

Interpolated Adaptive Linear Reduced Order Modeling for Deformation Dynamics

Yutian Tao^{1,2}, Maurizio Chiaramonte², Pablo Fernandez²

¹University of Wisconsin–Madison, USA

²Meta, USA

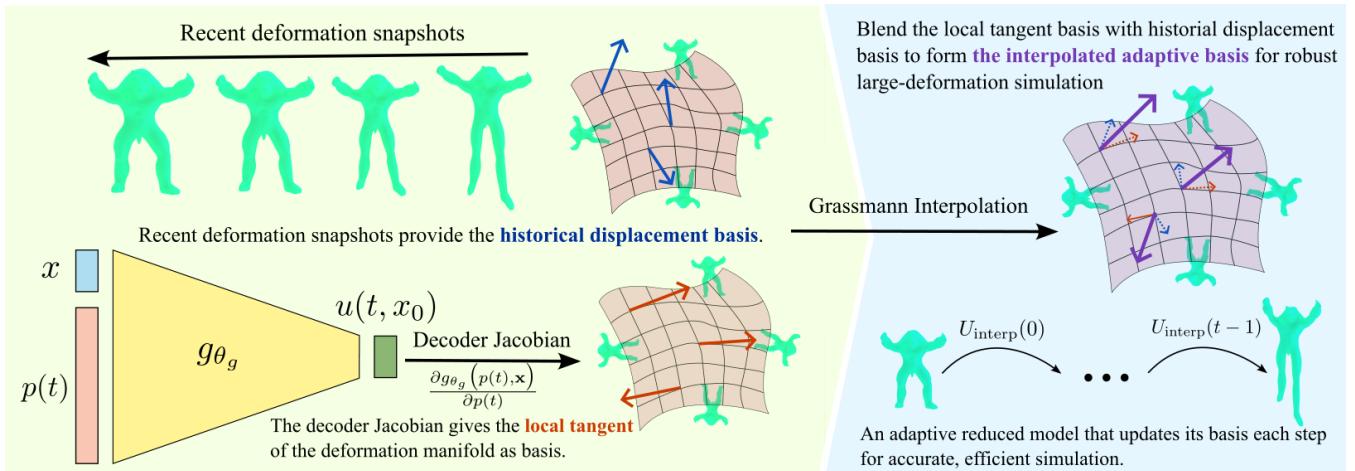


Figure 1: We propose a method that forms an adaptive basis for robust, efficient simulation under large deformations by blending the local tangent of the deformation manifold—learned from the decoder’s gradient—with recent displacement history. Our approach achieves 40–70% lower online error than PCA-based ROM, with per-step costs only 1.3–1.5× higher, while maintaining interactive runtimes.

Abstract

Linear reduced-order modeling (ROM) is widely used for efficient simulation of deformation dynamics, but its accuracy is often limited by the fixed linearization of the reduced mapping. We propose a new adaptive strategy for linear ROM that allows the reduced mapping to vary dynamically in response to the evolving deformation state, significantly improving accuracy over traditional linear approaches. To further handle large deformations, we introduce a historical displacement basis combined with Grassmann interpolation, enabling the system to recover robustly even in challenging scenarios. We evaluate our method through quantitative online-error analysis and qualitative comparisons with principal component analysis (PCA)-based linear ROM simulations, demonstrating substantial accuracy gains while preserving comparable computational costs.

CCS Concepts

- Computing methodologies → Physical Simulation; Dimensionality reduction and manifold learning;

1. Introduction

Efficient and accurate simulation of deformable objects remains a central challenge in computer graphics, with applications in animation, virtual and augmented reality, and robotics. A common strategy to achieve interactive performance is reduced-order modeling (ROM), which projects high-dimensional dynamics onto a low-dimensional subspace, dramatically reducing the number of

degrees of freedom (DOFs) while preserving important features. Traditional linear subspace models—constructed from modal analysis or data-driven bases such as PCA—are especially attractive for their simplicity and speed. However, their accuracy is fundamentally constrained by linearization: when deformations are large or highly nonlinear, a fixed reduced mapping cannot capture the

evolving state, leading to drift, loss of detail, and inability to recover to the undeformed rest configuration.

Researchers have explored nonlinear and data-driven approaches that learn more expressive low-dimensional manifolds to overcome these limitations. Autoencoder-based methods and implicit neural representations demonstrate that compact latent spaces can capture complex deformations beyond the reach of linear subspaces. Among these, Continuous ROM (CROM) [CXC²²] leverages encoder-decoder neural fields to map between latent coordinates and continuous displacement fields, offering discretization flexibility and improved fidelity. Despite their expressiveness, many nonlinear ROMs incur nontrivial per-step costs—such as evaluating full-order residuals or Jacobians—which limit their practical speed.

A complementary direction introduces adaptivity into the reduced space itself. Basis enrichment and localized corrections can recover accuracy when previously unseen deformation patterns arise, balancing efficiency and fidelity. However, triggering and integrating these adaptations robustly remains challenging; naive heuristics may over-refine, introduce instabilities, or erode the computational advantage.

To address these challenges, we propose an interpolated adaptive linear ROM for deformation dynamics that combines the efficiency of linear subspaces with a decoder-based representation inspired by CROM, enabling a time-varying reduced mapping while remaining compatible with continuous neural frameworks. Our method derives a time-dependent reduced basis from the decoder Jacobian (gradient) evaluated at the current latent state, which naturally aligns the subspace with the local tangent of the learned nonlinear manifold while retaining the computational structure of linear ROM. To further improve robustness under large deformations, we introduce a historical displacement basis and blend it with the current adaptive basis through Grassmann interpolation, which smoothly transitions between the two sets of basis vectors—like morphing one subspace into another—while preserving their geometric structure and capturing important features from both recent history and the current deformation. This history-aware interpolation enables accurate recovery to the rest state after extreme deformations, overcoming the drift that arises when using only the decoder gradient. In summary, our main contributions are:

- **Adaptive linear subspace method:** Extends CROM by efficiently using the decoder Jacobian to construct a time-varying reduced mapping aligned with the evolving deformation state, adding only negligible overhead to the standard ROM pipeline.
- **History-aware Grassmann interpolation:** Blends a historical displacement basis with the current adaptive basis to enable perfect recovery to the undeformed rest configuration even under large deformations.
- **Comprehensive evaluation:** Demonstrates **40–70%** lower online error than PCA-based ROM while maintaining interactive runtime with per-step costs within **1.3–1.5×** that of PCA, supported by quantitative metrics and qualitative visual comparisons across diverse objects and challenging scenarios.

2. Related Work

Linear Subspace Model Reduction. Linear subspace model reduction has long been a standard approach for accelerating physical simulations in computer graphics and engineering, particularly in deformation dynamics. Early work on modal analysis [PW89] showed that an object’s motion could be efficiently represented as a superposition of a small number of vibration modes, later extended with constraints [HSO03], modal warping for large rotations [CK05], and modal derivatives that enrich the basis with quadratic mode couplings to capture large deformations [BJ05]. These extensions established modal analysis as a practical real-time tool for deformation dynamics. At the same time, data-driven *Principal Component Analysis* (PCA)-based subspace methods emerged. Rather than relying on vibration modes alone, PCA is applied to simulation snapshots to extract dominant deformation patterns. Early papers demonstrated the feasibility of data-driven reductions for elastostatics [JP99], character animation [KJP02], and finite element simulations of soft bodies [KLM01], where PCA or other snapshot-based decompositions were used to construct low-dimensional bases.

Beyond deformation, linear subspace reduction has been successfully applied to other domains in graphics. In fluids, PCA-derived flow bases enable real-time incompressible simulation [TLP06]. Linear subspaces also support reduced sound simulation [OSG02, JBP06], shape deformation [VTSSH13, BvTH16], and computational design [XLCB15, UMK17, MHR^{*}16]. However, these methods suffer from critical accuracy limitations: if a deformation or motion lies outside the span of the chosen subspace basis, the reduced simulation simply cannot represent it.

Model Reduction with Neural Networks. Recent years have seen rapid progress in machine learning approaches for reduced-order modeling. These methods use neural networks to construct compact nonlinear manifolds that serve as effective reduced simulation spaces. A main class of methods relies on autoencoders to encode deformation snapshots into low-dimensional latent vectors. Nonlinear manifold reduction for PDEs was introduced through autoencoders [LC20], and this concept was extended to deformable solids by learning latent dynamics in reduced coordinates [FMD^{*}19]. To enhance accuracy for highly nonlinear deformation, a deep autoencoder with high-order differentiability—enabling computation of Hessian-based dynamics—was proposed [SYS^{*}21]. Correction techniques further refine these models; for instance, target-specific adjustments have been proposed to handle contact phenomena [RCCO23]. Beyond ROM for solids, neural networks have been applied to a wide range of graphics problems to learn expressive low-dimensional latent spaces. Latent-space physics frameworks use autoencoders and recurrent networks to evolve fluid states entirely in a low-dimensional space, achieving dramatic speedups while retaining physical accuracy [KAT^{*}19, WBT19]. Similar principles have been applied to deformable bodies, where neural subspace dynamics enable fast interactive simulation by learning physical updates directly in a reduced space [HDDN19, LLL^{*}25]. In character animation, periodic autoencoders learn compact motion-phase manifolds for realistic motion synthesis [SMK22].

A complementary path uses implicit neural representations, en-

abling discretization-independent model reduction. Some frameworks explore mesh-agnostic mappings that predict piecewise linear deformations without requiring shared triangulations [AGK^{*}22]. Continuous Reduced-Order Modeling proposed a continuous neural-field formulation for PDEs [CXC^{*}22], and subsequent work explored discretization-free subspace representations that allow latent dynamics to be learned independently of the underlying mesh [CCG^{*}23], linear continuous reduction models that directly learn spatially smooth mappings from undeformed to deformed states [CCW^{*}23], and neural stress field reduction that encodes high-dimensional stress responses into compact implicit neural fields [ZLL^{*}23]. Together, these approaches highlight the versatility of neural latent representations in graphics, which can operate not only on displacements but also on internal quantities such as strain, stress, fluid velocity, or motion phases. While these neural field-based methods generally achieve higher accuracy and flexibility than linear subspace models, they also tend to be more computationally demanding than traditional linear reduction methods.

Adaptive Subspace Methods for Dynamics. A major limitation of fixed subspace simulation is the inability to represent unanticipated local deformations or contact events. To address this, a variety of adaptive subspace methods have been proposed that dynamically update the reduced basis during simulation time. One family of approaches focuses on global basis expansion. An early example is the online reduction strategy, which interleaves subspace solves with occasional full-space solves to enrich the basis whenever the reduced model diverges significantly from the ground truth [KJ09]. This strategy minimizes expensive full solves while adaptively expanding the snapshot basis. In cloth simulation, a related technique precomputes a large dictionary of basis vectors and adaptively selects subsets depending on the garment’s pose, thereby reproducing diverse folding patterns with a small active subspace [HTC^{*}14].

Another direction is localized subspace enrichment, where the reduced basis is extended only in regions of high deformation or contact. A notable method augments modal bases with localized displacement fields generated on demand at collision sites [HZ13]. Hybrid approaches further combine subspace simulation with learned corrections: neural models can predict contact-driven deformations and inject them into a handle-based subspace, efficiently capturing both broad dynamics and fine-scale contact detail [RCPO21, RCCO22]. Adaptive rigidification offers a different perspective by dynamically reducing the number of elastic degrees of freedom and treating nearly static regions as rigid to accelerate computation [MAKWL22]. The Trading Spaces method generalizes these families by introducing an oracle-driven time integration scheme that dynamically switches between subspace simulation, nodal enrichments, and even full-space solves as necessary [TFLK24].

These adaptive methods generally deliver substantial speedups compared to fixed reduced spaces while improving accuracy for large deformations and contact. However, they also face challenges. Oracles used to decide when and where to adapt are often simplistic, focusing only on energy error [KJ09] or applying enrichment broadly at detected contacts [HZ13, TMDK15], which can either miss important events or lead to over-refinement. Furthermore, switching between models on a per-time-step basis can pro-

duce popping artifacts if transitions are not handled consistently. Unlike these enrichment strategies, our method adapts the reduced mapping continuously by following the local gradient of a learned decoder, avoiding the need for explicit oracle-based enrichment.

3. Technical Background

In this section we review the two principal ROM approaches that underpin our work: linear ROM and CROM. Both have been widely used for simulating soft-body dynamics, yet they differ fundamentally in their assumptions, expressiveness, and computational trade-offs.

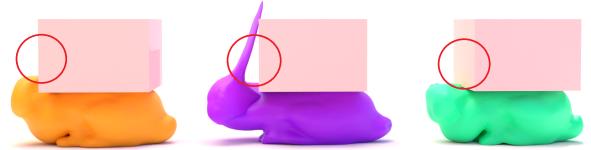


Figure 2: Bunny compression under large deformation: PCA (middle) exhibits noticeable artifacts compared to FOM and ours.

3.1. Linear Reduced-Order Modeling

The simulation of deformable soft bodies is a core problem in computer graphics and computational mechanics. When discretized using the finite element method (FEM), the motion of a soft body is governed by Newton’s second law

$$M\ddot{u}(t) = f_{\text{int}}(u(t)) + f_{\text{ext}}(t), \quad (1)$$

where $u(t) \in \mathbb{R}^{3n}$ is the vector of nodal displacements for a mesh with n vertices, $M \in \mathbb{R}^{3n \times 3n}$ is the mass matrix, $f_{\text{int}}(u(t))$ is the generally nonlinear internal force, and $f_{\text{ext}}(t)$ represents external forces such as gravity, collisions, or user interactions. For realistic models n can reach tens of thousands, making full-order model prohibitively expensive for interactive applications.

Linear ROM alleviates this cost by approximating the high-dimensional state $u(t)$ with a much lower-dimensional representation. The core idea is to project the dynamics onto a fixed linear subspace spanned by a small set of basis vectors,

$$u(t) = Uq(t), \quad (2)$$

where $U \in \mathbb{R}^{3n \times r}$ contains the basis vectors as columns and $q(t) \in \mathbb{R}^r$ are the reduced coordinates with $r \ll n$.

The basis U is typically constructed in a data-driven manner. Principal Component Analysis (PCA) is commonly applied to a set of simulated or captured deformations to extract the dominant modes of variation. These modes capture the most representative deformation patterns and provide an efficient subspace for runtime simulation.

Substituting (2) into (1) and premultiplying by U^T yields the reduced equations of motion

$$\tilde{M}\ddot{q}(t) = \tilde{f}_{\text{int}}(q(t)) + U^T f_{\text{ext}}(t), \quad (3)$$

where $\tilde{M} = U^T M U$ is the reduced mass matrix and $\tilde{f}_{\text{int}}(q) = U^T f_{\text{int}}(U q)$ is the reduced internal force.

This reduction dramatically lowers computational cost and enables real-time simulation and control of soft bodies. Linear ROM performs well when deformations remain moderate and can be represented accurately within a fixed subspace, such as small vibrations, bending, or stretching. However, accuracy degrades when the system exhibits strongly nonlinear behavior or large deformations as shown in Figure 2, since the fixed basis cannot capture the evolving dynamics. Moreover, because the basis is tied to a particular mesh, changes in topology or even vertex sampling require regenerating the basis, limiting flexibility and reusability. For a broader review of linear ROM in graphics and mechanics, see [SB12].

3.2. Continuous Reduced-Order Modeling

To overcome the limitations of fixed linear subspaces, recent work has introduced nonlinear and data-driven approaches. Continuous Reduced-Order Modeling (CROM) represents the deformation field with an implicit neural function that maps low-dimensional latent coordinates directly to continuous displacements.

In CROM, the displacement at a material point $x_i \in \mathbb{R}^3$ is expressed as

$$u(t, x_i) = g_{\theta_g}(q(t), x_i), \quad (4)$$

where $q(t) \in \mathbb{R}^r$ is a latent vector and g_{θ_g} is a neural decoder with parameters θ_g . An encoder e_{θ_e} maps a high-dimensional displacement field to the latent coordinates, while the decoder reconstructs displacements from q and x , as illustrated in Figure 3. The networks are trained jointly to minimize reconstruction error over a dataset of simulated deformations, after which the reduced dynamics can be integrated entirely in the latent space.

By operating on continuous spatial coordinates rather than a fixed discretization, CROM learns a latent manifold for the deformation field itself. This design allows training on heterogeneous meshes or sampling patterns and supports adaptive remeshing or variable-resolution evaluation at runtime. It also provides strong memory efficiency, since the latent dimension scales with the intrinsic complexity of the deformation rather than the number of mesh nodes, and experiments have shown order-of-magnitude speedups and accuracy improvements over PCA and autoencoder baselines.

Despite these advantages, CROM is not without limitations. Its expressiveness is bounded by the manifold captured during training and it cannot guarantee accurate prediction for dynamics outside that range. Each time step still requires evaluating the neural decoder and sampling the original PDE at a reduced set of spatial points, so CROM does not completely eliminate expensive PDE solves. In addition, obtaining a high-quality latent model demands an offline training phase on full-order simulation data, which can be computationally expensive.

Like other learning-based ROM, our approach requires high-quality training data and remains limited by the coverage of the learned deformation manifold, but within that manifold it offers a far better trade-off between efficiency and expressiveness than classical PCA-based ROM. While the adaptive basis is obtained from

the gradient of the CROM decoder, which may differ from the theoretically optimal subspace, this computation is extremely fast and adds negligible overhead compared with the cost of assembling or evaluating a full reduced system. By combining this fast gradient-based adaptation with a time-updated subspace, our method delivers rich nonlinear accuracy and real-time performance in a simple, linear ROM framework.

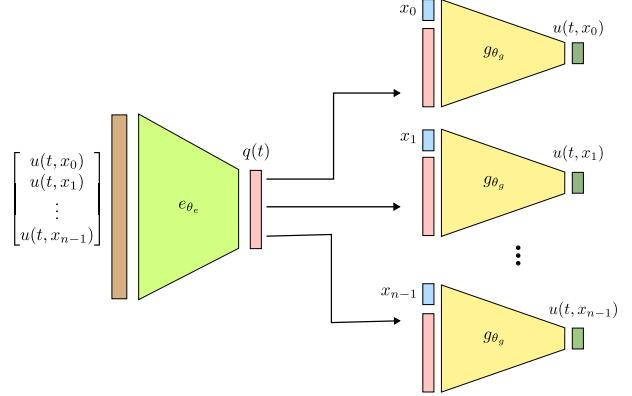


Figure 3: Continuous reduced-order modeling (CROM). The encoder e_{θ_e} maps a full displacement field $u(t, \mathbf{x})$ to latent coordinates q , and the decoder g_{θ_g} reconstructs continuous displacements from q and spatial position x_i .

4. Proposed Method

In this section, we present our method for deformation dynamics. We first describe the adaptive model-reduction framework, then detail the latent-space dynamics that govern reduced simulations, and finally introduce a history-aware interpolation strategy for handling large deformations.

4.1. Adaptive Linear Model Reduction

Before introducing adaptivity, we first describe how our implementation modifies the traditional linear reduced-order model (ROM). Instead of directly applying Equation 2, we adopt an incremental formulation

$$u(t) = u(t-1) + Uq(t), \quad (5)$$

where the displacement is updated relative to the previous step. This incremental form improves numerical stability and accuracy in scenarios with large deformations, where a direct global update may accumulate significant error.

To incorporate adaptivity, we generalize the basis matrix U to be time dependent:

$$u(t) = u(t-1) + U(t-1)q(t). \quad (6)$$

Here $U(t-1)$ is updated at every step to better capture the local deformation behavior. Because CROM represents deformations on a nonlinear manifold while our incremental update is formulated in a linear space, each state is projected onto the CROM manifold using the encoder

$$p(t) = e_{\theta_e}(u(t)), \quad (7)$$

where $p(t) \in \mathbb{R}^r$ is the latent variable with the same dimension r as $q(t)$.

The adaptive basis $U(t)$ is computed from the CROM decoder by differentiating its output with respect to the latent variable. Let $g_{\theta_g}(p(t), \mathbf{x}) \in \mathbb{R}^{3n}$ denote the stacked displacement field obtained by evaluating the decoder at all mesh nodes $\mathbf{x} = \{x_i\}_{i=1}^n$. Since $g_{\theta_g}(p(t), x_i)$ returns the displacement at a single spatial location x_i , the global basis matrix is assembled by concatenating the per-node decoder Jacobians:

$$U(t) \approx \frac{\partial g_{\theta_g}(p(t), \mathbf{x})}{\partial p(t)} \in \mathbb{R}^{3n \times r}. \quad (8)$$

This Jacobian aligns the reduced basis with the local tangent of the learned deformation manifold, enabling the adaptive model to capture strongly nonlinear dynamics while retaining the computational structure of a linear subspace.

The procedure is applied at every time step as shown in Algorithm 1: the current state is projected to the CROM latent space, the decoder gradient provides the updated basis, the reduced variables $q(t)$ are advanced by solving the latent dynamics, and the full displacement $u(t)$ is incrementally reconstructed. This adaptive construction produces a time-varying basis that tracks the evolving geometry of the deformation, enabling efficient yet accurate simulation of complex soft-body motions.

Algorithm 1 Adaptive Linear Model Reduction Procedure

```

1: Input: Initial displacement  $u(0) \in \mathbb{R}^{3n}$ 
2: for each time step  $t = 1, 2, \dots$  do
3:    $p(t-1) \leftarrow e_{\theta_e}(u(t-1))$ 
4:    $U(t-1) \leftarrow \frac{\partial g_{\theta_g}(p(t-1), \mathbf{x})}{\partial p(t-1)}$ 
5:   Compute  $q(t)$  using Newton solver (Algorithm 2)
6:    $u(t) \leftarrow u(t-1) + U(t-1)q(t)$ 
7: end for
```

4.2. Latent Space Dynamics

With the adaptive basis in place, the reduced coordinates $q(t)$ evolve directly in the latent space to capture deformation dynamics efficiently. At each time step, after projecting the current state onto the CROM manifold and updating $U(t)$, we solve for $q(t)$ by minimizing the total system energy—including elastic, inertial, gravitational, and contact terms—within the current adaptive subspace:

$$q(t) = \arg \min_q E(u(q)), \quad (9)$$

where E is the total energy evaluated at the configuration $u(q)$. In our method, we employ implicit time integration, formulating the energy E to include elastic potential, inertia, contact, and friction terms as required by the application. Specifically, we adopt the implicit energy-diminishing integrator [SH98], and handle contact and friction using the incremental potential contact method [LFS^{*}20]. The resulting minimization problem can be solved using standard numerical techniques, such as Newton's method or other gradient-based optimization algorithms, depending on the system's

nonlinearity. In our implementation, we utilize a Newton iteration of the form:

$$q^{k+1} = q^k - \alpha^k \left[\nabla_q^2 E(u(q^k)) \right]^{-1} \nabla_q E(u(q^k)), \quad (10)$$

where α^k is a step size determined by a line search or set to 1 for standard Newton's method. By applying the chain rule, the reduced gradient and Hessian can be expressed as

$$\nabla_q E(u(q^k)) = \left(\frac{\partial u}{\partial q} \right)^T \nabla_u E(u(q^k)) = U^T \nabla_u E(u(q^k)), \quad (11)$$

$$\nabla_q^2 E(u(q^k)) = \left(\frac{\partial u}{\partial q} \right)^T \nabla_u^2 E(u(q^k)) \left(\frac{\partial u}{\partial q} \right) = U^T \nabla_u^2 E(u(q^k)) U, \quad (12)$$

where $\nabla_u E$ and $\nabla_u^2 E$ are the gradient and Hessian of the energy with respect to the full displacement vector u .

After updating $q(t)$, the full state $u(t)$ is reconstructed and the simulation advances to the next time step. Importantly, the adaptive basis $U(t)$ is updated only once per time step while the inner Newton iterations for solving $q(t)$ proceed with this fixed basis. This separation keeps the per-step cost low and ensures stable convergence. The overall procedure for a single time step is summarized in Algorithm 2.

Algorithm 2 Latent Space Dynamics via Newton's Method (one time step t)

```

1: Input: Previous state  $u(t-1)$ , basis  $U(t-1) \in \mathbb{R}^{3n \times r}$ , energy
    $E(u)$ 
2: Initialize:  $q^0 \leftarrow 0$ 
3: for Newton iteration  $k = 0, 1, \dots$  do
4:    $u^k \leftarrow u(t-1) + U(t-1)q^k$ 
5:    $g^k \leftarrow U(t-1)^T \nabla_u E(u^k)$ 
6:    $H^k \leftarrow U(t-1)^T \nabla_u^2 E(u^k) U(t-1)$ 
7:   Solve  $H^k \Delta q^k = -g^k$ 
8:   Determine step size  $\alpha^k \in (0, 1]$ 
9:    $q^{k+1} \leftarrow q^k + \alpha^k \Delta q^k$ 
10:  if convergence criterion satisfied then
11:    break
12:  end if
13: end for
14:  $q(t) \leftarrow q^{k+1}, \quad u(t) \leftarrow u(t-1) + U(t-1)q(t)$ 
```

4.3. Grassmann Interpolation for Large Deformation

While the adaptive linear model reduction performs well for small deformations, it encounters significant challenges under large deformations. As illustrated in Figure 4, the object may fail to recover its original state after experiencing substantial deformation. This issue arises because the current basis adaptation scheme does not account for the history of previous displacements, leading to a loss of important information about the deformation trajectory.

To address this limitation, we incorporate historical displacement information into the basis construction. We formulate the following least-squares problem:

$$\min_{\Phi} \| \Phi P - D \|_F^2, \quad (13)$$

where $\Phi \in \mathbb{R}^{3n \times r}$ is the basis matrix to be estimated, $P = [p(t-m+1) | \dots | p(t)] \in \mathbb{R}^{r \times m}$ stacks the latent variables from the most recent m time steps, and $D = [u(t-m+1) | \dots | u(t)] \in \mathbb{R}^{3n \times m}$ collects the corresponding full displacements. Intuitively, this formulation seeks a basis that best captures the essential deformation patterns of the recent trajectory. When $\text{rank}(P) = r$, the problem has a closed-form solution

$$\Phi = DP^T(PP^T)^{-1}.$$

To guard against rank deficiency, we regularize by adding a small diagonal term (10^{-8} in our implementation) to PP^T . The computational cost of constructing Φ is dominated by the matrix multiplications and the inversion of the $r \times r$ system, yielding a per-time step complexity of $\mathcal{O}(nmr + r^3)$.

After obtaining the historical displacement basis Φ , we blend it with the current adaptive basis U using Grassmann interpolation to create the interpolated adaptive basis U_{interp} as shown in Algorithm 4. Intuitively, Grassmann interpolation allows us to smoothly transition between two sets of basis vectors—effectively “morphing” one subspace into another in the most natural way, while always remaining within the space of valid bases. The interpolation is governed by a parameter $\lambda \in [0, 1]$: when $\lambda = 0$, the result is exactly the same span as historical basis Φ ; when $\lambda = 1$, it is the same span as current basis U ; and for intermediate values, the interpolated basis U_{interp} combines information from both. This approach ensures that U_{interp} not only adapts to the current deformation but also retains important features from the recent deformation history. By performing the interpolation on the Grassmann manifold—the mathematical space of all r -dimensional subspaces—this method preserves the geometric structure of the bases, resulting in more robust and physically meaningful model reduction, especially under large deformations.

The Grassmann interpolation procedure involves several matrix operations: computing the thin QR factorization, computing the singular value decomposition (SVD) of $U^T \Phi$, matrix multiplications, and elementwise trigonometric functions on the principal angles. The overall computational complexity for the interpolation step is $\mathcal{O}(nr^2)$, which is efficient in practice since $r \ll n$.

By incorporating historical information and employing Grassmann interpolation, our method effectively mitigates the limitations of purely adaptive bases and enables the system to recover from large deformations more reliably. The final algorithm is summarized in Algorithm 3.

5. Results

We evaluate our method on a suite of benchmark deformable objects spanning a variety of shapes and material settings. Training data are generated from long-duration dynamic simulations in which each object is repeatedly dropped into a Pachinko-style environment and tumbled in a dryer-like setting to induce a wide range of deformations, resulting in roughly 300,000 displacement snapshots per object. All models share a Young’s modulus of 10^5 and a Poisson’s ratio of 0.45. For the CROM architecture, the encoder is a multilayer perceptron (MLP) with two hidden layers of 128 units each, and the decoder has eight hidden layers of 128 units; all hidden layers use ELU activations. Networks are trained with Adam

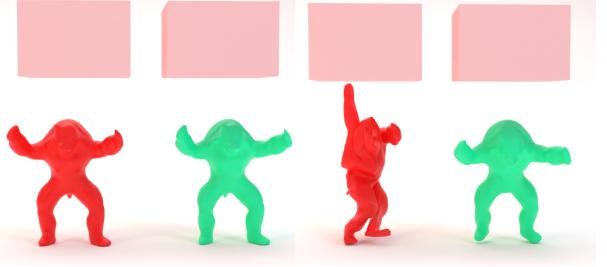


Figure 4: Comparison of adaptive linear model reduction under large deformation. **Left:** Two identical objects before compression. **Right:** Results after compression—**without historical interpolation** and **with historical interpolation**. The interpolation enables perfect recovery to the original state, whereas the standard one fails under large deformation.

Algorithm 3 Adaptive Linear Model Reduction with Grassmann Interpolation

```

1: Input: Initial displacement  $u(0) \in \mathbb{R}^{3n}$ 
2: for each time step  $t = 1, 2, \dots$  do
3:    $p(t-1) \leftarrow e_{\theta_e}(u(t-1))$ 
4:    $U(t-1) \leftarrow \frac{\partial g_{\theta_g}(p(t-1), \mathbf{x})}{\partial p(t-1)}$ 
5:   Construct historical basis  $\Phi$  using the most recent  $m$  latent codes and displacements
6:    $U_{\text{interp}}(t-1) \leftarrow \text{GRASSMANNINTERPOLATION}(\Phi, U(t-1), \lambda)$  (Algorithm 4)
7:   Compute  $q(t)$  with Newton solver (Algorithm 2) using basis  $U_{\text{interp}}(t-1)$ 
8:    $u(t) \leftarrow u(t-1) + U_{\text{interp}}(t-1)q(t)$ 
9: end for

```

Algorithm 4 Grassmann Interpolation

```

1: Input: Bases  $\Phi, U \in \mathbb{R}^{3n \times r}$ , interpolation parameter  $\lambda \in [0, 1]$ 
2: Output: Interpolated basis  $U_{\text{interp}} \in \mathbb{R}^{3n \times r}$ 
3: Thin QR:  $Q_\Phi R_\Phi \leftarrow \text{qr}(\Phi)$ ,  $Q_U R_U \leftarrow \text{qr}(U)$ 
4:  $M \leftarrow Q_\Phi^\top Q_U$ 
5: SVD:  $M = V \cos \Theta W^\top$ 
6:  $S \leftarrow (Q_U W - Q_\Phi V \cos \Theta) (\sin \Theta)^{-1}$ 
7:  $G \leftarrow Q_\Phi V \cos(\lambda \Theta) + S \sin(\lambda \Theta)$ 
8: Thin QR:  $U_{\text{interp}} \leftarrow \text{qr}(G)$ 
9: return  $U_{\text{interp}}$ 

```

and a scheduled learning rate that starts at 5 and decreases to 2, 1, and 0.5 over four phases of 100 epochs each (total 400 epochs), with batch size 64 and GPU acceleration. The training loss is the mean squared error between reconstructed and ground-truth displacements.

After establishing the experimental setup, we present an ablation study of key parameters, a comparison of accuracy and visual quality against PCA-based ROM and the full-order model (FOM), and a runtime analysis.

5.1. Ablation Study

We investigate the influence of three key parameters: the number of basis vectors r , the history window size m , and the interpolation parameter λ . The parameter r is standard in ROM, while m and λ are specific to our history-aware interpolation. For each configuration, we compute a relative online error by accumulating the per-time step ℓ_2 displacement difference between each ROM (PCA and ours) and the FOM over a given simulation interval, and normalizing by the accumulated FOM displacement magnitude over the same interval. To prevent long-horizon drift from dominating the **quantitative** error measurements, the ROM state is realigned to the FOM state at the end of each time step before proceeding. This alignment is used only for numerical error evaluation and is not applied in the qualitative visual comparisons.

5.1.1. Effect of Basis Size r

We assess sensitivity to r on the armadillo compression and bending examples (Figure 5). As shown in Figure 6, our method consistently achieves lower online error than PCA-based ROM for all tested $r \in \{4, 8, 16, 32\}$. While increasing r improves both methods and narrows the gap, our approach remains more accurate across settings. We adopt $r = 16$ as the default in subsequent experiments unless otherwise stated.

5.1.2. Effect of History Window Size m

We vary the history window m to evaluate its effect on recovery from large deformations. Table 1 reports total relative online error for Armadillo compression and bending. Very small windows (e.g., $m = 1$) underperform due to insufficient history. Windows in the range $m \in [5, 10]$ generally minimize error, though too-small m can reduce visual smoothness. In practice we select $m = 20$ as a good balance between accuracy and visual quality and use it by default thereafter.

5.1.3. Effect of Interpolation Parameter λ

The parameter λ controls the blend between the historical basis and the current adaptive basis. Lower λ values (more emphasis on history) tend to reduce error, but setting λ too small (e.g., $\lambda = 0$) can cause numerical issues by relying almost exclusively on historical information. Table 2 summarizes results; we find $\lambda \in [0.1, 0.3]$ robust across examples and use $\lambda = 0.2$ by default.

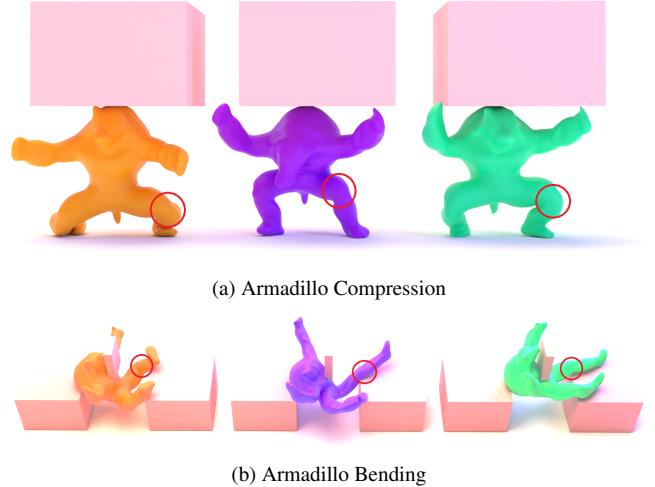


Figure 5: Ablation study scenarios: Each subfigure shows three armadillos—**FOM**, **PCA**, and **ours**. As highlighted by the red circle, **ours** exhibits greater similarity to **FOM**.

m	1	5	10	20	30	50
Armadillo Compression						
$r = 8$	78.90	31.84	32.18	41.77	45.79	58.57
$r = 16$	77.10	29.26	26.33	27.80	29.77	37.36
Armadillo Bending						
$r = 8$	88.25	38.25	40.50	47.03	56.12	75.55
$r = 16$	74.60	37.56	36.32	38.31	43.97	62.98

Table 1: Total relative online error (%) for different history window sizes m on the armadillo compression and bending examples. Smaller windows lack sufficient history and yield higher errors, while moderate windows ($m \in [5, 10]$) generally achieve the best accuracy. We adopt $m = 20$ as a practical default to balance accuracy and visual smoothness.

5.2. Accuracy and Visual Quality

We evaluate accuracy across a diverse set of objects and scenarios, comparing our approach to a PCA-based ROM baseline. The total relative online error is accumulated over the entire simulation, with per-step realignment to the FOM state to avoid divergence. Table 3 reports errors, showing that our method consistently outperforms PCA, particularly on challenging cases. Representative visual comparisons are given in Figure 7.

5.3. Performance

We assess computational efficiency by measuring the average runtime per simulation step across objects and scenarios, and compare against PCA-based ROM. All timings are recorded on a MacBook Pro with an M4 Pro processor, 48 GB RAM, and 4 threads. Results are summarized in Table 4. Although our adaptive basis introduces additional computations (basis updates and Grassmann interpolation),

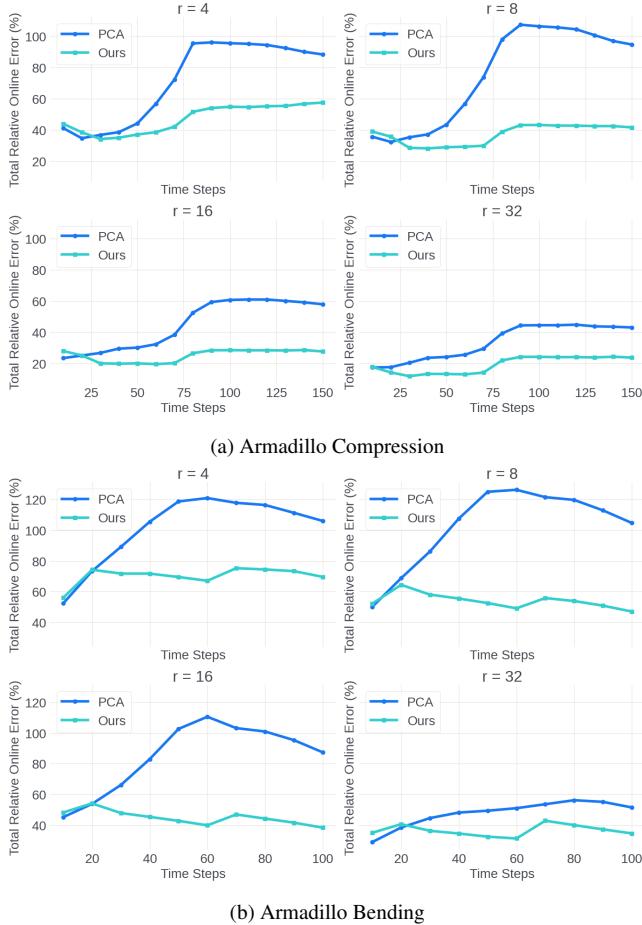


Figure 6: Effect of basis size r on relative online error for the two scenarios. Increasing r improves accuracy for both methods and reduces the performance gap, but our approach remains consistently more accurate than PCA.

tion), the implementation maintains interactive performance, with per-step runtimes within a factor of roughly 1.3–1.5 of PCA-based ROM.

6. Conclusions & Limitations

In this paper, we presented an adaptive linear model reduction framework enhanced with history-aware Grassmann interpolation, enabling robust and efficient simulation of soft-body dynamics under large deformations. By blending the computational efficiency of classical linear subspaces with the flexibility of a time-updated basis, our method not only improves accuracy and robustness over standard PCA-based ROM, but also maintains the ability to recover the undeformed rest state even after extreme deformations.

Despite these advances, several limitations remain. First, like most learning-based methods, the effectiveness of our approach depends on the quality and diversity of the training data used to construct the CROM manifold and the adaptive basis, but this challenge is well recognized across the field and can be mitigated with

λ	0.0	0.1	0.2	0.3	0.4	0.5
Armadillo Compression						
$r = 8$	36.12	36.72	41.77	45.97	49.21	55.08
$r = 16$	40.08	24.93	27.80	30.24	34.95	38.39
Armadillo Bending						
$r = 8$	44.58	44.48	47.03	51.37	56.80	63.94
$r = 16$	36.37	37.63	38.31	42.01	45.52	50.47

Table 2: Total relative online error (%) for different λ values on the armadillo compression and bending examples. Smaller λ values, which place more weight on the historical basis, generally reduced error, but setting λ too close to zero can cause numerical instability. Values in the range $\lambda \in [0.1, 0.3]$ provide robust performance and we use $\lambda = 0.2$ as the default.

Object	Scenario	PCA	Ours
Armadillo	Compression	58.02	27.80
Armadillo	Bending	87.44	38.31
Armadillo	Stretching	46.05	40.41
Armadillo	Twisting	31.09	21.59
Bunny	Compression	103.40	28.46
Bunny	Bending	102.10	32.86
Dragon	Stretching	37.68	17.9
Dragon	Bending	75.28	40.67
Pig	Compression	69.53	29.62
Pig	Stretching	95.13	86.64

Table 3: Total relative online error (%) for multiple objects and scenarios. Our method reduces error by 40-70% compared to PCA.

broader data coverage. Second, while the historical basis captures recent deformation trajectories, it is not yet clear whether this is the most effective way to encode long-term deformation history, making it a natural target for future improvements. Finally, the adaptive basis is derived from the gradient of the CROM decoder rather than an explicitly optimized subspace, yet this approximation is ex-

Object	Scenario	#Tets	PCA	Ours
Armadillo	Compression	6440	6.42 ms	8.87 ms
Armadillo	Bending	6440	6.17 ms	8.44 ms
Armadillo	Stretching	6440	5.70 ms	8.01 ms
Armadillo	Twisting	6440	4.65 ms	7.08 ms
Bunny	Compression	6566	6.28 ms	8.64 ms
Bunny	Compression	6566	6.35 ms	8.50 ms
Dragon	Stretching	3840	2.58 ms	5.66 ms
Dragon	Bending	3840	3.41 ms	5.62 ms
Pig	Compression	10926	10.58 ms	14.10 ms
Pig	Stretching	10926	9.18 ms	10.65 ms

Table 4: Average runtime per simulation step (milliseconds). Our method remains interactive while adding only a modest overhead of about 1.3x-1.5x compared to PCA.

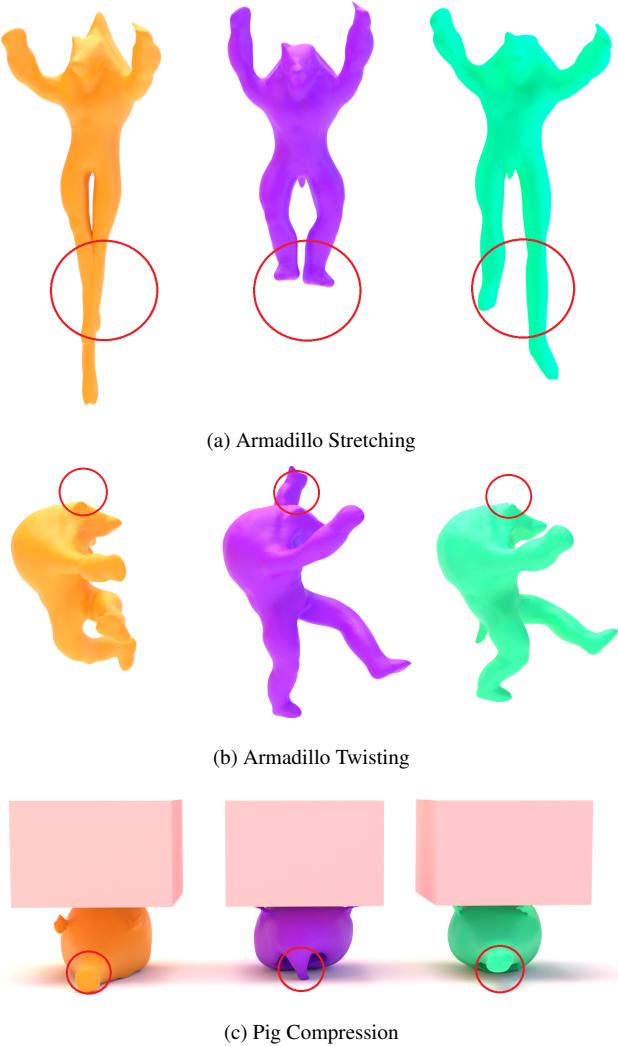


Figure 7: Representative visual comparisons across scenarios: **FOM**, **PCA**, and **ours**.

tremely fast to compute and keeps the method practical even in challenging configurations.

7. Acknowledgments

We gratefully acknowledge the use of mesh models provided by the Stanford 3D Scanning Repository and Keenan Crane. These datasets were instrumental in the evaluation and visualization of our proposed method.

References

- [AGK*22] AIGERMAN N., GUPTA K., KIM V. G., CHAUDHURI S., SAITO J., GROUEIX T.: Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *arXiv preprint arXiv:2205.02904* (2022). [3](#)
- [BJ05] BARBIĆ J., JAMES D. L.: Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM transactions on graphics (TOG)* 24, 3 (2005), 982–990. [2](#)
- [BvTH16] BRANDT C., VON TYCOWICZ C., HILDEBRANDT K.: Geometric flows of curves in shape space for processing motion of deformable objects. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 295–305. [2](#)
- [CCGC23] CHEN P. Y., CHIARAMONTE M. M., GRINSPUN E., CARLBERG K.: Model reduction for the material point method via an implicit neural representation of the deformation map. *Journal of Computational Physics* 478 (2023), 111908. [3](#)
- [CCW*23] CHANG Y., CHEN P. Y., WANG Z., CHIARAMONTE M. M., CARLBERG K., GRINSPUN E.: Licrom: Linear-subspace continuous reduced order modeling with neural fields. In *SIGGRAPH Asia 2023 Conference Papers* (2023), pp. 1–12. [3](#)
- [CK05] CHOI M. G., KO H.-S.: Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Transactions on Visualization and Computer Graphics* 11, 1 (2005), 91–101. [2](#)
- [CXG*22] CHEN P. Y., XIANG J., CHO D. H., CHANG Y., PERSHING G., MAIA H. T., CHIARAMONTE M. M., CARLBERG K., GRINSPUN E.: Crom: Continuous reduced-order modeling of pdes using implicit neural representations. *arXiv preprint arXiv:2206.02607* (2022). [2, 3](#)
- [FMD*19] FULTON L., MODI V., DUVENAUD D., LEVIN D. I., JACOBSON A.: Latent-space dynamics for reduced deformable simulation. In *Computer graphics forum* (2019), vol. 38, Wiley Online Library, pp. 379–391. [2](#)
- [HDDN19] HOLDEN D., DUONG B. C., DATTA S., NOWROUZEZAHRAI D.: Subspace neural physics: Fast data-driven interactive simulation. In *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2019), pp. 1–12. [2](#)
- [HSO03] HAUSER K. K., SHEN C., O'BRIEN J. F.: Interactive deformation using modal analysis with constraints. [2](#)
- [HTC*14] HAHN F., THOMASZEWSKI B., COROS S., SUMNER R. W., COLE F., MEYER M., DEROSE T., GROSS M.: Subspace clothing simulation using adaptive bases. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–9. [3](#)
- [HZ13] HARMON D., ZORIN D.: Subspace integration with local deformations. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10. [3](#)
- [JBP06] JAMES D. L., BARBIĆ J., PAI D. K.: Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 987–995. [2](#)
- [JP99] JAMES D. L., PAI D. K.: Artdefo: accurate real time deformable objects. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 65–72. [2](#)
- [KAT*19] KIM B., AZEVEDO V. C., THUERÉY N., KIM T., GROSS M., SOLENTHALER B.: Deep fluids: A generative network for parameterized fluid simulations. In *Computer graphics forum* (2019), vol. 38, Wiley Online Library, pp. 59–70. [2](#)
- [KJ09] KIM T., JAMES D. L.: Skipping steps in deformable simulation with online model reduction. In *ACM SIGGRAPH Asia 2009 papers*. 2009, pp. 1–9. [3](#)
- [KJP02] KRY P. G., JAMES D. L., PAI D. K.: Eigenskin: real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2002), pp. 153–159. [2](#)
- [KLM01] KRYSL P., LALL S., MARSDEN J. E.: Dimensional model reduction in non-linear finite element dynamics of solids and structures. *International Journal for numerical methods in engineering* 51, 4 (2001), 479–504. [2](#)
- [LC20] LEE K., CARLBERG K. T.: Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics* 404 (2020), 108973. [2](#)

- [LFS^{*}20] LI M., FERGUSON Z., SCHNEIDER T., LANGLOIS T. R., ZORIN D., PANZZO D., JIANG C., KAUFMAN D. M.: Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.* 39, 4 (2020), 49. [5](#)
- [LLL^{*}25] LI Y., LIN G. W.-C., LARIONOV E., BOZIC A., ROBLE D., KAVAN L., COROS S., THOMASZEWSKI B., STUYCK T., CHEN H.-Y.: Self-supervised learning of latent space dynamics. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 8, 4 (2025), 1–18. [2](#)
- [MAKWL22] MERCIER-AUBIN A., KRY P. G., WINTER A., LEVIN D. I.: Adaptive rigidification of elastic solids. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–11. [3](#)
- [MHR^{*}16] MUSIALSKI P., HAFNER C., RIST F., BIRSAK M., WIMMER M., KOBBELT L.: Non-linear shape optimization using local subspace projections. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–13. [2](#)
- [OSG02] O'BRIEN J. F., SHEN C., GATCHALIAN C. M.: Synthesizing sounds from rigid-body simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2002), pp. 175–181. [2](#)
- [PW89] PENTLAND A., WILLIAMS J.: Good vibrations: Modal dynamics for graphics and animation. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques* (1989), pp. 215–222. [2](#)
- [RCCO22] ROMERO C., CASAS D., CHIARAMONTE M. M., OTADUY M. A.: Contact-centric deformation learning. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–11. [3](#)
- [RCCO23] ROMERO C., CASAS D., CHIARAMONTE M., OTADUY M. A.: Learning contact deformations with general collider descriptors. In *SIGGRAPH Asia 2023 Conference Papers* (2023), pp. 1–10. [2](#)
- [RCPO21] ROMERO C., CASAS D., PÉREZ J., OTADUY M.: Learning contact corrections for handle-based subspace dynamics. *ACM Transactions on Graphics (ToG)* 40, 4 (2021), 1–12. [3](#)
- [SB12] SIFAKIS E., BARBIC J.: Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *Acm siggraph 2012 courses*. 2012, pp. 1–50. [4](#)
- [SH98] STUART A. M., HUMPHRIES A. R.: *Dynamical systems and numerical analysis*, vol. 2. Cambridge university press, 1998. [5](#)
- [SMK22] STARKE S., MASON I., KOMURA T.: Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–13. [2](#)
- [SYS^{*}21] SHEN S., YIN Y., SHAO T., WANG H., JIANG C., LAN L., ZHOU K.: High-order differentiable autoencoder for nonlinear model reduction. *arXiv preprint arXiv:2102.11026* (2021). [2](#)
- [TFLK24] TRUSTY T., FEI Y., LEVIN D., KAUFMAN D.: Trading spaces: Adaptive subspace time integration for contacting elastodynamics. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–16. [3](#)
- [TLP06] TREUILLE A., LEWIS A., POPOVIĆ Z.: Model reduction for real-time fluids. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 826–834. [2](#)
- [TMDK15] TENG Y., MEYER M., DEROSE T., KIM T.: Subspace condensation: Full space adaptivity for subspace deformations. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9. [3](#)
- [UMK17] ULU E., MCCANN J., KARA L. B.: Lightweight structure design under force location uncertainty. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13. [2](#)
- [VTSSH13] VON TYCOWICZ C., SCHULZ C., SEIDEL H.-P., HILDEBRANDT K.: An efficient construction of reduced deformable objects. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–10. [2](#)
- [WBT19] WIEWEL S., BECHER M., THUERÉY N.: Latent space physics: Towards learning the temporal evolution of fluid flow. In *Computer graphics forum* (2019), vol. 38, Wiley Online Library, pp. 71–82. [2](#)
- [XLCB15] XU H., LI Y., CHEN Y., BARBIĆ J.: Interactive material design using model reduction. *ACM Transactions on Graphics (TOG)* 34, 2 (2015), 1–14. [2](#)
- [ZLL^{*}23] ZONG Z., LI X., LI M., CHIARAMONTE M. M., MATUSIK W., GRINSPUN E., CARLBERG K., JIANG C., CHEN P. Y.: Neural stress fields for reduced-order elastoplasticity and fracture. In *SIGGRAPH Asia 2023 Conference Papers* (2023), pp. 1–11. [3](#)