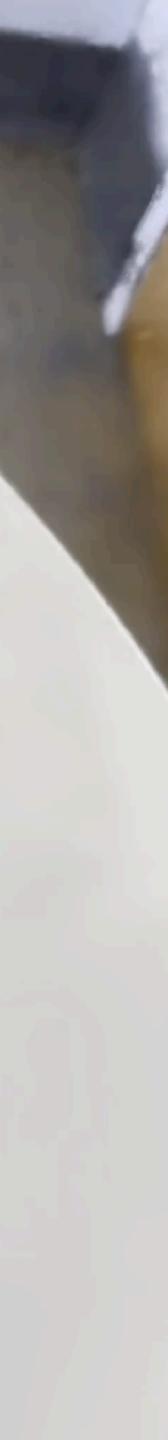
PhysGaussian: Physics-Integrated 3D Gaussians for Generative Dynamics

1P & So

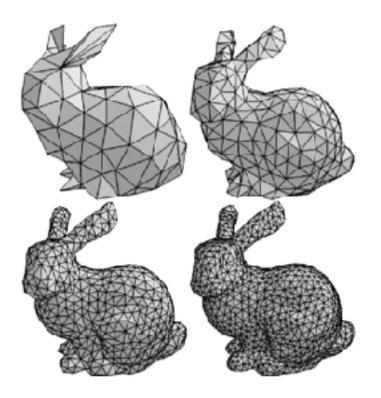
Tianyi Xie^{1*}, Zeshun Zong^{1*}, Yuxing Qiu^{1*}, Xuan Li^{1*}, Yutao Feng^{2,3}, Yin Yang³, Chenfanfu Jiang¹ ¹University of California, Los Angeles, ²Zhejiang University, ³University of Utah



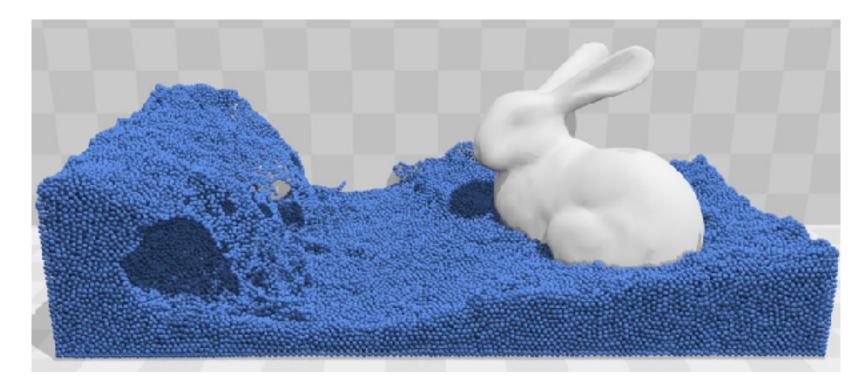
Traditional Physics-based Animation Pipeline



Real Object



Modeling



[Macklin et al. 2013]

Simulation

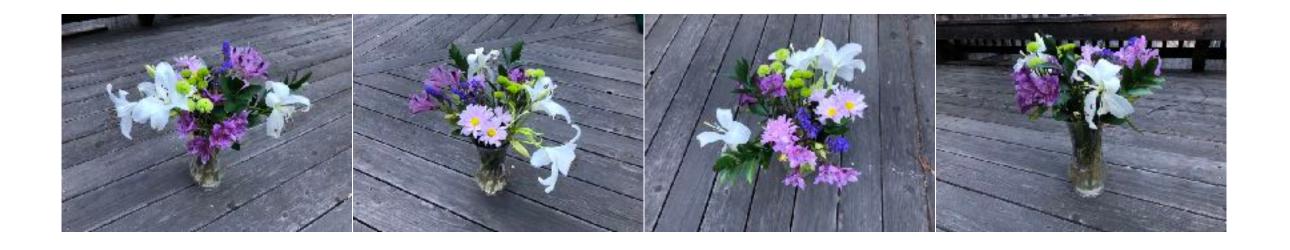


[Macklin et al. 2013]

Rendering

Task: Physics-aware Generative Dynamics

Multi-view Images

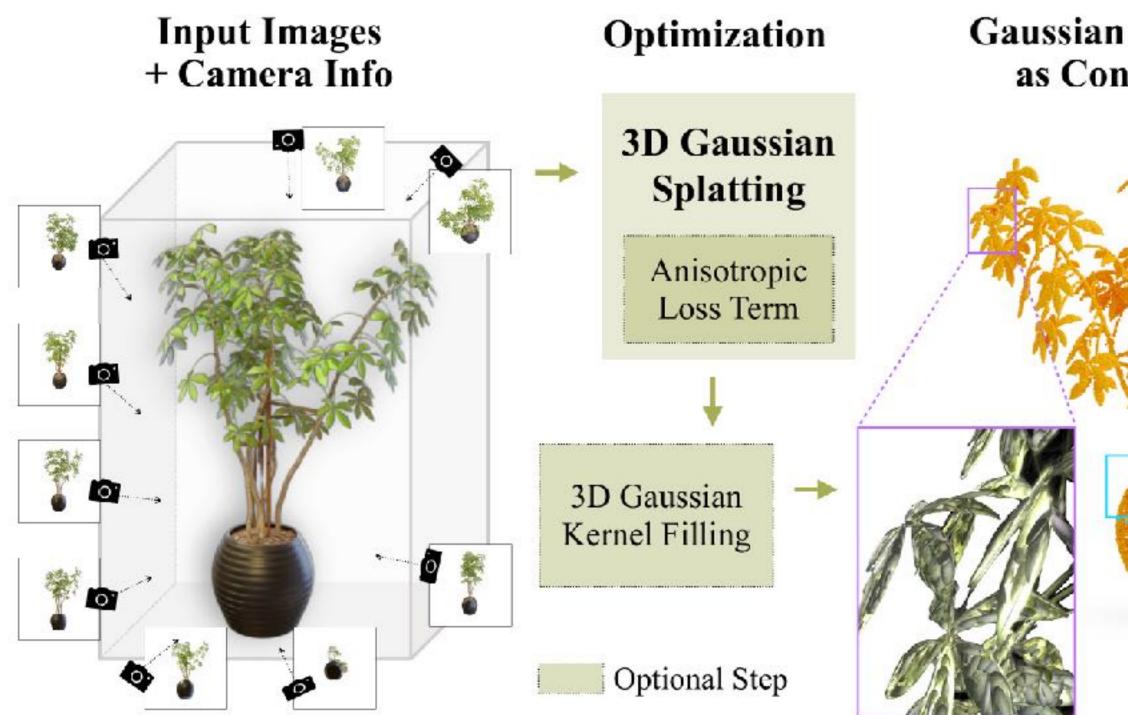


Physics-based Dynamics



Unified Modeling/Simulation/Rendering Pipeline

Method Overview



Modeling

Physics Integration Physics-grounded Gaussian Ellipsoids **Novel Motion** as Continuum **Kinematics** Gaussian Evolution Harmonics Transform **Dynamics Continuum Mechanics Time Integration** Multiple-Viewpoint Renderer

Simulation

Rendering



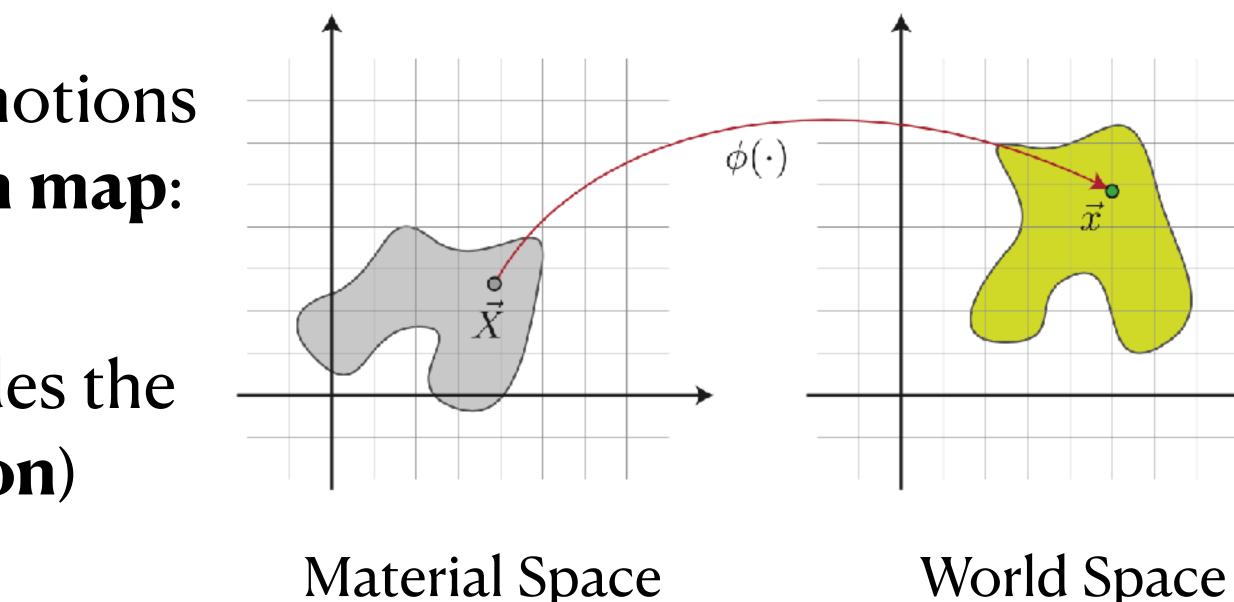
What is Dynamics?

 Continuum mechanics describes motions by a time-dependent **deformation map**:

 $x = \phi(X, t)$

- The deformation gradient encodes the local transformations (linearization) $F(X,t) = \nabla_X \phi(X,t)$
- The velocity change is governed by the momentum conservation law:

 $\rho(x,t)\dot{v}(x,t) = \nabla \cdot \sigma(x,t) + f^{ext}$ Acceleration Mass Force



Per-point Newton Second Law

Evolving Gaussian Kernels

Rest Shape

$$G_p(X) = e^{-\frac{1}{2}(X - X_p)^T A_p^{-1}(X - X_p)}$$

Assume local affine transformations (linearization):

$$\tilde{\phi}_{p}(X,t) = x_{p} + F_{p}(X - X_{p})$$

$$G_{p}(x,t) = e^{-\frac{1}{2}(x - x_{p})^{T}(F_{p}A_{p}F_{p}^{T})^{-1}(x - x_{p})}$$
Evolved Position x_{p}
Evolved Covariance a_{p}

Deformed Shape

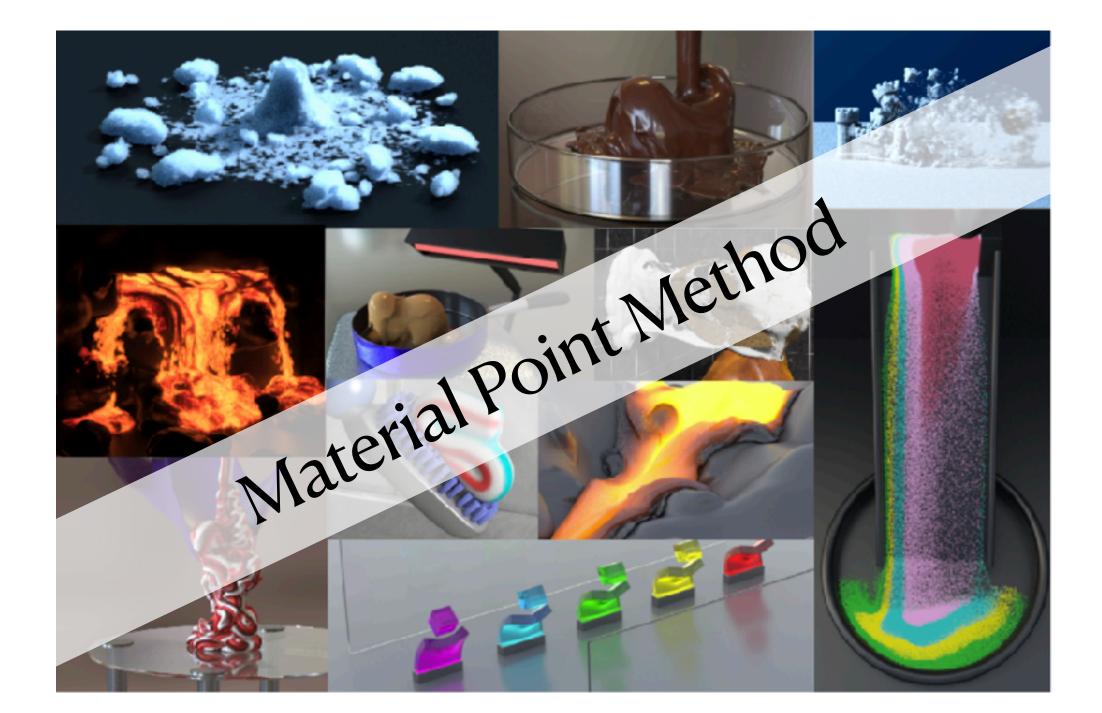
 $G_{p}(x,t) = e^{-\frac{1}{2}(\phi^{-1}(x,t) - X_{p})^{T}A_{p}^{-1}(\phi^{-1}(x,t) - X_{p})}$ Not Gaussian $oldsymbol{\phi}_1$ $(oldsymbol{x}_1,oldsymbol{a}_1)$ $(\boldsymbol{X}_1, \boldsymbol{A}_1)$ $ilde{oldsymbol{\phi}}_2$ Ω^0 $(\mathbf{X}_2, \mathbf{A}_2)$ $(oldsymbol{x}_2,oldsymbol{a}_2)$ $x_p(t) = \phi(X_p, t)$ $a_p(t) = F_p(t)A_pF_p(t)^T$

Simulator Choice

Requirement:

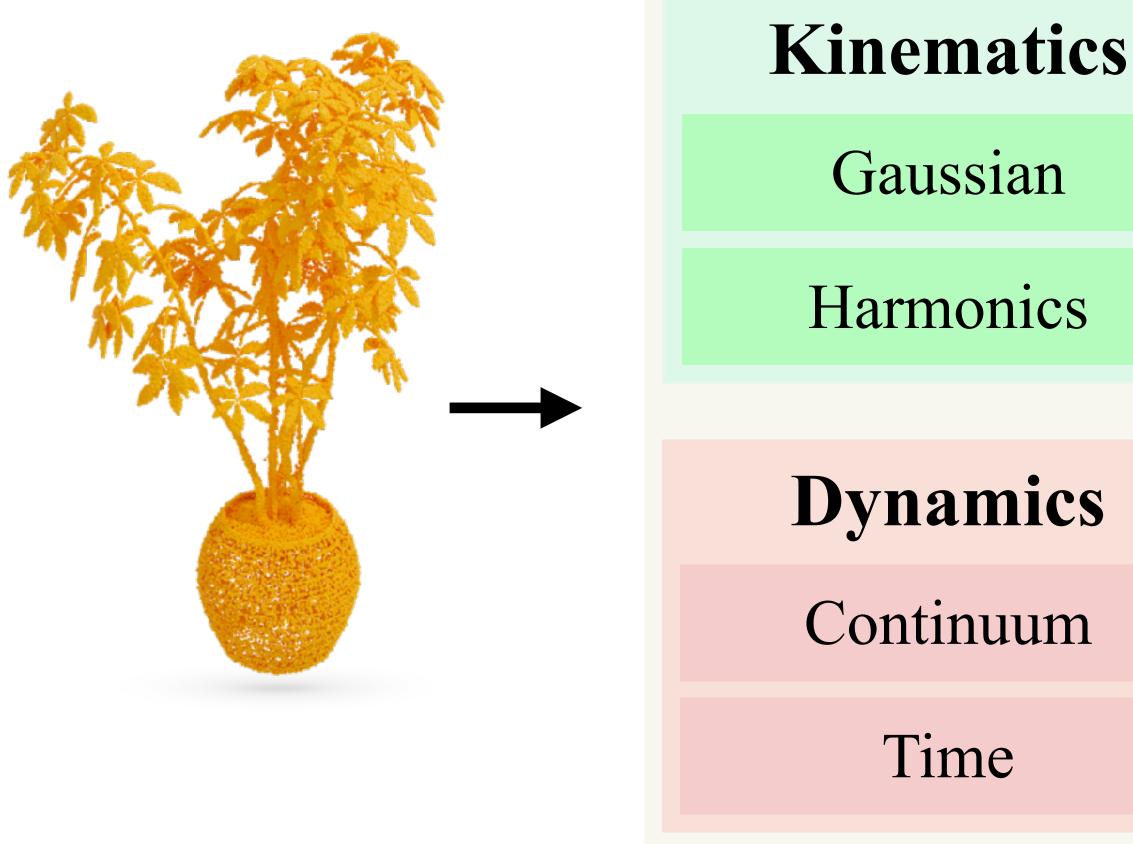
- Use particles as the geometry discretization.
- Easily obtain deformation gradients on particles.
- Support vast types of dynamics.

The only choice:



gradients on particles.

Summary of Generative Dynamics





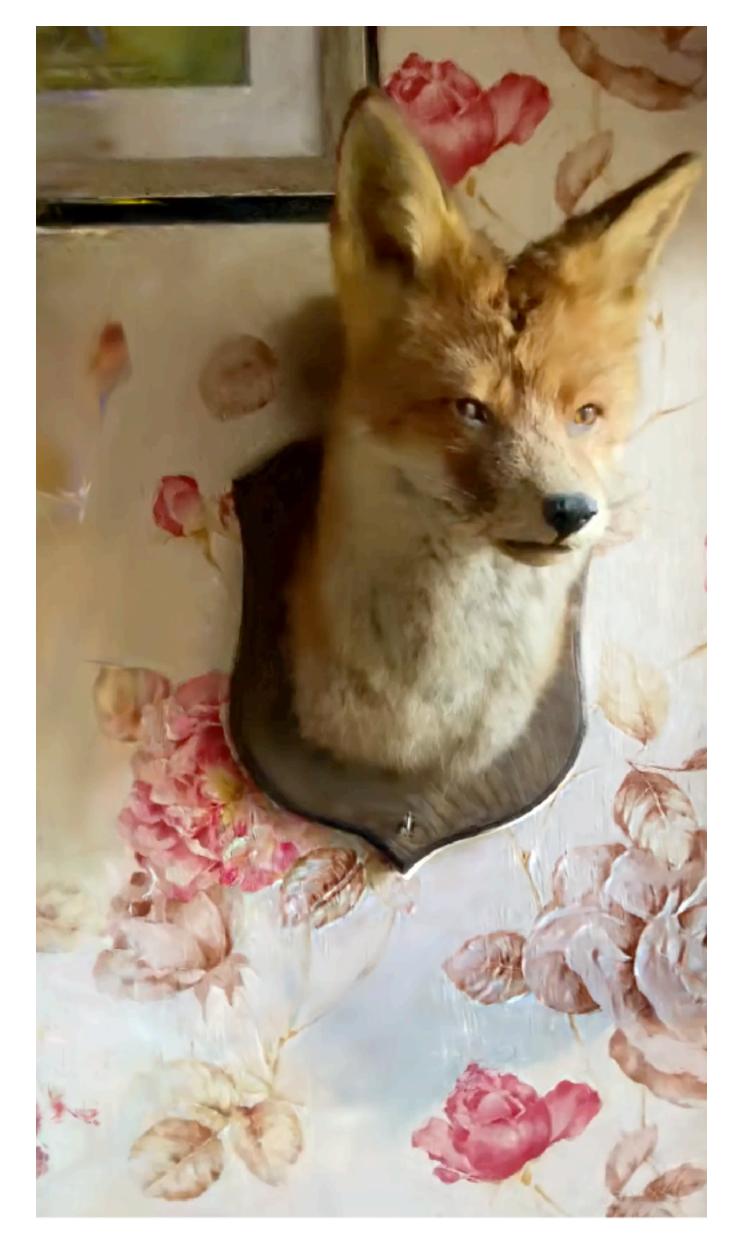




Rendering

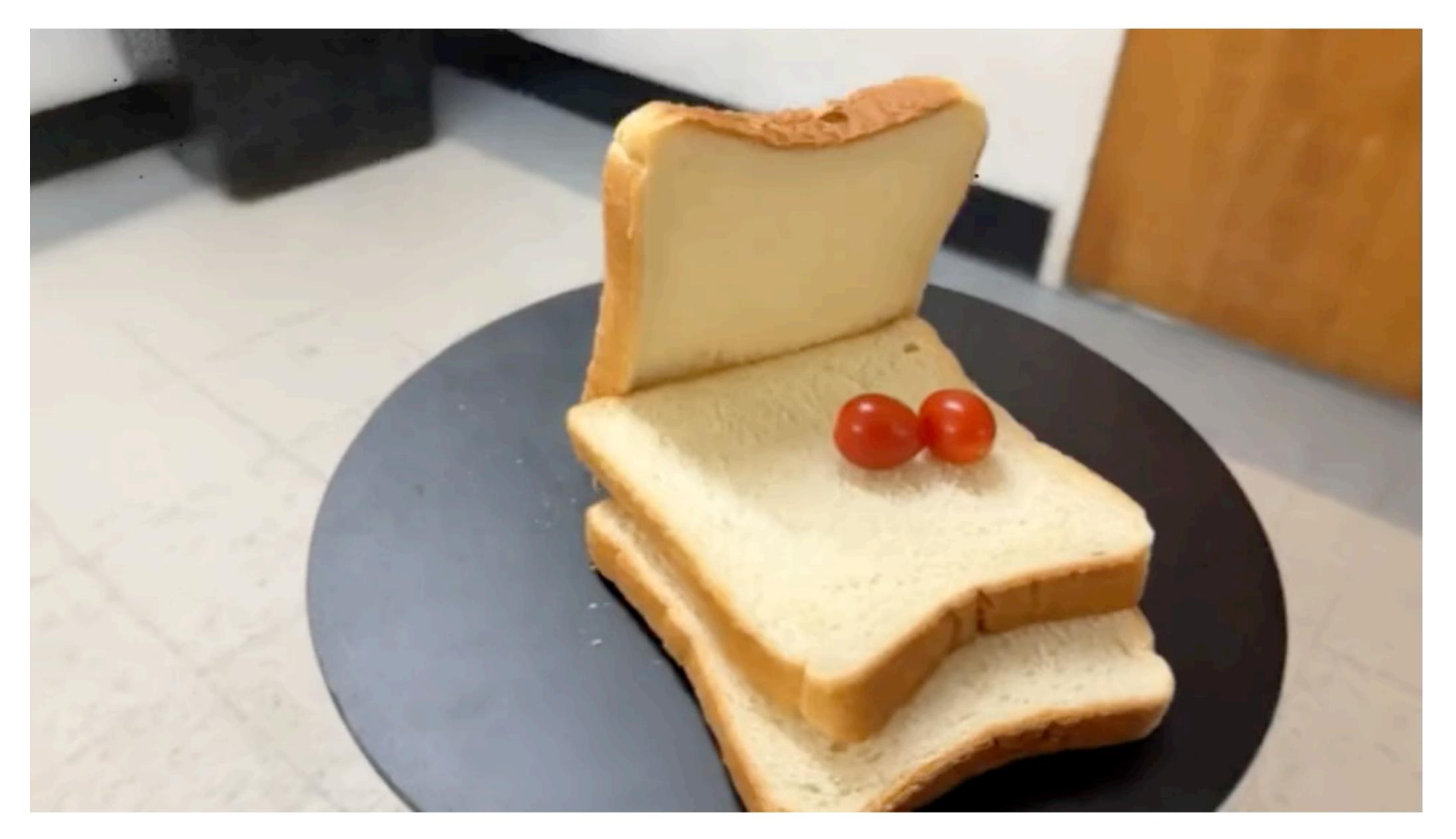
Results

Fox (Elasticity)

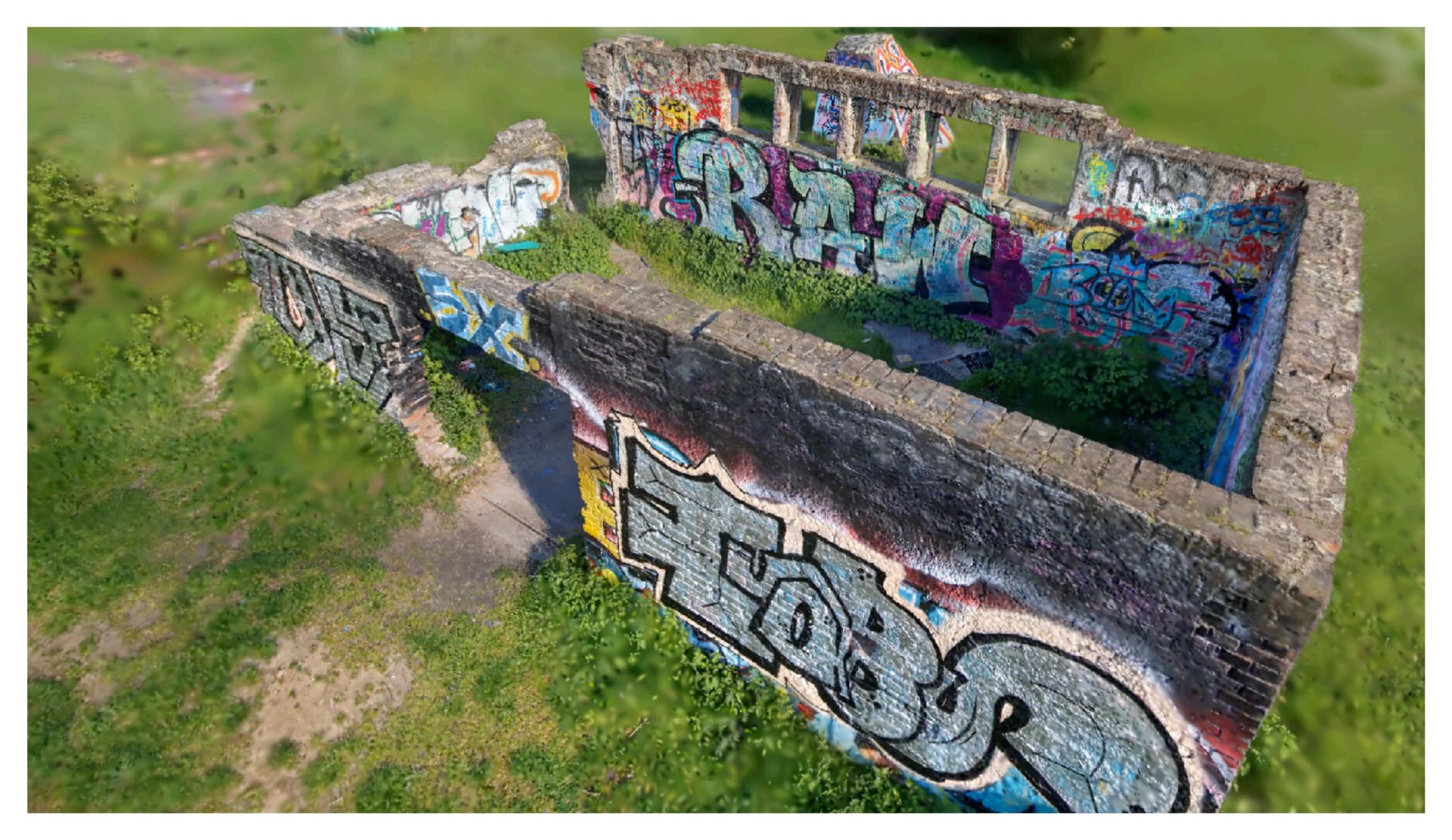


Plane (Metal)





Toast (Fracture)



Ruins (Sand)



Jam (Paste)





Thanks!



