AniMimic: Imitating 3D Animation from Video Priors

Supplementary Material

1. Differentiable Simulation Details

Forward Simulation The dynamics of deformable bodies follow Newton's Second Law, formulated as

$$\frac{d^2\boldsymbol{x}}{dt^2} = \boldsymbol{M}^{-1}\boldsymbol{f}(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{M}$ is the mass matrix, whose diagonal entries contain the masses of all vertices. The total force $\boldsymbol{f}(\boldsymbol{x})$ consists of the external forces $\boldsymbol{f}_{\text{ext}}(\boldsymbol{x})$ (e.g., gravity) and the internal elastic forces $\boldsymbol{f}_{\text{int}}(\boldsymbol{x})$ arising from deformation. To ensure consistency with skinning-based animation, we omit external forces and simulate only the internal elastic response. The internal force $\boldsymbol{f}_{\text{int}}(\boldsymbol{x})$ captures how the deformable object attempts to restore its rest shape by resisting deformation.

In continuum mechanics, this elastic behavior is characterized by defining an energy density function $\psi(\boldsymbol{F}(\boldsymbol{x}))$, which measures the strain energy per unit undeformed volume. We employ the Fixed Corotated constitutive model [5] to describe the material behavior. Its energy density is defined as

$$\psi(\boldsymbol{x}) = \mu\|\boldsymbol{F}_i - \boldsymbol{R}_i\|_F^2 + \frac{\lambda}{2}(\det(\boldsymbol{F}_i) - 1)^2, \tag{2}$$

where μ and λ are the Lamé parameters, $\boldsymbol{F}_i$ is the deformation gradient, and $\boldsymbol{R}_i$ is the rotational component extracted from the polar decomposition of $\boldsymbol{F}_i$. The total potential energy of the deformable body is computed as

$$\Psi(\boldsymbol{x}) = \sum_{i=1}^{N} \psi(\boldsymbol{F}_i)V_i, \tag{3}$$

where V_i denotes volume of i-th tetdrahedra. Note the internal resisting force is then defined as the negative gradient of the potential energy with respect to the vertex position

$$\boldsymbol{f}_{\text{int}}(\boldsymbol{x}) = -\frac{\partial\Psi(\boldsymbol{x})}{\partial\boldsymbol{x}}. \tag{4}$$

To advance the simulation, we use an optimization-based implicit integrator:

$$\boldsymbol{x}^{n+1} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_{\boldsymbol{M}}^2 + \Psi(\boldsymbol{x}), \tag{5}$$

and solve this minimization via Newton's method with line search. This yields the updated configuration $\boldsymbol{x}^{n+1}$ and allows us to compute the simulated trajectory of every vertex over time.

Backward Propagation Following Li et al. [1], once the optimization problem in Eq. 5 is solved at each time step, we can backpropagate through the implicit integrator. For any loss function L, let $\boldsymbol{G}$ denote the gradient of the objective $\frac{1}{2}\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_{\boldsymbol{M}}^2 + \Psi(\boldsymbol{x})$. The gradient can be propagated from time step $n+1$ to time step n as:

$$\frac{dL}{d\boldsymbol{x}^n} = -\mathcal{A}\frac{\partial\boldsymbol{G}}{\partial\boldsymbol{x}^n} - \frac{1}{h}\frac{dL}{d\boldsymbol{v}^{n+1}} \tag{6}$$

$$\left[\frac{dL}{d\boldsymbol{v}^n}, \frac{dL}{dE}\right] = -\mathcal{A}\left[\frac{\partial\boldsymbol{G}}{\partial\boldsymbol{v}^n}, \frac{\partial\boldsymbol{G}}{\partial E}\right], \tag{7}$$

where the velocity update is $\boldsymbol{v}^{n+1} = \frac{\boldsymbol{x}^{n+1} - \boldsymbol{x}^n}{\Delta t}$ and E is the Young's modulus. $\mathcal{A}$ is obtained by solving a linear system:

$$\mathcal{A} = \left[\frac{dL}{d\boldsymbol{x}^{n+1}} + \frac{1}{h}\frac{dL}{d\boldsymbol{v}^{n+1}}\right]\left[\frac{\partial\boldsymbol{G}}{\partial\boldsymbol{x}^{n+1}}\right]^{-1} \tag{8}$$

By combining these analytical gradients with Warp's differentiation operator [3], our simulation becomes fully differentiable with respect to material parameters. This enables direct optimization of material properties using losses computed at arbitrary time steps.

2. Additional Results

Additional Qualitative Comparison. In Fig. 1, we provide additional comparison results with the baseline methods, including Puppeteer [4], DreamMesh4D [2] and SC4D [6].

Diverse Motion. By leveraging the capability of video models, our method naturally supports generating diverse motions for the input 3D assets by producing multiple videos and optimizing the rigging model for each, as demonstrated in Fig. 2.

Novel Views. We present additional novel-view results of our generated motions in Fig. 3. Since the motions are reconstructed on a 3D model, our method naturally maintains cross-view consistency, removing the need to generate multi-view videos.

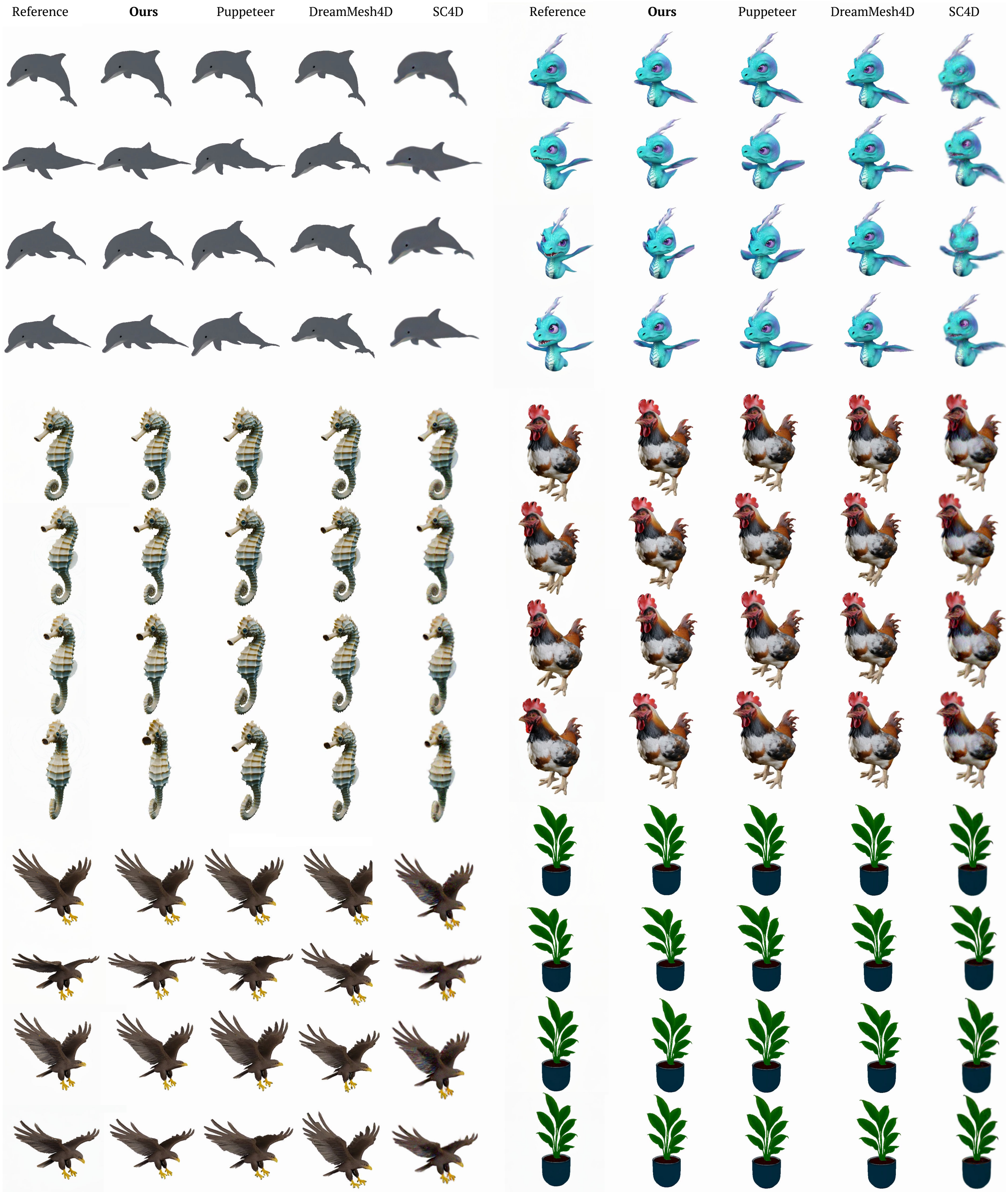

Figure 1. **Qualitative Comparison.** We provide additional qualitative comparison results with Puppeteer [4], DreamMesh4D [2], and SC4D [6].

Figure 2. **Varying Motion Generation.** With different guiding videos of the same character, our method generates the corresponding 4D trajectories containing varied motion sequences.

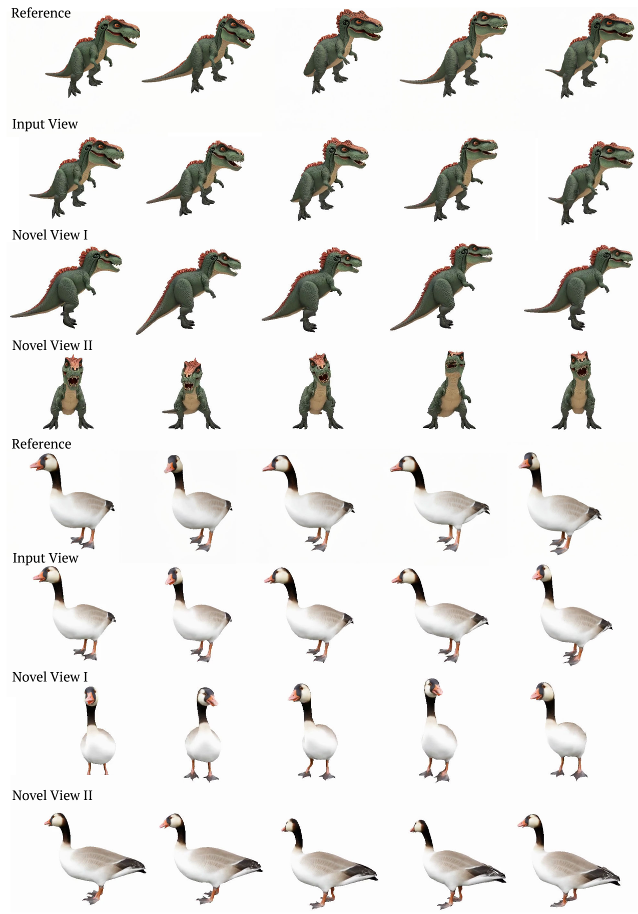

Figure 3. **Novel Views.** In addition to the input-view results, we render the generated animations from two novel viewpoints.

References

[1] Xuan Li, Chang Yu, Wenxin Du, Ying Jiang, Tianyi Xie, Yunuo Chen, Yin Yang, and Chenfanfu Jiang. Dress-1-to-3: Single image to simulation-ready 3d outfit with diffusion prior and differentiable physics. *ACM Transactions on Graphics (TOG)*, 44(4):1–16, 2025. 1

[2] Zhiqi Li, Yiming Chen, and Peidong Liu. Dreammesh4d: Video-to-4d generation with sparse-controlled gaussian-mesh hybrid representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2

[3] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. https://github.com/nvidia/warp, 2022. NVIDIA GPU Technology Conference (GTC). 1

[4] Chaoyue Song, Xiu Li, Fan Yang, Zhongcong Xu, Jiacheng Wei, Fayao Liu, Jiashi Feng, Guosheng Lin, and Jianfeng Zhang. Puppeteer: Rig and animate your 3d models. *Advances in Neural Information Processing Systems*, 2025. 1, 2

[5] Alexey Stomakhin, Russell Howes, Craig A Schroeder, and Joseph M Teran. Energetically consistent invertible elasticity. In *Symposium on Computer Animation*, 2012. 1

[6] Zijie Wu, Chaohui Yu, Yanqin Jiang, Chenjie Cao, Fan Wang, and Xiang Bai. Sc4d: Sparse-controlled video-to-4d generation and motion transfer. In *European Conference on Computer Vision*, pages 361–379. Springer, 2024. 1, 2