state. The highest 25% form $\mathcal{I}_s^{\mathrm{act}}$. The ranking covers the entire evaluated rollout rather than being recomputed independently at each step.

For Fold MSE, vertex normals are constructed from the incident ground-truth mesh faces. A curvature proxy at vertex i is the mean material-neighbor disagreement $1 - \mathbf{n}^*_{s,h,i} \cdot \mathbf{n}^*_{s,h,j}$. The highest 25% of all vertex–step pairs under this proxy form $\mathcal{I}_s^{\mathrm{fold}}$. Both selection masks depend on ground truth, not on the method being evaluated.

Writing $\mathcal{T}$ for the evaluated trajectories, the rollout-averaged coordinate metrics are

$$\text{Global MSE} = \frac{\sum_{s\in\mathcal{T}}\sum_{h=1}^{L}\sum_{i=1}^{N} d_{s,h,i}}{|\mathcal{T}|LN},$$

$$\text{Active MSE} = \frac{\sum_{s\in\mathcal{T}}\sum_{(h,i)\in\mathcal{I}_s^{\text{act}}} d_{s,h,i}}{\sum_{s\in\mathcal{T}}|\mathcal{I}_s^{\text{act}}|},$$

$$\text{Fold MSE} = \frac{\sum_{s\in\mathcal{T}}\sum_{(h,i)\in\mathcal{I}_s^{\text{fold}}} d_{s,h,i}}{\sum_{s\in\mathcal{T}}|\mathcal{I}_s^{\text{fold}}|}.$$

In particular, an "H15 Global MSE" in our tables averages prediction error over all 15 steps; it is not the error at step 15. A per-step curve instead reports the error at each individual step.

Let $\mathcal{E}_{\text{u}}$ contain each undirected material edge once, and let ℓ_e^0 be edge $e = (i, j)$'s template rest length. Its strain in a predicted or target state $\mathbf{q}$ is

$$\xi_{s,h,e}(\mathbf{q}) = \frac{\|\mathbf{q}_{s,h,i} - \mathbf{q}_{s,h,j}\|_2 - \ell_e^0}{\ell_e^0}.$$

Edge-strain MSE is

$$\text{Edge-strain MSE} = \frac{1}{|\mathcal{T}|L|\mathcal{E}_{\text{u}}|}\sum_{s\in\mathcal{T}}\sum_{h=1}^{L}\sum_{e\in\mathcal{E}_{\text{u}}}\left[\xi_{s,h,e}(\widehat{\mathbf{q}}) - \xi_{s,h,e}(\mathbf{q}^*)\right]^2.$$

The evaluator aggregates the underlying error sums and denominators before rounding. Global, Active, and Fold MSE are displayed in units of $10^{-5}\,\text{m}^2$; Edge-strain MSE is displayed in units of 10^{-3}.

B Model and Training Details

B.1 Mesh-Prior Features and Supervision

We retain the notation of Sec. 3: $\mathbf{M}$ denotes grasped vertices, $\mathbf{F} = \mathbf{1} - \mathbf{M}$ denotes free vertices, and $\widehat{\mathbf{Q}}_t^{\text{G}}$ and $\widehat{\Delta\mathbf{Q}}_t^{\text{G}}$ denote the Mesh Prior's position and displacement sequences. Each undirected material edge is represented in both directions. The SoftGym node and directed-edge features are

$$\mathbf{v}_{k,i}^{\text{act}} = (1 - M_i)(\mathbf{q}_{k,i} - \mathbf{q}_{k-1,i}) + M_i\mathbf{a}_{k+1},$$

$$\mathbf{x}_{k,i} = \left[\mathbf{v}_{k,i}^{\text{act}}, \text{onehot}_9(c_i)\right],$$

$$\mathbf{e}_{k,ij} = \left[\mathbf{q}_{k,i} - \mathbf{q}_{k,j}, \|\mathbf{q}_{k,i} - \mathbf{q}_{k,j}\|_2, \mathbf{u}_i - \mathbf{u}_j, \|\mathbf{u}_i - \mathbf{u}_j\|_2\right].$$

Here c_i is the implementation's vertex type, whose nine-category encoding distinguishes free and grasped vertices in these data; $\mathbf{u}_i$ is the fixed two-dimensional material coordinate. The node feature uses the previous observed or predicted displacement at a free vertex and the planned action at a grasped vertex.

Node, edge, and acceleration-target normalizers accumulate statistics on training data only. Separate MLPs encode the node and edge features, followed by 15 residual MeshGraphNets message-passing blocks of hidden width 128 and a node decoder. The decoder predicts normalized discrete acceleration; its physical scale is restored before the second-order position update in Sec. 3. The prior is supervised at free vertices, while grasped positions are overwritten by the commanded positions after every prior step.

During prior training, the current free-vertex state is perturbed by $\boldsymbol{\eta}_k = \mathbf{F} \odot \boldsymbol{\epsilon}_k$, where $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, 0.003^2\mathbf{I})$. The input state and adjusted next-state target are

$$\mathbf{q}_k^{\text{in}} = \mathbf{q}_k + \boldsymbol{\eta}_k, \qquad \mathbf{q}_{k+1}^{\text{tar}} = \mathbf{q}_{k+1}^* + 0.9\boldsymbol{\eta}_k.$$

The adjusted normalized acceleration target appearing in the main-text free-vertex loss is therefore

$$\mathbf{z}_k^{\text{tar}} = \mathcal{N}_\alpha\left(\mathbf{q}_{k+1}^{\text{tar}} - 2\mathbf{q}_k^{\text{in}} + \mathbf{q}_{k-1}\right).$$

The perturbation and target adjustment are used only during training; rollout uses unperturbed predicted states.

B.2 Deterministic Residual DiT

The Mesh Prior is frozen when training Residual DiT. Its prior displacements define the physical target $\mathbf{R}_t^* = \Delta\mathbf{Q}_t^* - \widehat{\Delta\mathbf{Q}}_t^{\mathrm{G}}$. During residual training, the frozen training-split prior predictions are read from a cache; evaluation recomputes the prior online from the current rollout history. This distinction prevents a cached ground-truth-history prior from entering autoregressive evaluation.

The fixed coordinate-wise scale $\mathbf{S}$ is computed as the absolute 99.5th percentile of prior residuals over 512 selected training windows, with a minimum scale of 10^{-6}. Its SoftGym value is

$$\mathbf{S} = (0.0107487,\ 0.0103373,\ 0.0094373).$$

As in Sec. 3, the target is divided coordinate-wise by $\mathbf{S}$, clipped to $[-1, 1]$, and set to zero at grasped vertices. The training loss is the normalized residual MSE over free vertices.

The DiT input contains the three historical position frames followed by five zero-filled future slots, with the grasp mask appended to each of the eight frames. Neither future ground truth nor a noisy residual is supplied to the predictor. Planned actions are encoded as condition tokens. Prior-condition tokens encode the five step-wise prior displacements, their five cumulative displacements, and the grasp indicator: $5(3) + 5(3) + 1 = 31$ feature channels per vertex. The predictor receives a fixed zero timestep and returns all five normalized residual frames in one forward pass.

The SoftGym DiT uses a 50×50 input layout, patch size 25, four Transformer layers, 16 attention heads, and attention-head dimension 72. The recorded backbone-plus-prior-encoder parameter count is 277,887,235. Residual training uses AdamW with learning rate 10^{-5} and weight decay 0.01, BF16, gradient clipping at 1, a global batch size of 2,048 across eight A100 GPUs, 1,000 warm-up updates, and a cosine schedule.

B.3 Position Correction and Grasp Constraints

The decoder smooths a *position* correction, not the raw DiT displacement output. The implementation first accumulates the composed raw displacements into $\widehat{\mathbf{q}}_{t+h}^{\mathrm{raw}}$, then subtracts the online Mesh Prior position $\widehat{\mathbf{q}}_{t+h}^{\mathrm{G}}$:

$$\widehat{\mathbf{e}}_{t+h}^{\mathrm{raw}} = \widehat{\mathbf{q}}_{t+h}^{\mathrm{raw}} - \widehat{\mathbf{q}}_{t+h}^{\mathrm{G}}.$$

For binary material adjacency $\mathbf{W}_{\mathcal{G}}$ and its degree matrix $\mathbf{D}$, one row-normalized Jacobi update gives

$$\widehat{\mathbf{e}}_{t+h}^{\mathrm{topo}} = (1-\lambda)\widehat{\mathbf{e}}_{t+h}^{\mathrm{raw}} + \lambda\mathbf{D}^{-1}\mathbf{W}_{\mathcal{G}}\widehat{\mathbf{e}}_{t+h}^{\mathrm{raw}}.$$

The corrected position is added to the Mesh Prior at free vertices. At grasped vertices, the final projection restores the raw commanded position after smoothing, so neighboring corrections cannot displace the controlled boundary. The evaluated SoftGym operating point uses $\gamma = 1$ and $\lambda = 0.8$, selected without TEST tuning.

Genesis adaptation. Genesis preserves the prior–residual architecture and direct residual-regression objective, but replaces the single-grasp SoftGym control interface. Its per-step arm commands have shape 2×3; its two arm masks identify the moving and holding grasp clusters; and the future boundary supplies commanded positions for controlled vertices. The Genesis GNN projects controlled vertices to those positions. Its DiT condition additionally includes the arm masks and boundary increments. The Genesis prior and DiT are trained from scratch on that dataset.

C Comparison Methods and Protocol Matching

The principal SoftGym comparison contains nine *evaluation settings*, not nine independently trained models: three settings from external methods, the shared GNN-Only component control, four Flow Matching-based internal settings, and the primary deterministic MeshPriorDiT. The two Prior-residual DiT operating points share one trained Flow checkpoint and differ only in validation-selected inference parameters.

Why these external baselines. UniClothDiff directly addresses action-conditioned cloth dynamics with a Transformer diffusion model and reports substantially lower long-horizon error than earlier dynamics methods in its own experiments (Tian et al., 2025). PGND models deformable-object dynamics, including cloth, through interacting particles and a spatial grid; its authors report strong results against learned and physics-based simulators in their sparse-view RGB-D setting (Zhang et al., 2025). They are therefore recent, closely related, and competitive representatives of generative cloth prediction and particle–grid dynamics, respectively. Our comparisons start from the authors' released code: the UniClothDiff reproduction retains its dynamics network and DDPM sampling path, while the PGND adaptation uses the released particle–grid computation. We adapt their data and action interfaces to the shared state-based SoftGym protocol, train on its training split, and evaluate the resulting models on the same TEST trajectories as MeshPriorDiT. The values below are our protocol-matched results.

External methods. UniClothDiff-Dynamics (our reproduction) generates future motion directly, without a Mesh Prior or residual target. Following the released dynamics sampler, it receives the grasp mask as an input condition and accumulates the generated displacements into positions *without an additional hard-grasp projection*. This is a protocol difference from MeshPriorDiT, not an omitted step in our reproduction.

PGND is retrained on our trajectories in two grasp settings. PGND (point grasp) forces the selected grasp index, whereas PGND (radius grasp) retains the method's pre-step forcing of particles in the gripper neighborhood. Both settings retain the point-based dynamics and spatial interaction mechanism and use three observed states and planned commands. Training uses a five-step position loss over all vertices. Neither setting uses our Mesh Prior, residual target, or topology-aware decoder. Point and radius are two settings of one external method, not two separately published methods.

Shared component control and internal designs. GNN-Only is the same frozen Mesh Prior used inside MeshPriorDiT without residual correction. It is both a structured reference and a component control, rather than an independent external reproduction.

DiT-Only generates full future motion without a Mesh Prior. Prior-conditioned DiT receives the prior as a condition but still generates full future motion. Prior-residual DiT instead generates motion relative to the prior and composes the correction with it. These three internal designs are trained with Flow Matching and sampled using 50 second-order Heun steps. One five-frame window consequently requires 100 velocity-network evaluations, excluding any Mesh Prior evaluations. The Prior-residual DiT operating points are Balanced ($\gamma = 0.5, \lambda = 0.8$) and MinMSE ($\gamma = 1, \lambda = 0.8$), chosen on validation data.

D Extended SoftGym Results

D.1 Full-TEST H15 Comparison

Table 3 extends the main-text Global-MSE comparison to Active, Fold, and Edge-strain MSE. All nine settings were evaluated on the same 3,106 SoftGym TEST trajectories under the H15 autoregressive protocol; entries are computed from the matched, unrounded per-trajectory records.

Relative to its shared GNN-Only prior, MeshPriorDiT reduces full-TEST H15 rollout-averaged Global MSE by 78.23% (paired 95% confidence interval: 75.61–80.76%). The corresponding reductions are 70.34% (66.69–73.96%) for Active MSE, 57.43% (53.00–61.79%) for Fold MSE, and 12.75% (11.63–13.83%) for Edge-strain MSE. These comparisons establish the effect of correcting the shared prior. Across the nine settings, MeshPriorDiT has the lowest aggregate value for each of the four metrics. Its Global MSE is 60.56% below PGND (point grasp), the strongest external method in this comparison, and 88.66% below our UniClothDiff-Dynamics reproduction. Relative to the lowest-Global-MSE Flow alternative, Prior-residual DiT (MinMSE), it lowers Global MSE by 44.52% and Edge-strain MSE by 35.80%.

Table 2: **Comparison settings and protocol differences.** Residual motion is defined relative to the Mesh Prior. NFE counts DiT or velocity-network calls per five-frame window, excluding Mesh Prior calls.

Setting	Training target	Prior use	Inference
UniClothDiff-Dynamics (repr.)	Full motion; DDPM noise	None	DDPM, 1,000 steps
PGND (point grasp)	Multi-step position MSE	None	Iterative PGND
PGND (radius grasp)	Multi-step position MSE	None	Iterative PGND
GNN-Only	Normalized 1-step acceleration MSE	Prior itself	GNN rollout
DiT-Only	Full motion; Flow Matching	None	Heun-50, 100 NFE
Prior-conditioned DiT	Full motion; Flow Matching	Condition	GNN + Heun-50, 100 NFE
Prior-residual DiT (Balanced)	Residual motion; Flow Matching	Condition + reference	GNN + Heun-50, 100 NFE
Prior-residual DiT (MinMSE)	Residual motion; Flow Matching	Condition + reference	GNN + Heun-50, 100 NFE
MeshPriorDiT (ours)	Residual motion; direct MSE	Condition + reference	GNN + one DiT evaluation

Grasp handling. UniClothDiff-Dynamics uses a mask input without hard output projection. PGND forces either the selected point or particles within the gripper neighborhood, as named above. GNN-Only and all internal DiT settings apply hard grasp projection.

Table 3: **SoftGym H15 results on all 3,106 TEST trajectories.** Entries average all 15 rollout steps. Global, Active, and Fold MSE are in units of $10^{-5}\,\mathrm{m}^2$; dimensionless Edge-strain MSE is in units of 10^{-3}. Lower is better; **bold** and underlined mark the best and second-best results, respectively, among the nine settings.

Setting	Global	Active	Fold	Edge-strain
External baselines				
UniClothDiff-Dynamics (repr.)	14.898	40.901	34.443	65.729
PGND (point grasp)	4.283	11.535	11.132	14.846
PGND (radius grasp)	5.867	17.293	16.251	52.254
Shared component control				
GNN-Only	7.758	16.395	10.815	5.675
Flow-based internal comparisons				
DiT-Only	4.762	13.997	13.229	138.173
Prior-conditioned DiT	6.680	21.625	18.971	179.709
Prior-residual DiT (Balanced)	3.756	9.304	7.499	5.637
Prior-residual DiT (MinMSE)	3.045	8.219	7.677	7.712
Primary method				
MeshPriorDiT (ours)	**1.689**	**4.863**	**4.604**	**4.951**

D.2 Per-Step Rollout Behavior

Table 4 gives selected step-wise Global MSE values from the same matched 3,106-trajectory H15 evaluation. Unlike Table 3, each entry uses only the indicated step rather than averaging all 15 steps. MeshPriorDiT has the lowest error at each reported step, including the later blocks that depend on predicted history.

Table 4: **Step-wise SoftGym H15 Global MSE.** All methods use the same 3,106 TEST trajectories. Values are in units of $10^{-5}\,\mathrm{m}^2$; lower is better. **Bold** and underlined mark the best and second-best results at each step among the four methods shown.

Setting	$t{+}1$	$t{+}5$	$t{+}10$	$t{+}15$
GNN-Only	0.010	4.170	10.830	14.575
PGND (point grasp)	0.101	2.118	5.584	8.675
Prior-residual DiT (MinMSE)	0.011	1.211	4.218	6.578
MeshPriorDiT (ours)	**0.005**	**0.592**	**2.262**	**4.021**

E Flow Ablations and the Prior–Residual Design

The design question is whether the mesh rollout is more useful as an explicit, correctable reference than as an input condition alone. Within a Flow Matching formulation, we evaluate three ways to use the same cloth-prediction interface: DiT-Only predicts full future displacements without a prior; Prior-conditioned DiT receives the mesh rollout but still predicts full displacements; Prior-residual DiT predicts displacements relative to the rollout and composes the correction with it. All three use the same full-TEST H15 trajectories, action commands, metric definitions, and Heun-50 inference schedule.

The corresponding Global MSEs are 4.762, 6.680, and 3.045, respectively, in units of $10^{-5}\,\mathrm{m}^2$; the last value is the validation-selected MinMSE operating point. Thus, the tested Prior-residual DiT has 36.07% lower Global MSE than DiT-Only and 54.42% lower Global MSE than Prior-conditioned DiT. Merely supplying the prior as context does not produce the same benefit in these controls. This is independent, within-Flow evidence for treating the graph rollout as a prediction reference that can be corrected explicitly.

The main deterministic model provides complementary evidence: MeshPriorDiT lowers Global MSE by 78.23% relative to its *identical trained* GNN-Only prior and reaches $1.689 \times 10^{-5}\,\mathrm{m}^2$, below every Flow-based setting. Together, these results support the utility of the prior–residual interface under both direct regression and Flow-based prediction.

They support the complete design choice, not a single-factor causal claim about the residual target or the deterministic training objective.

F Alternative Residual Formulation and Decoder Sensitivity

F.1 Conditional Residual Flow

The Flow-based Prior-residual DiT is an alternative residual predictor, not the primary MeshPriorDiT training procedure. Its Balanced and MinMSE settings share one trained checkpoint and differ only in validation-selected inference parameters. Let $\overline{\mathbf{R}}_t^*$ denote the normalized free-vertex residual target defined in the Method section. We sample $\boldsymbol{\epsilon}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\tau \sim \mathcal{U}(0,1)$, and construct the linear path

$$\begin{aligned} \boldsymbol{\epsilon} &= \mathbf{F}_H \odot \boldsymbol{\epsilon}_0, \\ \mathbf{X}_\tau &= (1-\tau)\boldsymbol{\epsilon} + \tau \overline{\mathbf{R}}_t^*, \\ \mathbf{u}_\tau^* &= \overline{\mathbf{R}}_t^* - \boldsymbol{\epsilon}. \end{aligned} \tag{1}$$

Conditioned on historical states, planned actions, prior displacements, and the grasp mask, the DiT predicts a velocity field. Its effective output is restricted to free vertices:

$$\widehat{\mathbf{v}}_{t,\tau} = \mathbf{F}_H \odot \mathbf{v}_\phi\Big(\mathbf{X}_\tau, \tau \mid \mathbf{Q}_t^-, \mathbf{A}_t, \widehat{\Delta\mathbf{Q}}_t^{\mathrm{G}}, \mathbf{M}\Big). \tag{2}$$

Writing $N_{\mathrm{F}} = \sum_i F_i$, the training objective is

$$\mathcal{L}_{\mathrm{flow}}(\phi) = \mathbb{E}_{t,\tau,\boldsymbol{\epsilon}_0}\left[\frac{\|\widehat{\mathbf{v}}_{t,\tau} - \mathbf{u}_\tau^*\|_F^2}{3HN_{\mathrm{F}}}\right]. \tag{3}$$

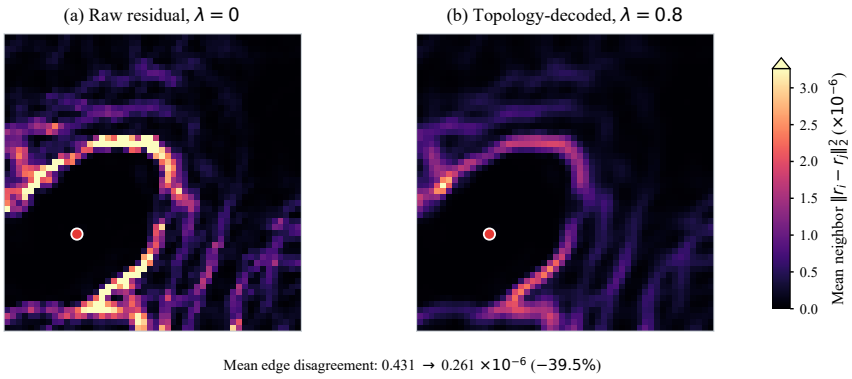

Figure 6: **Local consistency of position residuals after topology-aware decoding.** One selected SoftGym VALID Vertical Lift trajectory at $t+10$ is shown using the frozen deterministic checkpoint with $\gamma = 0.5$. The two panels compare the raw position residual ($\lambda = 0$) with the decoded residual ($\lambda = 0.8$) on a shared color scale. Color gives mean squared material-neighbor disagreement at each vertex; the red point marks the grasped vertex. Mean adjacent-vertex residual disagreement falls by 39.5% in this example. This is a single-case residual diagnostic, not Edge-strain MSE or an aggregate TEST result.

At inference, integration starts from masked Gaussian noise and solves

$$\frac{\mathrm{d}\mathbf{X}_\tau}{\mathrm{d}\tau} = \widehat{\mathbf{v}}_{t,\tau}, \qquad \widehat{\mathbf{R}}_t^{\mathrm{phy}} = \mathbf{S} \odot \mathbf{X}_1. \tag{4}$$

The noise, intermediate states, and velocity are projected onto free vertices. The evaluated configuration uses 50 second-order Heun steps, or 100 velocity-network evaluations per five-step window. Residual gain, topology-aware decoding, and grasp constraints are applied during composition. This Flow formulation describes an alternative experiment, not the deterministic training objective of the primary MeshPriorDiT model (Lipman et al., 2023).

F.2 Residual Gain and Topology-Aware Decoding

We examine the residual gain γ and topology-smoothing coefficient λ using one frozen deterministic Residual DiT checkpoint. The operating point $(\gamma, \lambda) = (1.0, 0.8)$ was selected on validation data before TEST evaluation. The diagnostic grid contains $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$ and $\lambda \in \{0, 0.2, 0.35, 0.5, 0.65, 0.8, 1\}$. All 35 settings were subsequently evaluated on the same 3,106 SoftGym TEST trajectories.

The full-TEST macro trends appear in the main paper's Figure 5. Setting $\gamma = 0$ removes the residual, making λ immaterial. At $\gamma = 1$, removing smoothing yields a slightly lower TEST macro Global MSE than the frozen setting (1.819 versus 1.882, in units of 10^{-5}), but a higher Edge-strain MSE (4.719 versus 4.257, in units of 10^{-3}). Reducing the gain to $\gamma = 0.5$ further lowers Edge-strain MSE at $\lambda = 0.8$ while increasing Global MSE. These observations characterize an accuracy–structure trade-off; they do not justify reselecting parameters on TEST data. These equal-action macro averages are distinct from the trajectory-weighted main comparison.

To visualize the decoder's local effect, we measure the disagreement of a position-residual field $\mathbf{e}$ over material neighbors. For vertex i with material neighborhood $\mathcal{N}(i)$, the heat-map value is

$$c_i(\mathbf{e}) = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \|\mathbf{e}_i - \mathbf{e}_j\|_2^2.$$

The summary value shown in the figure averages the same squared differences once per undirected material edge.

In Figure 6, mean undirected-edge disagreement decreases from 0.431 to 0.261 in units of $10^{-6}\,\mathrm{m}^2$, a 39.5% reduction for this example. Its illustrative $\gamma = 0.5$ setting is separate from the validation-selected $(1.0, 0.8)$ operating point used in the main comparisons.

Table 5: **Genesis H15 TEST results.** Global MSE is in units of $10^{-5}\,\mathrm{m}^2$; lower is better. Methods within each task use identical trajectories. Bold and underlined values mark the lower and higher result within each task, respectively.

Task	Traj.	Method	Global MSE
Overall	3,106	GNN-Only	<u>24.963</u>
		MeshPriorDiT	**4.461**
Corner-to-Side Fold	795	GNN-Only	<u>8.880</u>
		MeshPriorDiT	**0.471**
Diagonal Fold	795	GNN-Only	<u>31.294</u>
		MeshPriorDiT	**11.763**
Vertical Lift	1,516	GNN-Only	<u>30.077</u>
		MeshPriorDiT	**2.725**

G Genesis Results

Genesis is a second, independently constructed action-conditioned cloth dataset. Its mesh prior and Residual DiT are trained separately from the SoftGym models; the evaluation is not zero-shot transfer between simulators. The Genesis training split contains 300,000 five-step windows from 18,616 trajectories, while validation and TEST each contain 50,000 windows. The frozen H15 TEST evaluation uses 3,106 trajectories without teacher forcing.

Genesis also differs in its action boundary. It has two grasp groups and two planned hand motions, with commanded positions specified for the corresponding controlled vertices. Thus, the single-grasp SoftGym action equation is not reused verbatim for Genesis. The Residual DiT checkpoint was selected using Genesis validation data before TEST evaluation.

Table 5 reports the H15 comparison with the Genesis GNN-Only prior. Both methods use the same 3,106 TEST trajectories. MeshPriorDiT reduces rollout-averaged Global MSE by 82.13% overall. A paired trajectory-cluster bootstrap with 2,000 resamples gives a 95% confidence interval of 80.60–83.68% for the relative reduction. The Global MSE reduction is present in all three task groups: 94.70% on Corner-to-Side Fold, 62.41% on Diagonal Fold, and 90.94% on Vertical Lift.

H Computational Cost

H.1 Inference Schedules

The primary MeshPriorDiT performs five online mesh-prior updates and one deterministic DiT evaluation per five-step window. GNN-Only uses the same prior updates without a DiT correction. Each Flow-based DiT setting uses Heun-50, or 100 velocity-network evaluations per five-step window. The reproduced UniClothDiff-Dynamics baseline retains its 1,000-step DDPM sampling schedule. Across three H15 windows these correspond to 3, 300, and 3,000 DiT evaluations, respectively; prior-based settings also perform 15 online GNN updates. Network-call counts alone do not measure wall-clock cost.

H.2 Nine-Setting End-to-End H15 Timing

Table 6 joins two distinct measurements: rollout-averaged Global MSE on the full SoftGym TEST set and inference latency on one preselected SoftGym VALID trajectory. The batch-one timing uses a dedicated NVIDIA A100-SXM4-80GB, with each frozen setting measured serially in a separate process. The same three observed frames and 15 planned displacements are supplied to every setting. Inputs and checkpoints are preloaded on the GPU; after 10 warm-up invocations, 30 complete H15 predictions are timed with CUDA synchronization.

Each timed path includes online prior computation where applicable, model inference, grasp handling, topology-aware decoding where applicable, and history feedback across H15; it excludes model loading, disk I/O, metric computation, and visualization. PGND retains its direct H15 inter-

Table 6: **SoftGym H15 accuracy and inference cost across nine settings.** TEST Global MSE averages all 3,106 trajectories and is in units of $10^{-5}\,\mathrm{m}^2$ (lower is better). VALID latency is measured on one preselected trajectory; median and P95 are over 30 batch-one runs after 10 warmups on one A100-SXM4-80GB. DiT NFE counts network evaluations over all 15 steps, excluding online GNN updates; a dash denotes a method without a DiT. Bold and underlined values mark the lowest and second-lowest entries in each accuracy or latency column.

Setting	TEST Global MSE	VALID H15 (s) Median	VALID H15 (s) P95	DiT NFE /H15
UniClothDiff-Dynamics (repr.)	14.898	61.348	62.747	3,000
PGND (point grasp)	4.283	<u>0.145</u>	<u>0.163</u>	–
PGND (radius grasp)	5.867	0.149	0.167	–
GNN-Only	7.758	**0.127**	**0.127**	0
DiT-Only	4.762	5.967	6.050	300
Prior-conditioned DiT	6.680	7.265	7.373	300
Prior-residual DiT (Balanced)	3.756	7.235	7.310	300
Prior-residual DiT (MinMSE)	<u>3.045</u>	7.380	7.491	300
MeshPriorDiT (ours)	**1.689**	0.196	0.201	3

face rather than the three five-step DiT windows. On this VALID timing input, MeshPriorDiT is 30.4–37.7 times faster than the four Flow-based DiT settings and 313.1 times faster than the reproduced DDPM setting, while having the lowest full-TEST Global MSE. GNN-Only and both PGND settings are faster than MeshPriorDiT but less accurate on the full TEST set. These cross-method times describe the evaluated systems, whose architectures and rollout interfaces differ. Peak Torch-allocated memory is approximately 2.25 GiB for MeshPriorDiT and Prior-residual DiT (MinMSE); PGND's Warp allocations are not covered by that memory measurement.

I Qualitative Rollouts

Figures 7 and 8 extend the qualitative comparison to all three action families. Within each selected trajectory, the observed state and later predictions are aligned across ground truth, GNN-Only, and deterministic MeshPriorDiT. The two models were trained separately for their respective simulators.

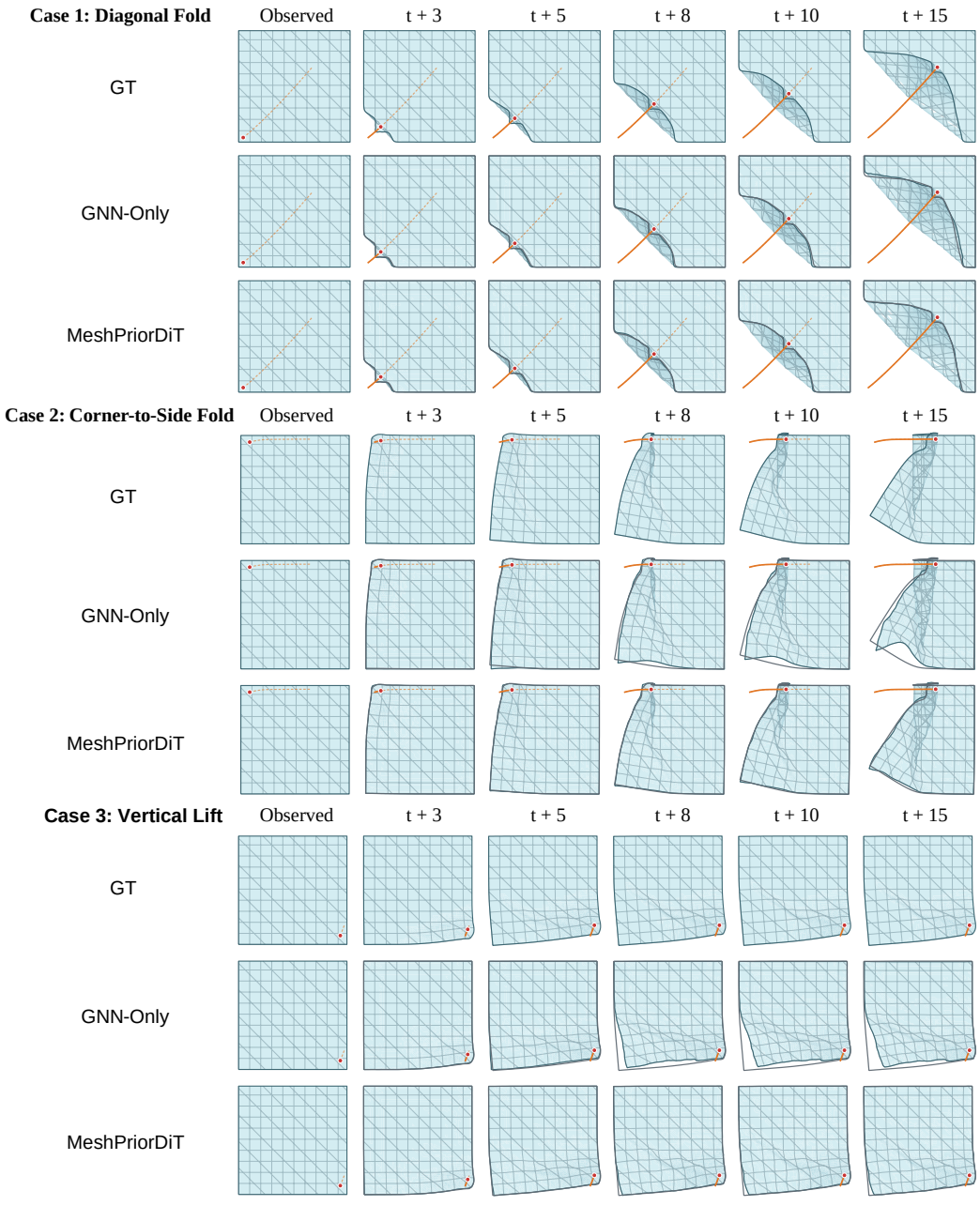

Figure 7: **Selected SoftGym cloth rollouts.** Diagonal Fold, Corner-to-Side Fold, and Vertical Lift each show a matched TEST trajectory at the last observed state and at $t+3, 5, 8, 10, 15$. Rows compare ground truth, GNN-Only, and deterministic MeshPriorDiT. Orange traces indicate commanded grasp motion; gray outlines in prediction rows indicate the corresponding ground-truth boundary.

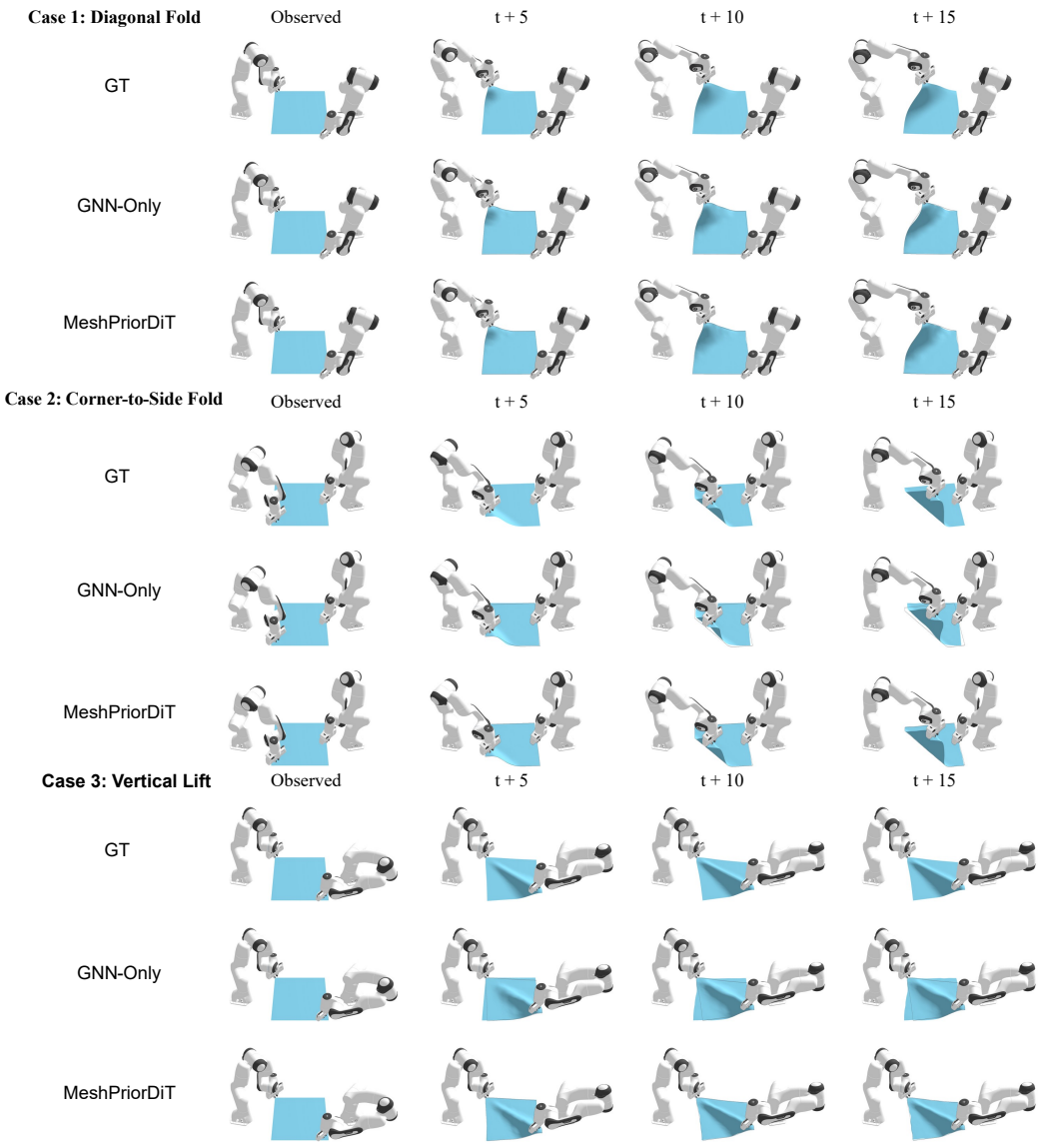

Figure 8: **Selected Genesis cloth rollouts.** Diagonal Fold, Corner-to-Side Fold, and Vertical Lift each show a matched VALID trajectory at the last observed state and at $t + 5, 10, 15$. Ground truth, GNN-Only, and the separately trained deterministic MeshPriorDiT use synchronized robot poses and a fixed camera within each action. Gray outlines in prediction rows indicate the corresponding ground-truth boundary. These illustrative examples are not a zero-shot transfer experiment or aggregate TEST result.