Super-Resolution Cloth Animation with Spatial and Temporal Coherence

JIAWANG YU, Style3D Research, China and Zhejiang University, China
ZHENDONG WANG*, Style3D Research, China

(a) Loungewear (b) Overcoat (c) Tea Dress (d) Ruffled Dress (e) Dress

Fig. 1. In these examples, our super-resolution method significantly enhances mesh quality. From the initial coarse resolution of 20mm, it advances to a finer resolution of 2.5mm, marking an 8× improvement in resolution. The resulting super-resolution meshes exhibit smoothness in wrinkle areas and closely resembles the complex wrinkle patterns seen in the high-resolution simulation ground truth.

Creating super-resolution cloth animations, which refine coarse cloth meshes with fine wrinkle details, faces challenges in preserving spatial consistency and temporal coherence across frames. In this paper, we introduce a general framework to address these issues, leveraging two core modules. The first module interleaves a simulator and a corrector. The simulator handles cloth dynamics, while the corrector rectifies differences in low-frequency features across various resolutions. This interleaving ensures prompt correction of spatial errors from the coarse simulation, effectively preventing their temporal propagation. The second module performs mesh-based super-resolution for detailed wrinkle enhancements. We decompose garment meshes into overlapping patches for adaptability to various styles and geometric continuity. Our method achieves an 8× improvement in resolution for cloth animations. We showcase the effectiveness of our method through diverse animation examples, including simple cloth pieces and intricate garments.

CCS Concepts: • **Computing methodologies → Physical Simulation**; **Supervised Learning**.

Additional Key Words and Phrases: Cloth animation, wrinkle enhancement, super-resolution, temporal coherence

*Corresponding author.

Authors' addresses: Jiawang Yu, Style3D Research, China and Zhejiang University, China, j.w.yu@hotmail.com; Zhendong Wang, Style3D Research, China, wang.zhendong.619@gmail.com.

0730-0301/2024/7-ART1 $15.00
https://doi.org/10.1145/3658143

ACM Reference Format:
Jiawang Yu and Zhendong Wang. 2024. Super-Resolution Cloth Animation with Spatial and Temporal Coherence. *ACM Trans. Graph.* 43, 4, Article 1 (July 2024), 14 pages. https://doi.org/10.1145/3658143

1 INTRODUCTION

Physics-based cloth simulation is vital for digital garment design where balancing the need for accurate wrinkles and efficiency in interactive applications is crucial. Since coarse meshes capture low-frequency features effectively but struggle with fine wrinkles, combining efficient methods for high-frequency feature generation suggests a solution: super-resolution for cloth animation, i.e. simulating low-frequency features with coarse meshes and complement them with generated high-frequency details on fine meshes. However, adding fine details to a coarse cloth simulation, poses significant challenges, mainly due to inadequate spatial and temporal coherence across various resolutions. Previous methods for quasi-static wrinkle generation fall short as they overlook the significant differences in low-frequency features across resolutions due to discretization errors and the locking issue with linear finite elements.

In this paper, we present a general framework for spatially and temporally coherent super-solution cloth animations with intricate wrinkles. The framework consists of two basic modules, three functional components, and a crucial low-dimensional intermediate variable. The variable links the two modules and is important in generating high-frequency details by encapsulating low-frequency features. The first module integrates two functional components: a simulator and a corrector. The simulator produces coarse mesh sequences that might diverge from corresponding ground-truth fine sequences in terms of low-frequency features. Hence, the corrector

plays a pivotal role in maintaining spatial coherence by rectifying biases among these features across different resolutions. To ensure temporal coherence, we interleave the two components. The simulator advances at every frame from the corrected result rather than the previous simulated coarse mesh, enabling immediate rectification of simulator-generated errors and minimizing error accumulation throughout the animation sequence. The core of the second module is the third functional component, focusing on the mesh-based super-resolution (MSR) task: recovering fine meshes with spatially coherent high-frequency details from the low-dimensional variables. Additionally, we propose an additional initialization strategy to improve temporal coherence in the generated details.

In our framework, the representation of low-frequency features offers flexibility; it can be realized through a low-dimensional subspace or an explicit coarse mesh. Additionally, the three components of our framework offer various options for implementation. Specifically, we explicitly represent low-frequency features using a coarse mesh as the low-dimensional variable, ensuring consistency between the input and output of the corrector. Our corrector is a graph neural network (GNN), while the core of our MSR task is an image-based super-resolution (ISR) neural network. Moreover, we enhance our MSR component's adaptability to various garment styles by decomposing garment meshes into patches, ensuring geometry continuity and overall smoothness through overlapping adjacent patches. The first module sets our method apart from previous methods as it rectifies discrepancies among low-frequency features across various resolutions. Our second module differs by using mesh positions directly instead of normal maps for fine wrinkle generation, enabling the ISR neural network to perceive both in-plane deformations and out-of-plane curvatures. As a result, complex habilimentations like gatherings can be properly handled. We summarize our main contributions as:

- we propose a general framework for super-resolution cloth animations by interleaving a simulator and a corrector to align low-frequency features across different resolutions, ensuring spatial and temporal coherence of generated wrinkles from the MSR module;
- we utilize a GNN to prevent the simulation error from prolongation and an ISR neural network to generate high-quality wrinkle details on triangular garment meshes;
- we introduce an initialization strategy for the ISR neural network to improve the temporal coherence of generated wrinkles and utilize patches to enhance ISR's adaptability to diverse garment styles and implement overlapping between adjacent patches to ensure continuity and smoothness.

2 RELATED WORK

The last three decades have seen a great progress in physics-based cloth simulations from the pioneer works [Baraff and Witkin 1998; Choi and Ko 2002] on implicit integration for linearized simulation to Projective Dynamics [Bouaziz et al. 2014] and the descent methods [Lan et al. 2023; Wang and Yang 2016] for nonlinear simulation.

High-resolution simulation methods. To achieve detailed folds and wrinkles in garment simulation, high-resolution cloth simulation is essential. Numerous approaches have enhanced its performance: multi-resolution techniques [Kircher and Garland 2005; Lee et al. 2010; Müller 2008; Wang et al. 2010b], including multigrid methods (geometric [Liu et al. 2021b; Wang et al. 2018; Wiersma et al. 2023; Xian et al. 2019] or algebraic [Shao et al. 2022; Tamstorf et al. 2015]), multilevel methods [Chen et al. 2021b; Wu et al. 2022], and domain decomposition methods [Li et al. 2019] have been instrumental. Moreover, parallel algorithms [Lan et al. 2022; Li et al. 2020] tailored for GPUs exploit their computing power, pushing the efficiency boundaries of high-resolution simulation. However, instability and complexity with hierarchies make them impractical for complex garments. Direct high-quality clothing simulation on such meshes is computationally intensive and falls short for real-time applications.

Optimization-based and data-driven methods. Studies have shown that separating wrinkle formation from the overall clothing motion doesn't cause visual artifacts. Optimization-based techniques [Chen et al. 2023b, 2021a, 2023a; Gillette et al. 2015; Kim and Farbiz 2013; Müller and Chentanez 2010; Rohmer et al. 2010; Wang 2021; Zhang et al. 2022b] rely on simulation but may struggle to replicate wrinkle patterns seen in direct high-resolution simulations. Data-driven methods [Wang et al. 2010a; Zurdo et al. 2013] use high-resolution wrinkle data for runtime synthesis but lack a direct connection to the ground truth fine simulations, posing challenges in reproducing temporally coherent wrinkle patterns. Some argue that clothing's appearance is captured through few degrees of freedom [de Aguiar et al. 2010; Feng et al. 2010; Li et al. 2018], while others model motion spaces with body pose, low-dimensional, and local movements [Guan et al. 2012]. Data-driven transformations using rotation-invariant cloth quantities have been explored [Feng et al. 2010].

Learning-based methods. Learning-based methods offer an efficient trade-off between precomputation and runtime performance, making them appealing for interactive applications. An early attempt involved learning linear upsampling operators to enhance details for coarse cloth simulation [Kavan et al. 2011]. Zhang et al. [2021b] learned dynamic features for rendering cloth appearance changes like folds and wrinkles. Oh et al. [2018] proposed to generated wrinkle details through DNN models across multiple levels to enrich coarse-level physics. Global shape deformations, representing low-frequency motions in clothing, can be inferred from learned subspace motion models [Lahner et al. 2018] or human body parametric motion models [Pan et al. 2022]. The method proposed by Zhang et al. [2022a] learns generative spaces for garment geometries, predicting per-frame local displacements with collision handling. On a similar note, Zhao et al. [2023] deform garment meshes using rigid transformations and nonlinear displacements, maintaining position and direction consistency. Several deep learning methods for garment animations, based on physics-based loss schemes, have been proposed, including self-supervised approaches [Santesteban et al. 2022, 2021] and unsupervised methods [Bertiche et al. 2022]. MESHGRAPHNETS [Pfaff et al. 2020] employ graph neural networks (GNNs) for adaptable mesh-based simulations, allowing dynamic mesh discretization and scalability across complex state spaces. Alet et al. [2019] explore GNNs for spatial processes without predefined structures. Learning-based methods [Han et al. 2018; Ma et al. 2023]

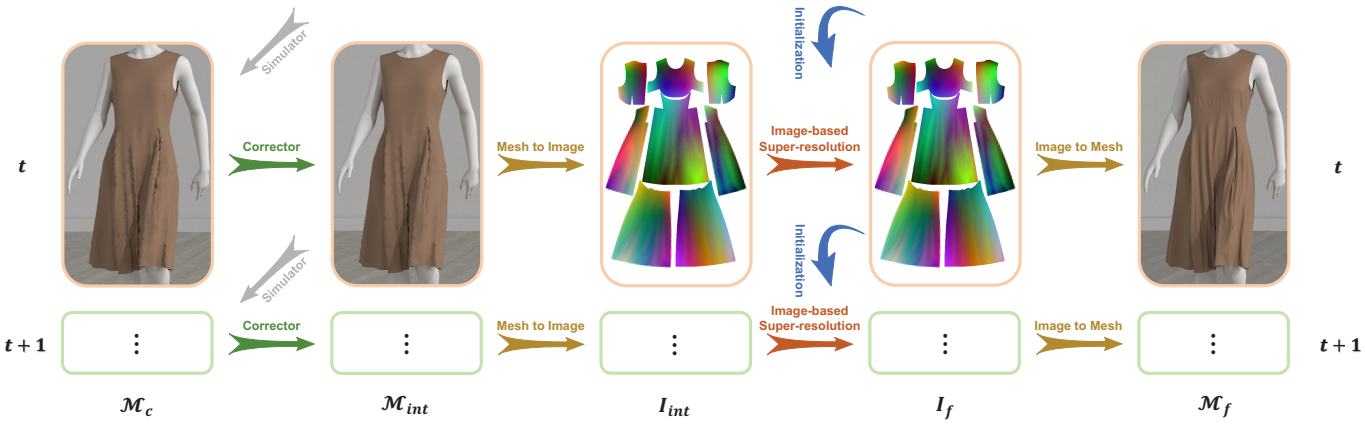


Fig. 2. Our method comprises five nodes within the data stream. A coarse mesh $\mathcal{M}_c$ serves as the input for the corrector to rectify low-frequency features, producing another coarse mesh, $\mathcal{M}_{int}$, as a superior input for the mesh-based super-resolution task. Transitioning to an image-based super-resolution network, $\mathcal{M}_{int}$ is transformed into a high-resolution image, $\mathcal{I}_{int}$, which in turn generates an output image, $\mathcal{I}_f$, of the same size but with more details. Subsequently, we reconstruct the targeted fine mesh, $\mathcal{M}_f$, from $\mathcal{I}_f$ by employing an image-to-mesh transfer. To enhance temporal coherence, we advance the simulation forward a timestep, deriving $\mathcal{M}_c^{t+1}$ from $\mathcal{M}_{int}^t$ instead of $\mathcal{M}_c^t$. For improved efficacy of the image-based SR network, we utilize $\mathcal{I}_f^t$ as an initialization for $\mathcal{I}_f^{t+1}$.

have also been proposed to address PDE dynamics through implicit numerical time integration for physics-based simulation.

Recently, Zhang et al. [2021a] employed an ISR neural network to improve the quality of normal maps, recovering intricate wrinkle details on fine garment meshes. Our super-resolution method for cloth animations also builds upon the advancements in ISR from the computer vision community. We leverage the state-of-the-art ISR method known as SwinIR [Liang et al. 2021], which utilizes transformers with shifted windows [Liu et al. 2021a]. SwinIR comprises shallow and deep feature extractions, akin to the low-frequency and high-frequency features in cloth animations. As our method doesn't primarily focus on ISR, those interested in more literature on image super-resolution can refer to comprehensive surveys [Anwar et al. 2020; Liu et al. 2022; Wang et al. 2020] on this subject.

3 METHODOLOGY

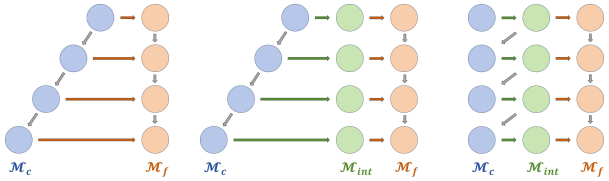
The motivation driving our method stems from a crucial observation: a cloth simulator effectively captures low-frequency global motions using a coarse mesh, while high-frequency local wrinkles can be separately generated and incorporated into this coarse simulation mesh. In our pursuit of enriching coarse garments with intricate wrinkle, we aim for high spatial and temporal coherence within the generated fine meshes. In the domain of mesh-based super-resolution (MSR) in computer graphics, previous methods [Lahner et al. 2018; Zhang et al. 2021a] usually take a sequence of coarse meshes as input and produces a sequence of fine meshes embedded with realistic wrinkle details. However, we notice that the divergence of low-frequency features across different resolutions in cloth simulations undermines the effectiveness of a MSR task, as illustrated in Fig. 3. To enhance the spatial and temporal coherence of the generated fine meshes with high-quality wrinkles, we introduce a novel framework.

As illustrated in Fig. 2, the framework of our method consists of two modules: one interleaving a simulator and a corrector to generate a coarse mesh sequence, and an MSR module focused on refining mesh details. Specifically, we use a coarse mesh as the low-dimensional variable representing low-frequency features, employ a graph neural network as the corrector, and utilize an ISR neural network in our MSR module. Considering the frame t, the goal of our method is to transform the coarse garment mesh, $\mathcal{M}_c^t$, into a super-resolution fine mesh, $\mathcal{M}_f^t$. To achieve spatial coherence, we first rectify $\mathcal{M}_c^t$ to an intermediate coarse mesh, $\mathcal{M}_{int}^t$, which aligns with $\mathcal{M}_f^t$ in terms of low-frequency features. In this correction process, we utilize a GNN to build a mapping from $\mathcal{M}_c^t$ to $\mathcal{M}_{int}^t$. Given the suitability of 2D uniform grids resolution manipulation over 3D triangular meshes, we employ an ISR neural network for our MSR task. This requires a mesh-to-image operator, transforming $\mathcal{M}_{int}^t$ into an image $\mathcal{I}_{int}^t$ as the input for the ISR neural network to generate an output super-resolution image $\mathcal{I}_{sp}^t$ with enhanced details. Then, employing an image-to-mesh operator, we obtain the final high-resolution mesh $\mathcal{M}_f^t$ from $\mathcal{I}_{sp}^t$, showcasing intricate folds and wrinkles in three-dimensional space. The next challenge involves determining the suitable input, $\mathcal{M}_c^{t+1}$, to produce $\mathcal{M}_f^{t+1}$ and progress the animation. Initially, the instinct is to advance $\mathcal{M}_c^t$ forward by a time step using the simulator to obtain $\mathcal{M}_c^{t+1}$. However, we anticipate that independently advancing coarse and fine sequences will diverge in terms of global motions. To address this, we propose a more effective strategy: advancing $\mathcal{M}_{int}^t$ forward by a time step to obtain $\mathcal{M}_c^{t+1}$ using the simulator. Consequently, in our approach, the simulator and the corrector are interleaved to generate the coarse mesh sequence, significantly improving the temporal coherence of the input mesh sequence for the MSR module.

Overall, our method ensure the spatial and temporal coherence of super-resolution clothing animations: the first module aligns low-frequency features in coarse mesh sequences, while the second module focuses on generating high-quality fine mesh sequences with temporally coherent intricate wrinkles.

(a) Coarse simulation (b) Our result (c) Fine simulation

Fig. 3. Rabbit. A large piece of cloth falls onto a rabbit, resulting in numerous intricate wrinkles. Due to the limitation of a coarse discretization, the wrinkles generated by the coarse simulation (a) appear coarse-grained and notably deviate from the fine-grained wrinkles observed in the ground truth fine simulation (c). In contrast, our result (b) adeptly replicates wrinkles resembling those of the ground truth simulation.



(a) Previous method (b) No interleaving (c) Interleaving

Fig. 4. Various temporal models exist for super-resolution cloth animations. The gray arrow symbolizes the simulator, the green arrow for the corrector, and the orange arrow for the mesh-based super-resolution module.

3.1 Coarse Mesh Sequence Generator

Our first module generates inputs for the MSR task, ensuring spatial and temporal coherence regarding low-frequency features.

3.1.1 The simulator. For driving cloth animation dynamics, utilizing a simulator—be it physics-based or neural network-based—is a straightforward and highly effective method. For improved collision handling and enhanced generality, we opt for a physics-based simulator. Initially, the instinct for preparing input for the MSR task was to employ the simulator for generating a sequence of coarse garment meshes, i.e. the model (a) in Fig. 4. However, as showcased in Fig. 5, despite variations in detailed wrinkles, a noticeable discrepancy exists in the global shape between the coarse and fine meshes. This discrepancy has also been discussed in [Wang et al. 2010b], indicating different strain limiting properties of finite elements at various resolution levels. Such differences, rooted in discretization errors and the locking issue inherent in linear finite elements in coarse meshes, result in notable variations in low-frequency features across different simulation resolutions. Consequently, this divergence significantly impacts the performance of the MSR module, highlighting the inadvisability of establishing a direct link between coarse and fine simulation results.

This problem has also been noticed in [Zhang et al. 2021a], where they aimed to establish connections between fine meshes and their corresponding downsampled coarse versions using a super-resolution neural network. We agree with this strategy as the low-frequency features remain consistent between the fine mesh and its corresponding downsampled counterpart. Hence, we adopt the downsampled coarse mesh as the intermediate variable in our framework. However, Zhang et al. [2021a] insisted on directly using simulated coarse meshes as input for their super-resolution approach to generate wrinkle details on fine meshes, which led to diminish performance.

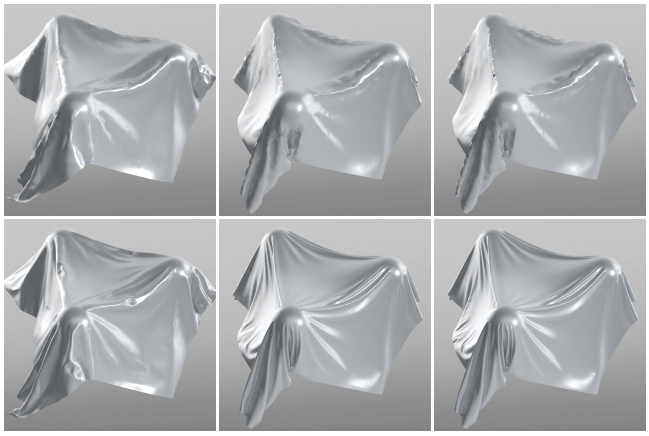
3.1.2 The corrector. To address differences in low-frequency features across resolutions, our approach involves a corrector. By utilizing a coarse mesh as the intermediate variable, the corrector minimizes the disparity between a simulated coarse mesh and the intermediate representation, generating a correction velocity as the output. Optimization-based methods often struggle due to the challenge in formulating an explicit optimization target for this task. Instead, we opt for a data-driven approach where a neural network learns this correction. Unlike cloth animation driven by body motions, this task is static. Although the input simulation-derived coarse mesh is free of penetrations and intersections, downsampling a fine mesh might lead to self-intersections in wrinkle areas. Thus, the output from the corrector might exhibits such issues, and we can omit collision handling in this phase.

Given the significance of mesh topology in addressing the locking issue on coarse meshes, we utilize a graph neural network as the corrector to obtain corrections for velocities, following an Encode-Process-Decode framework. We start by utilizing an encoder to transform the input coarse mesh $\mathcal{M}_c$ into a graph $G = (V, E)$, in which V represents the set of mesh vertices and E represents the set of mesh edges. Notably, unlike MESHGRAPHNET [Pfaff et al. 2020], self-collisions and external collisions are not considered in our case. We encode the relative displacement vectors in both 2D material and 3D world spaces into the mesh edges $e_{ij} \in E$, i.e. $u_{ij} = u_i - u_j$ and its norm $|u_{ij}|$, and x_{ij} and its norm $|x_{ij}|$ respectively. The remaining velocity features q_i is provided as node features in v_i. Subsequently, the processor iteratively update all edge e_{ij} and node v_i embeddings to e'_{ij}, v'_i respectively by

$$e'_{ij} \leftarrow f_e(e_{ij}, v_i, v_j), \quad v'_i \leftarrow f_v(v_i, \sum_j e'_{ij}),$$

where f_e and f_v are implemented using MLPs with a residual connection. Specifically, the processor consists of several identical message passing blocks, which generalize Graph-Net blocks [Sanchez-Gonzalez et al. 2018] to multiple edge sets. Each block has independent parameters and is applied in sequence to the output of the previous block. After the final processing step, the decoder extracts the target velocity correction for each node by using an MLP to transform the latent node features v_i into the output features p_i. Finally, the output mesh nodes V are updated using the corrected velocity $q'_i = q_i + p_i$ to produce $\mathcal{M}_{int}$. Our corresponding loss function quantifies the difference between the predicted correction p_i and the ground truth correction $\bar{p}_i$, i.e. $\mathcal{L}_{GNN} = |p_i - \bar{p}_i|$.

3.1.3 Temporal coherence. As the simulation advances forward, errors accumulate, contributing to a growing disparity in temporal



(a) Coarse simulation (b) Our result (c) Ground truth

Fig. 5. Comparison among different inputs in the first row for the mesh-based super-resolution module. (a) $\mathcal{M}_c$ generated by direct coarse simulation; (b) $\mathcal{M}_{int}$ generated by our method; (c) the ground truth $\mathcal{M}_{int}$. The second row are their corresponding super-resolution results except for (c) the ground truth $\mathcal{M}_f$.

consistency. This divergence causes the coarse animation sequence to progressively deviate from the ground-truth fine animation sequence. Initially, employing a corrector to address errors in each coarse frame appears effective. However, this accumulation results in increasingly divergent errors that the learning-based corrector struggles to rectify. As depicted in Figure 13, as errors accumulate, the corrector's ability to rectify them diminishes. Alternatively, generating accurate low-frequency features aligned with fine simulation results often requires approaches beyond physics-based simulations, relying on learning-based methods. While suitable for tight garments driven by body motions, this approach might not be ideal for general cloth animations like curtains or loose dresses. To resolve this issue, we propose interleaving the simulator and the corrector, as illustrated in Fig. 2. At frame t, our strategy involves advancing the coarse simulation to $\mathcal{M}_c^{t+1}$ from the intermediate rectified coarse mesh $\mathcal{M}_{int}^t$ instead of $\mathcal{M}_c^t$, enhancing the temporal accuracy of the coarse sequence. This approach effectively interrupts the error accumulation between $\mathcal{M}_c^t$ and $\mathcal{M}_c^{t+1}$ by using $\mathcal{M}_{int}^t$. Notably, this strategy aids in the training of the corrector as it focuses solely on the immediate error produced by the simulator rather than dealing with accumulated errors.

In the framework of our method, depicted in Fig. 2, the simulator models dynamics while the GNN corrector aligns low-frequency features between coarse and fine simulations. Indeed, the GNN corrector introduces a non-physical energy, which acts as supplementary compression energy and minimally affects dynamics but is crucial for addressing the locking issue—a challenge in computer graphics. This energy is dissipated by the simulator and re-adjusted by the GNN corrector. Our approach, mirroring Position-based Dynamics (PBD), uses this interplay to maintain system progression towards the target position with low-frequency features consistent with those in fine simulations.

3.2 Mesh-based Super-Resolution

Upon receiving an input coarse garment mesh $\mathcal{M}_c$, our second module, mesh-based super-resolution, is tasked with generating high-quality wrinkles on a fine mesh $\mathcal{M}_f$. To facilitate this task, the mesh topology of both meshes remain fixed throughout the entire process. Theoretically, our MSR module establishes a mapping from $\mathcal{M}_c$ to $\mathcal{M}_f$, an 8× improvement in resolution from 20mm to 2.5mm. Based on our experiments, we find that meshes at a resolution of 2.5mm offer sufficient degrees of freedom to simulate garment wrinkles, even in complex clothing configurations like gatherings, as demonstrated in Fig. 15. Furthermore, meshes with coarser resolutions than 20mm lack the necessary granularity to capture wrinkle details, preventing the induction of intricate features.

Our MSR module is based on ISR methodology. The challenge in 3D space for the MSR task lies in the complexity of surface mesh shapes, especially concerning out-of-plane curvatures. An effective approach involves encoding 3D mesh data into a 2D material space, facilitating more manageable downsampling and upsampling. Using a structured uniform grid for the material space proves advantageous over irregular, unstructured triangle meshes, allowing for streamlined utilization of established methods for ISR tasks.

3.2.1 Mesh-Image transfer. To facilitate input for ISR, our garment meshes are transformed into images. Given a coarse garment mesh $\mathcal{M}_c$, its corresponding image $\mathcal{I}_c$ serves as the input for the ISR module. The ISR generates an output $\mathcal{I}_f$, representing a fine garment mesh $\mathcal{M}_f$. In ISR, both input and output images have a high resolution, with a pixel size of 2.5mm, enough for capturing wrinkle details from fine garment meshes. In our mesh-to-image transformation, we establish a uniform background grid in the 2D material space, with each grid intersection representing a pixel of size 2.5mm. Pixel values are calculated by identifying the triangle that encloses the pixel, determining the value through linear interpolation. Conversely, in the image-to-mesh transformation, vertices within a mesh are positioned by identifying the grid area defined by four neighboring pixels, also utilizing linear interpolation to obtain vertex values.

Coordinate map. An image with three color channels can effectively represent a set of 3D points or vectors. The choice of mesh information encoding significantly influences this representation. While the conventional approach involves using a *normal map* [Lahner et al. 2018; Zhang et al. 2021a] that captures surface curvature to depict wrinkle areas, we propose an alternative: the *coordinate map* that directly encodes 3D vertex coordinates into the image [Gu et al. 2002]. Our preference against the normal map stems from the complexities involved in reconstructing mesh geometry from it, potentially leading to artifacts and requiring additional computation, as demonstrated in Fig. 16(b). The coordinate map empowers our MSR module to perceive both out-of-plane curvatures and in-plane strains, augmenting its effectiveness in handling garment crafting. Specifically, given $\mathbf{p}_{i,j}$ represents a pixel at the intersection of ith row and the jth column in the coordinate map, we can compute the deformation gradients using finite difference,

$$\mathbf{g}_{i,j}^u = (\mathbf{p}_{i+1,j} - \mathbf{p}_{i-1,j})/h,\ \mathbf{g}_{i,j}^v = (\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j-1})/h,$$

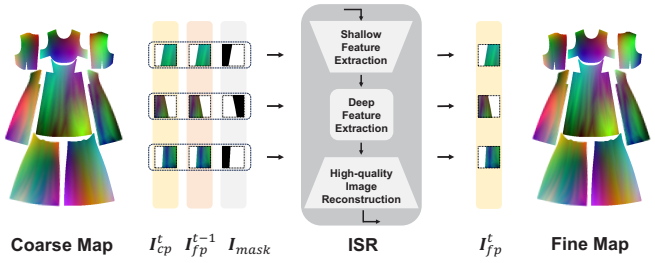


Fig. 6. A coarse map will be divided into patches before processed by the ISR neural network. The input consists of coarse patches from the current frame, $\mathcal{I}^t_{cp}$, fine initialization patches from the last frame, $\mathcal{I}^{t-1}_{fp}$, and mask patches. The output fine patches, $\mathcal{I}^t_{fp}$, reconstruct a fine map.

and Hessian,

$$\mathbf{H}^{uu}_{i,j} = (\mathbf{g}^u_{i+1,j} - \mathbf{g}^u_{i-1,j})/h, \quad \mathbf{H}^{uv}_{i,j} = (\mathbf{g}^u_{i,j+1} - \mathbf{g}^u_{i,j-1})/h,$$
$$\mathbf{H}^{vu}_{i,j} = (\mathbf{g}^v_{i+1,j} - \mathbf{g}^v_{i-1,j})/h, \quad \mathbf{H}^{vv}_{i,j} = (\mathbf{g}^v_{i,j+1} - \mathbf{g}^v_{i,j-1})/h,$$

in which h is twice the pixel size. Additionally, before calculating the pixel values in a coordinate map, we independently normalize the mesh's three dimensions—x, y, and z coordinates— to uniform distributions. This strategy ensures that the ISR neural network has equal sensitivity to changes across all three dimensions.

Boundary mask. In a single pattern image, certain areas may lack information. To address this concern, we employ a mask that labels whether a pixel contains useful information or not.

3.2.2 ISR neural network. Our MSR module is based on the advanced ISR method called SwinIR [Liang et al. 2021], which comprises three critical components: shallow feature extraction, deep feature extraction, and high-quality image reconstruction. In our MSR task, we equate shallow features and deep features in the image space to low-frequency and high-frequency features in the mesh space, respectively. Starting with a high-resolution image $\mathcal{I}_c$ containing coarse details, we derive latent shallow features $\mathcal{F}_s$ using convolution layers, represented as $\mathcal{F}_s = f(\mathcal{I}_c, \mathbf{p})$, where $\mathbf{p}$ denotes input parameters. Subsequently, latent deep features $\mathcal{F}_d$ are derived as a function of shallow features, expressed as $\mathcal{F}_d = g(\mathcal{F}_s, \mathbf{p})$. Finally, the integrated shallow features $\mathcal{F}_s$ and deep features $\mathcal{F}_d$ reconstruct the ultimate high-quality image, enriching it with finer details, denoted as $\mathcal{I}_f = h(\mathcal{F}_s, \mathcal{F}_d)$. Neural networks learn a mapping between $\mathcal{I}_c$ and $\mathcal{I}_f$ using these functions. For a detailed understanding of the ISR neural network SwinIR, please refer to [Liang et al. 2021]. The loss function is composed of two key terms. The first term, $\mathcal{L}_{dist} = \alpha|\mathcal{I}_f - \mathcal{I}_{gt}|$, measures the discrepancy between the predicted result, $\mathcal{I}_f$, and the ground truth, $\mathcal{I}_{gt}$, using a weighted norm. Additionally, we aim to constrain the pixel gradients of $\mathcal{I}_f$ and $\mathcal{I}_{gt}$, forming the second term of the loss function, $\mathcal{L}_{cons} = \beta|\mathcal{G}_f - \mathcal{G}_{gt}|$. Hence, the final loss function is $\mathcal{L}_{sp} = \mathcal{L}_{dist} + \mathcal{L}_{cons}$.

Initialization. Training a neural network involves solving a nonlinear optimization problem, where a well-chosen initialization often accelerates the optimization process significantly. Considering the continuous temporal and spatial motion of a mesh, the shape of the mesh in the last frame closely resembles that of the current frame.

Fig. 7. In this example of multi-layer cloth animation driven by a walking human, our super-resolution method is able to generate realistic high-quality wrinkles without notable penetrations and intersections.

As a result, the output from the previous frame serves as an effective initialization for predicting the current frame.

Overlapping patches. To enhance generalization across diverse 2D pattern parameterizations, our approach involves utilizing small patches as both input and output for the ISR neural network. Rather than encoding all patterns into a single composite image, each pattern is individually transferred to an image. Subsequently, each resulting image is divided regularly into multiple small image patches of equal size. The ISR neural network's output consists of image patches that can collectively form a complete image, as depicted by Fig. 6. As each patch undergoes independent processing by the ISR neural network, the shared boundary of neighboring patches may vary, potentially causing discontinuities. To alleviate this concern, adjacent patches overlap by two pixel lines, ensuring smoother transitions and addressing any potential discontinuities. During the reconstruction of the fine mesh from generated coordinate map patches, a vertex situated in the overlapped areas might possess multiple position coordinates. Instead of averaging different positions from distinct patches, we opt to average overlapped pixel values in image space to maintain smoothness. Specifically, pixel lines a, b, c are part of the left patch and pixel lines b, c, d belong to the right patch. After super-resolution processing, the pixel coordinates can be denoted as $\mathbf{x}_a, \mathbf{x}^l_b, \mathbf{x}^l_c$ for the left patch and $\mathbf{x}^r_b, \mathbf{x}^r_c, \mathbf{x}_d$ for the right patch. We noted that merely averaging the coordinates of overlapping pixels, such as $\hat{\mathbf{x}}_b = (\mathbf{x}^l_b + \mathbf{x}^r_b)/2$, results in non-smoothness between pixel lines a and b, and pixel lines c and d. To address this, we implement Laplacian smoothing on overlapping pixel lines. For instance, for pixel line b, the smoothing procedure computes the position as $\tilde{\mathbf{x}}_b = (\mathbf{x}_a + \hat{\mathbf{x}}_c)/2$. The final position for pixel line b is calculated as $\mathbf{x}_b = \sigma\tilde{\mathbf{x}}_b + (1-\sigma)\hat{\mathbf{x}}_b/2$, where $\sigma = 0.5$ serves as the weighting factor to balance the impact of Laplacian smoothing and the original averaging.

3.3 Networks Training

We require to train two neural networks in our method: a GNN as the corrector and an ISR neural network for the MSR task.

3.3.1 Data set construction. Given the 2D patterns of a garment, we discretize all patterns into triangle meshes at resolutions of 2.5mm and 20mm, respectively. In constructing our dataset, we utilize the simulator to generate ground truth mesh sequences at 2.5mm, subsequently downsampling them to 20mm to create the ground truth intermediate coarse mesh. In the training process for the ISR neural network, we convert both coarse and fine meshes into coordinate maps at a shared high resolution with a pixel size of 2.5mm. To interleave the simulator and the GNN corrector in the training, we advance each intermediate coarse mesh forward by one timestep using the simulator, resulting in a coarse simulation mesh.

3.3.2 GNN training. The architecture of our GNN corrector is similar to that of MESHGRAPHNET [Pfaff et al. 2020], consisting of an encoder, a processor and a decoder. We do not consider dynamic collision edges, and all the three components are for static topology edges. The MLPs of the three components are ReLU-activated two-hidden-layer MLPs with layer and output size of 64, except for the decoder whose output size matches the prediction. All MLPs outputs except the decoder are normalized by a LayerNorm. All input and target features are normalized to zero-mean, unit variance, using dataset statistics. Models are trained with the Adam optimizer for 400K training steps, using an exponential learning rate decay from 10^{-4} to 10^{-6} over 400K steps. We also use the conventional Graph Attention Networks [Brody et al. 2022; Veličković et al. 2017] to improve the performance of our GNN corrector.

Rollout training. For training, we only supervise on the velocity corrections. Even though the simulator is responsible for cloth dynamics, the GNN correcting can also leave errors. These errors can accumulate temporally and make a big difference. To make our model robust to rollout of hundreds of steps, we used the same training noise strategy as in GNS [Sanchez-Gonzalez et al. 2020]. However, the GNN cannot eliminate all errors. To enhance the accuracy of a long animation, we implement the window training strategy as detailed in *Deep Fluid* [Kim et al. 2019] for a second-phase training. Specifically, upon completing the first-phase naïve training, we select a random ground-truth intermediate coarse mesh at time t as the start point. We then proceed with a rollout of ω steps, employing the simulator-corrector interleaving strategy. The loss function for this rollout training is calculated using the formula $\mathcal{L}_{rollout} = \frac{1}{\omega}\sum_{i=t+1}^{t+\omega} |\mathbf{p}_i - \bar{\mathbf{p}}_i|$, which represents the averaged step-wise error. This metric is employed to update the configuration of the GNN corrector, which is essential for refining the network's capability to effectively reduce accumulated errors in animation sequences. The window rollout training process is iterative, continuing until the rollout loss function decreases below a specified threshold. As depicted in Fig.8, after approximately 1000 frames of rollout, our method with rollout training consistently produces coarse meshes that closely resemble the ground truth. These accurate coarse meshes are crucial inputs for the MSR module. In contrast, the inference result from the naïvely trained network in Fig.8(a) deviates significantly from the ground truth.

3.3.3 ISR training. We leverage SwinIR [Liang et al. 2021] as the ISR model for our MSR task. Our ISR training mirrors the approach employed in the training of SwinIR. Additionally, we incorporate an initialization strategy to enhance temporal coherence and improve training convergence. During training, we smooth the initialization data using a Gaussian kernel to prevent the accumulation of inappropriate mesh smoothing during inference, ensuring our trained ISR neural network can counteract excessive smoothing and avoid its temporal propagation. Additionally, we enhance the smoothed initialization data by introducing noise from a normal distribution with zero mean and a standard deviation of 0.1mm, thereby improving the robustness of our method against local discontinuities.

(a) Naïve training (b) Rollout training (c) Ground truth

Fig. 8. Rollout training is essential for the first module of our method, ensuring the temporal coherence of low-frequency features in the predicted coarse mesh sequences.

3.3.4 Patch-based training. We employ a patch-based training approach for both neural networks. In the case of a coarse mesh, we randomly extract overlapping sub-meshes for training the GNN. Simultaneously, for a coordinate map, we randomly sample overlapping image patches to train the ISR model. This patch training strategy directs our neural networks to emphasize local features, thereby enhancing their generalization to unseen data. During runtime inference, we input an entire coarse mesh into the GNN, whereas regularly arranged image patches are fed into the ISR network.

3.3.5 Training Performance. Although we have demonstrated in Fig. 19 that our method has the capability to enable one-time training to cover different kinds of garments with various fashion styles using the same fabric, it is important to note that we need to train independent networks for garments with different fabrics. In our method, both the GNN corrector and the ISR neural network were trained on a machine equipped with an Intel Core i9-13900K CPU, 64GB RAM memory, and a single NVIDIA RTX4090 GPU with 24GB memory. The naïve training of the GNN corrector for all the examples takes about 2 days on average. The rollout training also takes about 2 days on average to converge. The ISR neural network takes about 2 to 4 days to converge. To expedite the overall training process, we distribute the training of the two networks to two machines for independent parallel training. Consequently, it takes up to 4 days using two machines to train our method.

4 RESULTS

We showcase the effectiveness of our method through diverse cloth simulation scenarios, including single cloth piece animations on different static or dynamic supports, as illustrated in Fig. 3 and 5, and complex garment animations driven by walking or dancing humans, as illustrated in Fig. 1. In cases involving static collision objects,

(a) Loungewear (b) Overcoat (c) Tea Dress (d) Ruffled Dress

Fig. 9. Vertex-wise Euclidean distances to the ground truth have been visualized on our super-resolution garments. In the four examples, most areas are depicted in blue, indicating minimal errors. Although complex wrinkle areas in the Ruffled Dress and Tea Dress exhibit higher errors, they still showcase high-quality wrinkle details.

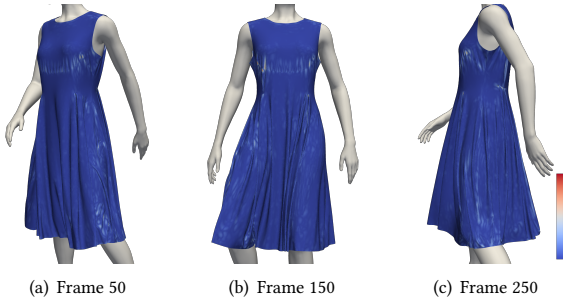
such as Three-Ball, Rabbit, and Table, we modify the initial states of the cloth pieces to generate distinct simulation data for inference. In scenarios featuring dynamic collision objects, like human body motions, the motions differ between the training and inference data.

4.1 Evaluation

We aim to evaluate the spatial and temporal coherence of super-resolution cloth animations generated by our method and the corresponding runtime performance.

4.1.1 Spatial coherence. We want to illustrate that our method adeptly generates super-resolution cloth meshes with wrinkle patterns closely mirroring those of the corresponding ground truth fine simulation meshes. As evidenced in other figures, including cloth examples in Fig. 3 and 5, and garment examples in Fig.15, 19, and 16, the shapes of our super-resolution garments closely resemble the ground truth visually. To quantitatively assess the disparity, we visualize vertex-wise Euclidean distances on various garment meshes using color maps. As illustrated in Fig. 9, various garments on different human bodies, each with distinct gestures, exhibit minimal errors when compared to the ground truths.

4.1.2 Temporal coherence. We also want to demonstrate that the cloth mesh sequences produced by our super-resolution method exhibit a similarity in temporal dynamics to the ground truth fine simulation mesh sequences. This encompasses not only capturing low-frequency global motions but also preserving high-frequency wrinkle details across successive frames, ensuring a temporal consistency that mirrors the characteristics of the ground truth. For visual demonstration, we highlight the differences between the SR meshes and the ground truth using the Dress example at frames 50, 150, and 250, respectively. As depicted in Fig. 10, the errors consistently remain minimal across a wide range of frames. For statistic demonstration, we illustrate the temporal errors of the Three-Ball example in Fig. 11. After about 1000 frames of rollout, both the mean errors and corresponding standard derivations of the GNN corrector and the ISR neural network remain at a low level, showcasing robust temporal coherence in our super-resolution cloth animation.



(a) Frame 50 (b) Frame 150 (c) Frame 250

Fig. 10. In the Dress example, the vertex-wise Euclidean distances to the ground truth remain consistently minimal over time, spanning from frame 50 to frame 250.

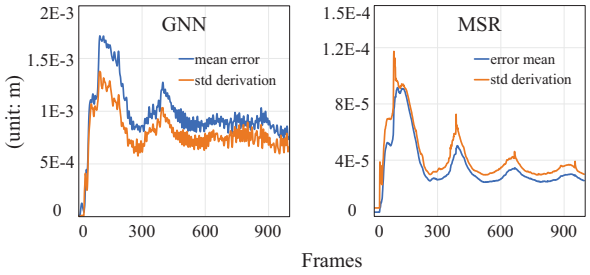


Fig. 11. In the rollout of the Three-Ball example, both the mean errors and standard derivations of the GNN and the ISR network remain at a low level, showcasing temporal coherence in our super-resolution cloth animation.

4.1.3 Runtime Performance. We present a performance summary for each example conducted on an Nvidia RTX 4090 GPU in Table 1. Our GNN corrector demonstrates comparable speed to the simulator on coarse meshes. However, the ISR module, slower than fine simulations on high-resolution meshes, emerges as the bottleneck. Nonetheless, the ISR's efficacy significantly enhances the overall efficiency of high-resolution cloth animations. For instance, applying ISR every ten time steps as a frame in simulations with a small time step, such as 1/300s, enhances animation timing and reduces runtime budgets. Additionally, future enhancements or replacements of the ISR, coupled with the potential for Multi-GPU acceleration—which is more suited to ISR than to high-resolution simulations—hold the promise of further performance improvements. Furthermore, our method facilitates the training of the ISR using real garment data from 3D scans, yielding more realistic super-resolution animations. This represents a significant advancement over purely physics-based methods, which struggle with this task.

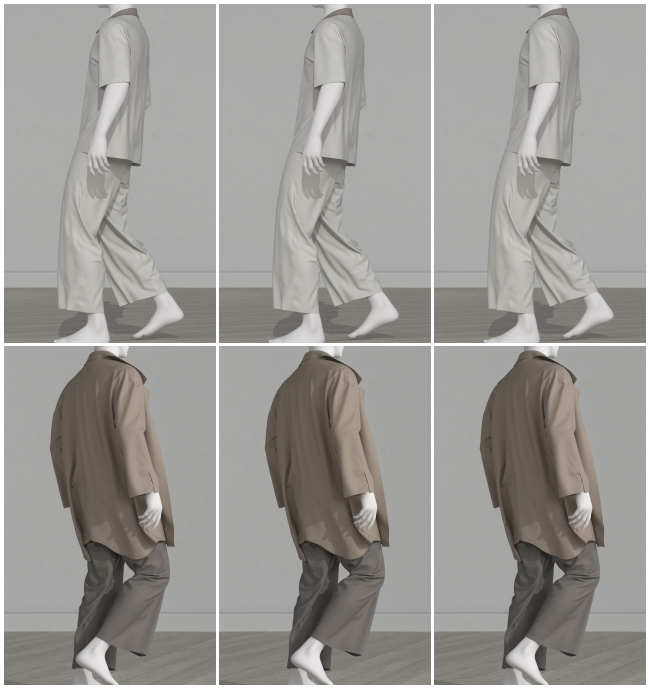
4.2 Ablation Studies

To demonstrate the importance of some strategies adopted in our method, we perform some ablation studies.

4.2.1 Interleaving simulator and corrector. In our first attempt, we trained the GNN corrector to rectify the coarse mesh sequences produced by the simulator. However, we observed that the result from standalone simulation could not be effectively corrected by a

Table 1. The statistics of our examples and their performances on a single Nvidia RTX 4090 GPU. We use a small time step $\Delta t = 1/300s$ for both coarse and fine simulations. Our GNN corrector shows comparable speed to the simulator on coarse meshes. The ISR runs slower than fine simulations on high-resolution meshes. By applying ISR every ten time steps as a frame, the timing of super-resolution cloth animations can be reduced by about 2-3 times. All timing are provided in milliseconds.

Example	HR		LR		Coarse Step	GNN	ISR	Fine Step	SR Frame	Fine Frame
	#Vert.	#Tri.	#Vert.	#Tri.	t_c	t_g	t_i	t_f	$t_{sf} = 10(t_c + t_g) + t_i$	$t_{ff} = 10t_f$
Three-Ball	184.4 K	367.6 K	2.9 K	5.8 K	6.04	6.01	312.81	116.17	433.31	1131.7
Rabbit	883.0 K	1.76 M	14.1 K	27.7 K	24.52	26.89	1272.24	483.7	1786.34	4837
Table	1.23 M	2.45 M	19.4 K	38.3 K	32.4	40.27	1802.64	684.35	2529.34	6843.5
Loungewear	591.6 K	1.18 M	10 K	18.8 K	22.93	24.57	867.11	514.85	1342.11	5148.5
Overcoat	1.03 M	2.06 M	17.2 K	32.3 K	47.21	38.97	1515.51	1300.9	2377.31	13009
Jacket	314.4 K	622.5 K	5.4 K	9.9 K	8.34	10.14	530.76	157.81	715.56	1578.1
Tea Dress	210.7 K	416.5 K	3.6 K	6.6 K	8.59	7.19	354.49	167.63	512.29	1676.3
Ruffled Dress	516.0 K	1.02 M	8.6 K	16 K	20.97	20.43	750.08	626.46	1164.08	6264.6
Dress	372.0 K	738 K	6.1 K	11.5 K	15.66	11.87	628.78	307.52	904.08	3075.2



(a) Frame SR (b) Time-step SR (c) Ground truth

Fig. 12. When applying ISR every ten time steps as a frame to enhance fine wrinkle details in examples Loungewear and Overcoat, the outcomes (a) are comparable to those obtained with every time step strategy (b) and closely resemble the ground truth (c).

neural network, i.e. the training failed to converge to a small loss value. As illustrated in Fig. 13, without our interleaving strategy, the loss value of the GNN corrector cannot decrease below 0.5mm during both the training and evaluation processes.

4.2.2 ISR initialization. This strategy plays a crucial role in enhancing the convergence speed of the ISR neural network. As depicted in

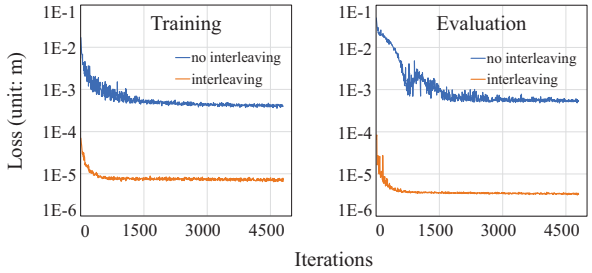


Fig. 13. Interleaving the simulator and the corrector significantly improves the training convergence of the first module of our method.

Fig. 14, the descent rate of the loss function experiences accelerated improvement, both during training and evaluation.

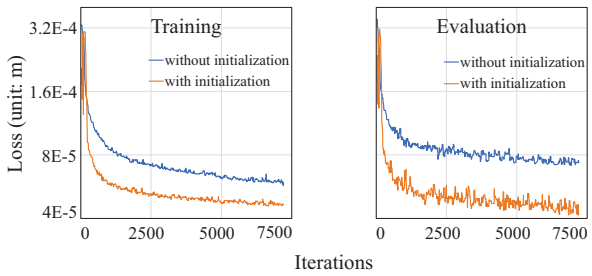


Fig. 14. The introduction of initialization significantly accelerates the convergence of both training and evaluation processes.

4.2.3 Gradient loss. Integrating a gradient loss is essential for the ISR neural network to grasp local constraints, as demonstrated by the elastic band on the Ruffled Dress. Without this addition, the elastic band fails to be faithfully reproduced, resulting in generated wrinkle patterns that significantly deviate from the ground truth, as depicted in Fig. 15(b) where the elastic band appears loose. Conversely, the incorporation of a gradient loss enables the reproduction of fine-grained wrinkles along the elastic band, even though they may appear smoother than the ground truth. In the Ruffled Dress

(a) Coarse (b) No gradient loss (c) Gradient loss (d) Ground truth

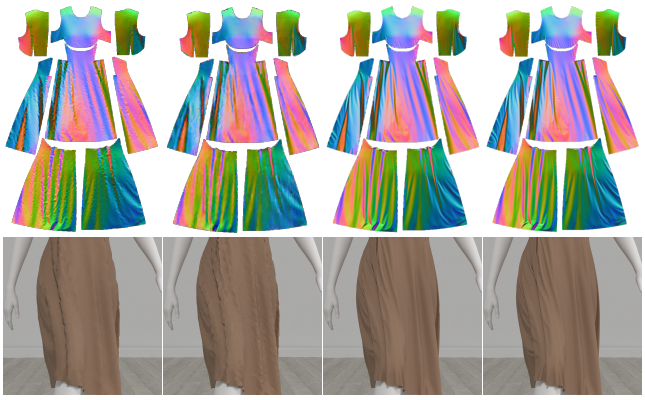
Fig. 15. Ruffled Dress. In the coarse simulation (a), wrinkles are entirely absent. Without the inclusion of a gradient loss in the MSR module (b), the elastic band becomes loose, and the fine-grained wrinkles fail to be accurately reproduced. However, upon adding the gradient loss (c), fine-grained wrinkles become evident.

example, we compute the gradient loss specifically in the region of the elastic band, with the loss weights set as $\alpha = 1$ and $\beta = 10$.

4.3 Comparisons

Based on our comprehensive literature review, we identify *Deep Details Enhancement* (DDE) [Zhang et al. 2021a] as the state-of-the-art super-resolution (SR) method specifically designed for enhancing garment details through neural networks. To ensure a fair comparison, our method takes a coarse mesh as input and produces a SR mesh. In contrast, DDE takes the normal map of the same coarse mesh as input and outputs a SR normal map. We conduct comparisons by evaluating the normal map of our SR mesh against DDE's SR normal map and also comparing our SR mesh to the fine mesh reconstructed from DDE's SR normal map. Remarkably, as depicted in Fig 16, the normal maps of our generated SR mesh closely align with the ground truth normal maps, indicated by a smaller mean error and standard deviation, specifically the error distributions are $\mathcal{N}_{ours}(0.0090, 0.0940^2)$ and $\mathcal{N}_{DDE}(0.0136, 0.1104^2)$. Furthermore, our SR mesh is visually almost identical to the ground truth, while the reconstructed mesh from DDE exhibits notable non-smoothness.

What's more, we compare our method with pure neural garment simulation to showcase the benefits of our method on generating high-quality high-resolution garment animations. Because we integrate a physics-based simulator in the framework of our method, multi-layer collision handling is much easier for our method than pure neural network-based method. In addition, though neural network-based methods can produce pretty coarse garment animations, they face challenges in high-resolution cases. We evaluate our approach against two established pure neural cloth simulation methods: *Neural Cloth Simulation* (NCS) [Bertiche et al. 2022] and *Motion Guided Deep Dynamic Garments* (MGDDG) [Zhang et al. 2022a]. As depicted in Figure 17, we simulate a high-resolution T-shirt on a dancing avatar. The result from NCS shows an overly stiff garment lacking appropriate wrinkles. The wrinkles generated by MGDDG appear significantly coarser than those produced by our method. Additionally, both NCS and MGDDG struggle with unseen avatar motions, evidenced by noticeable artifacts on the left arm. These comparisons highlight the enhanced adaptability and realism of our method in dynamic cloth simulation.



(a) Coarse input (b) DDE (c) Our result (d) Ground truth

Fig. 16. Comparison to DDE. Given an input coarse mesh or its normal map (a), the super-resolution normal map from DDE (b) derivate from the fine ground truth normal map (d). In contrast, the normal map of our super-resolution mesh is close to the ground truth.

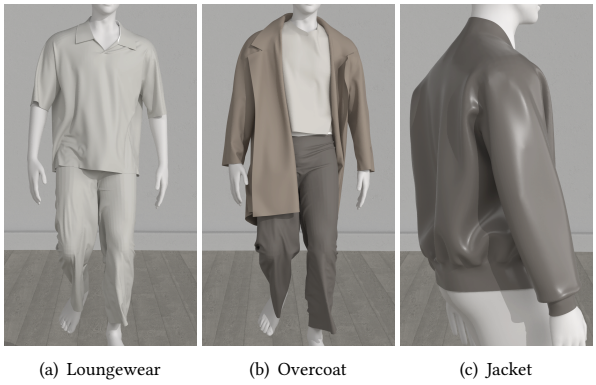
(a) Our method (b) NCS (c) MGDDG

Fig. 17. Comparisons to pure neural simulation methods. In the illustration of a high-resolution T-shirt on a dancing avatar, (a) our method produces plausible wrinkles. (b) NCS yields an overly stiff T-shirt, while the wrinkles from (c) MGDDG appear much coarser compared to our method's results.

4.4 Generalization Tests

We evaluate the generalization capability of our method across various avatar movements, clothing styles, and fabric thicknesses using a network trained on a specific garment. Additionally, we assess its generalization across different simulators using a network trained specifically for each simulator.

4.4.1 Avatar Motions. For each garment, during runtime inference as illustrated in Fig. 1, we utilize distinct avatar motions compared to those employed during the training of the GNN corrector and the ISR neural network, as displayed in Fig. 18. The training and testing motion sequences for each example have been listed in Table. 2. For garments on the male avatar, we use "CatWalk-Man" sequence to train models and "Walking-Man" sequence for inference testing, with the latter displaying more complex motions. "CatWalk-Man"



(a) Loungewear (b) Overcoat (c) Jacket

(d) Tea Dress (e) Ruffled Dress (f) Dress

Fig. 18. For the training of our GNN corrector and ISR network, we employ different avatar sequences for each examples from those used during runtime inference, as illustrated in Fig. 1.

simply walks and turns around, whereas "Walking-Man" involves turning, walking straight and backward, and multiple turn-arounds. For dresses on the female avatar, we use "Walking-Woman" and "Tryon-Woman" sequences in training and testing. "Tryon-Woman" involves clockwise and anticlockwise turns with arm movements, ending with a straight walk. "Walking-Woman" features straight walking, turning around to look back, and then turning back. Each sequence lasts over ten seconds. As the simulator handles garments' global dynamics, the GNN corrector efficiently adapts to diverse avatar motions by focusing on low-frequency motion residuals of garments.

Table 2. Training and testing motion sequences for each garment example.

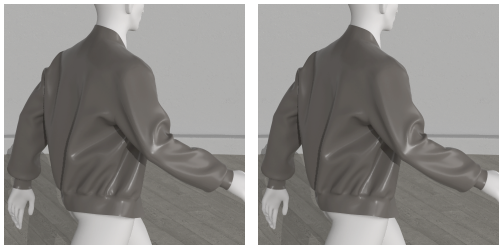
Example	Training motion	Testing motion
Loungewear	Cat-Walk Man	Walking Man
Overcoat	Cat-Walk Man	Walking Man
Jacket	Cat-Walk Man	Walking Man
TeaDress	Walking Woman	Try-on Woman
RuffledDress	Try-on Woman	Walking Woman
Dress	Try-on Woman	Walking Woman

(a) Our result (b) Ground truth (c) Our result (d) Ground truth

Fig. 19. To apply the ISR neural network trained for the Dress example to the Loungewear and Overcoat examples, we uniformly set all cloth fabrics to match that of the Dress. Our outputs for both Loungewear and Overcoat closely resemble their respective ground truth, showcasing the robustness of our method across different garment styles.

4.4.2 Garment styles. By adopting the patch-based training strategy, we aim to assess the generalization capability of our method across various garment styles. However, we noticed that the generalization capability of our GNN corrector is somewhat limited to specific garment mesh topologies due to inherent constraints in GNN technology. Presently, we train separate GNN correctors for each garment despite identical fabric properties. This limitation suggests a promising direction for future research: developing a more versatile neural network capable of adapting to varying mesh topologies. It is noteworthy that the ISR neural network trained for the MSR task on the Dress example can be seamlessly applied to the Loungewear and Overcoat examples by simply adjusting their fabric to match that of the Dress, as depicted in Fig. 19, showcasing the versatility of our method across different garment styles. The error for the Dress rollout is characterized by a normal distribution $\mathcal{N}_{dr}(0.08438, 0.01195^2)$. Similarly, for the rollouts of Loungewear and Overcoat, their error distributions are described by $\mathcal{N}_{lw}(0.1443, 0.02605^2)$ and $\mathcal{N}_{oc}(0.1337, 0.02107^2)$, respectively. Here, errors are measured in millimeters.

4.4.3 Fabric types. Fabric thickness is indicative of the stiffness in fabrics. In our experiments, we choose silk-like fabrics with small thickness to generate fine wrinkles thereby showcasing the efficacy of our super-resolution method. In theory, thicker fabrics, producing fewer wrinkles, should present less challenge for our method. As illustrated in Fig. 20, our method is also effective in the Jacket example, which features thick leather fabric. However, when a jacket made from thin silk fabric is processed by the ISR model trained specifically for the thick leather jacket, the outcomes are not satisfactory. At present, our mesh-based super-resolution module does not account for fabric properties. As a result, garments made from different fabrics require separate ISR networks to be trained specifically for each. Addressing this limitation by incorporating fabric characteristics into the super-resolution process for cloth animations represents a significant direction for future research.



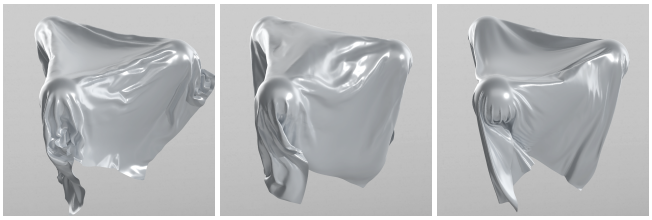
(a) SR: leather (b) Ground truth: leather

(c) SR: silk (d) Ground truth: silk

Fig. 20. The ISR network, trained on a leather jacket, effectively produces super-resolution results akin to the ground truth from fine simulations. However, this success does not extend to silk jackets, where the generated wrinkles markedly differ from the ground truth, highlighting the network's limitations across fabric types.

4.4.4 Simulators. In our method's framework, the simulator plays a pivotal role in generating the dynamics of cloth motions. Our simulator relies on implicit Euler time integration and employs the Baraff-Witkin [Baraff and Witkin 1998] model for anisotropic elasticity energy, along with the Discrete Shell [Grinspun et al. 2003] model for dihedral angle-based bending energy and analytic eigenvalue filtering [Wang et al. 2023] improving bending stability. For collision detection, we utilize BVH on GPUs [Karras 2012], detecting both self-collisions and external collisions through unilateral quadratic repulsion energies. We use the simulator to generate training data. In our method, we use a timestep of 1/300s to maintain the stability of high-resolution simulations. Our simulator is instrumental in producing ground truth fine simulation cloth animations at a resolution of 2.5mm.

Additionally, the same simulator is employed in the first module of our method to drive cloth dynamics. While our method is technically tailored to this specific simulator, we posit that it can be adapted for use with alternative simulators, such as those implementing Projective Dynamics [Bouaziz et al. 2014] which also adapts Baraff-Witkin model for elasticity energy but Quadratic Bending [Bergou et al. 2006] for bending energy. We replace the simulator by Projective Dynamics during the inference time. As depicted in Fig.21(b), the super-resolution result of our method significantly deviates from the ground truth (Fig.21(c)) fine mesh produced by the PD simulator. However, it still shows a substantial improvement over the coarse simulation (Fig. 21(a)).



(a) From coarse simulation (b) Our result (c) Ground truth

Fig. 21. When combining the PD simulator with our trained GNN corrector, our method still produces plausible super-resolution results. Although these results may deviate from the ground truth, they still outperform the super-resolution results obtained using the coarse simulation as input.

4.5 Limitations

Nevertheless, our method does exhibit certain limitations. As stated in Section 3.3.5, the training processes for both the GNN and the ISR neural network are not highly efficient. In terms of runtime inference efficiency on the training machine, the first module's performance, which involves interleaving the simulator and the GNN corrector during inference rollout, is sufficient for real-time interactive applications at around 30 FPS. This is attributed to the high efficiency of both the simulator and the GNN when processing a coarse mesh. However, the efficiency of our MSR module currently falls short of supporting real-time applications, achieving only around 3 FPS. The performance bottleneck in our method lies with the ISR module, namely SwinIR. To enable real-time high-quality cloth animations, it is imperative to minimize the runtime computational budget of the ISR neural network in our method. It is worth noting that the advanced method proposed by Zhang et al. [2022c] may potentially enhance the performance of SwinIR by up to 4 times without sacrificing quality. Unfortunately, we have not had the opportunity to implement it in our method at this time.

Regarding collision handling, the presence of intersections within the intermediate mesh eliminates the necessity for collision handling in the GNN corrector. The main challenge lies within the MSR module. We circumvent the issue of wrinkle penetration and intersection handling by converting the MSR task into an ISR problem. With the assistance of the physics-based simulator, which adeptly handles multi-layer collisions on coarse meshes, the resulting super-resolution meshes exhibit no evident penetrations and intersections. To thoroughly showcase the capabilities of our method, we intentionally omit a post-processing step to eliminate penetrations and self-intersections in our super-resolution cloth animations. As demonstrated in the supplemental video, while most cases avoid intersections, a few instances of penetrations into human bodies and self-intersections are observed, as illustrated in Fig. 22(a). Additionally, at present, there is no anti-aliasing when transforming a mesh into the image space, resulting in zigzag artifacts along folding lines, such as the collar of the Loungewear example.

Furthermore, fabric properties, such as membrane stiffness and bending stiffness, are not considered in our method at present. If cloth fabrics change, both neural networks in our method need to be retrained.

(a) Penetrations (b) Artifacts zoom-in (c) Zigzag artifact

Fig. 22. Limitations. In the Overcoat example, low probability penetrations have been observed near the left shoulder, while in the Loungewear example, zigzag artifacts are present at the collar.

5 CONCLUSIONS AND FUTURE WORK

In conclusion, our method stands as a versatile framework for super-resolution cloth animations, intricately capturing wrinkle details that closely resemble those in the ground truth fine simulation. A key distinction from previous wrinkle enhancement methods lies in our design of a corrector, aligning low-frequency features across different resolution levels to ensure spatial coherence. Furthermore, the strategy of interleaving the simulator and the corrector guarantees temporal coherence. Pioneering the application of state-of-the-art ISR methods to the challenging MSR task for high-quality cloth animations, our approach incorporates an initialization strategy that reinforces the temporal coherence of generated fine wrinkles.

Looking ahead, addressing key challenges is essential to enhancing the efficiency and precision of learning-based methods for detailed clothing animations. Combining a learning-based differentiable simulator with the learning-based corrector could integrate the first module of our method into a unified neural network. Furthermore, leveraging recent advancements, such as those presented in [Zhang et al. 2022c], holds promise for further improving the inference efficiency of our method.

ACKNOWLEDGMENTS

We wish to thank anonymous reviewers for valuable comments. We thank Yuying Wu and Rui Chen from the Digital Product Content (DPC) team and Kevin Xu and Linghui Fu from the Digital Avatar (DA) team at Style3D for helping with rendering and scene creation. We thank Ruiyang Liu and Qian He for technical support on neural network training. We also thank Tianqi Gao for encouragement and support. This work was partly supported by Key R&D Program of Zhejiang (No. 2023C01047).

REFERENCES

Ferran Alet, Adarsh Keshav Jeewajee, Maria Bauza Villalonga, Alberto Rodriguez, Tomas Lozano-Perez, and Leslie Kaelbling. 2019. Graph element networks: adaptive, structured computation and memory. In *International Conference on Machine Learning*. PMLR, 212–222.

Saeed Anwar, Salman Khan, and Nick Barnes. 2020. A Deep Journey into Super-resolution: A Survey. *ACM Comput. Surv.* 53, 3, Article 60 (may 2020), 34 pages.

David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 43–54.

Miklos Bergou, Max Wardetzky, David Harmon, Denis Zorin, and Eitan Grinspun. 2006. A Quadratic Bending Model for Inextensible Surfaces. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Cagliari, Sardinia, Italy) *(SGP '06)*. Eurographics Association, Goslar, DEU, 227–230.

Hugo Bertiche, Meysam Madadi, and Sergio Escalera. 2022. Neural Cloth Simulation. *ACM Trans. Graph.* 41, 6, Article 220 (nov 2022), 14 pages.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph. (SIGGRAPH)* 33, 4, Article 154 (July 2014), 11 pages.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How Attentive are Graph Attention Networks?. In *International Conference on Learning Representations*.

Jiong Chen, Florian Schäfer, Jin Huang, and Mathieu Desbrun. 2021b. Multiscale Cholesky Preconditioning for Ill-Conditioned Problems. *ACM Trans. Graph.* 40, 4, Article 81 (jul 2021), 13 pages.

Yunuo Chen, Tianyi Xie, Cem Yuksel, Danny Kaufman, Yin Yang, Chenfanfu Jiang, and Minchen Li. 2023b. Multi-Layer Thick Shells. In *ACM SIGGRAPH 2023 Conference Proceedings (SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 25, 9 pages.

Zhen Chen, Hsiao-Yu Chen, Danny M. Kaufman, Mélina Skouras, and Etienne Vouga. 2021a. Fine Wrinkling on Coarsely Meshed Thin Shells. *ACM Trans. Graph.* 40, 5, Article 190 (aug 2021), 32 pages.

Zhen Chen, Danny Kaufman, Mélina Skouras, and Etienne Vouga. 2023a. Complex Wrinkle Field Evolution. *ACM Trans. Graph.* 42, 4, Article 72 (jul 2023), 19 pages.

Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but responsive cloth. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (San Antonio, Texas) *(SIGGRAPH '02)*. Association for Computing Machinery, New York, NY, USA, 604–611.

Edilson de Aguiar, Leonid Sigal, Adrien Treuille, and Jessica K. Hodgins. 2010. Stable spaces for real-time clothing. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) *(SIGGRAPH '10)*. Association for Computing Machinery, New York, NY, USA, Article 106, 9 pages.

Wei-Wen Feng, Yizhou Yu, and Byung-Uck Kim. 2010. A Deformation Transformer for Real-Time Cloth Animation. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) *(SIGGRAPH '10)*. Association for Computing Machinery, New York, NY, USA, Article 108, 9 pages.

Russell Gillette, Craig Peters, Nicholas Vining, Essex Edwards, and Alla Sheffer. 2015. Real-Time Dynamic Wrinkling of Coarse Animated Cloth. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '15)*. Association for Computing Machinery, New York, NY, USA, 17–26.

Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete Shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03)*. Eurographics Association, Goslar, DEU, 62–67.

Xianfeng Gu, Steven J. Gortler, and Hugues Hoppe. 2002. Geometry images. *ACM Trans. Graph.* 21, 3 (jul 2002), 355?361.

Peng Guan, Loretta Reiss, David A. Hirshberg, Alexander Weiss, and Michael J. Black. 2012. DRAPE: DRessing Any PErson. *ACM Trans. Graph.* 31, 4, Article 35 (jul 2012), 10 pages.

Jiequn Han, Arnulf Jentzen, and Weinan E. 2018. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences* 115, 34 (2018), 8505–8510.

Tero Karras. 2012. Maximizing parallelism in the construction of BVHs, octrees, and k-d trees. In *Proceedings of the Fourth ACM SIGGRAPH/Eurographics conference on High-Performance Graphics*. 33–37.

Ladislav Kavan, Dan Gerszewski, Adam W. Bargteil, and Peter-Pike Sloan. 2011. Physics-Inspired Upsampling for Cloth Simulation in Games. , Article 93 (2011), 10 pages.

Byungsoo Kim, Vinicius C Azevedo, Nils Thuerey, Theodore Kim, Markus Gross, and Barbara Solenthaler. 2019. Deep fluids: A generative network for parameterized fluid simulations. 38, 2 (2019), 59–70.

Byung-Uck Kim and Farzam Farbiz. 2013. Wrinkle Flow for Compact Representation of Predefined Clothing Animation. In *ACM SIGGRAPH 2013 Posters* (Anaheim, California) *(SIGGRAPH '13)*. Association for Computing Machinery, New York, NY, USA, Article 15, 1 pages.

Scott Kircher and Michael Garland. 2005. Progressive Multiresolution Meshes for Deforming Surfaces. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '05)*. Association for Computing Machinery, New York, NY, USA, 191?200.

Zorah Lahner, Daniel Cremers, and Tony Tung. 2018. Deepwrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European conference on computer vision (ECCV)*. 667–684.

Lei Lan, Minchen Li, Chenfanfu Jiang, Huamin Wang, and Yin Yang. 2023. Second-Order Stencil Descent for Interior-Point Hyperelasticity. *ACM Trans. Graph.* 42, 4, Article

108 (jul 2023), 16 pages.
Lei Lan, Guanqun Ma, Yin Yang, Changxi Zheng, Minchen Li, and Chenfanfu Jiang. 2022. Penetration-Free Projective Dynamics on the GPU. *ACM Trans. Graph.* 41, 4, Article 69 (jul 2022), 16 pages.
Yongjoon Lee, Sung eui Yoon, Seungwoo Oh, Duksu Kim, and Sunghee Choi. 2010. Multi-Resolution Cloth Simulation. *Computer Graphics Forum (Pacific Graphics)* 29, 7 (2010), 2225–2232.
Cheng Li, Min Tang, Ruofeng Tong, Ming Cai, Jieyi Zhao, and Dinesh Manocha. 2020. P-Cloth: Interactive Complex Cloth Simulation on Multi-GPU Systems Using Dynamic Matrix Assembly and Pipelined Implicit Integrators. *ACM Trans. Graph.* 39, 6, Article 180 (nov 2020), 15 pages.
Minchen Li, Ming Gao, Timothy Langlois, Chenfanfu Jiang, and Danny M. Kaufman. 2019. Decomposed Optimization Time Integrator for Large-Step Elastodynamics. *ACM Trans. Graph.* 38, 4, Article 70 (July 2019), 10 pages.
Yinxiao Li, Yan Wang, Yonghao Yue, Danfei Xu, Michael Case, Shih-Fu Chang, Eitan Grinspun, and Peter K Allen. 2018. Model-driven feedforward prediction for manipulation of deformable objects. *IEEE Transactions on Automation Science and Engineering* 15, 4 (2018), 1621–1638.
Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE international conference on computer vision*. 1833–1844.
Hongying Liu, Zhubo Ruan, Peng Zhao, Chao Dong, Fanhua Shang, Yuanyuan Liu, Linlin Yang, and Radu Timofte. 2022. Video super-resolution based on deep learning: a comprehensive survey. *Artificial Intelligence Review* 55, 8 (2022), 5981–6035.
Hsueh-Ti Derek Liu, Jiayi Eris Zhang, Mirela Ben-Chen, and Alec Jacobson. 2021b. Surface Multigrid via Intrinsic Prolongation. *ACM Trans. Graph.* 40, 4, Article 80 (jul 2021), 13 pages.
Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021a. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE international conference on computer vision*. 10012–10022.
Pingchuan Ma, Peter Yichen Chen, Bolei Deng, Joshua B Tenenbaum, Tao Du, Chuang Gan, and Wojciech Matusik. 2023. Learning Neural Constitutive Laws From Motion Observations for Generalizable PDE Dynamics. *arXiv preprint arXiv:2304.14369* (2023).
Matthias Müller. 2008. Hierarchical Position Based Dynamics. In *Proceedings of VRIPHYS*. 1–10.
Matthias Müller and Nuttapong Chentanez. 2010. Wrinkle Meshes. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Madrid, Spain) *(SCA '10)*. Eurographics Association, Goslar, DEU, 85–92.
Young Jin Oh, Tae Min Lee, and In-Kwon Lee. 2018. Hierarchical Cloth Simulation Using Deep Neural Networks. In *Proceedings of Computer Graphics International 2018* (Bintan, Island, Indonesia) *(CGI 2018)*. Association for Computing Machinery, New York, NY, USA, 139–146.
Xiaoyu Pan, Jiaming Mai, Xinwei Jiang, Dongxue Tang, Jingxiang Li, Tianjia Shao, Kun Zhou, Xiaogang Jin, and Dinesh Manocha. 2022. Predicting Loose-Fitting Garment Deformations Using Bone-Driven Motion Networks. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 11, 10 pages.
Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. 2020. Learning Mesh-Based Simulation with Graph Networks. In *International Conference on Learning Representations*.
Damien Rohmer, Tiberiu Popa, Marie-Paule Cani, Stefanie Hahmann, and Alla Sheffer. 2010. Animation Wrinkling: Augmenting Coarse Cloth Simulations with Realistic-Looking Wrinkles. *ACM Trans. Graph.* 29, 6, Article 157 (dec 2010), 8 pages.
Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. 2020. Learning to simulate complex physics with graph networks. In *International conference on machine learning*. PMLR, 8459–8468.
Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. 2018. Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning*. PMLR, 4470–4479.
Igor Santesteban, Miguel A Otaduy, and Dan Casas. 2022. Snug: Self-supervised neural dynamic garments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8140–8150.
Igor Santesteban, Nils Thuerey, Miguel A Otaduy, and Dan Casas. 2021. Self-supervised collision handling via generative 3d garment models for virtual try-on. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11763–11773.
Han Shao, Libo Huang, and Dominik L. Michels. 2022. A Fast Unsmoothed Aggregation Algebraic Multigrid Framework for the Large-Scale Simulation of Incompressible Flow. *ACM Trans. Graph.* 41, 4, Article 49 (jul 2022), 18 pages.
Rasmus Tamstorf, Toby Jones, and Stephen F. McCormick. 2015. Smoothed Aggregation Multigrid for Cloth Simulation. *ACM Trans. Graph. (SIGGRAPH Asia)* 34, 6, Article 245 (Oct. 2015), 13 pages.
Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
Huamin Wang. 2021. GPU-Based Simulation of Cloth Wrinkles at Submillimeter Levels. *ACM Trans. Graph.* 40, 4, Article 169 (jul 2021), 14 pages.
Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James F. O'Brien. 2010a. Example-Based Wrinkle Synthesis for Clothing Animation. *ACM Trans. Graph.* 29, 4, Article 107 (jul 2010), 8 pages.
Huamin Wang, James O'Brien, and Ravi Ramamoorthi. 2010b. Multi-Resolution Isotropic Strain Limiting. *ACM Trans. Graph. (SIGGRAPH Asia)* 29, 6, Article 156 (Dec. 2010), 156:1–156:10 pages.
Huamin Wang and Yin Yang. 2016. Descent Methods for Elastic Body Simulation on the GPU. *ACM Trans. Graph. (SIGGRAPH Asia)* 35, 6, Article 212 (Nov. 2016), 10 pages.
Zhihao Wang, Jian Chen, and Steven CH Hoi. 2020. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence* 43, 10 (2020), 3365–3387.
Zhendong Wang, Longhua Wu, Marco Fratarcangeli, Min Tang, and Huamin Wang. 2018. Parallel Multigrid for Nonlinear Cloth Simulation. *Computer Graphics Forum (Pacific Graphics)* 37, 7 (2018), 131–141.
Zhendong Wang, Yin Yang, and Huamin Wang. 2023. Stable Discrete Bending by Analytic Eigensystem and Adaptive Orthotropic Geometric Stiffness. *ACM Trans. Graph.* 42, 6, Article 175 (dec 2023), 16 pages.
Ruben Wiersma, Ahmad Nasikun, Elmar Eisemann, and Klaus Hildebrandt. 2023. A Fast Geometric Multigrid Method for Curved Surfaces. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 1, 11 pages.
Botao Wu, Zhendong Wang, and Huamin Wang. 2022. A GPU-Based Multilevel Additive Schwarz Preconditioner for Cloth and Deformable Body Simulation. *ACM Trans. Graph.* 41, 4, Article 63 (jul 2022), 14 pages.
Zangyueyang Xian, Xin Tong, and Tiantian Liu. 2019. A Scalable Galerkin Multigrid Method for Real-Time Simulation of Deformable Objects. *ACM Trans. Graph.* 38, 6, Article 162 (nov 2019), 13 pages.
Jiayi Eris Zhang, Jérémie Dumas, Yun (Raymond) Fei, Alec Jacobson, Doug L. James, and Danny M. Kaufman. 2022b. Progressive Simulation for Cloth Quasistatics. *ACM Trans. Graph.* 41, 6, Article 218 (nov 2022), 16 pages.
Meng Zhang, Duygu Ceylan, and Niloy J. Mitra. 2022a. Motion Guided Deep Dynamic 3D Garments. *ACM Trans. Graph.* 41, 6, Article 219 (nov 2022), 12 pages.
Meng Zhang, Tuanfeng Wang, Duygu Ceylan, and Niloy J Mitra. 2021a. Deep detail enhancement for any garment. *Computer Graphics Forum* 40, 2 (2021), 399–411.
Meng Zhang, Tuanfeng Y. Wang, Duygu Ceylan, and Niloy J. Mitra. 2021b. Dynamic Neural Garments. *ACM Trans. Graph.* 40, 6, Article 235 (dec 2021), 15 pages.
Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. 2022c. Efficient long-range attention network for image super-resolution. In *European Conference on Computer Vision*. Springer, 649–667.
Fang Zhao, Zekun Li, Shaoli Huang, Junwu Weng, Tianfei Zhou, Guo-Sen Xie, Jue Wang, and Ying Shan. 2023. Learning Anchor Transformations for 3D Garment Animation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 491–500.
Javier S Zurdo, Juan P Brito, and Miguel A Otaduy. 2013. Animating wrinkles by example on non-skinned cloth. *IEEE Transactions on Visualization and Computer Graphics* 19, 1 (2013), 149–158.