

---

# MPMAvatar: Learning 3D Gaussian Avatars with Accurate and Robust Physics-Based Dynamics

---

Changmin Lee   Jihyun Lee   Tae-Kyun Kim

KAIST

{lcm914, jyun.lee, kimtaekyun}@kaist.ac.kr

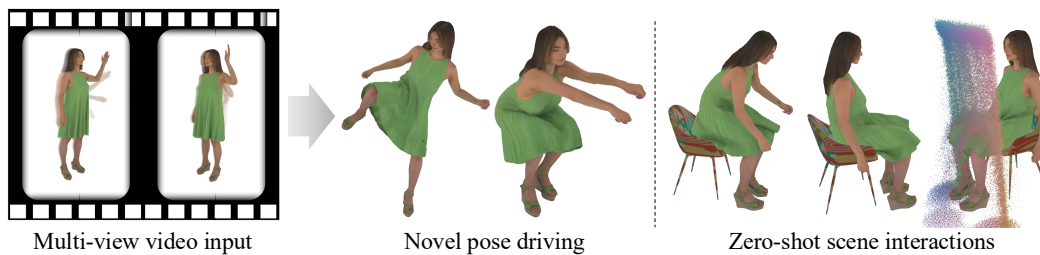


Figure 1. **3D Gaussian avatars with accurate and robust physics-based animation.** We present MPMAvatar, a framework for creating 3D Gaussian avatars from multi-view videos, that support *physically accurate and robust animations*, especially for loose garments. Ours is also zero-shot generalizable to novel scene interactions.

## Abstract

While there has been significant progress in the field of 3D avatar creation from visual observations, modeling physically plausible dynamics of humans with loose garments remains a challenging problem. Although a few existing works address this problem by leveraging physical simulation, they suffer from limited accuracy or robustness to novel animation inputs. In this work, we present MPMAvatar, a framework for creating 3D human avatars from multi-view videos that supports highly realistic, robust animation, as well as photorealistic rendering from free viewpoints. For accurate and robust dynamics modeling, our key idea is to use a Material Point Method-based simulator, which we carefully tailor to model garments with complex deformations and contact with the underlying body by incorporating an anisotropic constitutive model and a novel collision handling algorithm. We combine this dynamics modeling scheme with our canonical avatar that can be rendered using 3D Gaussian Splatting with quasi-shadowing, enabling high-fidelity rendering for physically realistic animations. In our experiments, we demonstrate that MPMAvatar significantly outperforms the existing state-of-the-art physics-based avatar in terms of (1) dynamics modeling accuracy, (2) rendering accuracy, and (3) robustness and efficiency. Additionally, we present a novel application in which our avatar generalizes to unseen interactions in a zero-shot manner—which was not achievable with previous learning-based methods due to their limited simulation generalizability. Our project page is at: <https://KAISTChangmin.github.io/MPMAvatar/>.

## 1 Introduction

Creating 3D human avatars has been an important research problem due to their broad range of applications, including virtual and augmented reality, computer games, and content creation. The key goals of this field include modeling and learning avatars that support (1) photorealistic rendering from free viewpoints and (2) realistic animation driven by sparse motion inputs (e.g., SMPL-X [51])

parameters). On the rendering side, recent approaches based on 3D Gaussian Splatting [22] have shown remarkable progress in enabling high-fidelity avatar rendering from free viewpoints. However, realistically animating avatars under novel motion inputs remains a challenging problem, especially for loose garments. Existing methods typically model these garment motions using piecewise linear transformations [1, 3, 6, 48, 61, 72, 47, 73] or pose-dependent geometric correctives [5, 35, 38, 39, 65], but they are known to be limited in accurately capturing complex deformations and tend to overfit to motions observed during training, as discussed in [78].

To address these limitations, incorporating physics-based prior via *physically simulating* the avatar dynamics can be an effective approach to enhance the realism and generalizability of animation. There are only a few works towards this direction; Xiang *et al.* [69] proposes an avatar simulated using X-PBD [41], but their method relies on a time-consuming manual parameter search to approximate reasonable cloth behavior. PhysAvatar [78] is the most recent physics-based avatar that adopts C-IPC [31] for garment simulation. However, the simulator fails when the animation inputs, SMPL-X [51] meshes, have a small degree of self-penetration, which can occur as they are practically obtained via *estimation* (e.g., fitting the parametric model to input observations). This causes C-IPC to fail to resolve the collision in the Continuous Collision Detection (CCD) stage [31], thus the driving body meshes are *manually adjusted* to avoid simulation failures in this method. In terms of appearance, PhysAvatar relies on mesh-based rendering, which limits its ability to capture fine-grained appearance details.

In this paper, we propose MPMAvatar, a framework for creating 3D human avatars from multi-view videos that enables (1) physically accurate and robust animation especially for loose garments, as well as (2) high-fidelity rendering based on 3D Gaussian Splatting [22]. For garment dynamics modeling, our key idea is to use a Material Point Method [17] (MPM)-based simulator, which is capable of simulating objects under complex contacts without failure cases via feedforward velocity projection. However, directly adopting the existing MPM simulator [17], mainly used for general object dynamics modeling, introduces two challenges for effective garment dynamics modeling for our avatar.

First, garments typically exhibit a codimensional manifold structure, and their physical properties vary significantly depending on the direction (e.g., in-manifold vs. normal directions). Second, the existing collision handling algorithm [17] of MPM is mainly designed for *colliders* that are analytically represented using level sets (e.g., simple geometries such as spheres), which is not applicable in our scenario where the collider is SMPL-X [51] body mesh underlying the garments. To address these issues, we tailor our MPM simulator by (1) adopting an anisotropic constitutive model [16] to better model the manifold-dependent dynamics of garments, and (2) by introducing a novel collision handling algorithm that can handle more general colliders represented as meshes. We combine this dynamics modeling scheme with our canonical avatar that can be rendered using 3D Gaussian Splatting with quasi-shadowing, enabling high-fidelity rendering for physically realistic animations.

In our experiments, we demonstrate that MPMAvatar outperforms the current state-of-the-art physics-based avatar [78] in terms of (1) dynamics modeling accuracy, (2) rendering accuracy, and (3) simulation robustness and efficiency. Additionally, we present a novel application in which our avatar generalizes to novel scene interactions in a zero-shot manner – demonstrating the generalizability of our physics simulation-based dynamics modeling. Our code will be also publicly available to allow full reproducibility.

Overall, our contributions can be summarized as follows:

- We present MPMAvatar, a novel framework for creating 3D clothed human avatars from multi-view videos. Our avatar supports physically realistic and robust animations, especially for loose garments, as well as high-quality rendering.
- We present an MPM [17]-based simulation method carefully tailored for effective garment dynamics modeling, based on an anisotropic constitutive model [16] and mesh-based collider handling.
- We empirically demonstrate that MPMAvatar achieves superior performance than the existing SOTA physics-based avatar (PhysAvatar [78]) in terms of (1) dynamics modeling accuracy, (2) rendering accuracy, and also (3) simulation robustness and efficiency.
- We show that our physics-based simulation method is *zero-shot generalizable* to interactions with an unseen external object. To the best of our knowledge, MPMAvatar is the first to empirically demonstrate this ability.

## 2 Related Work

**Physics-based simulation methods.** Physics-based simulation methods [46, 41, 21, 30, 31, 17] simulate object dynamics by solving the governing differential equations (e.g., conservation of momentum). Position-Based Dynamics (PBD) [46, 41] directly manipulates the positions of particles to satisfy physical constraints. While it is fast and stable, it is limited in accurately modeling complex material behaviors due to the lack of explicit physical modeling (e.g., plasticity). Variational integrators [21], such as IPC [30] and its cloth-specific extension C-IPC [31], formulates simulation as the iterative minimization of a total energy potential, which is possibly augmented with barrier terms to robustly handle contact and friction. While C-IPC is effective for simulating cloth and was thus adopted in the current SOTA physics-based avatar [78], it fails to resolve collisions for *noisy* colliders during Continuous Collision Detection (CCD), as discussed in Sec. 1. Some recent works [37, 33] explore differentiable cloth simulation for inverse problems. However, they primarily operate in simplified settings without complex colliders, making them less applicable to collision-heavy scenarios like clothed human avatars. Material Point Method (MPM) [17] simulates continuum materials by hybridly representing them with Lagrangian particles and an Eulerian grid, and is known for robust handling of large deformations and self-collisions. Due to these advantages crucial for simulation stability and accuracy, our work adopts MPM, while further tailoring its simulation scheme for more effective garment dynamics modeling for our avatar. In a later section (Sec. 5), we demonstrate that our simulation method outperforms C-IPC used in the SOTA physics-based avatar [78] in both dynamics modeling accuracy and robustness.

**Learning-based simulation methods.** Some existing methods [8, 7] use neural network-based simulators trained on large datasets to *implicitly* encode physics. However, they have limited generalizability beyond the training dynamics and cannot *guarantee* physically plausible deformations, as object dynamics are *predicted* by the network. As discussed in [8], this leads to failure cases when objects move faster or more erratically than in the training data. Thus, directly following the motivation of the recent physics-based avatar [78], we adopt a physics-based simulation. In Sec. 5, we additionally show that our MPM-based simulation method is *zero-shot generalizable* to novel scene interactions — which is not achievable through learning-based simulation.

**Dynamic avatar reconstruction from visual inputs.** Learning 3D human avatars from visual inputs has been an active area of research in computer vision and graphics. Earlier methods rely on (1) *a mesh representation* [51, 9, 11, 56, 71, 10, 24, 29, 23, 62], which is computationally efficient but struggles to capture details for rendering, or (2) *an implicit representation* [13, 52, 68, 77, 53, 18, 27, 28], which captures high-frequency geometry and appearance details but complicates dynamics modeling due to the absence of explicit geometry. Most recent avatar reconstruction methods [20, 36, 45, 49, 79] use 3D Gaussian Splats (3DGS) [22], which address these limitations by enabling high-fidelity rendering while using explicit representation, facilitating more effective dynamics modeling. For modeling avatar dynamics, existing methods often represent garment motions using piecewise linear transformations [1, 3, 6, 48, 61, 72, 47, 73] or pose-dependent geometric correctives [5, 35, 38, 39, 65]. However, these approaches are limited in their ability to capture complex deformations and tend to overfit to the training motion data, as discussed in [78]. To enable more realistic and physically accurate animations, a few works [58, 69, 78] have integrated simulators such as XPBD [41, 58], C-IPC [31, 70]. However, they require manual parameter search, or manual adjustment for body mesh colliders to avoid simulation failures, as discussed in Sec. 1. A related line of work, such as DiffAvatar [34], focuses on generating physically plausible garments from static 3D scans using differentiable simulation. However, their formulation omits appearance modeling and is tailored for scan-based asset preparation, rather than dynamic avatar reconstruction from visual observations. Note that some existing works on garment-only modeling (not the clothed avatars as in our work) adopt a neural network-based simulator [8, 7, 55], but they do not *guarantee* physically based behavior and are known to have limited generalizability compared to physics-based simulators [78].

## 3 Preliminaries: Material Point Method (MPM)

Material Point Method (MPM) [17] models an object as a continuum, enabling the simulation of diverse materials including solids, liquids, and gases. MPM advances the simulation by representing the continuum using both Lagrangian particles and an Eulerian grid and solving two governing equations: (a) conservation of mass and (b) conservation of momentum:

$$(a) \quad \frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0, \quad (b) \quad \rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}, \quad (1)$$

where  $\rho$  and  $\mathbf{v}$  are density and velocity, respectively.  $\frac{D}{Dt}$  denotes the material derivative,  $\boldsymbol{\sigma}$  is the cauchy stress tensor, and  $\mathbf{g}$  is the gravitational acceleration. During simulation, physical quantities such as mass and momentum are two-way transferred between the particles and the grid. Here, mass conservation is straightforwardly achieved due to the invariant mass carried by Lagrangian particles, while momentum conservation is performed on the Eulerian grid to efficiently approximate the spatial derivatives. When solving Eq. 1b for updating momentum on the grid, computing the cauchy stress tensor  $\boldsymbol{\sigma}$  is the key in capturing the material behavior. More specifically, it is defined as  $\boldsymbol{\sigma} = \frac{1}{\det(\mathbf{F})} \frac{\partial \psi}{\partial \mathbf{F}} \mathbf{F}^T$ , where  $\mathbf{F}$  is the deformation gradient that linearly approximates local deformations, and the strain-energy density function  $\psi$ , which depends on  $\mathbf{F}$ , quantifies the energy stored through the deformation. For defining  $\psi$ , various *constitutive models* [59, 25, 60] have been developed to define  $\psi$ , such that it can effectively model various material behaviors (e.g. for jelly, snow, sand, and fluids).

## 4 MPMAvatar: Photorealistic Avatars with Physics-Based Dynamics

In this section, we present MPMAvatar, a framework that learns 3D human avatars from multi-view videos that support (1) physically accurate and robust animation and (2) high-quality rendering. In the following sections, we first describe our avatar representation (Sec. 4.1). We then present our physics-based approach to modeling avatar dynamics, which is particularly effective for realistically animating loose garments (Sec. 4.2). Finally, we explain how the proposed physically-based dynamic avatar can be learned from multi-view video inputs (Sec. 4.3).

### 4.1 Avatar Representation

To enable both physically realistic animation and high-fidelity rendering, we use a hybrid representation that combines (1) a mesh with physical parameters to enable physically based animation, and (2) 3D Gaussian Splats [22] for high-quality rendering. Formally, we represent the canonical geometry of an avatar with a 3D triangular mesh  $\mathcal{M}_1 = (\mathbf{V}_1, \mathbf{F})$ , where  $\mathbf{V}_1$  and  $\mathbf{F}$  are mesh vertices and faces, respectively.<sup>1</sup> To model the physics-based dynamics of the geometry, the avatar is also represented with physical parameters  $\mathcal{P} = (E, \nu, \gamma, \kappa, \rho, \alpha)$ . It consists of material parameters used for traditional Material Point Method (MPM) [17] simulation (Young’s modulus  $E$  and Possion’s ratio  $\nu$ ), additional material parameters for anisotropic dynamics modeling (shear stiffness  $\gamma$  and normal stiffness  $\kappa$ , which will be introduced in Sec. 4.2.1), density  $\rho$ , and the rest geometry parameter  $\alpha$ .

To enable high-fidelity rendering, we represent the avatar appearance with 3D Gaussian Splats [22]  $\mathcal{G} = \{\mathbf{g}_i\}_{i=1 \dots N_G}$ . Each Gaussian Splat  $\mathbf{g}_i$  is parameterized by a translation vector  $\mathbf{t}_i \in \mathbb{R}^3$ , a quaternion  $\mathbf{q}_i \in \mathbb{R}^4$ , a scale vector  $\mathbf{s}_i \in \mathbb{R}^3$ , an opacity  $o_i \in \mathbb{R}$ , and color  $\mathbf{c}_i$  represented by spherical harmonics [22]. Note that our Gaussian Splats are *attached* to the canonical avatar mesh  $\mathcal{M}_1$ , where each  $\mathbf{g}_i$  is associated with a face of  $\mathcal{M}_1$ . More specifically, following [54], we define all spatial parameters of  $\mathbf{g}_i$  in the local coordinate system with respect to its associated mesh face, allowing our Gaussian Splats to naturally deform according to the underlying mesh deformations.

### 4.2 Physics-Based Dynamics Modeling

We now explain how we model the dynamics of our avatar discussed in Sec. 4.1 – to achieve highly realistic and physically grounded animations. Following the existing physics-based avatar [78], we animate the body (non-garment regions) of the avatar using Linear Blend Skinning [15], while animating its garments driven by the underlying body motions (represented with SMPL-X [51] meshes) via physical simulation. For the simulation, we adopt Material Point Method (MPM) [17] due to its effectiveness in modeling large deformations and robustly handling collisions (Sec. 2). While MPM is actively adopted in recent 3D scene simulation methods [4, 76, 70], it is mainly used for modeling the dynamics of general objects (e.g., flower pots, elastic torus).

<sup>1</sup>While the deformation gradient is also denoted by  $\mathbf{F}$ , we allow a slight abuse of notation to remain consistent with notation conventions used in related work.

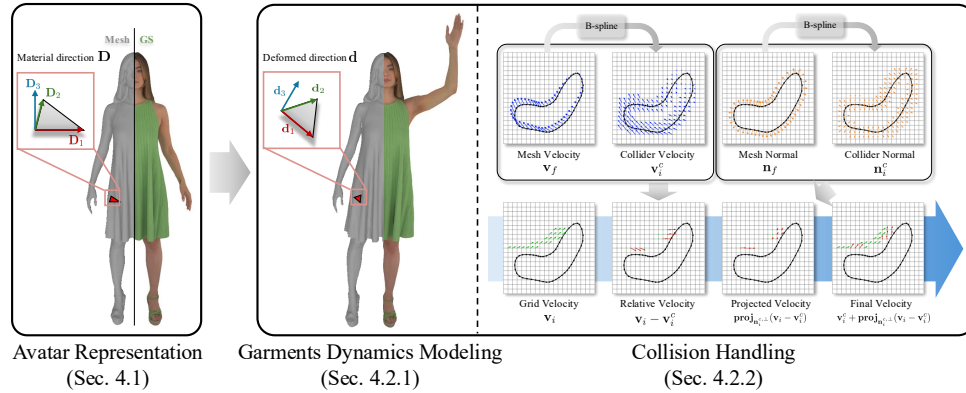


Figure 2. **Overview of our dynamic avatar modeling.** We hybridly represent our canonical avatar with (1) a mesh with physical parameters for geometry and dynamics modeling, and (2) 3D Gaussian Splats [22] for appearance modeling (Sec. 4.1). This avatar can be animated via linear blend skinning for non-garment regions and physical simulation for garment regions (Sec. 4.2.1) with our novel collision handling algorithm (Sec. 4.2.2). *Visualization key.* Blue arrows indicate body grid velocities, green arrows denote garment grid velocities, and red arrows show colliding grid regions where velocity projection is applied.

In this work, we carefully tailor the simulator [17] to achieve more effective modeling of *garment dynamics* in our avatar. In particular, we (1) adopt the anisotropic constitutive model [16] to better model the manifold-dependent dynamics of garments, and (2) introduce a collision handling algorithm designed to effectively resolve garment-body collisions. In what follows, we further elaborate on these two modifications.

#### 4.2.1 Anisotropic Dynamics Modeling for Garments

Garments typically exhibit a codimensional manifold structure, and their physical properties vary depending on it. For example, garments can easily stretch along in-manifold directions, but not along the normal directions. To accurately model this behavior, we adopt the *anisotropic* constitutive model proposed by Jiang *et al.* [16] underlying our MPM simulator. In particular, they propose to model the strain energy for anisotropic material depending on each Lagrangian particle’s material directions, which are approximated using Lagrangian mesh. More specifically, the deformation gradient  $\mathbf{F}$  at each particle is computed as  $\mathbf{F} = \mathbf{d}\mathbf{D}^{-1}$ , where  $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3] \in \mathbb{R}^{3 \times 3}$  is the original material direction and  $\mathbf{d} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3] \in \mathbb{R}^{3 \times 3}$  is the deformed material direction (see Fig. 2). Since the strain-energy density function  $\psi$  must be invariant under rotations, they apply QR-decomposition  $\mathbf{F} = \mathbf{Q}\mathbf{R}$ , and  $\psi$  is reparameterized as  $\psi(\mathbf{F}) = \hat{\psi}(\mathbf{R})$ , such that:

$$\hat{\psi}(\mathbf{R}|E, \nu, \kappa, \gamma) = \hat{\psi}_{\text{normal}}(\mathbf{R}_{33}|\kappa) + \hat{\psi}_{\text{shear}}(\mathbf{R}_{13}, \mathbf{R}_{23}|\gamma) + \hat{\psi}_{\text{in-plane}}(\mathbf{R}_{11}, \mathbf{R}_{12}, \mathbf{R}_{22}|E, \nu),$$

where  $\hat{\psi}_{\text{normal}}$ ,  $\hat{\psi}_{\text{shear}}$ , and  $\hat{\psi}_{\text{in-plane}}$  are functions for penalizing normal deformation, shearing, and in-plane deformation, respectively (refer to [16] and Appendix C for details). Note that this constitutive model [16] requires a Lagrangian mesh representing the codimensional object to track the material directions, which had motivated our avatar representation (Sec. 4.1) based on a hybrid mesh and Gaussian Splats. As the official implementation of the MPM solver for this constitutive model is not publicly available, we re-implemented it using PyTorch [50] and Warp [40], and plan to release our code to facilitate future research.

#### 4.2.2 Collision Handling

We additionally introduce a collision handling algorithm designed to effectively resolve our garment-body collisions.

**Collision handling of MPM [17].** The existing collision handling algorithm of MPM is designed to be effective for an external object (i.e., a collider) *represented as a dynamic level set*. In a nutshell, MPM resolves collisions by projecting the grid velocity of the *colliding region* of the object onto the

collider’s tangent space, preventing penetration while modeling tangential motion. Technically, this requires evaluating the velocity and normal of the collider at the Eulerian grids *nearby* the colliding regions, which are not strictly on the collider surface. MPM originally enables these evaluations by considering colliders whose geometry and velocity fields are analytically defined over all points  $\mathbf{x} \in \mathbb{R}^3$  in the ambient 3D space, e.g., a sphere geometry  $\phi(\mathbf{x}) = \|\mathbf{x} - \mathbf{c}\| - r$ , where  $\mathbf{c}$  and  $r$  denote center and radius, respectively. However, our collider is a SMPL-X [51] mesh representing the body underlying the garment, thus it is not trivial to directly adopt this algorithm.

**Our algorithm.** To address the above limitation, we introduce a simple yet effective collision handling algorithm for MPM to support colliders represented as meshes. Note that, in this case, normal or velocity is defined only on the collider surface, not at all Euclidean grid nodes. To address this, our idea is to transfer these quantities to nearby grid nodes using B-spline weights, directly analogical to *particle-to-grid transfer* in MPM simulation. In Fig. 2, we outline our overall collision handling procedure consisting of two stages: (1) mesh-to-grid transfer (upper row) and (2) relative velocity projection (lower row). In the mesh-to-grid transfer stage, we transfer each collider face’s velocity  $\mathbf{v}_f$  and normal  $\mathbf{n}_f$  to nearby grid nodes using B-Spline weights, producing the extended collider velocity  $\mathbf{v}_i^c$  and normal  $\mathbf{n}_i^c$  at each grid node  $i$ . In the relative velocity projection stage, we first transform the grid velocity into the collider mesh’s reference frame by subtracting the collider velocity  $\mathbf{v}_i^c$ . If the relative velocity points inward, we project out its normal component. Finally, we transform the corrected velocities back into the world frame. It is worth noting that the complexity of the existing collision handling algorithm of MPM is  $O(N_{\text{grid}}^3)$ , where  $N_{\text{grid}}$  is the grid resolution, as it requires evaluating the level set function at all grid nodes for collision check. In contrast, the complexity of our collision handling algorithm is  $O(N_f)$ , where  $N_f$  is the number of the collider mesh faces. As it is usually  $N_f \ll N_{\text{grid}}^3$  (note that  $N_f \approx 20\text{K}$  and  $N_{\text{grid}}^3 \approx 8\text{M}$  in our case), our approach based on a mesh-based collider is extremely more efficient, as well as reflecting a more practical scenario.

**Summary.** Using our tailored MPM-based simulator, we can effectively model the anisotropic dynamics of garments under complex collisions with the underlying body meshes. Later in the experiments (Sec. 5), we show that our simulation scheme leads to SOTA dynamics accuracy.

### 4.3 Learning from Multi-View Videos

We now explain how our avatar outlined in the previous sections can be learned from multi-view video inputs. As a preprocessing step, we first perform 3D mesh tracking on the input frames to capture dense temporal geometry correspondences, which are used to supervise the subsequent learning stages. In particular, we use the mesh tracking algorithm of the existing physics-based avatar work [78], which assumes that a mesh at the first frame  $\mathcal{M}_1 = (\mathbf{V}_1, \mathbf{F})$  is given (e.g., from an off-the-shelf static scene reconstruction method), and optimizes the deformed meshes at the subsequent frames  $(\mathcal{M}_i)_{i=2\dots T}$ , where  $\mathcal{M}_i = (\mathbf{V}_i, \mathbf{F})$ , based on a rendering loss. For more details on the tracking algorithm, we refer the reader to [78]. In the following, we focus on explaining how the physical dynamics (Sec.4.3.1) and appearance (Sec.4.3.2) of our avatar can be learned.

#### 4.3.1 Physical Parameters Learning

We now explain how we learn the physical parameters used to model our garment dynamics. Given the canonical avatar mesh  $\mathcal{M}_1$  obtained from the prior mesh tracking stage, the set of physical parameters associated with  $\mathcal{M}_1$  is defined as  $\mathcal{P} = (E, \nu, \gamma, \kappa, \rho, \alpha)$ , as previously discussed in Sec. 4.1. Note that  $E, \nu, \gamma$ , and  $\kappa$  are material parameters used for our Material Point Method (MPM) [17] simulation. Following the recent inverse physics work based on MPM [76], which found that Young’s modulus  $E$  is the key parameter dominating the dynamic behavior and thus fixed the rest of the parameters, we also fix  $\nu, \gamma, \kappa$  to their default values and focus on learning the other parameters— $E, \rho$ , and  $\alpha$  to mitigate over-parameterization. In the following, we specifically focus on elaborating on our newly introduced parameter  $\alpha$ , used for rest geometry modeling.

**Rest geometry modeling.** MPM [17] internally computes forces based on the deformation gradient relative to the object’s *rest geometry* (i.e., the canonical geometry in the unstressed state, without external forces such as gravity). Note that existing approaches [4, 76] based on MPM typically assume ideal conditions, i.e., having the initial frame correspond to an undeformed rest state, limiting

their applicability in real-world scenarios. In our case, the canonical geometry  $\mathcal{M}_1$  is obtained from real-world observations and is therefore already deformed by gravity. To correct this, we additionally introduce a simple parameter  $\alpha \in [0, 1]$  to compensate for gravity-induced deformation and learn the unseen rest geometry of the avatar. Formally, for each edge vector  $\mathbf{e}$  in  $\mathcal{M}_1$ , we decompose it into two components:  $\mathbf{e}_g$ , the projection of  $\mathbf{e}$  onto the gravity direction  $\mathbf{g}$ , and  $\mathbf{e}_\perp$ , the component orthogonal to  $\mathbf{g}$ , such that  $\mathbf{e} = \mathbf{e}_g + \mathbf{e}_\perp$ . Using  $\alpha$ , we simply model each edge  $\mathbf{e}_{\text{rest}}$  in the rest geometry as:  $\mathbf{e}_{\text{rest}} = \mathbf{e}_\perp + \alpha \mathbf{e}_g$ , where  $\alpha$  determines the extent to which the stretch in the gravity direction is compensated.  $\alpha$  is optimized end-to-end along with other physical parameters via inverse physics.

**Learning physical parameters.** We aim to learn  $E$ ,  $\rho$ , and  $\alpha$  such that the simulated mesh dynamics closely model the real-world garment motions. Given the canonical mesh at the first frame  $\mathcal{M}_1$ , we simulate its future states using our MPM-based simulator (Sec. 4.2) and the physical parameters  $\mathcal{P}$ , resulting in simulated meshes  $(\hat{\mathcal{M}}_i)_{i=2,\dots,T}$  over frames  $[2, T]$ . We then optimize the parameters  $E$ ,  $\rho$ , and  $\alpha$  by minimizing the vertex-to-vertex L2 loss between the simulated meshes and the tracked meshes  $(\mathcal{M}_i)_{i=2,\dots,T}$  capturing geometric dynamics from the input video. Following [78], we perform this optimization using a finite-difference approach. For more implementation details, we kindly refer the reader to Appendix B.1.

### 4.3.2 Appearance Learning

For learning the appearance of our avatar, we optimize the parameters of 3D Gaussian Splats  $\mathcal{G}$  defined on the canonical geometry  $\mathcal{M}_1$  at  $t = 1$ . As discussed in Sec. 4.1, the spatial parameters of  $g_i$  are defined in the local coordinate system with respect to its parent mesh face, allowing it to naturally deform according to the underlying mesh deformations. Using this, we first deform  $\mathcal{G}$  to the other frames at  $t \in \{2, \dots, T\}$  based on the tracked mesh deformations  $(\mathcal{M}_i)_{i=1,\dots,T}$ , and render them across all input views and timesteps. We then compute the loss by measuring the photometric discrepancy between the rendered images and the ground-truth images across all training frames and views. We finally optimize the parameters of  $\mathcal{G}$  via gradient descent to minimize this loss.

Note that the preprocessing mesh tracking stage, which we adopt from [78], also employs 3D Gaussian Splats as a surrogate representation to incorporate rendering loss for optimizing meshes. However, it learns the Gaussian colors *independently for each frame*<sup>1</sup>. In contrast, we learn  $\mathcal{G}$  from all input frames and views to better capture regions that are occluded in some views but visible in others, while still leveraging the previously learned surrogate Gaussian Splats for parameter initialization. Please refer to Appendix B.2 for the details.

**Quasi-shadowing.** When rendering our avatar using  $\mathcal{G}$  via 3D Gaussian Splatting, we additionally apply quasi-shadowing to enhance rendering fidelity. Following the prior work [2], we model self-shadowing by leveraging a neural network trained on ambient occlusion features extracted from the mesh in our hybrid avatar representation. Specifically, we modulate the color of each Gaussian Splat  $g_i$  using a shading scalar  $w_p \in [0, 1]$ , predicted by the network, to obtain the final color.

## 5 Experiments

### 5.1 Experimental Setup

For experimental comparisons, we mainly follow the setup used in the state-of-the-art physics-based avatar work (PhysAvatar [78]) to perform fair comparisons.

**Dataset.** We perform our main evaluations on **(1) ActorsHQ [14]**. In particular, we select four subjects used in [78]: two characters in loose dresses and two characters in two-piece outfits. For training each subject, we use 24 frames with large cloth dynamics for physical parameter learning and 200 frames for appearance learning. For testing, we use 200 unseen frames per subject. Whereas the existing work [70] only uses ActorsHQ for evaluation, we additionally include four sequences from **(2) 4D-DRESS [66]** dataset, to perform more extensive comparisons. We use two subjects in tops and skirts and two in tops and tight jeans. For training, we use 11 frames for physical parameter learning and 100 frames for appearance learning, while testing was carried out on 100 unseen frames.

<sup>1</sup>Note that the existing work [78] with this mesh tracking method discards the surrogate Gaussian Splats and resorts to mesh-based rendering for its avatar.

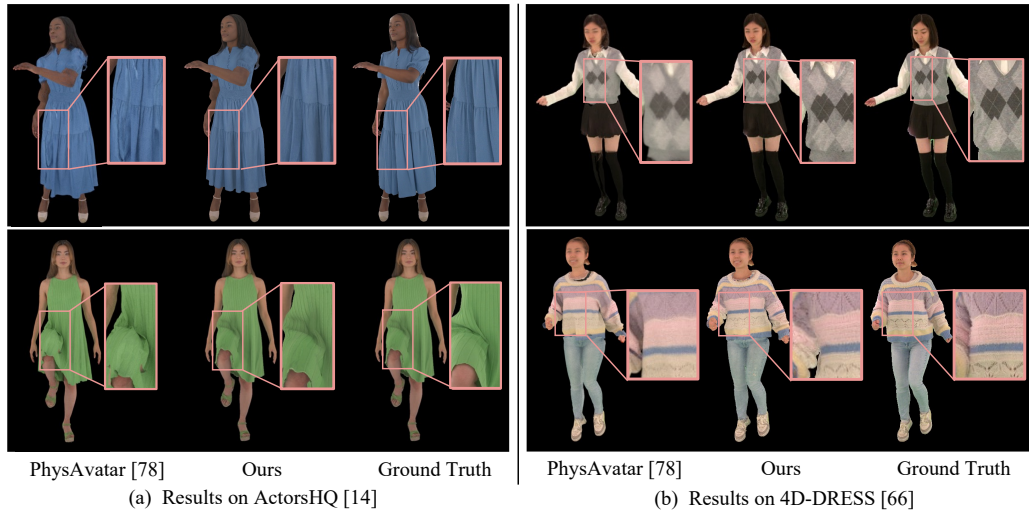


Figure 3. **Qualitative results on test frames in the ActorsHQ [14] and 4D-DRESS [66] datasets.** Our method outperforms PhysAvatar [78] in appearance by rendering sharper, less blurred textures with finer detail and in geometry by recovering folds and wrinkles that more closely match the ground truth.

Table 1. **Quantitative comparisons on ActorsHQ [14] and 4D-DRESS [66] datasets.** Bold indicates the best scores, and underline indicates the second best scores. Our proposed method achieves the best results across all geometry and appearance metrics on both benchmarks.

| Method                                |                         | Geometry<br>CD ( $\times 10^3$ ) ↓ | F-Score ↑   | LPIPS ↓      | Appearance<br>PSNR ↑ | SSIM ↑       |
|---------------------------------------|-------------------------|------------------------------------|-------------|--------------|----------------------|--------------|
| (a) Results on ActorsHQ [14] dataset. |                         |                                    |             |              |                      |              |
| A                                     | ARAH [65]               | 1.12                               | 86.1        | 0.055        | 28.6                 | 0.957        |
| B                                     | TAVA [32]               | 0.66                               | 92.3        | 0.051        | 29.6                 | <u>0.962</u> |
| C                                     | GS-Avatar [12]          | 0.91                               | 89.4        | 0.044        | 30.6                 | <u>0.962</u> |
| D                                     | PhysAvatar [78]         | <u>0.55</u>                        | <u>92.9</u> | <u>0.035</u> | 30.2                 | 0.957        |
| E                                     | <b>MPMAvatar (Ours)</b> | <b>0.42</b>                        | <b>95.7</b> | <b>0.033</b> | <b>32.0</b>          | <b>0.963</b> |
| F                                     | — Anisotropy            | 6.24                               | 90.3        | 0.039        | 28.7                 | 0.957        |
| G                                     | — Physics               | 0.69                               | 92.9        | 0.039        | 31.0                 | 0.962        |
| H                                     | — Shadow                | -                                  | -           | <b>0.033</b> | <u>31.8</u>          | <b>0.963</b> |
| (b) Results on 4D-DRESS [66] dataset. |                         |                                    |             |              |                      |              |
| A                                     | PhysAvatar [78]         | <u>0.37</u>                        | <u>96.6</u> | 0.022        | 33.2                 | 0.976        |
| B                                     | <b>MPMAvatar (Ours)</b> | <b>0.33</b>                        | <b>97.2</b> | <b>0.018</b> | <b>34.1</b>          | <b>0.977</b> |

**Baselines.** We use the same baselines as in [78], which are four open-sourced avatar reconstruction methods: ARAH [65], TAVA [32], GS-Avatar [12], and PhysAvatar [78]. Here, PhysAvatar is the most related baseline to ours, as it is the state-of-the-art work on physics-based avatar. As we already discussed in Sec. 1, PhysAvatar’s simulator fails when driving body mesh colliders have self-penetrations. While the original work manually adjusted the body meshes to avoid simulation failures, these meshes are not publicly available after requests. Therefore, we minimally excluded the faces of the body mesh collider during collision check to prevent their simulation failure.

**Evaluation Metrics.** To assess our dynamics modeling accuracy, we compute Chamfer Distance (CD) [44] and F-Score [64] between the simulated and the ground truth meshes. For F-Score, we set the threshold  $\tau$  to 0.001. For evaluating our rendering accuracy, we measure Learned Perceptual Image Patch Similarity (LPIPS) [75], Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM) between the rendered and the ground truth images.

## 5.2 Experimental Comparisons

**Simulation and rendering accuracy.** Tab. 1(a) (Rows A-E) and Tab. 1(b) show our main comparison results on the ActorsHQ [14] and 4D-DRESS [66] datasets, respectively. Ours achieves the best results across all *geometry* metrics on both benchmarks, validating that the animated geometry using our MPM [17]-based simulation models the most accurate avatar dynamics. Ours also achieves

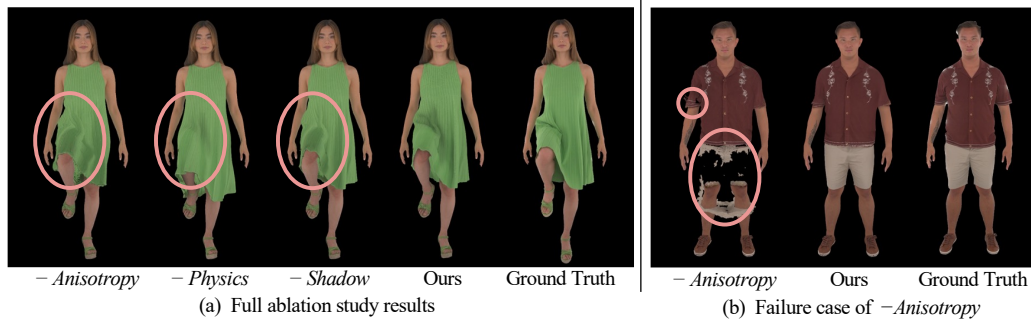


Figure 4. **Qualitative ablation study results.**

the best results across all *appearance* metrics, which shows that 3D Gaussian Splatting [22] with quasi-shadowing with the ambient occlusion prior extracted from a mesh can be an effective rendering scheme. In Fig. 3, we also show the qualitative comparisons against our most competitive baseline: PhysAvatar [78]. Our method captures complex cloth deformations (e.g., subtle wrinkles) more closely to the ground truth, and our rendering based on 3D Gaussian Splatting [22] can more effectively capture high-frequency appearance details (e.g., complex cloth patterns) than PhysAvatar based on mesh-based rendering.

**Robustness and efficiency.** In Tab. 2, we show the simulation success rate and per-frame simulation time on the ActorsHQ [14] benchmark – compared to PhysAvatar [78]. The success rate measures the average ratio of successfully simulated frames to the total number of evaluation frames, where we mark a frame as a failure if the simulation does not terminate within 20 hours for the single frame. For this evaluation, we disabled the *manual* relaxation of the collision check for PhysAvatar, which we originally applied to prevent its simulation failures (Sec. 5.1). The success rate of PhysAvatar is 37.6% while ours is 100%, validating that ours exhibit significantly higher robustness.

For simulation time, we report the average per-frame simulation time over the test sequence. Notably, we re-applied the *manual* relaxation of the collision check in PhysAvatar, as its simulation fails to terminate without this adjustment. As shown in the table, our method achieves a simulation time of 1.1 seconds per frame, compared to 170.0 seconds for PhysAvatar, demonstrating significantly better efficiency. Note that PhysAvatar [78]’s iterative solver (C-IPC [31]) takes a long time to converge for resolving complex collisions, whereas our feed-forward MPM simulator runs much faster.

Table 2. **Simulation robustness and efficiency comparisons with PhysAvatar [78].** Bold indicates the best scores. All scores are evaluated on the Actors-HQ dataset [14].

|   | Method                  | Success Rate (%) $\uparrow$ | Simulation Time (s) $\downarrow$ |
|---|-------------------------|-----------------------------|----------------------------------|
| A | PhysAvatar [78]         | 37.6                        | 170.0                            |
| B | <b>MPMAvatar (Ours)</b> | <b>100.0</b>                | <b>1.1</b>                       |

### 5.3 Ablation Study

**Anisotropic constitutive model.** – *Anisotropy* denotes our method variant which does not use an anisotropic constitutive model [16]. As shown in Tab. 1 (Row F), this variant results in large degradation in dynamics modeling accuracy, as it does not effectively model the manifold-dependent behaviors of cloths. Few cases even have severe tearing artifacts, as shown in Fig. 4b.

**Physical parameters learning.** – *Physics* denotes our method variant where all the physical parameters are fixed to their default values without learning. As shown in Tab. 1 (Row G), this results in suboptimal dynamics modeling accuracy, highlighting the importance of our inverse physics. In Fig. 4, we visually show that this variant leads to less accurate estimation of cloth deformations.

**Quasi-shadowing.** – *Shadow* denotes our method variant where quasi-shadowing is not used for rendering. As shown in Tab. 1 (Row H), this results in degradation in PSNR. In Fig. 4, we qualitatively show that its rendering result exhibits significantly less realism than ours with shadowing.

## 5.4 Application: Zero-shot Scene Interaction

As an additional application, we showcase that our physics-based simulator is zero-shot generalizable to interactions with external objects unseen during training. In the right subfigures of Fig. 1, our avatar garments are naturally deformed as interacting with a chair or sand. This generalizability can be achieved as our physics-based simulator explicitly leverages the physics prior, unlike in learning-based simulators [8, 7] known to be less effective in modeling unseen dynamics. We also note that, owing to the versatility of MPM [17] in handling diverse materials, our framework supports interactions with deformable particles (e.g., sand), while simulators like C-IPC [31] are limited to mesh-based simulations. Please see Appendix A for more interaction examples of ours.

## 6 Conclusion

We presented MPMAvatar, a framework for creating 3D human avatars from multi-view videos that supports (1) physically accurate and robust animation, as well as (2) high-fidelity rendering. Our Gaussian Splat-based avatar is animated based on a carefully tailored MPM-based simulator designed for effective garment dynamics modeling, enabling physically grounded animations.

**Limitations.** Although our avatar outperformed the existing state-of-the-art physics-based avatar method [78] in both appearance and geometry, it does not support relighting as in [78]. Also, for animation, we directly followed [78] and modeled the dynamics of non-garment regions via linear blend skinning, but this can be further improved, e.g., by using strand-based simulation for hair. We refer to Appendix D for a more detailed discussion of limitations and future works.

## 7 Acknowledgement

This work was supported by NST grant (CRC21011, MSIT), IITP grant (RS-2019-II190075, RS-2023-00228996, RS-2024-00459749, RS-2025-25443318, RS-2025-25441313, MSIT) and KOCCA grant (RS-2024-00442308, MCST)

## References

- [1] Rameen Abdal, Wang Yifan, Zifan Shi, Yinghao Xu, Ryan Po, Zhengfei Kuang, Qifeng Chen, Dit-Yan Yeung, and Gordon Wetzstein. Gaussian shell maps for efficient 3d human generation. In *CVPR*, 2024.
- [2] Timur Bagautdinov, Chenglei Wu, Tomas Simon, Fabian Prada, Takaaki Shiratori, Shih-En Wei, Weipeng Xu, Yaser Sheikh, and Jason Saragih. Driving-signal aware full-body avatars. *ACM TOG*, 2021.
- [3] Alexander Bergman, Petr Kellnhofer, Wang Yifan, Eric Chan, David Lindell, and Gordon Wetzstein. Generative neural articulated radiance fields. *NeurIPS*, 2022.
- [4] Junhao Cai, Yuji Yang, Weihao Yuan, Yisheng He, Zilong Dong, Liefeng Bo, Hui Cheng, and Qifeng Chen. Gaussian-informed continuum for physical property identification and simulation. *NeurIPS*, 2024.
- [5] Yue Chen, Xuan Wang, Xingyu Chen, Qi Zhang, Xiaoyu Li, Yu Guo, Jue Wang, and Fei Wang. Uv volumes for real-time rendering of editable free-view human performance. In *CVPR*, 2023.
- [6] Zijian Dong, Xu Chen, Jinlong Yang, Michael J Black, Otmar Hilliges, and Andreas Geiger. Ag3d: Learning to generate 3d avatars from 2d image collections. In *ICCV*, 2023.
- [7] Artur Grigorev, Giorgio Becherini, Michael Black, Otmar Hilliges, and Bernhard Thomaszewski. Contourcraft: Learning to resolve intersections in neural multi-garment simulations. 2024.
- [8] Artur Grigorev, Michael J Black, and Otmar Hilliges. Hood: Hierarchical graphs for generalized modelling of clothing dynamics. In *CVPR*, 2023.
- [9] Chen Guo, Xu Chen, Jie Song, and Otmar Hilliges. Human performance capture from monocular video in the wild. In *3DV*, 2021.
- [10] Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Real-time deep dynamic characters. *ACM TOG*, 2021.

- [11] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *CVPR*, 2020.
- [12] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *CVPR*, 2024.
- [13] Tao Hu, Tao Yu, Zerong Zheng, He Zhang, Yebin Liu, and Matthias Zwicker. Hvtr: Hybrid volumetric-textural rendering for human avatars. In *3DV*, 2022.
- [14] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM TOG*, 2023.
- [15] Alec Jacobson, Zhigang Deng, Ladislav Kavan, and John P Lewis. Skinning: Real-time shape deformation (full text not available). In *ACM SIGGRAPH 2014 Courses*. 2014.
- [16] Chenfanfu Jiang, Theodore Gast, and Joseph Teran. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM TOG*, 2017.
- [17] Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. The material point method for simulating continuum materials. In *ACM SIGGRAPH 2016 Courses*. 2016.
- [18] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Instantavatar: Learning avatars from monocular video in 60 seconds. In *CVPR*, 2023.
- [19] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *CVPR*, 2024.
- [20] HyunJun Jung, Nikolas Brasch, Jifei Song, Eduardo Perez-Pellitero, Yiren Zhou, Zhihao Li, Nassir Navab, and Benjamin Busam. Deformable 3d gaussian splatting for animatable human avatars. *arXiv preprint arXiv:2312.15059*, 2023.
- [21] Couro Kane, Jerrold E Marsden, Michael Ortiz, and Matthew West. Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. *IJNME*, 2000.
- [22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023.
- [23] Minje Kim and Tae-Kyun Kim. Bitt: Bi-directional texture reconstruction of interacting two hands from a single image. In *CVPR*, 2024.
- [24] Minje Kim and Tae-Kyun Kim. Srhand: Super-resolving hand images and 3d shapes via view/pose-aware neral image representations and explicit 3d meshes. In *NeurIPS*, 2025.
- [25] Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. Drucker-prager elastoplasticity for sand animation. *ACM TOG*, 2016.
- [26] Youngjoong Kwon, Baole Fang, Yixing Lu, Haoye Dong, Cheng Zhang, Francisco Vicente Carrasco, Albert Mosella-Montoro, Jianjin Xu, Shingo Takagi, Daeil Kim, Aayush Prakash, and Fernando De la Torre. Generalizable human gaussians for sparse view synthesis. In *ECCV*, 2024.
- [27] Jihyun Lee, Junbong Jang, Donghwan Kim, Minhyuk Sung, and Tae-Kyun Kim. Fourierhandflow: Neural 4d hand representation using fourier query flow. In *NeurIPS*, 2023.
- [28] Jihyun Lee, Minhyuk Sung, Honggyu Choi, and Tae-Kyun Kim. Im2hands: Learning attentive implicit representation of interacting two-hand shapes. In *CVPR*, 2023.
- [29] Jihyun Lee, Weipeng Xu, Alexander Richard, Shih-En Wei, Shunsuke Saito, Shaojie Bai, Te-Li Wang, Minhyuk Sung, Tae-Kyun Kim, and Jason Saragih. Rewind: Real-time egocentric whole-body motion diffusion with exemplar-based identity conditioning. In *CVPR*, 2025.
- [30] Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM TOG*, 2020.
- [31] Minchen Li, Danny M Kaufman, and Chenfanfu Jiang. Codimensional incremental potential contact. *ACM TOG*, 2020.

- [32] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhöfer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. In *ECCV*, 2022.
- [33] Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM TOG*, 2022.
- [34] Yifei Li, Hsiao yu Chen, Egor Larionov, Nikolaos Sarafianos, Wojciech Matusik, and Tuur Stuyck. Diffavatar: Simulation-ready garment optimization with differentiable simulation. In *CVPR*, 2024.
- [35] Zhe Li, Zerong Zheng, Yuxiao Liu, Boyao Zhou, and Yebin Liu. Posevocab: Learning joint-structured pose embeddings for human avatar modeling. In *SIGGRAPH*, 2023.
- [36] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *CVPR*, 2024.
- [37] Junbang Liang, Ming C. Lin, and Vladlen Koltun. Differentiable cloth simulation for inverse problems. In *NeurIPS*, 2019.
- [38] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM TOG*, 2021.
- [39] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeonwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural rendering and reenactment of human actor videos. *ACM TOG*, 2019.
- [40] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. <https://github.com/nvidia/warp>, March 2022. NVIDIA GPU Technology Conference (GTC).
- [41] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. In *MIG*, 2016.
- [42] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, 2019.
- [43] Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quanfeng Lu, Kaipeng Zhang, Yu Cheng, Dianqi Li, Yu Qiao, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. In *ICML*, 2025.
- [44] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019.
- [45] Arthur Moreau, Jifei Song, Helisa Dhamo, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Human gaussian splatting: Real-time rendering of animatable avatars. In *CVPR*, 2024.
- [46] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Rattcliff. Position based dynamics. *J. Vis. Commun. Image Represent.*, 2007.
- [47] Atsuhiko Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *ICCV*, 2021.
- [48] Atsuhiko Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Unsupervised learning of efficient geometry-aware neural articulated representations. In *ECCV*, 2022.
- [49] Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. Ash: Animatable gaussian splats for efficient and photoreal human rendering. In *CVPR*, 2024.
- [50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: an imperative style, high-performance deep learning library. In *NeurIPS*, 2019.
- [51] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019.
- [52] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, 2021.

- [53] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021.
- [54] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *CVPR*, 2024.
- [55] Boxiang Rong, Artur Grigorev, Wenbo Wang, Michael J Black, Bernhard Thomaszewski, Christina Tsalicoglou, and Otmar Hilliges. Gaussian garments: Reconstructing simulation-ready clothing with photorealistic appearance from multi-view video. In *3DV*, 2024.
- [56] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *ICCV*, 2021.
- [57] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. In *CVPR*, 2024.
- [58] Shota Sasaki, Jane Wu, and Ko Nishino. Pbdyg: Position based dynamic gaussians for motion-aware clothed human avatars. *arXiv preprint arXiv:2412.04433*, 2024.
- [59] Alexey Stomakhin, Russell Howes, Craig A Schroeder, and Joseph M Teran. Energetically consistent invertible elasticity. In *SCA*, 2012.
- [60] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. A material point method for snow simulation. *ACM TOG*, 2013.
- [61] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *NeurIPS*, 2021.
- [62] Jiaze Sun, Zhixiang Chen, and Tae-Kyun Kim. Mapconnet: Self-supervised 3d pose transfer with mesh and point contrastive learning. In *ICCV*, 2023.
- [63] Jiapeng Tang, Davide Davoli, Tobias Kirschstein, Liam Schoneveld, and Matthias Niessner. Gaf: Gaussian avatar reconstruction from monocular videos via multi-view diffusion. In *CVPR*, 2025.
- [64] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *CVPR*, 2019.
- [65] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdf. In *ECCV*, 2022.
- [66] Wenbo Wang, Hsuan-I Ho, Chen Guo, Boxiang Rong, Artur Grigorev, Jie Song, Juan Jose Zarate, and Otmar Hilliges. 4d-dress: A 4d dataset of real-world human clothing with semantic annotations. In *CVPR*, 2024.
- [67] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.
- [68] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, 2022.
- [69] Donglai Xiang, Timur Bagautdinov, Tuur Stuyck, Fabian Prada, Javier Romero, Weipeng Xu, Shunsuke Saito, Jingfan Guo, Breannan Smith, Takaaki Shiratori, et al. Dressing avatars: Deep photorealistic appearance for physically simulated clothing. *ACM TOG*, 2022.
- [70] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *CVPR*, 2024.
- [71] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. Monoperfcap: Human performance capture from monocular video. *ACM TOG*, 2018.
- [72] Yinghao Xu, Wang Yifan, Alexander W Bergman, Menglei Chai, Bolei Zhou, and Gordon Wetzstein. Efficient 3d articulated human generation with layered surface volumes. In *3DV*, 2024.
- [73] Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *CVPR*, 2024.
- [74] Youyi Zhan, Tianjia Shao, Yin Yang, and Kun Zhou. Real-time high-fidelity gaussian human avatars with position-based interpolation of spatially distributed mlps. In *CVPR*, 2025.

- [75] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.
- [76] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. In *ECCV*, 2024.
- [77] Fuqiang Zhao, Wei Yang, Jiakai Zhang, Pei Lin, Yingliang Zhang, Jingyi Yu, and Lan Xu. Humannerf: Efficiently generated human radiance field from sparse inputs. In *CVPR*, 2022.
- [78] Yang Zheng, Qingqing Zhao, Guandao Yang, Wang Yifan, Donglai Xiang, Florian Dubost, Dmitry Lagun, Thabo Beeler, Federico Tombari, Leonidas Guibas, et al. Physavatar: Learning the physics of dressed 3d avatars from visual observations. In *ECCV*, 2024.
- [79] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars. In *3DV*, 2025.

## A Additional Results

In the supplementary video (available on the project page), we show additional qualitative results of our experiments.

**Qualitative Comparisons (Sec. A.1).** We present our video results in comparison with PhysAvatar [78], a state-of-the-art physics-based avatar method. Our approach consistently achieves more accurate garment dynamics and higher rendering quality.

**Additional Qualitative Results (Sec. A.2).** We also include additional qualitative results of our method on (1) novel pose driving and (2) zero-shot scene interactions. For novel pose driving, our method accurately models garment deformation driven by motions from the AMASS [42] dataset, which contains relatively more dynamic motions than our training motions in [14, 66]. For novel scene interactions, our method models plausible interactions between garments and a variety of materials (e.g., cushion, sand), as well as mesh-based colliders (e.g., chair, rotor) that were unseen during training. This generalization is attributed to (1) the versatility of MPM [17] and (2) our effective mesh-based collision handling method.

**Ablation Results (Sec. A.3)** In the supplementary video, we additionally validate each of the key components of our method: (1) the constitutive model for anisotropic elastoplasticity [16], (2) physical parameter learning, (3) quasi-shadowing, and (4) rest-geometry modeling. In the fourth ablation, we directly use the canonical geometry as the rest geometry. Specifically, we optimize only Young’s modulus  $E$  and density  $\rho$ , while keeping the rest geometry parameter  $\alpha$  fixed at 1. Each component is shown to be critical for accurate dynamics and appearance modeling.

**Quantitative Evaluation of Physical Plausibility (Sec. A.4).** While we focuses on evaluating geometric and appearance fidelity which are the standard metrics in recent physics-based avatar methods [78, 69], the physical plausibility and contact accuracy are also important aspects to assess. However, quantitatively measuring these properties is challenging due to the lack of ground-truth physical annotations in real-world RGB datasets [14, 66].

Nevertheless, to provide additional insights, we report two metrics that we find to reflect physical plausibility and contact quality in practice. First, we measure the average cloth-body *Penetration Depth*, defined as the mean of  $\max(0, -d)$  where  $d$  is the signed distance from each cloth vertex to the SMPL-X [51] body. As shown in Tab. 3, our method significantly reduces penetration depth over  $6\times$  lower than PhysAvatar [78], indicating better contact handling.

For the second metric, we adopt the *Key Physical Phenomena Detection* metric proposed in Phy-GenBench [43], which computes a plausibility score using a VLM based on how well a given video aligns with physical plausibility prompts. In Tab. 3, we show that our method achieves a plausibility score closer to the ground-truth upper bound than PhysAvatar [78], indicating better alignment with the physical plausibility prompts.

Table 3. **Quantitative evaluation of physical plausibility and contact quality.** To complement standard geometry and appearance metrics, we additionally report two metrics that we find to reflect physical plausibility and contact behavior in practice. Specifically, we measure the average penetration depth (mm) and the Key Physical Phenomena Detection score [43] on the ActorsHQ [14] dataset. Our method achieves significantly lower penetration and a higher plausibility score compared to PhysAvatar [78], indicating improved contact handling and physical realism.

|   | Method           | Penetration Depth (mm) ↓ | Key Physical Phenomena Detection ↑ |
|---|------------------|--------------------------|------------------------------------|
| A | PhysAvatar [78]  | 0.294                    | 1.78                               |
| B | MPMAvatar (Ours) | 0.047                    | 1.83                               |
| C | Ground Truth     | -                        | 1.86                               |

**Additional Ablation Results on Hyperparameters (Sec. A.5)** To demonstrate that our pipeline remains robust across different simulation setups, we conducted an additional ablation study on the ActorsHQ [14] benchmark, where we evaluated the framework’s performance while varying key hyperparameters. In Tab. 4, we observe that variations in (1) time substeps (Rows C-D), (2) physical parameter initialization (Rows E-I), and (3) mesh triangle numbers (Rows J), do not significantly affect the final performance; notably, all variants outperform PhysAvatar [78] by a clear margin across all metrics. Note that for grid resolution, the grid and particle resolutions should be roughly aligned to enable stable momentum transfer between the two representations during MPM simulation, which is also a common convention adopted in many existing MPM-based methods [4, 70]. Therefore, we omitted further ablation on this aspect.

Table 4. **Ablation study on simulation hyperparameters.** To examine the robustness of our pipeline, we ablate key simulation hyperparameters on the ActorsHQ [14] dataset. Specifically, we vary (1) the number of time substeps  $N$  (Rows C–D), testing half and double our default value ( $N = 400$ ), (2) the initialization of physical parameters  $\rho$  and  $E$  (Rows E–I), including  $2\times$  and  $0.5\times$  our defaults ( $\rho = 1.0$ ,  $E = 100$ ) as well as random initialization within plausible ranges, and (3) the number of mesh triangles (Row J), reducing it to one quarter of the original resolution. Across all variants, our method consistently outperforms the prior state of the art [78], showing strong robustness to hyperparameter configurations. Bold indicates the best scores, and underline indicates the second best scores.

|   | Method                            | Geometry               |             | LPIPS ↓      | Appearance  |              |
|---|-----------------------------------|------------------------|-------------|--------------|-------------|--------------|
|   |                                   | CD ( $\times 10^3$ ) ↓ | F-Score ↑   |              | PSNR ↑      | SSIM ↑       |
| A | PhysAvatar [78]                   | 0.55                   | 92.9        | 0.035        | 30.2        | 0.957        |
| B | <b>MPMAvatar (Ours)</b>           | <b>0.42</b>            | <b>95.7</b> | <b>0.033</b> | <u>32.0</u> | <u>0.963</u> |
| C | Ours ( $N = 800$ )                | <b>0.42</b>            | 95.6        | <b>0.033</b> | <u>32.0</u> | 0.963        |
| D | Ours ( $N = 200$ )                | <b>0.42</b>            | <u>95.6</u> | <b>0.033</b> | <u>32.0</u> | <u>0.963</u> |
| E | Ours ( $\rho = 0.5$ , $E = 100$ ) | <b>0.42</b>            | <b>95.7</b> | <u>0.034</u> | <u>32.0</u> | 0.963        |
| F | Ours ( $\rho = 2.0$ , $E = 100$ ) | <b>0.42</b>            | 95.6        | <u>0.034</u> | <u>32.0</u> | <u>0.963</u> |
| G | Ours ( $\rho = 1.0$ , $E = 50$ )  | <b>0.42</b>            | <u>95.6</u> | <u>0.034</u> | <u>32.0</u> | <u>0.963</u> |
| H | Ours ( $\rho = 1.0$ , $E = 200$ ) | <b>0.42</b>            | <b>95.7</b> | <u>0.034</u> | <u>32.0</u> | <u>0.963</u> |
| I | Ours (randomly initialized)       | <u>0.43</u>            | 95.5        | <b>0.033</b> | <u>32.0</u> | <u>0.963</u> |
| J | Ours ( $0.25\times$ triangles)    | 0.44                   | 95.4        | <b>0.033</b> | <b>32.1</b> | <b>0.964</b> |

Table 5. **Quantitative comparison against concurrent baselines on the ActorsHQ [14] dataset.** Bold and underlined values indicate the best and second-best scores, respectively. Our method consistently outperforms recent baselines across all geometry and appearance metrics, highlighting the advantage of physics-based simulation.

|   | Method                  | Geometry               |             | LPIPS ↓      | Appearance  |              |
|---|-------------------------|------------------------|-------------|--------------|-------------|--------------|
|   |                         | CD ( $\times 10^3$ ) ↓ | F-Score ↑   |              | PSNR ↑      | SSIM ↑       |
| A | Gaussian Garments [55]  | 2.39                   | 86.5        | 0.042        | <u>29.5</u> | <u>0.959</u> |
| B | MMLPHuman [74]          | <u>0.47</u>            | <u>94.9</u> | <u>0.039</u> | 29.3        | 0.954        |
| C | <b>MPMAvatar (Ours)</b> | <b>0.42</b>            | <b>95.7</b> | <b>0.033</b> | <b>32.0</b> | <b>0.963</b> |

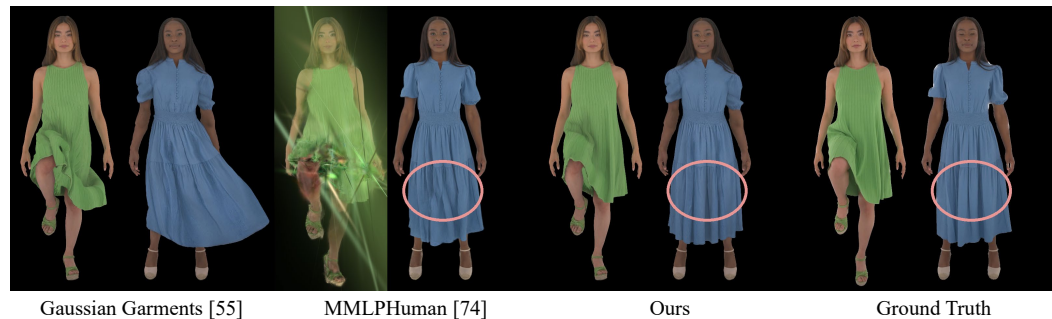


Figure 5. **Qualitative comparison against concurrent baselines on the ActorsHQ [14] dataset.** We compare our method with two recent concurrent methods: Gaussian Garments [55] and MMLPHuman [74]. Gaussian Garments [55] struggles to produce physically accurate deformations, while MMLPHuman [74] exhibits unnatural surface artifacts or discontinuities under challenging poses. In contrast, our method yields more realistic and plausible garment dynamics and geometry.

**Comparison with Additional Concurrent SOTA Baselines (Sec. A.6)** To further validate our approach, we additionally compare our method against two recent concurrent state-of-the-art avatar reconstruction methods: **Gaussian Garments** [55] and **MMLPHuman** [74]. Gaussian Garments replaces explicit simulation with a learned dynamics module based on graph neural networks, while MMLPHuman models geometry and appearance solely as a function of pose using multiple MLPs.

For both Gaussian Garments [55] and MMLPHuman [74] experiments, we ensure a fair comparison by aligning key experimental components with our method. For Gaussian Garments [55], we replace our MPM-based simulation with the learning-based garment simulator from ContourCraft [7], which serves as the simulation backend of Gaussian Garments [55], while keeping the tracked meshes and rendering pipeline identical to ours. Since Gaussian Garments [55] models the human body using only SMPL-X [51] without explicit body appearance modeling, we also use our body modeling setup to ensure a consistent evaluation environment. For MMLPHuman [74], we follow the official

implementation and evaluation protocol, and additionally use the same template mesh as in our method.

As shown in Fig. 5 and Tab. 5, our method outperforms both baselines across all geometry and appearance metrics. The learned simulator in Gaussian Garments [55] struggles to capture physical laws under our setting, where physical parameters must be estimated from only one second of motion, leading to high geometric error. Meanwhile, MMLPHuman [74] lacks explicit surface modeling and physical understanding, producing unrealistic surface artifacts or broken geometry when encountering unseen poses. These results demonstrate the advantage of our physics-based modeling pipeline for reconstructing accurate and plausible dynamic avatars.

## B Implementation Details

### B.1 Physics-Based Dynamics Modeling

**Collision handling.** In Alg. 1, we outline our collision handling algorithm. After parameter initialization (lines 1-3), we iterate over every face  $f$  in the collider mesh and transfer its velocity  $\mathbf{v}_f$  and normal  $\mathbf{n}_f$  to nearby grid nodes based on B-Spline weights (lines 5-11). This yields the extended velocity  $\mathbf{v}_i^c$  and normal  $\mathbf{n}_i^c$  of the collider at grid node  $i$ . Then, given the velocity of the simulating object  $\mathbf{v}_i$  at grid node  $i$ , if the vector  $\mathbf{v}_i - \mathbf{v}_i^c$  points inward, we project out the normal component — keeping only the tangential part — to model collision (lines 13–24). Note that this entire procedure runs in  $O(N_f)$  time, as collision checks become simple B-Spline weight lookups rather than costly level-set queries at all grid nodes.

---

**Algorithm 1** Collision Handling

---

```

1: for each  $i$  in a set of grid node indices do
2:    $\mathbf{v}_i^c, \mathbf{n}_i^c \leftarrow \mathbf{0}, \mathbf{0}$ ; ▷ Initialize the zero-value grids for collider
3: end for
4:
5: for each  $f$  in a set of collider mesh faces do
6:   for  $i$  in a set of neighboring grid nodes of  $\mathbf{x}_f$  do
7:      $w_{if}^c \leftarrow \text{Bspline}(\mathbf{x}_f, \mathbf{x}_i)$ ; ▷ Compute interpolation weights using B-spline kernel
8:      $\mathbf{v}_i^c \leftarrow \mathbf{v}_i^c + w_{if}^c \mathbf{v}_f$ ;
9:      $\mathbf{n}_i^c \leftarrow \mathbf{n}_i^c + w_{if}^c \mathbf{n}_f$ ;
10:   end for
11: end for
12:
13: for each  $i$  in a set of grid node indices do
14:    $w_i^c \leftarrow \sum_f w_{if}^c$ 
15:   if  $w_i^c > 0$  then ▷ Detect whether a collision has occurred
16:      $\mathbf{v}_i^r \leftarrow \mathbf{v}_i^c / w_i^c$ ;
17:      $\mathbf{n}_i^c \leftarrow \mathbf{n}_i^c / \|\mathbf{n}_i^c\|$ ;
18:      $\mathbf{v}_i^{rel} \leftarrow \mathbf{v}_i - \mathbf{v}_i^r$ ; ▷ Transform velocities into the collider's reference frame
19:     if  $\mathbf{v}_i^{rel} \cdot \mathbf{n}_i^c < 0$  then ▷ Check if the relative velocity points inward toward the collider
20:        $\mathbf{v}_i^{rel} \leftarrow \mathbf{v}_i^{rel} - (\mathbf{v}_i^{rel} \cdot \mathbf{n}_i^c) \mathbf{n}_i^c$ ; ▷ Project relative velocity onto the collider's tangent space
21:     end if
22:      $\mathbf{v}_i \leftarrow \mathbf{v}_i^{rel} + \mathbf{v}_i^r$  ▷ Transform velocities back into world frame
23:   end if
24: end for

```

---

**Physical parameters learning.** As discussed in Sec. 4.3.1 in the paper, we optimize Young’s modulus  $E$ , density  $\rho$ , and rest geometry parameter  $\alpha$  by simulating the first-frame canonical mesh  $\mathcal{M}_1 = (\mathbf{V}_1, \mathbf{F})$  and minimizing the vertex-wise  $L_2$  error with respect to the tracked meshes  $(\mathcal{M}_i)_{i=2, \dots, T}$ , where  $\mathcal{M}_i = (\mathbf{V}_i, \mathbf{F})$ .

In particular, let the initial mesh vertices be  $\hat{\mathbf{V}}_1 = \mathbf{V}_1$ . Then, for each subsequent frame, we obtain the simulated mesh vertices via

$$\hat{\mathbf{V}}_{i+1} = \text{MPM}(\hat{\mathbf{V}}_i, \mathbf{V}_1, \mathbf{F}, \mathcal{P}), \quad (2)$$

where  $\mathcal{P} = (E, \nu, \gamma, \kappa, \rho, \alpha)$  are our physical parameters.

Here, the canonical vertices  $\mathbf{V}_1$  and faces  $\mathbf{F}$  provide the fixed mesh topology used to compute edge vectors and material directions, which in turn define the deformation gradients for the anisotropic constitutive model inside the MPM simulation.

We then perform gradient-based optimization for  $\rho$ ,  $E$  and  $\alpha$ , such that they minimize the loss

$$\mathcal{L}_{\text{phys}}(\mathcal{P}) = \sum_{i=2}^T \|\hat{\mathbf{V}}_i - \mathbf{V}_i\|^2. \quad (3)$$

Here, each parameter's gradient is approximated using finite differences – following PhysAvatar [78]. For example, the gradient with respect to density  $\rho$  is computed as

$$\frac{\partial \mathcal{L}_{\text{phys}}}{\partial \rho} \approx (\mathcal{L}_{\text{phys}}(E, \nu, \gamma, \kappa, \rho + \Delta\rho, \alpha) - \mathcal{L}_{\text{phys}}(E, \nu, \gamma, \kappa, \rho, \alpha)) / \Delta\rho, \quad (4)$$

where  $\Delta\rho$  is the perturbation size.

### Training details.

Our MPM simulation uses a time step of  $\Delta t = 0.04$  with  $N = 400$  substeps and a grid resolution of 200. We optimize the physical parameters over 200 iterations using the Adam optimizer. For finite-difference gradient estimation, the perturbation sizes are set to  $\Delta\rho = 0.05$ ,  $\Delta E = 5$ , and  $\Delta\alpha = 0.005$ . The corresponding learning rates are 0.01 for  $\rho$ , 0.3 for  $E$ , and 0.01 for  $\alpha$ . All parameters are initialized as  $\rho = 1.0$ ,  $E = 100$ , and  $\alpha = 1.0$  for physical parameter learning, while  $\nu$ ,  $\gamma$ , and  $\kappa$  are fixed at their default values of 0.3, 500, and 500, respectively.

Following PhysAvatar [78], we adopt a stage-wise training scheme: the physical parameters are optimized first and the tracked meshes remain fixed throughout. Appearance learning is then performed independently based on the same tracked meshes (see Sec. 4.3).

**Simulation time and computing resource.** As noted in the main paper (Sec. 5.2), our simulation runs at approximately 1.1 seconds per frame on a single NVIDIA GeForce RTX 4090.

## B.2 Appearance Learning

For the appearance learning (Sec. 4.3.2 in the paper), we leverage the dense temporal correspondences from the tracked meshes  $(\mathcal{M}_i)_{i=1\dots T}$  to transform the canonical Gaussians  $\mathcal{G}$  (defined in the first frame) into each subsequent frame. Following [54], we compute the transformations that carries every Gaussian from its parent triangle in the canonical mesh to the corresponding triangle in the target frame. This allows us to render all training frames  $[1, \dots, T]$  using a single shared appearance model  $\mathcal{G}$ , such that it can be learned jointly from all input views and frames by minimizing:

$$\mathcal{L}_{\text{app}} = \mathcal{L}_{\text{rgb}} + \lambda_p \mathcal{L}_{\text{position}} + \lambda_s \mathcal{L}_{\text{scaling}}. \quad (5)$$

Here,  $\mathcal{L}_{\text{rgb}}$  measures the photometric discrepancy between the rendered and the ground truth images over all frames and views, and is defined as a weighted sum of L1 loss  $\mathcal{L}_1$ , SSIM [67] loss  $\mathcal{L}_{\text{SSIM}}$ , and LPIPS [75] loss  $\mathcal{L}_{\text{LPIPS}}$ :

$$\mathcal{L}_{\text{rgb}} = \lambda_1 \mathcal{L}_1 + \lambda_{\text{SSIM}} \mathcal{L}_{\text{SSIM}} + \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}}. \quad (6)$$

In Eq. 5,  $\mathcal{L}_{\text{position}} = \|\max(\mu, \epsilon_p)\|_2$  and  $\mathcal{L}_{\text{scaling}} = \|\max(s, \epsilon_s)\|_2$  regularize the location offset  $\mu$  and the scale  $s$  of each Gaussian not to exceed the thresholds  $\epsilon_p = 1.0$  and  $\epsilon_s = 0.6$ . This is to encourage the Gaussians to remain closely aligned with their parent triangle structures.

When optimizing the parameters of  $\mathcal{G}$ , we follow the standard 3DGS optimization procedure [22] and employ adaptive density control to increase the number of Gaussians in regions with high reconstruction error. For the loss weighting hyperparameters, we set  $\lambda_1 = 0.8$ ,  $\lambda_{\text{SSIM}} = 0.2$ ,  $\lambda_{\text{LPIPS}} = 0.2$ ,  $\lambda_p = 1.0$ , and  $\lambda_s = 1.0$ .

## C Anisotropic Constitutive Model

In this section, we provide additional details on the anisotropic constitutive model used in our MPM simulation framework, to complement the explanation provided in Sec. 3 and Sec. 4.2.1 of the main paper. This elaboration aims to help readers better understand how our method captures direction-dependent garment dynamics.

As explained in Sec. 3, we employ the Material Point Method (MPM) [17] to simulate the evolution of deformable objects by solving two governing equations: conservation of mass and conservation of momentum (Eq. 1). Among these, conservation of momentum governs the time evolution of velocity, and its simulation hinges on the computation of the Cauchy stress tensor  $\boldsymbol{\sigma}$ . This stress tensor depends on the deformation gradient  $\mathbf{F}$  and the strain-energy density function  $\psi$  via

$$\boldsymbol{\sigma} = \frac{1}{\det(\mathbf{F})} \frac{\partial \psi}{\partial \mathbf{F}} \mathbf{F}^\top,$$

where  $\psi$  is defined by a material-specific constitutive model.

As introduced in Sec. 4.2.1, we adopt the anisotropic constitutive model proposed by Jiang et al. [16], which is particularly well-suited for modeling thin, codimensional structures like garments. This model captures how cloth exhibits strong resistance to compression and shearing along the surface normal while remaining flexible along in-plane directions.

To compute the deformation gradient  $\mathbf{F}$  at each particle, the model uses local material directions derived from a Lagrangian mesh. Specifically,

$$\mathbf{F} = \mathbf{d}\mathbf{D}^{-1},$$

where  $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3]$  denotes the canonical (undeformed) material directions and  $\mathbf{d} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]$  denotes the corresponding deformed directions. Since the strain energy function  $\psi$  must be invariant under rotations, the model applies QR decomposition  $\mathbf{F} = \mathbf{Q}\mathbf{R}$  and reparameterizes the energy as a function of the upper-triangular matrix  $\mathbf{R}$ :

$$\psi(\mathbf{F}) = \hat{\psi}(\mathbf{R}) = \hat{\psi}_{\text{normal}} + \hat{\psi}_{\text{shear}} + \hat{\psi}_{\text{in-plane}},$$

where each term independently penalizes a specific type of deformation.

The normal component penalizes compression along the surface normal:

$$\hat{\psi}_{\text{normal}}(\mathbf{R}_{33}|\kappa) = \begin{cases} \frac{\kappa}{3}(1 - \mathbf{R}_{33})^3 & \text{if } \mathbf{R}_{33} \leq 1, \\ 0 & \text{otherwise,} \end{cases}$$

reflecting the assumption that cloth is typically surrounded by air and thus can freely expand but should resist compression.

The shear component penalizes off-diagonal shear deformation between in-plane and normal directions:

$$\hat{\psi}_{\text{shear}}(\mathbf{R}_{13}, \mathbf{R}_{23}|\gamma) = \frac{\gamma}{2}(\mathbf{R}_{13}^2 + \mathbf{R}_{23}^2),$$

which stabilizes the material frame by discouraging bending or tilting out of plane.

The in-plane component models isotropic stretching within the tangent plane using a fixed-corotated formulation:

$$\hat{\psi}_{\text{in-plane}}(\mathbf{R}_{11}, \mathbf{R}_{12}, \mathbf{R}_{22}|E, \nu) = \frac{E}{2(1+\nu)}((\sigma_1 - 1)^2 + (\sigma_2 - 1)^2) + \frac{E\nu}{2(1+\nu)(1-2\nu)}(\sigma_1\sigma_2 - 1)^2,$$

where  $\sigma_1, \sigma_2$  are the singular values of the in-plane matrix

$$\mathbf{R}^{2 \times 2} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ 0 & \mathbf{R}_{22} \end{bmatrix}.$$

This overall anisotropic formulation provides the strain-energy density  $\psi$  needed to compute the stress tensor  $\boldsymbol{\sigma}$  in the momentum equation (Eq. 1b), thereby enabling our simulator to capture realistic garment behavior with directionally varying stiffness. This detailed model plays a central role in achieving the accurate and physically plausible dynamics demonstrated in our results.

## D Limitations and Future Work

While our method achieves state-of-the-art performance in both appearance and physical dynamics modeling, we acknowledge several limitations and outline potential directions for future work.

**Scalability of Finite-Difference Optimization.** Our physical parameter optimization adopts a finite-difference scheme, which scales linearly with the number of parameters. While this remains practical for our current setting, where per-garment material parameters suffice due to limited intra-garment heterogeneity, extending to fine-grained parameterizations (e.g., per-vertex) would increase computational cost. As a mitigation strategy, incorporating differentiable simulators [37, 33] may improve scalability in future applications.

**Relighting.** Our current framework does not support relightable rendering. However, recent methods [19, 57] have proposed relighting-aware extensions for Gaussian avatars, and our hybrid representation is compatible with such techniques. We consider this a promising direction to further enhance rendering realism.

**Occlusion-Aware Generalization.** Our pipeline directly optimizes appearance only in regions visible in the multi-view training frames. Consequently, when previously occluded or unseen parts (e.g., the back side of the avatar) become visible under novel poses or viewpoints, rendering quality may degrade. Recent works have explored generative priors to inpaint unobserved regions [26], or diffusion-based view synthesis to generate pseudo multi-view supervision from monocular videos [63]. Incorporating such approaches into our pipeline could improve generalization to occluded or unseen regions.

## E Societal Impact

Our method enables physically accurate dynamic human avatar reconstruction from multi-view videos, supporting a wide range of applications in virtual reality, digital fashion, and entertainment. However, the capability to generate lifelike avatars also introduces potential risks, such as the misuse of the technology for creating deepfakes or other forms of deceptive content. When publishing our code, we will consider embedding traceable digital watermarks or developing authentication mechanisms to ensure the responsible use of generated avatars.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: Yes, we summarized the paper's contributions and scope in the abstract and the introduction, and our claims match the experimental results.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We explicitly acknowledge in the conclusion (Sec. 6) that our method does not support relighting and that our dynamics modeling for non-garment regions can be further improved.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our work is application-focused and does not present any new theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our primary contributions are the novel simulation-based avatar framework. All components of this framework are described in the Method section (Sec. 4), and the implementation details are provided in the supplementary document.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: While the code was not submitted with the paper due to the need for further cleanup, we plan to release it before publication to allow full reproducibility of our experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Our training and test protocols are outlined in the Experimental Setup section (Sec. 5.1), with further details provided in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We were unable to perform comprehensive significance testing or include error estimates due to computational resource limitations. However, we will include them in the revision.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report our per-frame simulation time in Sec. 5 and include the full details of our resources (including memory specifications) in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Our study uses publicly available datasets under proper access agreements and does not involve sensitive personal data or human subjects. All experiments were conducted with standard computational resources and no practices violate privacy, fairness, or environmental guidelines outlined in the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: As our work enables realistic human avatar creation, we acknowledge that it can be potentially misused for deepfake generation. We discuss these concerns in the supplementary material.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not involve releasing any high-risk pre-trained models or sensitive datasets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited all the assets used in this work, including the datasets.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: Our work does not involve releasing new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

#### 16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Our core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.