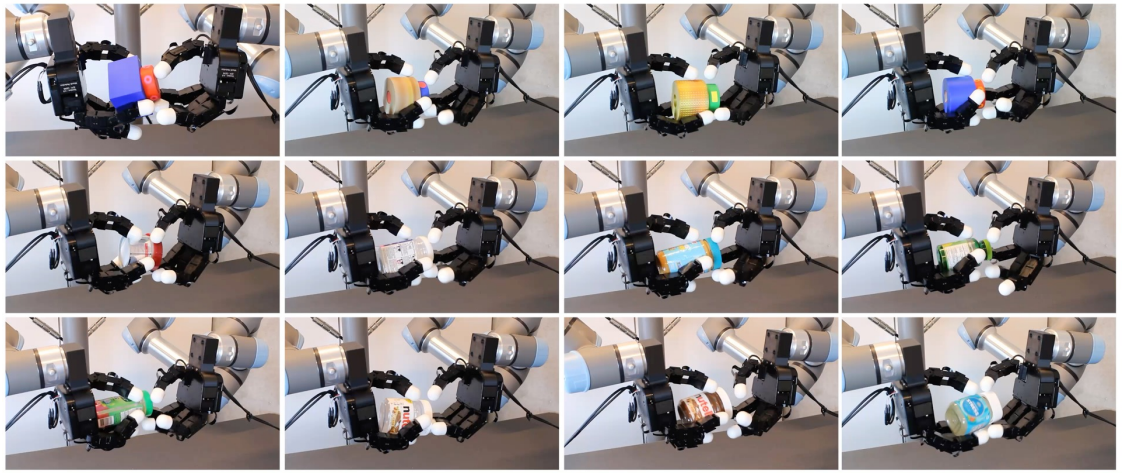
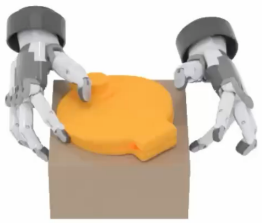
Scalable Neural Simulation with Lagrangian Mechanics

Chen Geng
Stanford
08/11/2026

Background: making simulation faithful to the real world is increasingly important...

- Roboticists struggle with building simulation models and sim-to-real gaps





[1] Mandi et al., DexMachina: Functional Retargeting for Bimanual Dexterous Manipulation. ICML 2026.

[2] Lin et al., Twisting Lids Off with Two Hands, CoRL 2024.

Background: making simulation faithful to the real world is increasingly important…

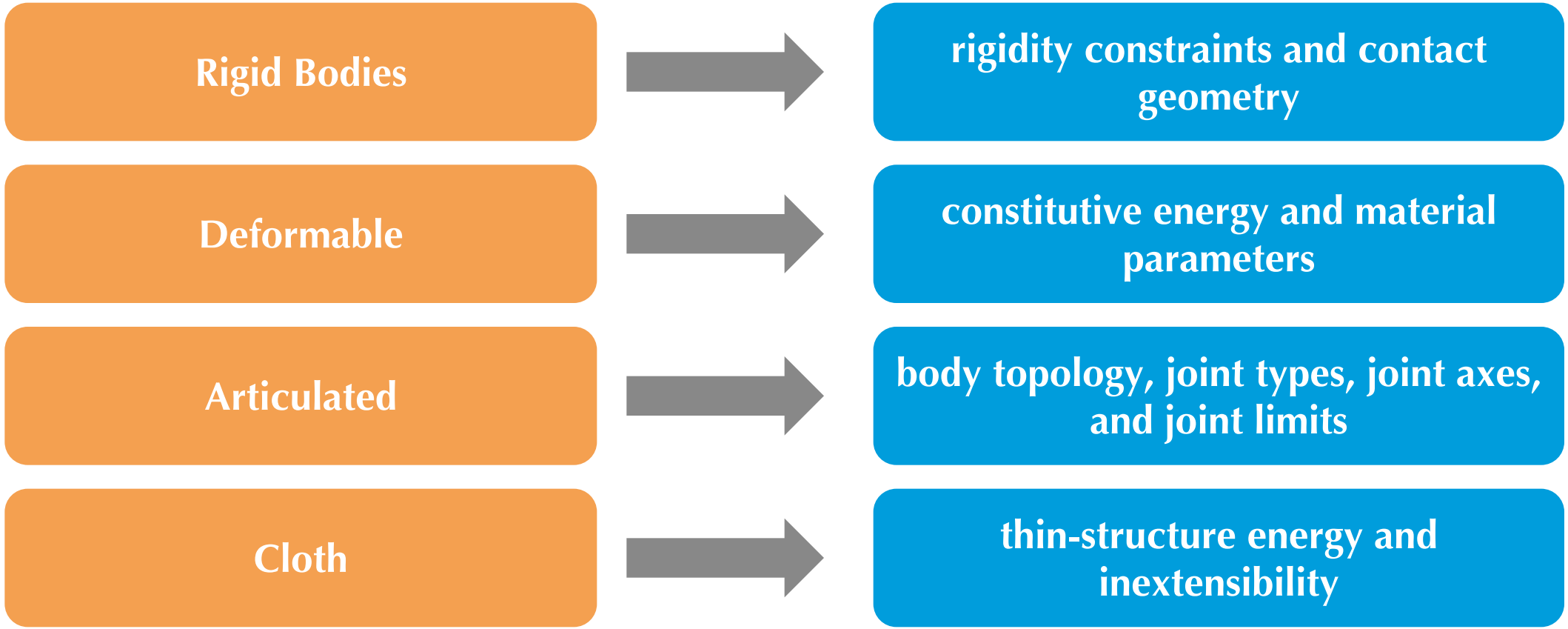
Background: making simulation faithful to the real world is increasingly important… But hard.

How to build simulatable models for these objects?

Inverse simulation: predict dynamic models without knowing (all) underlying physics.

Input: 3D shapes of static objects

Building accurate physical models is challenging and category-specific.

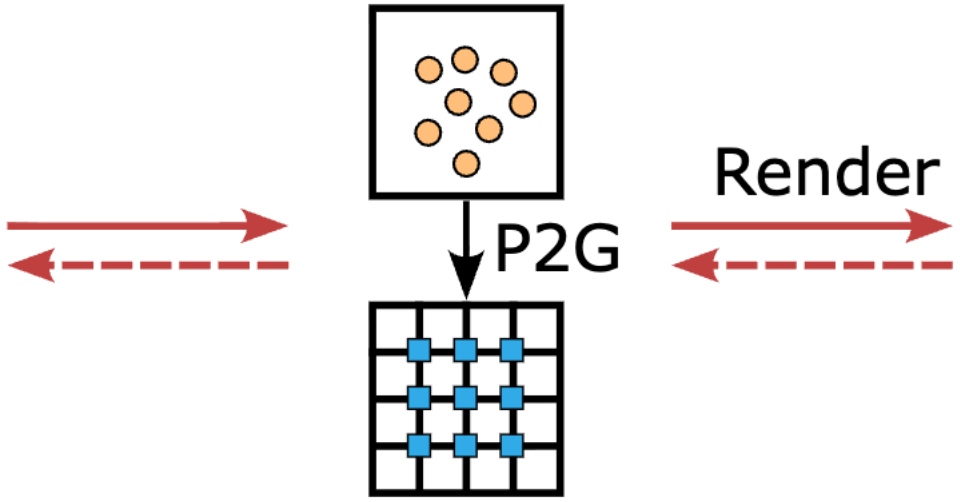


Is this fragmentation fundamental, or an artifact of how we set up the problem?

Inverse simulation solves inverse problem: finding physical models from observational data

Existing methods mostly follow a two-step approach:

Step 1: Finding an existing physical model.



E.g., Differentiable MPM
-> Category-specific

Zhang et al., PhysDreamer, ECCV 2024.

Existing methods mostly follow a two-step approach:

Step 1: Finding an existing physical model.

Step 2: Determining parameters w/ system ID.

Often fragile and not robust 😔

Objects can be even more complicated

Existing methods are hard to scale

Step 1: Finding an existing physical model.

Category-specific prior

Step 2: Determining parameters w/ system ID.

Often fragile and not robust 😔

Hard to scale to cross-category datasets!

Can we simulate diverse objects without category-specific physical equations?

Keep the universal principles. Avoid the category-specific equations.

Yes! It is possible, with a classical idea (latent space dynamics), and some data.

Back to basics: What does "physical model" actually mean?

Physical Principles
(universal)

- Lagrangian mechanics
  $L = T - V$ on the configuration manifold
- Euler-Lagrange equations
  $d/dt\ (\partial L/\partial \dot{q}) - \partial L/\partial q = Q$
- Energy conservation
- Newton's laws (equivalent)

Shared governing structure across many mechanical systems.

Category-Specific Equations
(category-specific)

Constitutive models:
- Hookean, St. VK, ARAP, corotational
  Maps deformation → stress

Constraint equations:
- Rigidity ($R^TR = I$), joints, inextensibility
  Defines what the object IS

Boundary / initial conditions

Different for every object category.

*Principles are **universal**. Category-specific equations are what make a rigid body different from cloth.*

Category-Specific physical equation: constraint forces are the hidden cost of over-parameterization.

Rigid body in Cartesian coords

- n vertices → 3n coordinates
- Rigid configurations form a 6-DOF manifold
- → 3n – 6 independent constraints
- naïvely using all pairwise distances introduces substantial redundancy
- **Intrinsic DOF = 6**
- **Gap: 3n – 6 wasted dimensions**

Three costs

1. **Labor-intensive modeling**
   - Each category needs expert setup
2. **No model for many objects**
   - Crumpled paper? Wilting flower?
3. **System ID is hard**
   - Inverse problem for E, ν, damping...

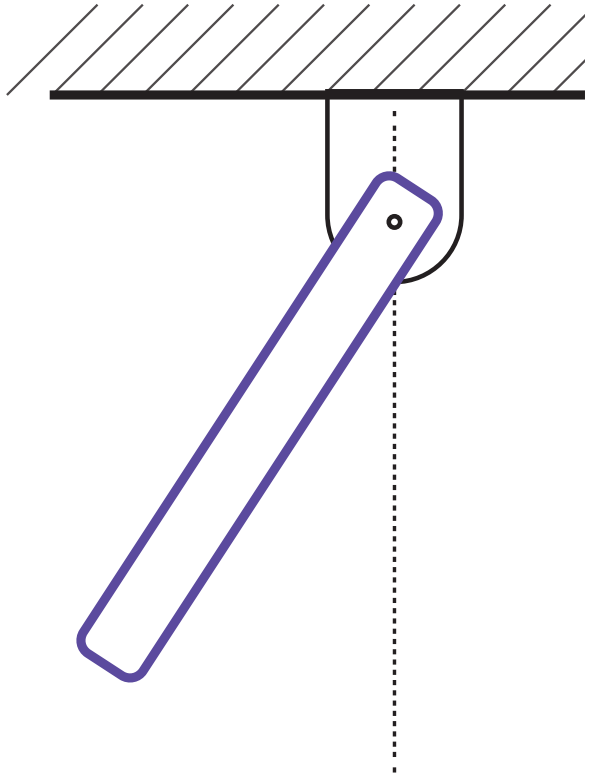
Constraint forces are the computational and modeling burden of over-parameterization.

D'Alembert's Principle:
The right coordinates make physical structure intrinsic

In the right generalized coordinates — coordinates that parameterize the constraint manifold — constraint forces vanish from the equations of motion.

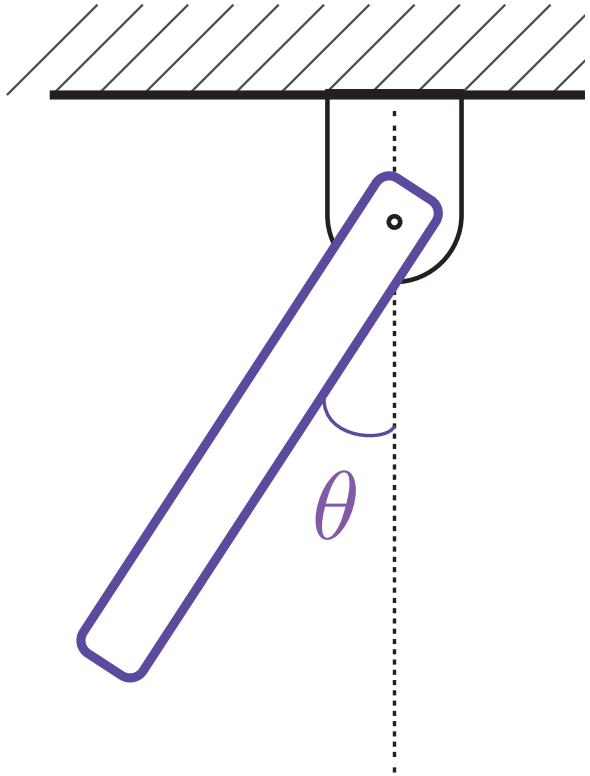
"With a proper choice of generalized coordinates, the problem of constraint may be avoided..." — Goldstein

A proper state parameterization scheme can simplify the modeling of physical systems.



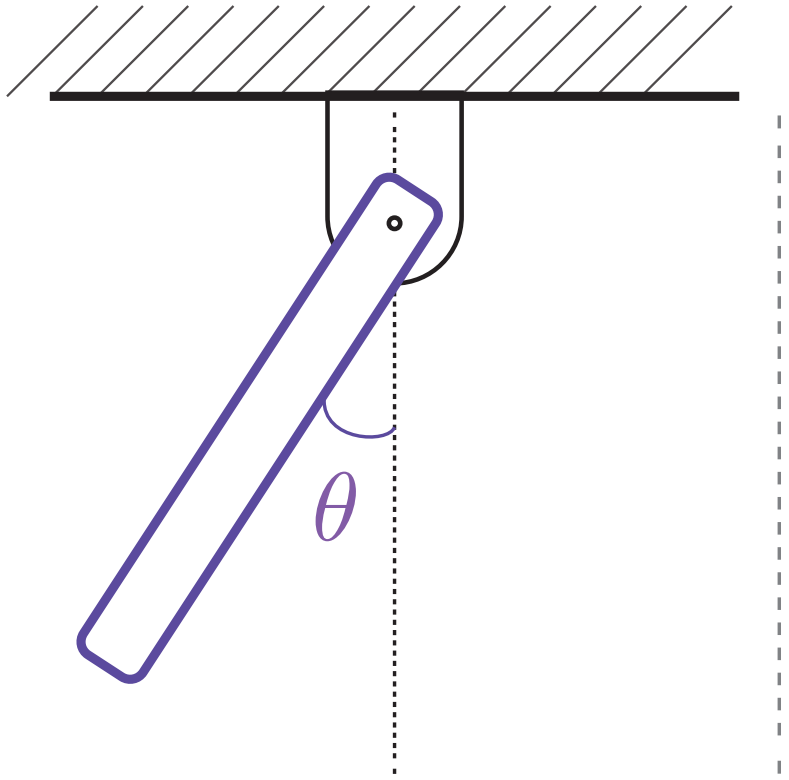
How to model the dynamics of this pendulum?

Symbolic representations are compact, yet not accessible

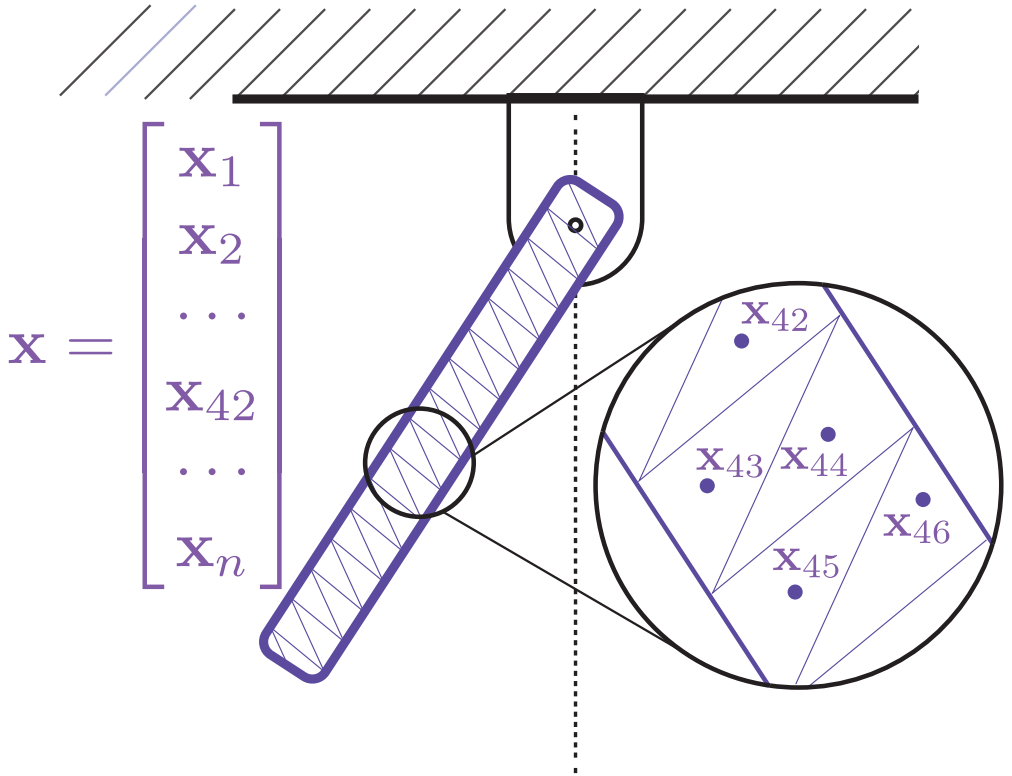


Physicists use symbolic representations.

Therefore, existing inverse modeling approaches use over-parameterized parameterizations

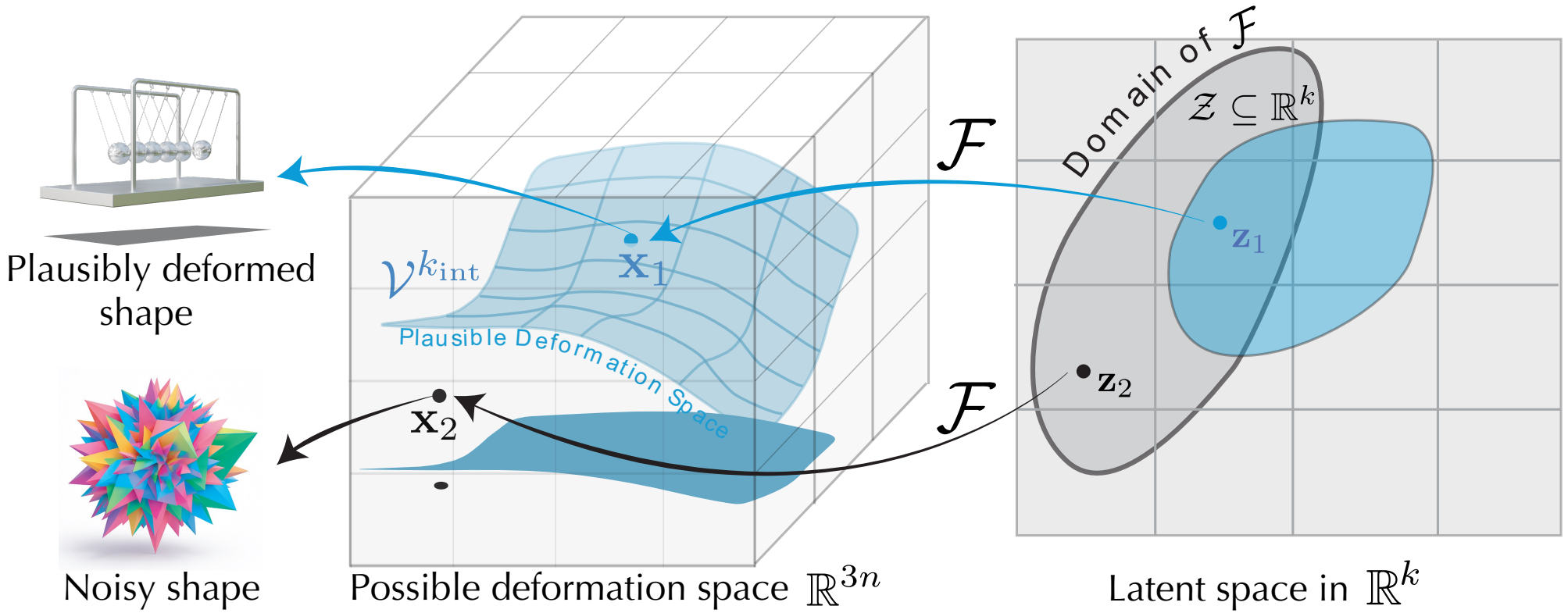


Physicists use symbolic representations.



Geometry-inherited parameterizations.

Rethinking generalized coordinates for object simulation:
Kinematic state parameterization



Lagrangian mechanics provides a common variational structure for a broad class of mechanical systems.

$$d/dt\ (\partial L/\partial \dot{q}) - \partial L/\partial q = Q$$

Formulating inverse simulation as a generative learning problem

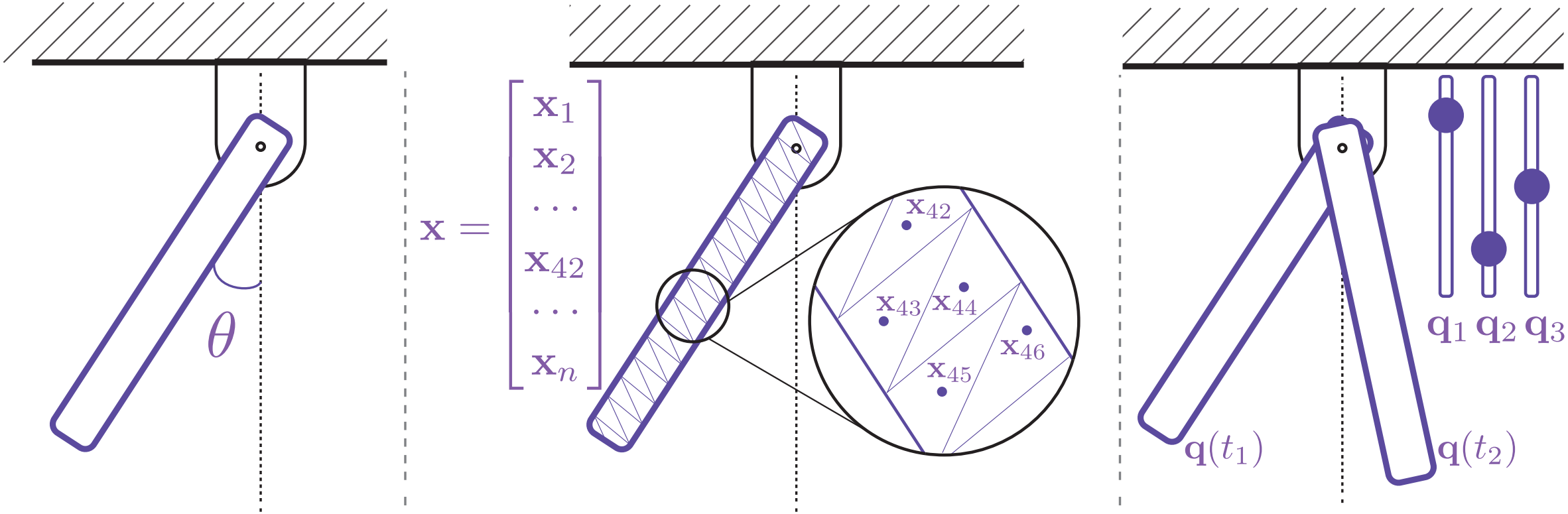
$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = Q$$

Instead of manually designing the coordinate map x=D(q), we learn it from data!

How do we find the right generalized coordinates for an arbitrary object?

Learn them from data.

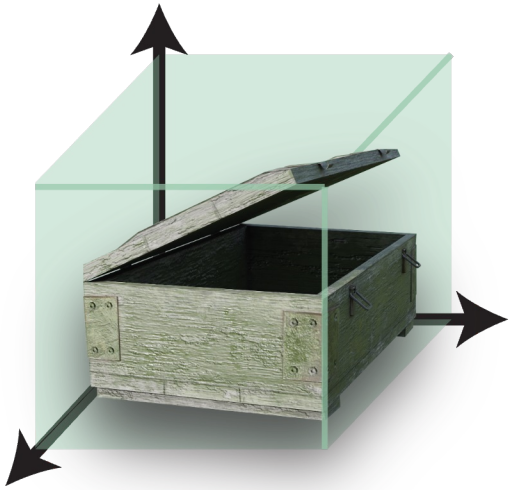
We introduce NeuROK: a generative, implicit kinematic state parameterization scheme.



Physicists use symbolic representations. Geometry-inherited parameterizations. Our NeuROK

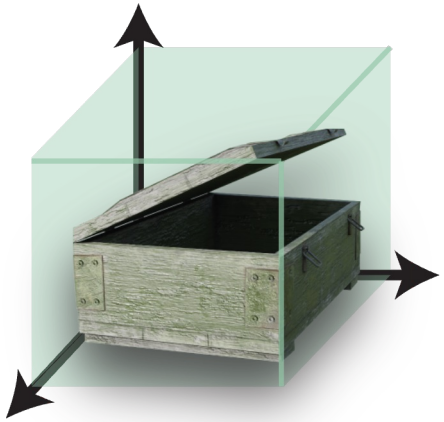
Geng*, He*, Gao*, Zhang, Wu, Wu. NeuROK: Generative 4D Neural Object Kinematics. CVPR 2026.

NeuROK maps each input static 3D object to an instance-specific latent kinematic state space

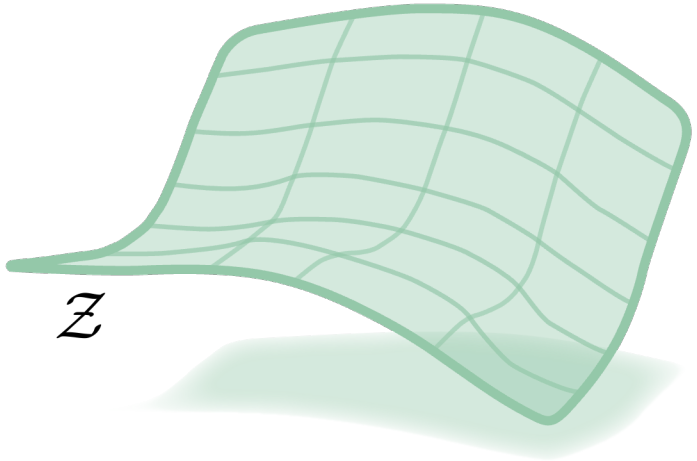


Input: Static 3D Object

NeuROK maps each input static 3D object to an instance-specific latent kinematic state space



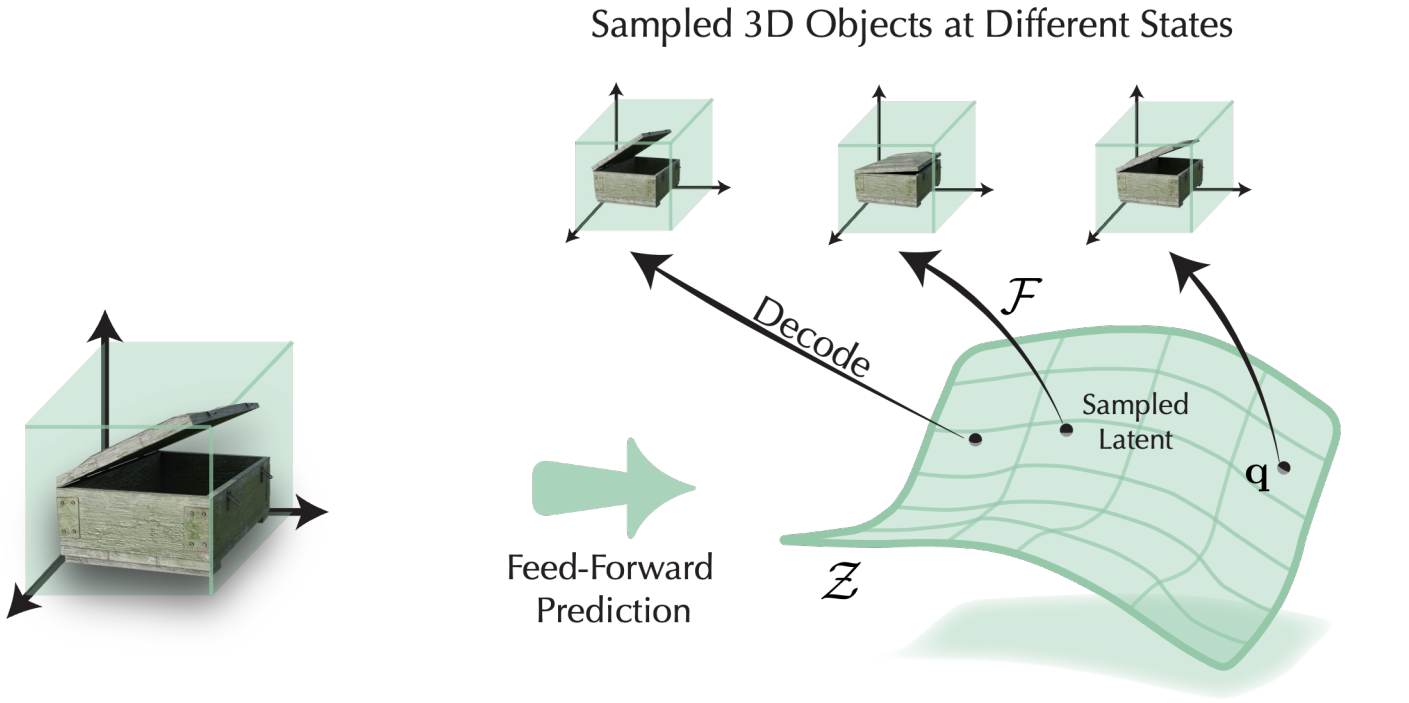
Feed-Forward Prediction



Input: Static 3D Object

Neural **O**bject **K**inematics (NeuROK)

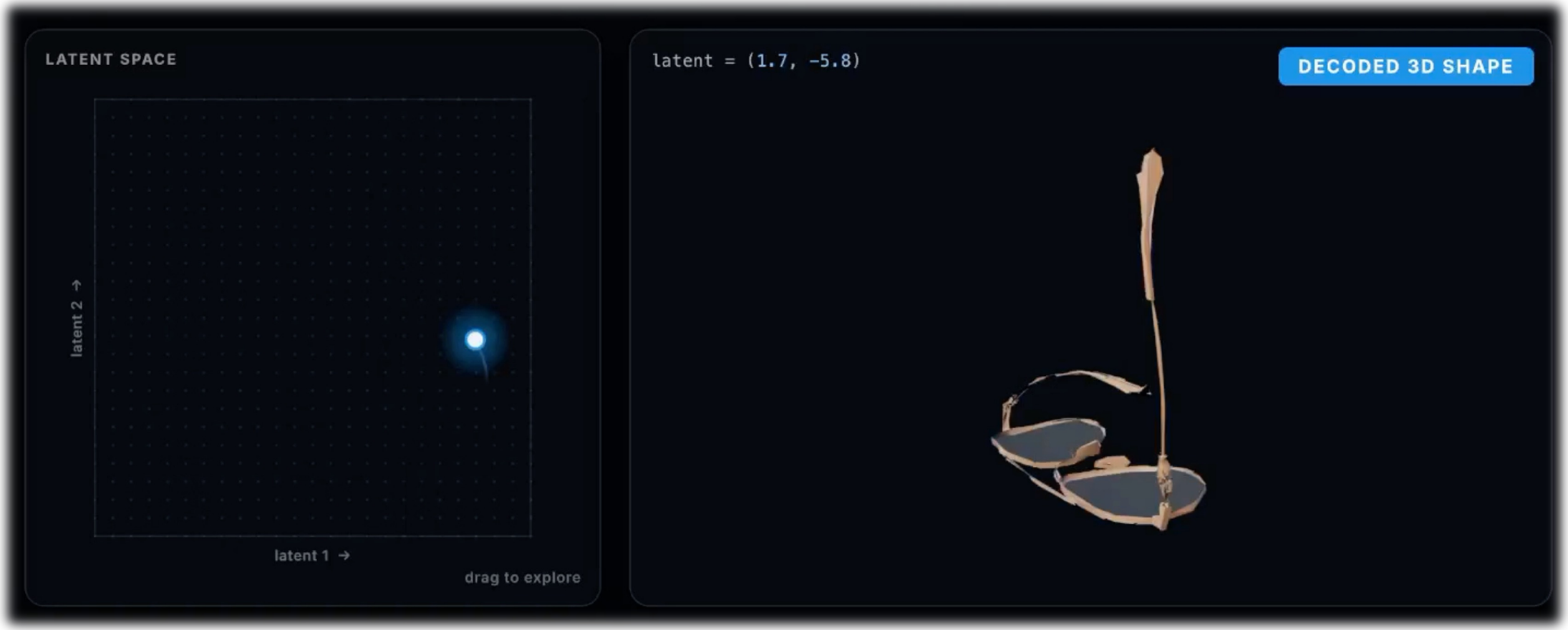
Each sampled latent on the predicted NeuROK can be mapped to a plausibly deformed kinematic state.



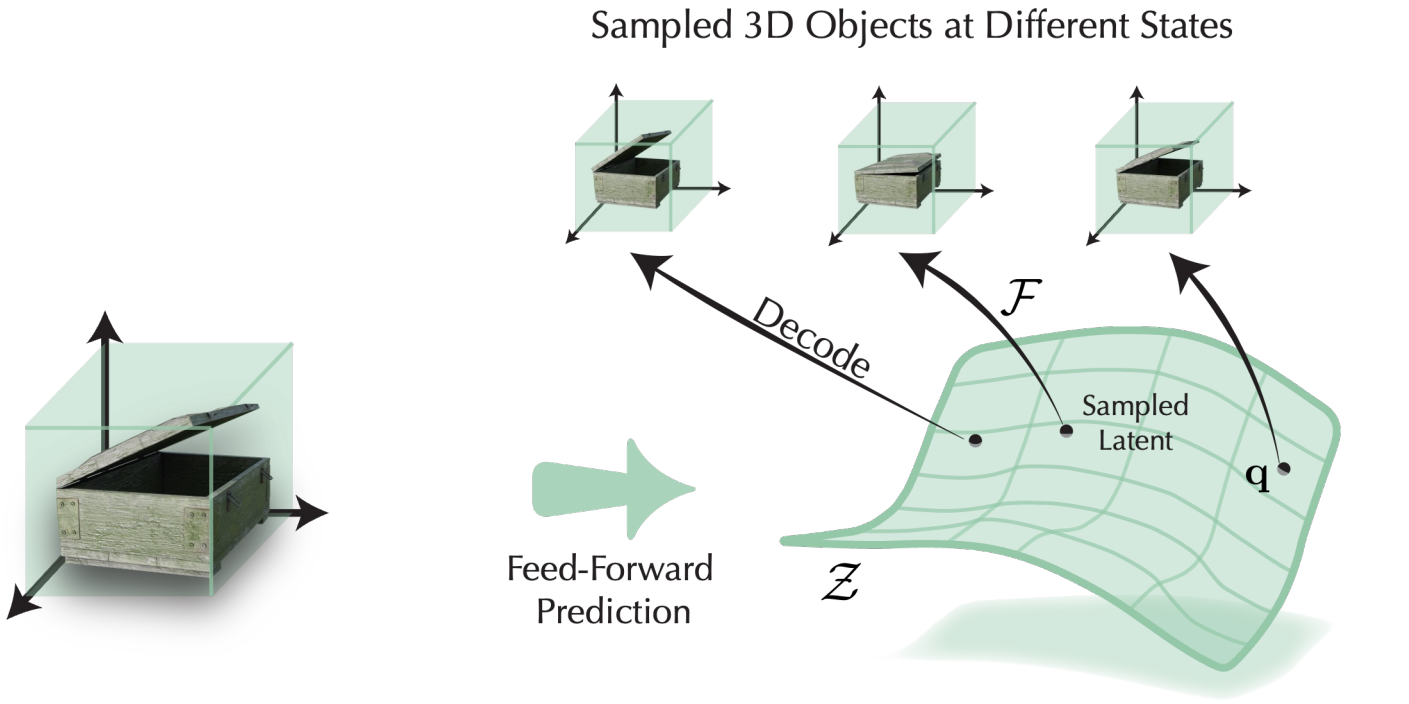
Input: Static 3D Object

Neural **O**bject **K**inematics (NeuROK)

Each sampled latent on the predicted NeuROK can be mapped to a plausibly deformed kinematic state.



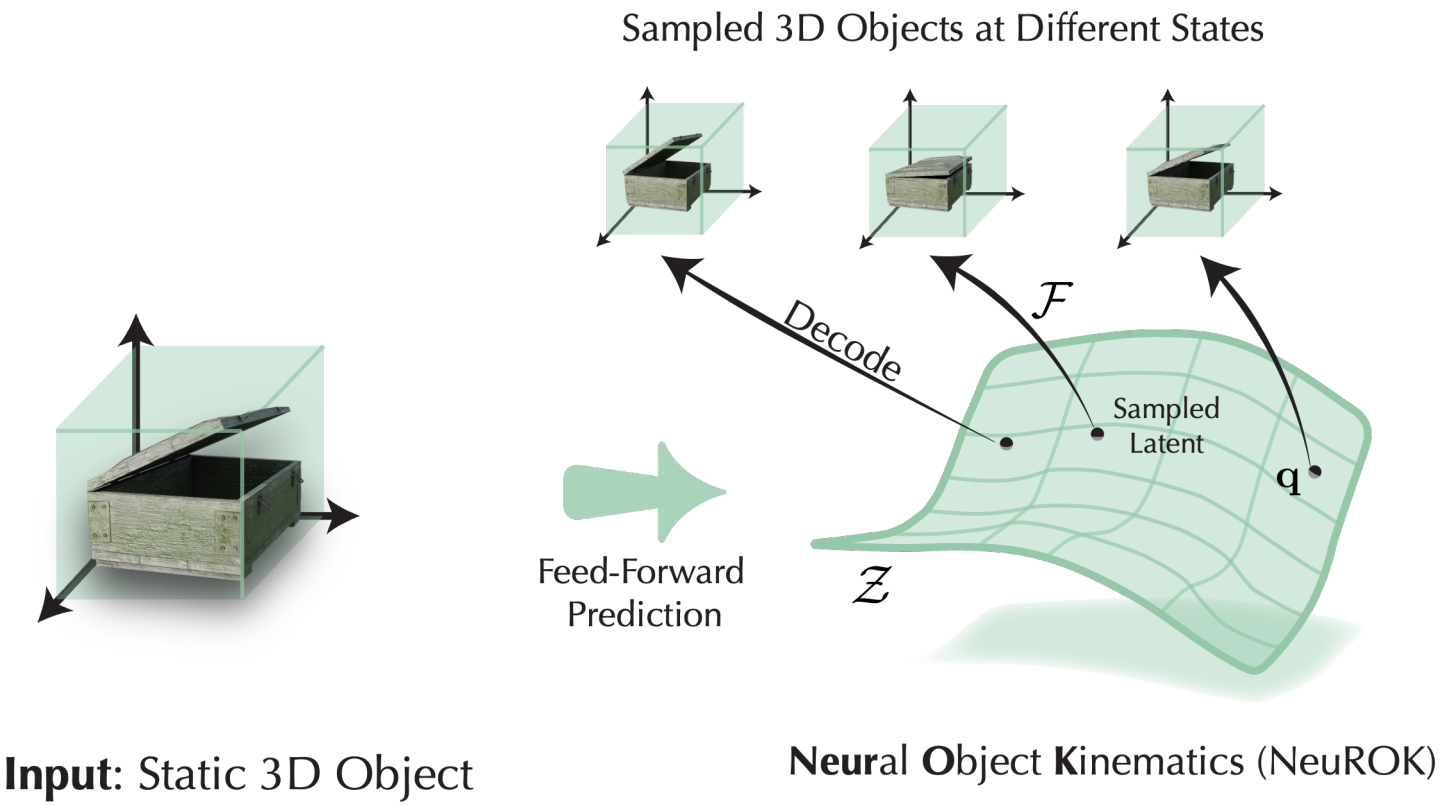
Each sampled latent on the predicted NeuROK can be mapped to a plausibly deformed kinematic state.



Input: Static 3D Object

Neural **O**bject **K**inematics (NeuROK)

NeuROK can greatly simplify the modeling of simulative 4D dynamics.



Input: Static 3D Object

Neural **O**bject **K**inematics (NeuROK)

A shared potential energy: the shared formula for every object in NeuROK.

$$V(z) = V_{elastic}(z) + V_{gravity}(z)$$

Rigid body

- Decoder D preserves edge lengths
- → $V_{elastic}(z) \equiv 0$ for all z
- **Rigidity is not imposed — it emerges.**

Deformable object

- $V_{elastic}$ captures elastic energy
- Decoder determines which deformations are "costly"
- **No separate constitutive model needed**

D'Alembert in action:
The learned coordinates implicitly define a constitutive model of admissible deformation.

Deriving the equations of motion

Starting point: The Lagrangian in Cartesian coordinates is standard:

$$L = T - V = \tfrac{1}{2}\, m\, ||\dot{x}||^2 - V(x)$$

But $x = D(z)$, so $\dot{x} = J_z\, \dot{z}$ where $J_z = \partial D/\partial z$ is the decoder Jacobian

$$T = \tfrac{1}{2}\, m\, ||\dot{x}||^2 \;=\; \tfrac{1}{2}\, m\, (J_z\, \dot{z})^T (J_z\, \dot{z}) \;=\; \tfrac{1}{2}\, m\, \dot{z}^T\, \mathbf{G(z)}\, \dot{z}$$

Key equation: the Euler-Lagrange equation (general form)

$$\mathbf{d/dt\ (\partial L/\partial \dot{q}) - \partial L/\partial q = Q}$$

Universal for any $L(q, \dot{q})$.
q = generalized coordinates, Q = external forces.

Apply E-L equation: $d/dt\ (\partial L/\partial \dot{z}) - \partial L/\partial z = 0$. $\partial L/\partial \dot{z} = m\, G(z)\, \dot{z}$; d/dt introduces Christoffel symbols via z-dependence of G

$$\mathbf{m\, G(z)\, \ddot{z} + C(z, \dot{z}) + \nabla_z\, V = 0}$$

G from decoder Jacobian • C from decoder Hessian (Christoffel symbols) • V is generic

Do learned coordinates need to be minimal?
Covariance of E-L equations

The concern: What if $k > k_{int}$? What if the learned coordinates are redundant?

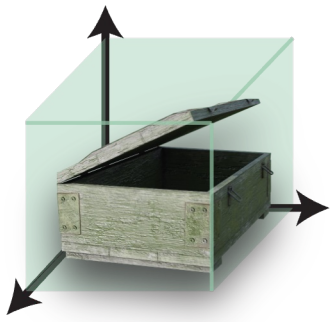
We prove that: Covariance of the Euler-Lagrange equations.

The E-L equations are form-invariant under smooth coordinate transformations.
If q and $\tilde{q}$ are two coordinate systems with $\tilde{q} = \tilde{q}(q)$, the E-L equations in $\tilde{q}$ have the same form, if:
- The transformation is C2 smooth,
- And the transformation is (locally) surjective.

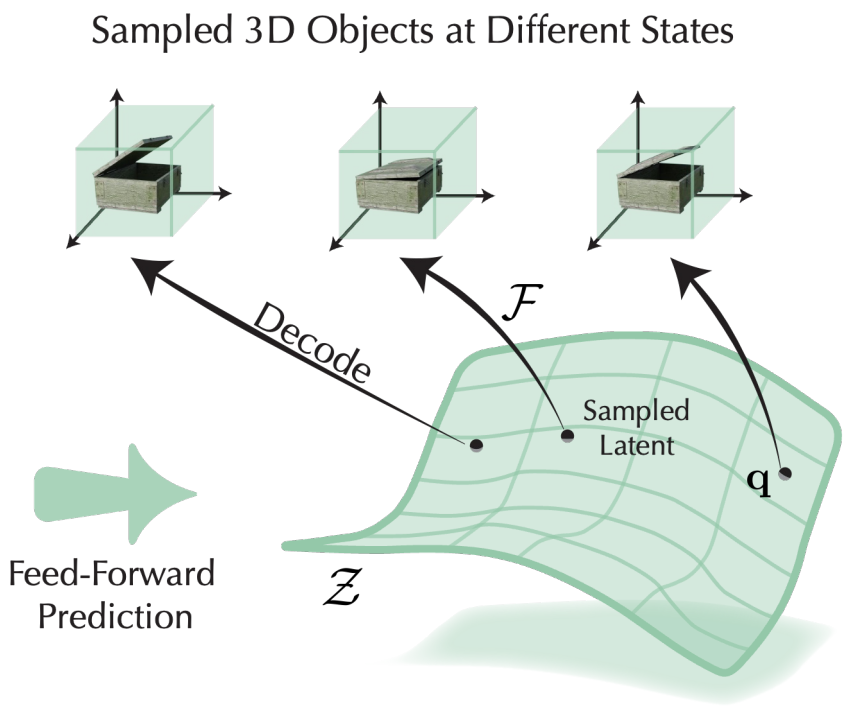
What this theoretical result means for NeuROK:

- Even if learned z-space has **redundant dimensions**, the E-L equations on (Z, G) still produce correct dynamics.
- We don't need **canonical** or **minimal** coordinates — ANY smooth parameterization covering the constraint manifold yields correct physics.
- Because minimal coordinates are unnecessary, we can learn a generative, potentially redundant deformation space.
- This naturally yields a generative modeling problem: same objective!

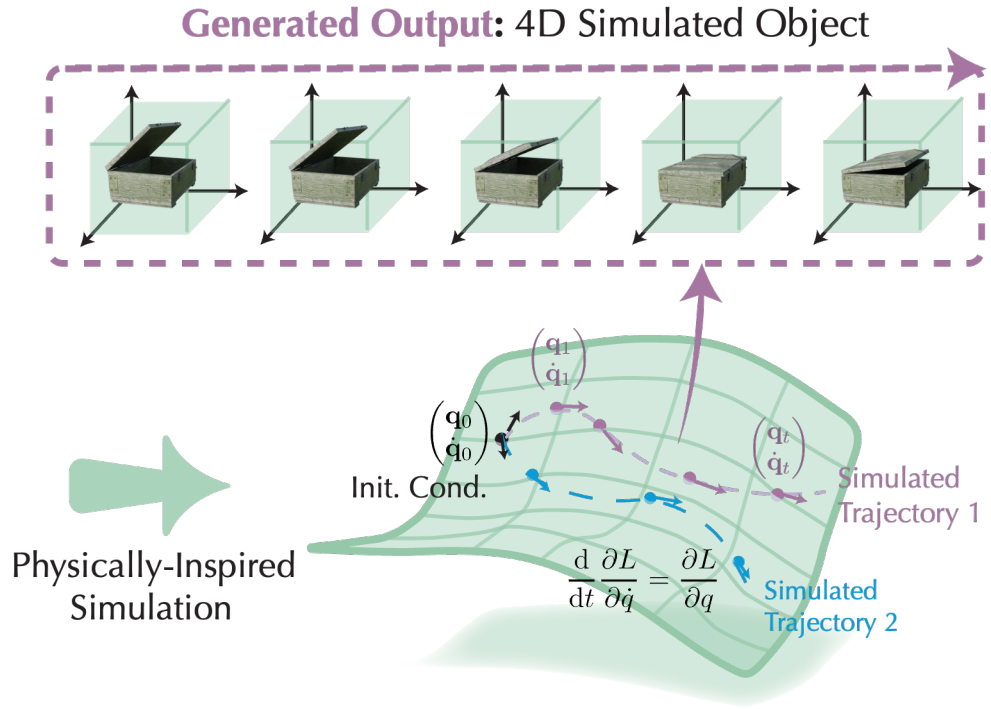
Simulating with the equation



Input: Static 3D Object

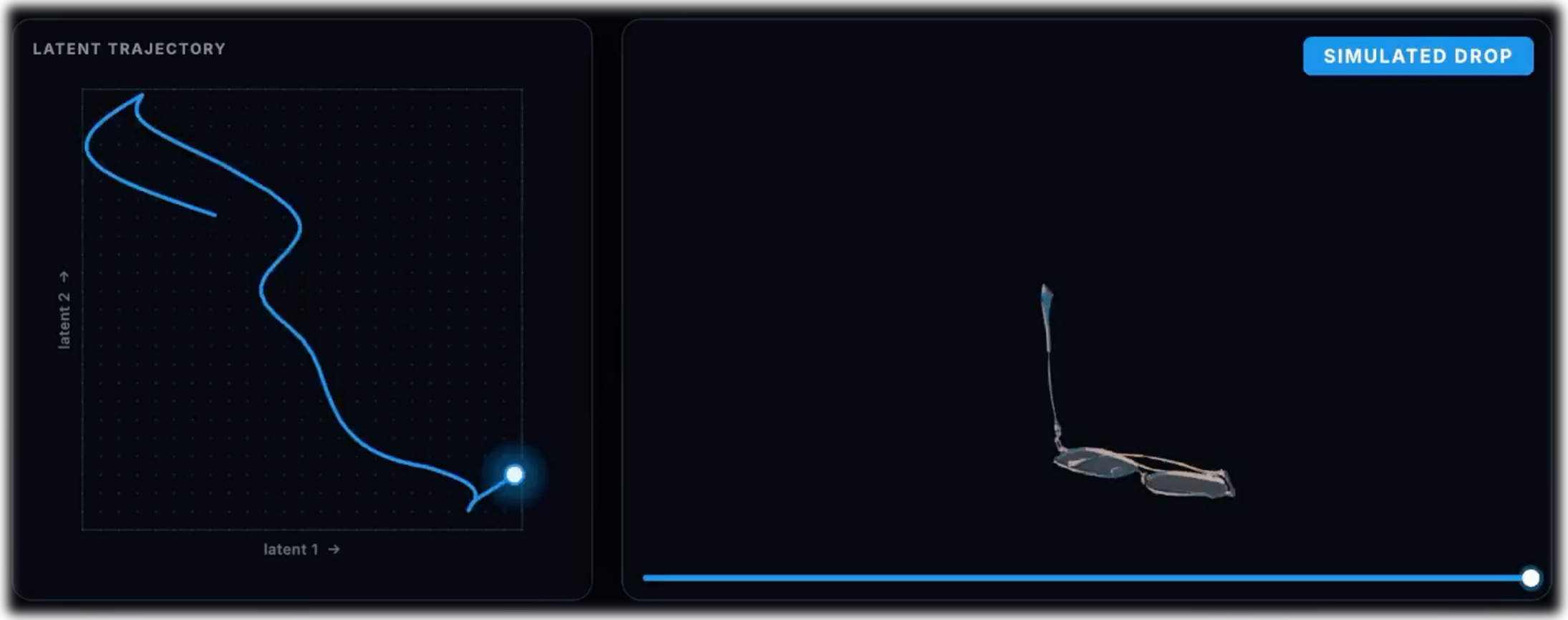


Neural **O**bject **K**inematics (NeuROK)

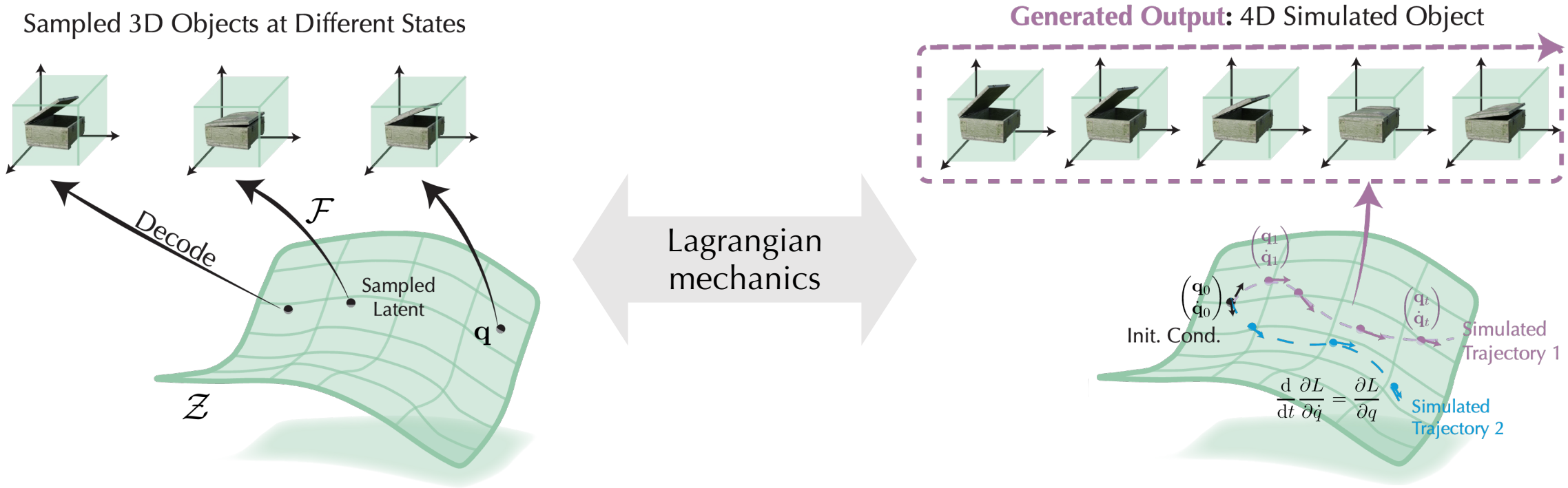


Generative Dynamics of NeuROK

Simulating with the equation



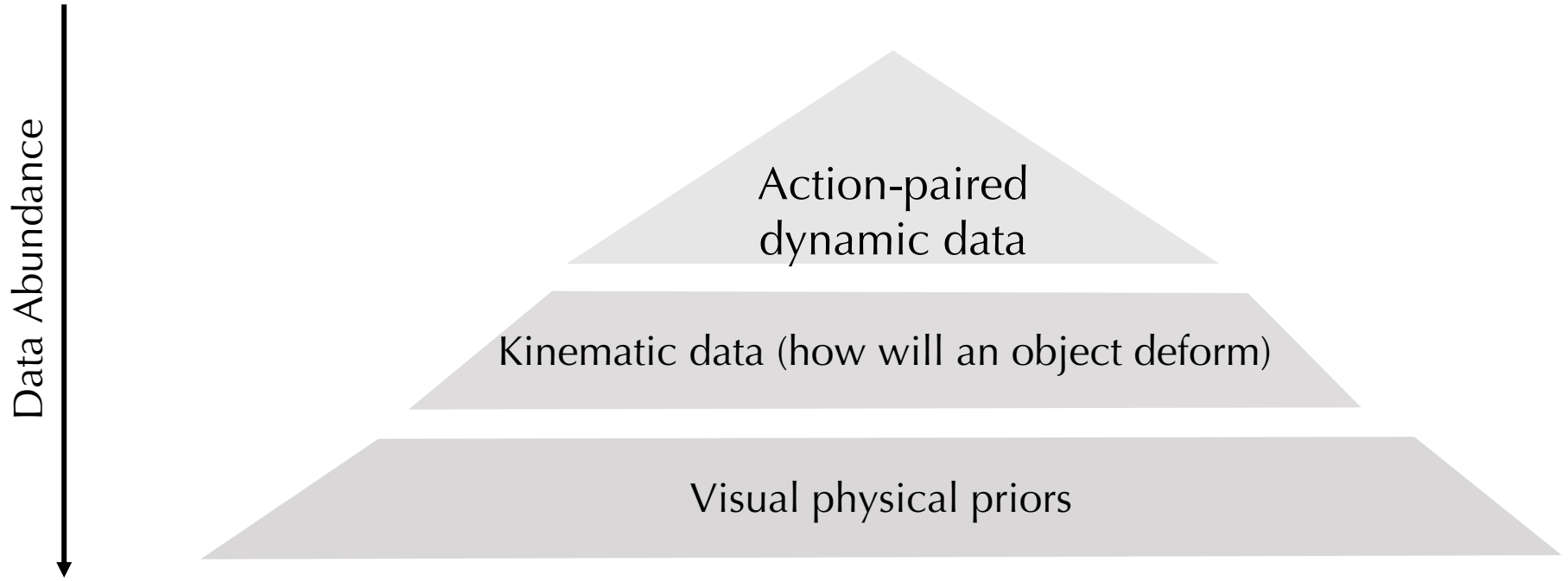
Lagrangian mechanics is a bridge between learned space and physical conditions.



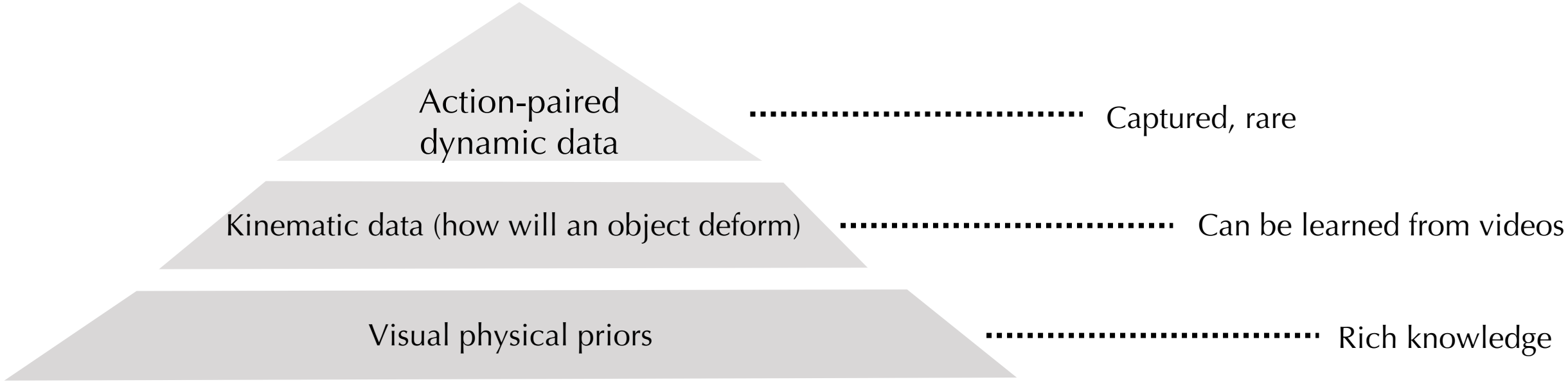
Learning-based Kinematics

Physical conditions

Data pyramid for neural simulation



Data pyramid for neural simulation

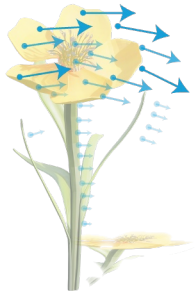


How do we learn the NeuROK?

From a large-scale 4D dataset, and a transformer-based autoencoder.

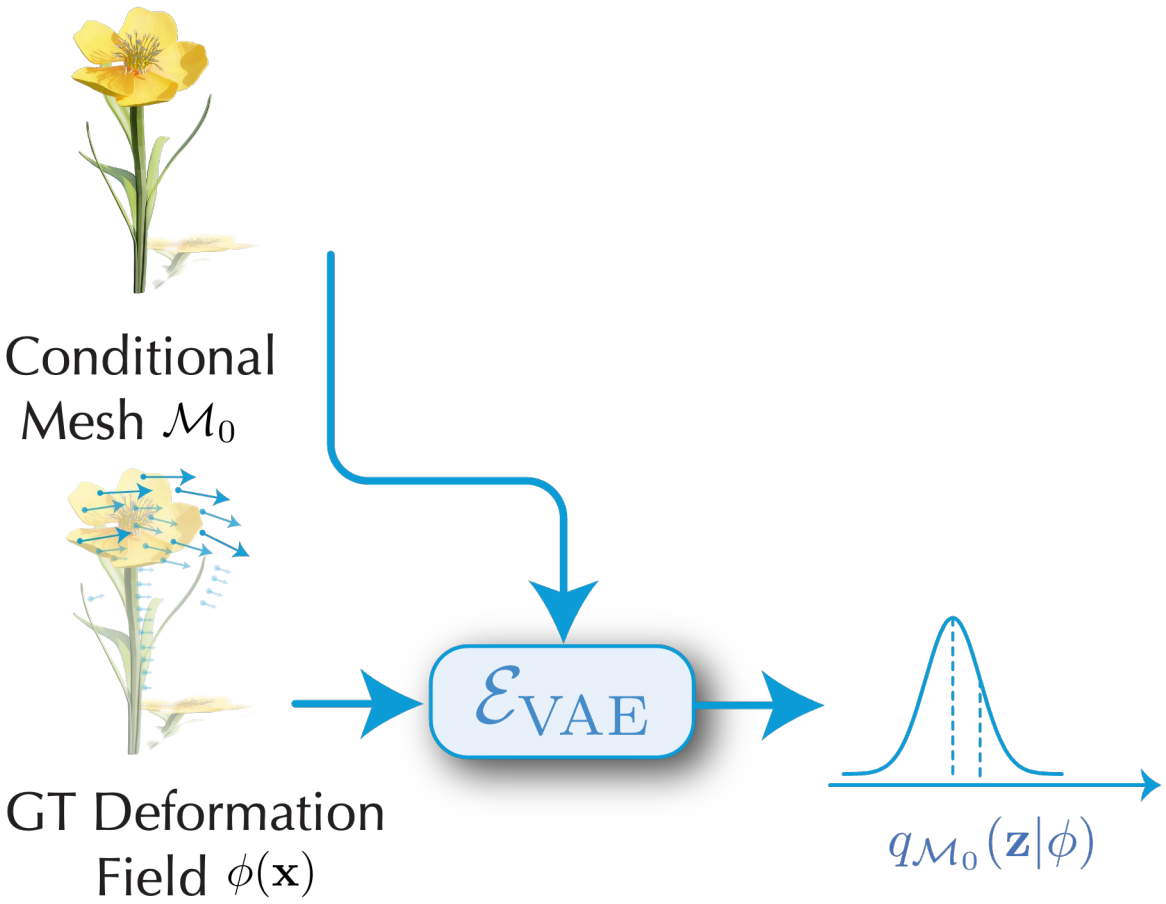
NeuROK is trained with a transformer-based encoder-decoder framework.

Conditional
Mesh $\mathcal{M}_0$

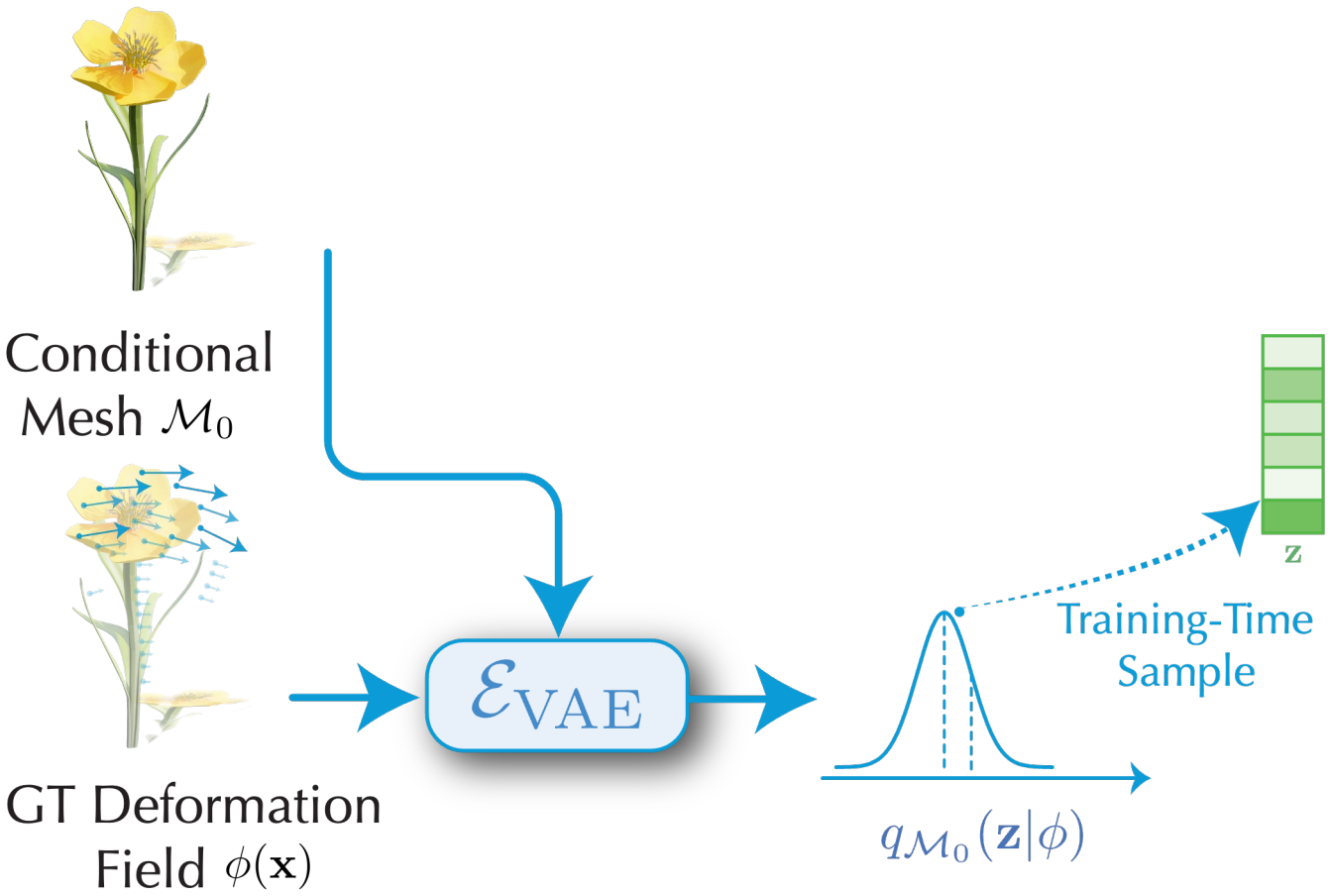


GT Deformation
Field $\phi(\mathbf{x})$

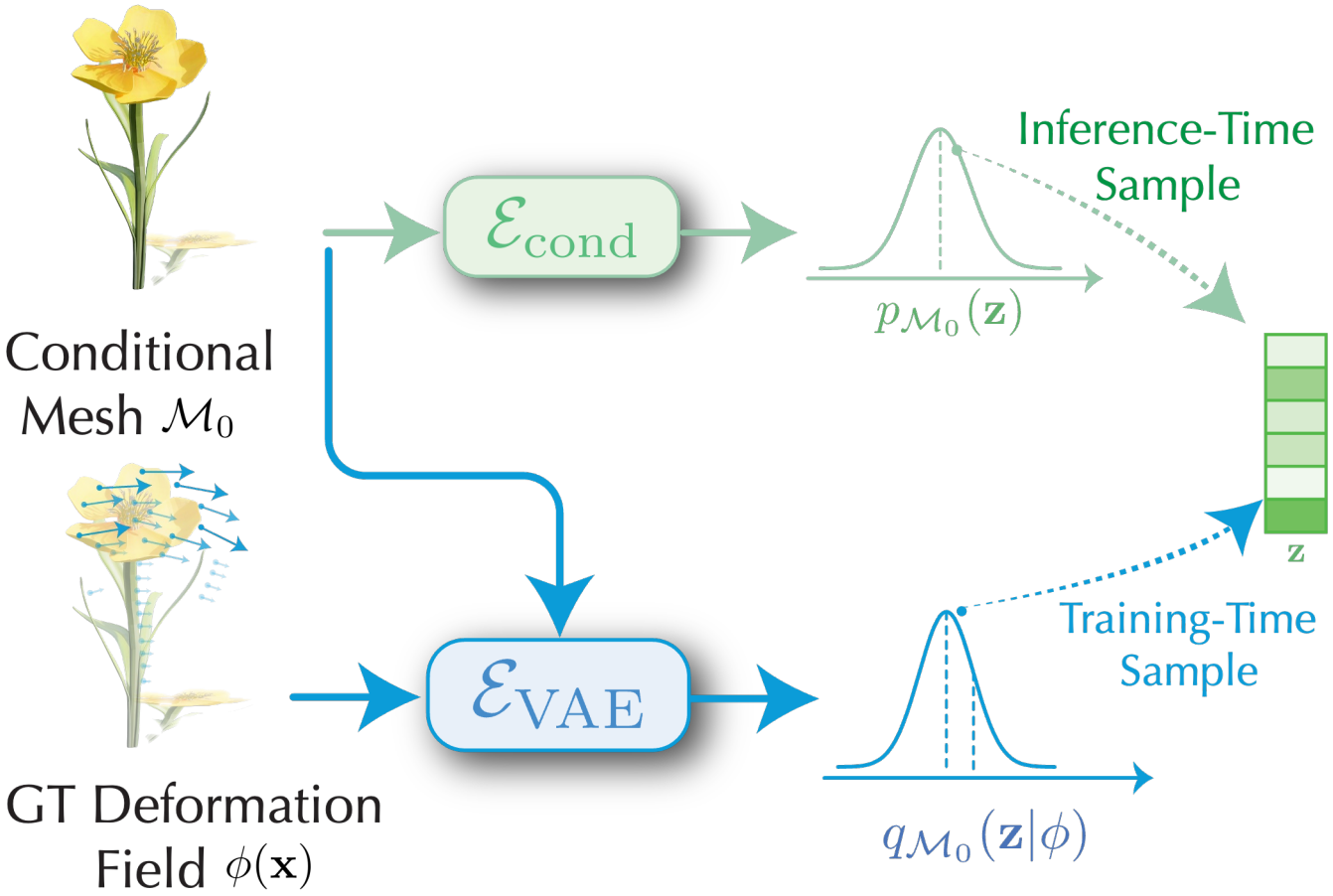
We train NeuROK with a generative target.



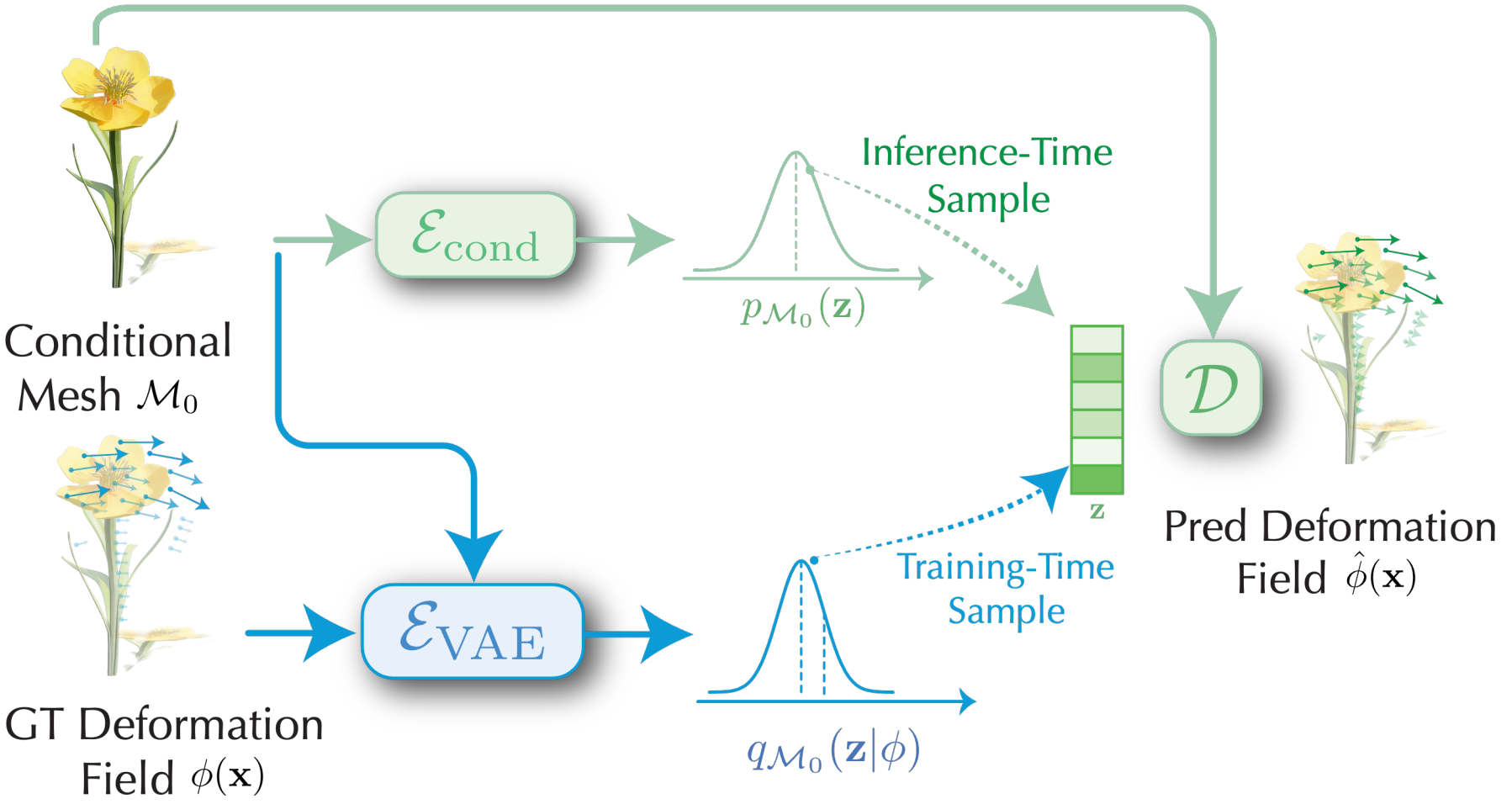
During training, we sample latent from the predicted posterior.



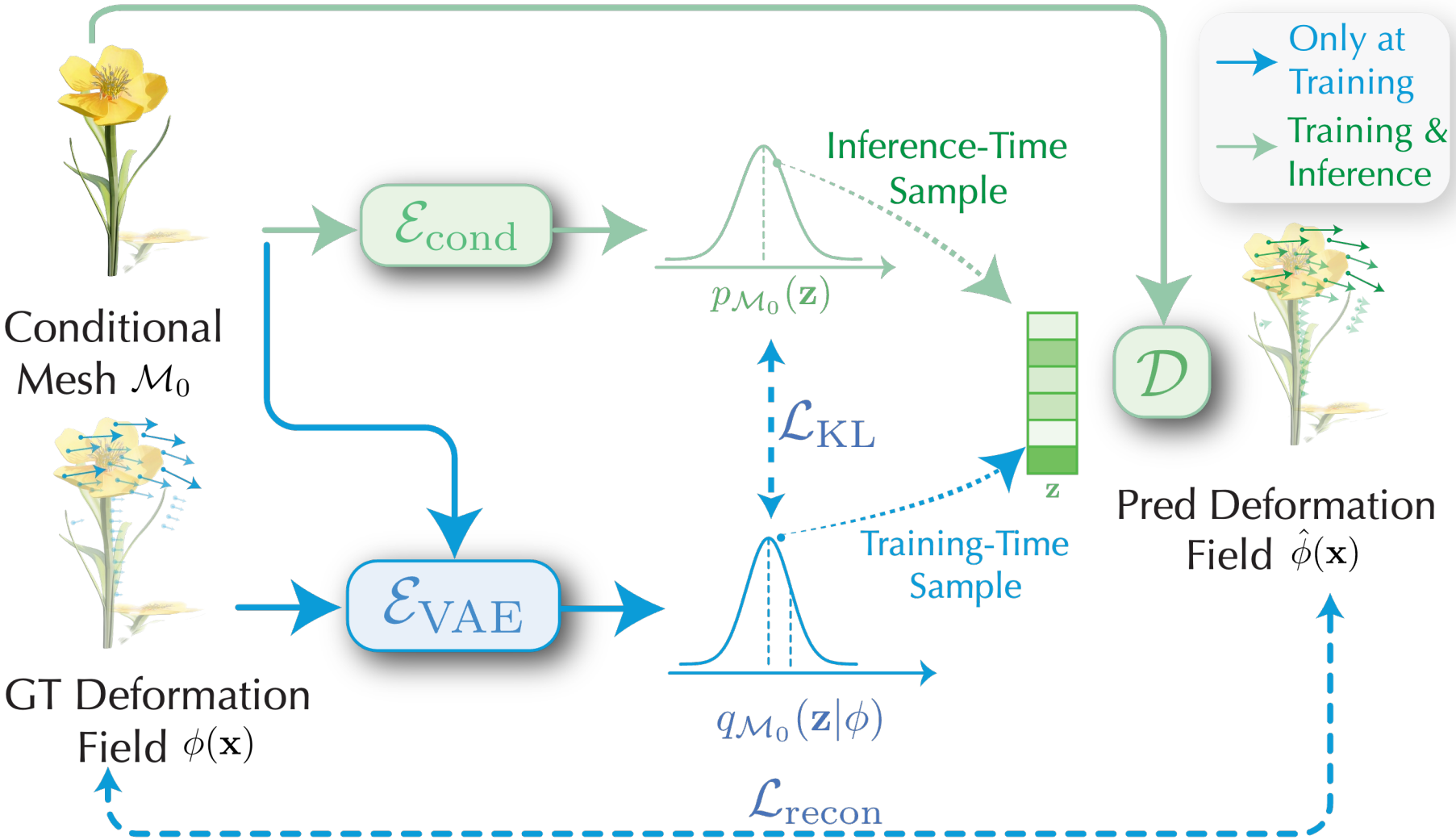
We also learn a conditional encoder that predicts an instance-specific prior distribution that allows inference-time sampling.



The sampled latent is decoded with a learned decoder.



We supervise with reconstruction loss and KL regularization.

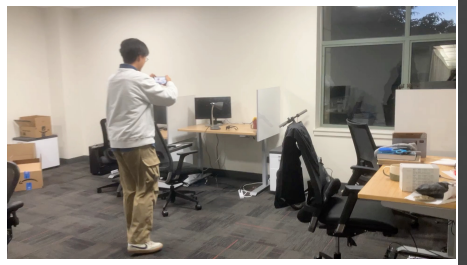


Scan your room and make it interactable!

Scanning our Stanford office with an iPhone…

Scanning process

… and interacting with it!



Scanning process

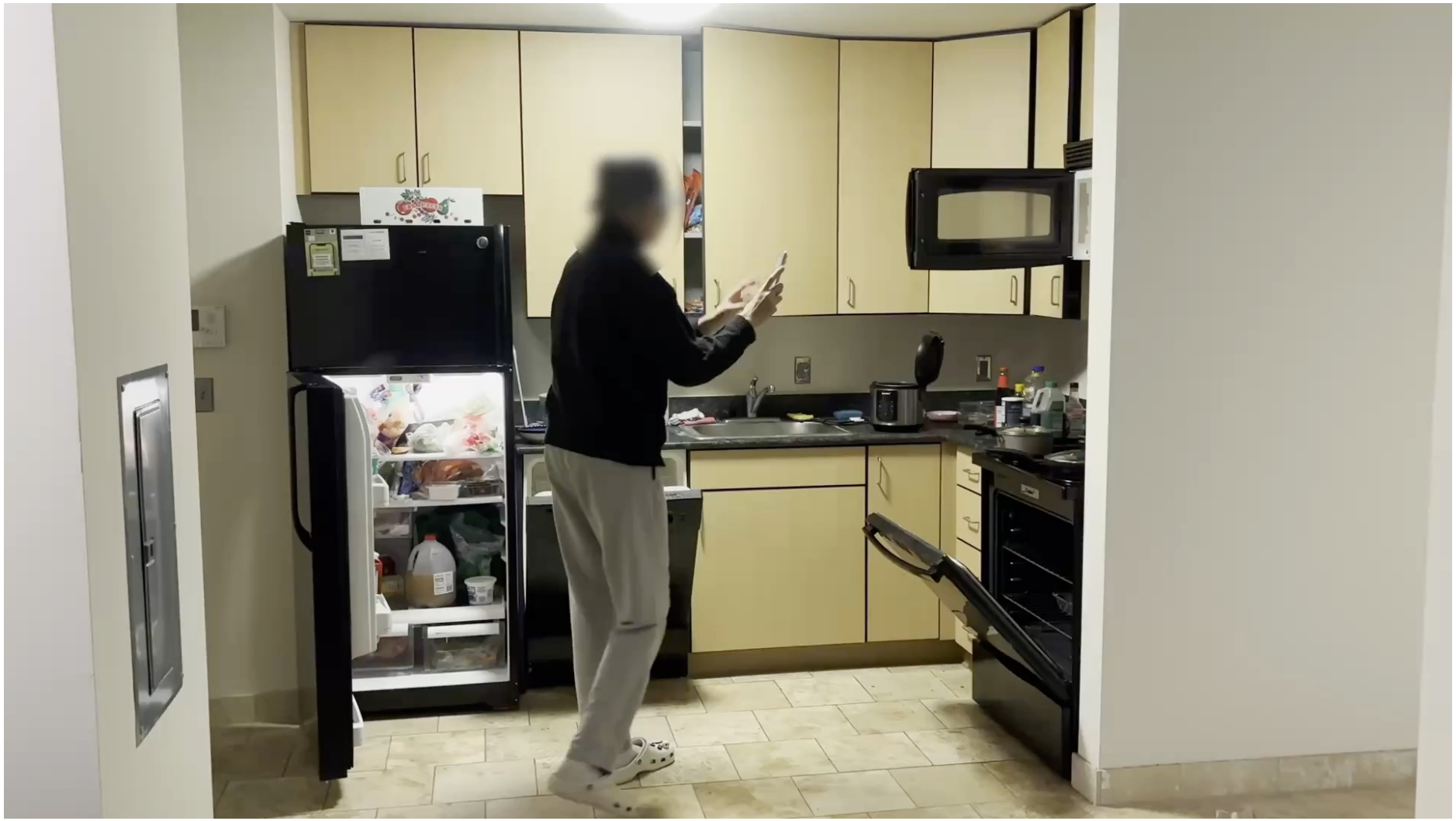
Scanning our Stanford kitchen with an iPhone…

Scanning process

… and interacting with it!

Scanning process

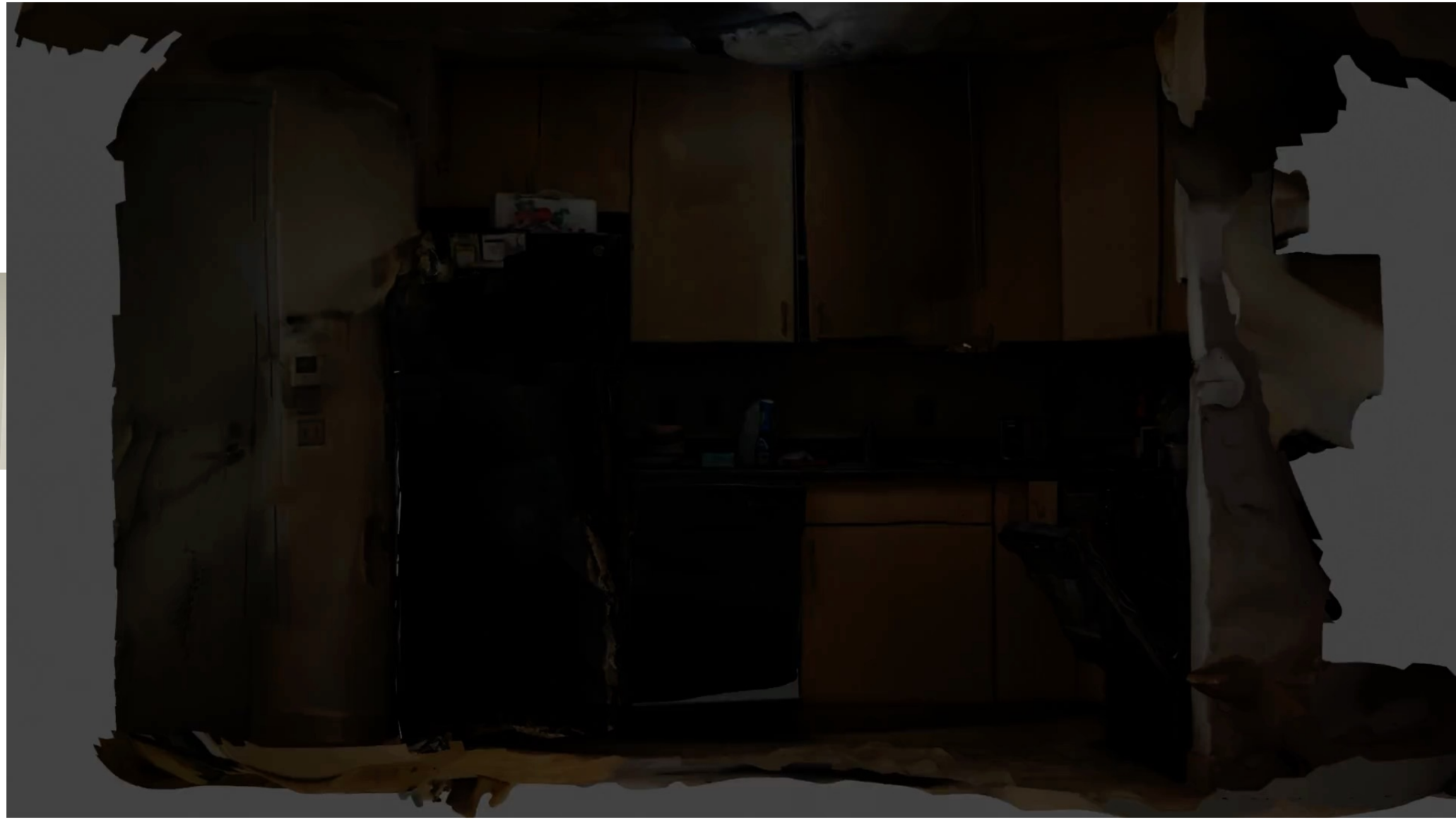
Scanning your kitchen with an iPhone…



Scanning process

… and interacting with it!

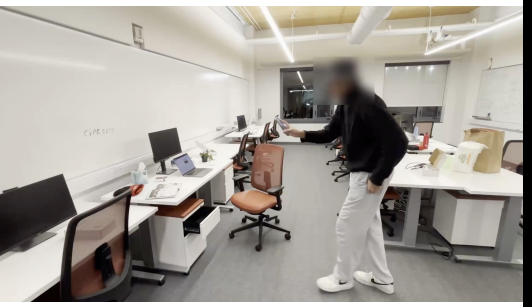
Scanning process



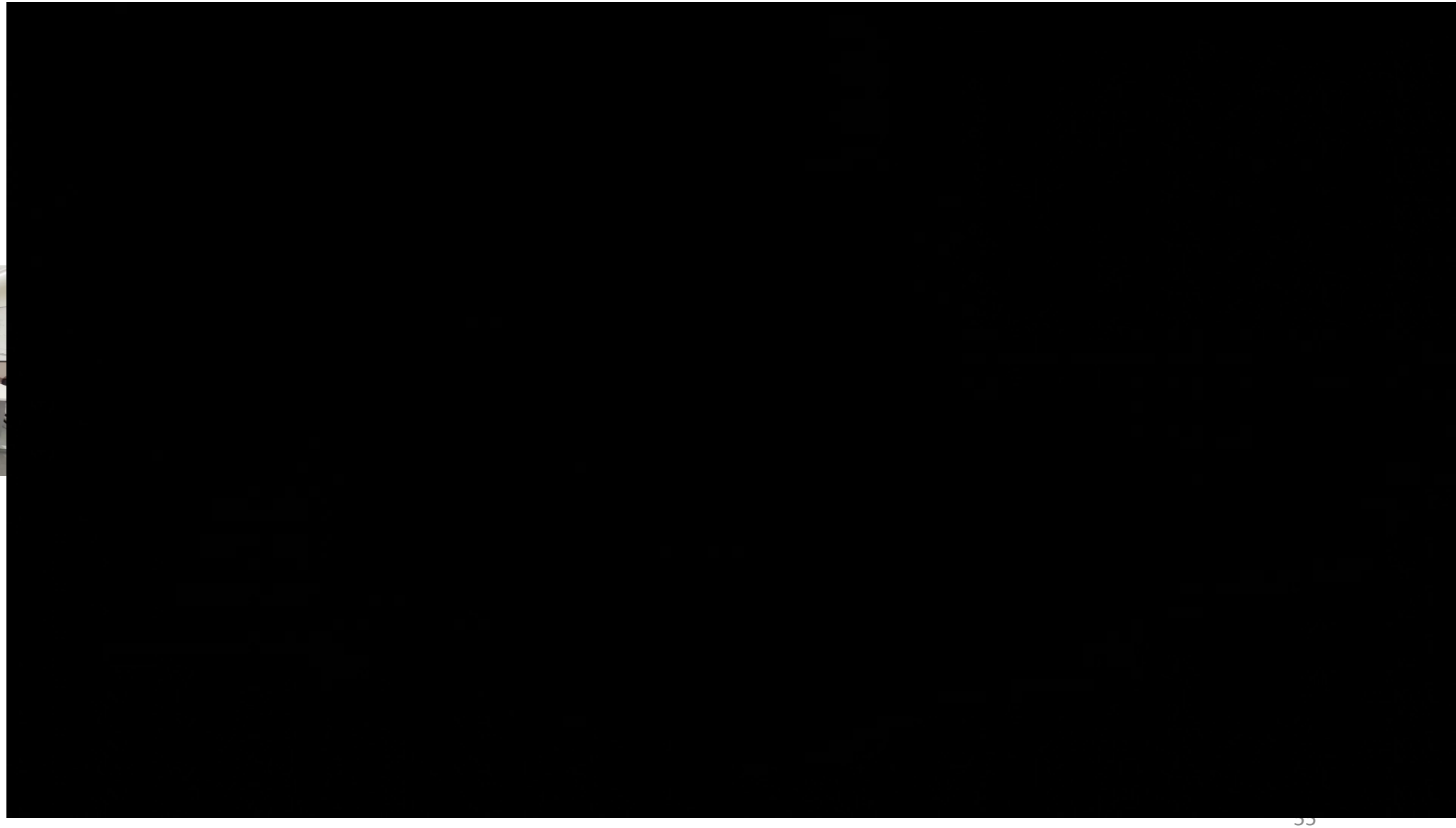
Scanning your office with an iPhone…

Scanning process

Scanning your office with an iPhone…



Scanning process

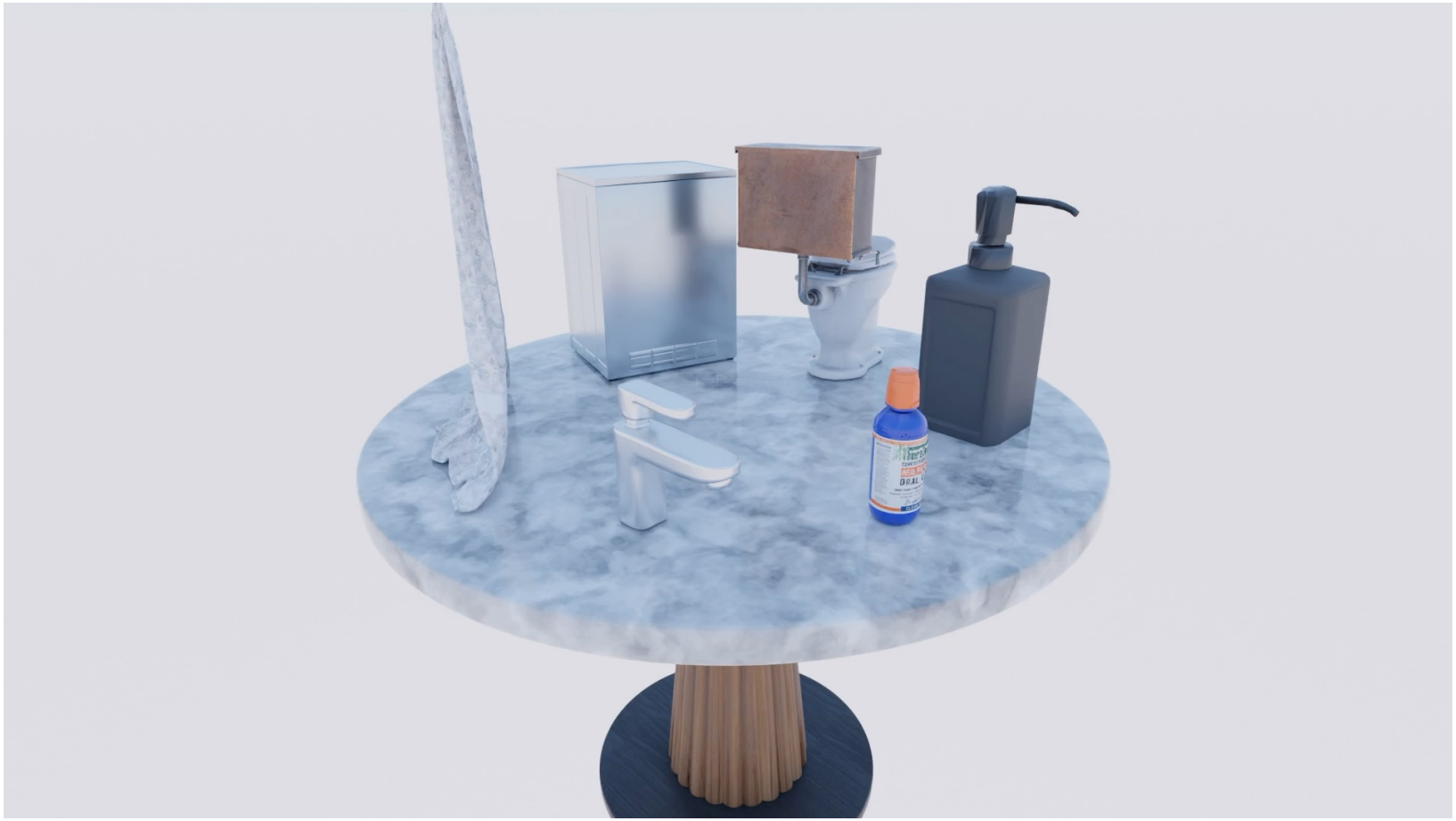


Applying to generated 3D shapes

Let's interact with objects in an office!

Input: Static 3D shapes and actions

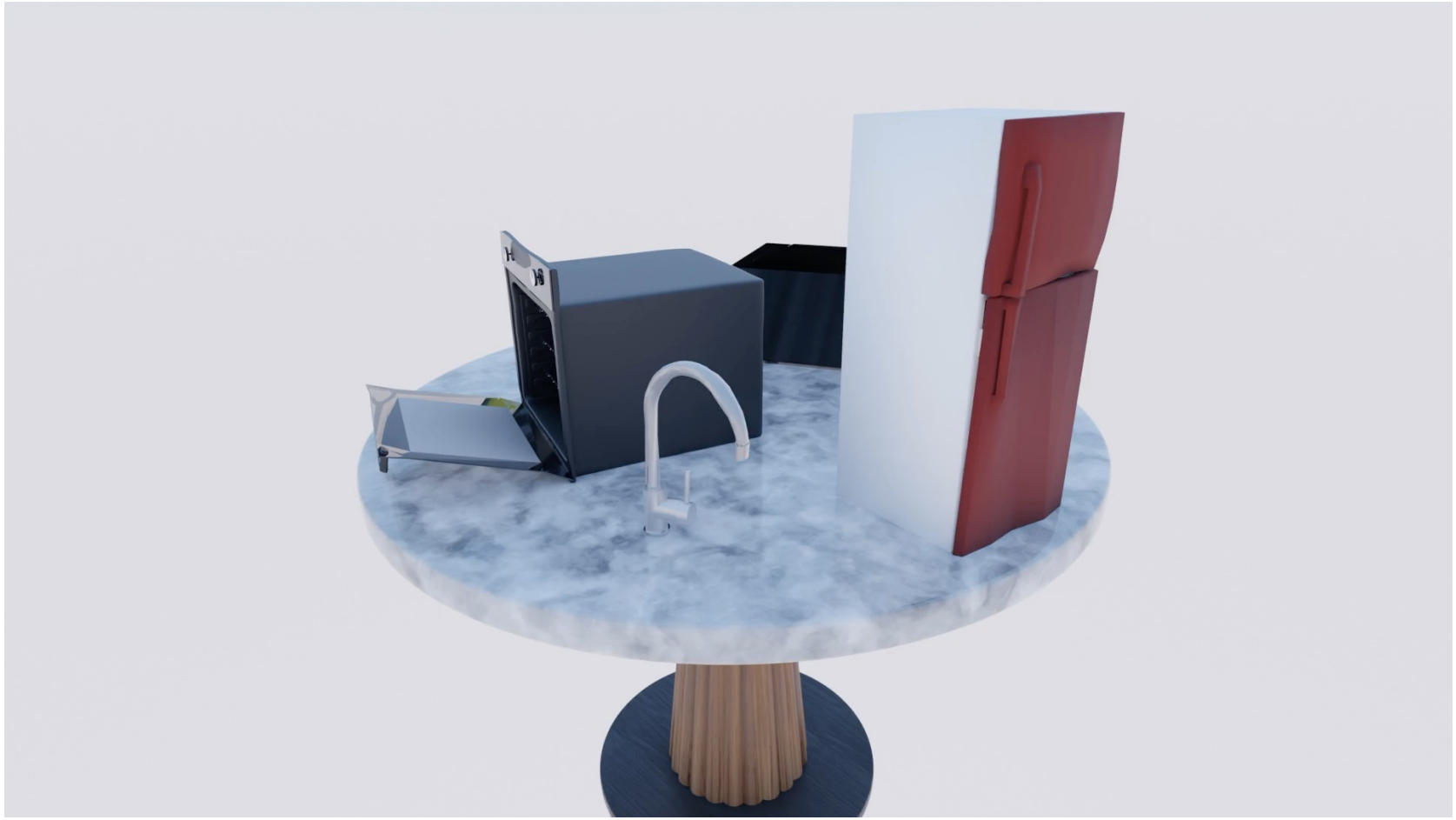
Let's interact with objects in a bathroom!



Input: Static 3D shapes and actions

Let's interact with objects in a bathroom!

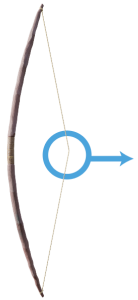
Let's interact with objects in a kitchen!



Input: Static 3D shapes and actions

Let’s interact with objects in a kitchen!

Comparison with Baselines



Input
"Bow"

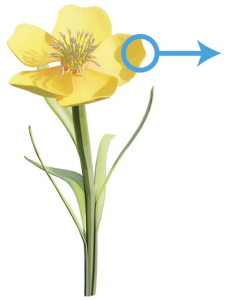
Ours

AnimateAnyMesh

PhysDreamer

Pixie

OmniphysGS



Input
"Flower"

Ours

AnimateAnyMesh

PhysDreamer

Pixie

OmniphysGS

Input
"Box"

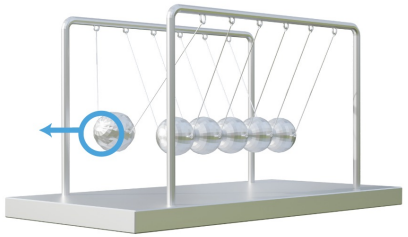
Ours

AnimateAnyMesh

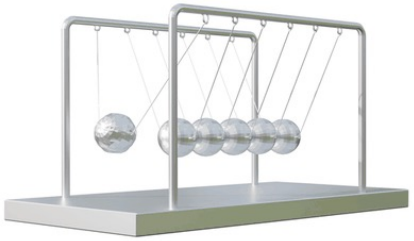
PhysDreamer

Pixie

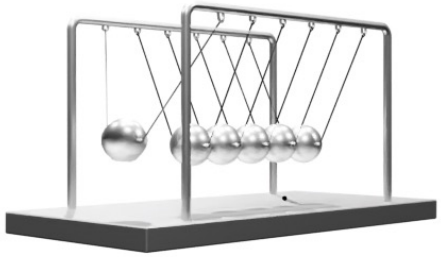
OmniphysGS



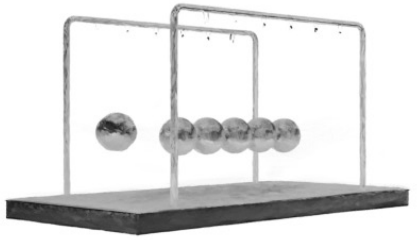
Input
"Newton"



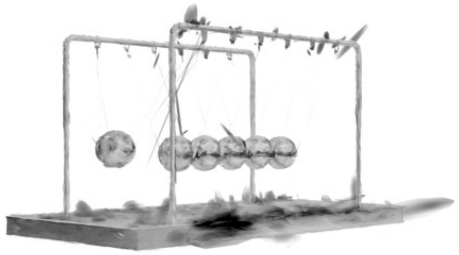
Ours



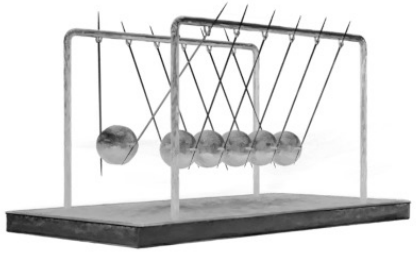
AnimateAnyMesh



PhysDreamer

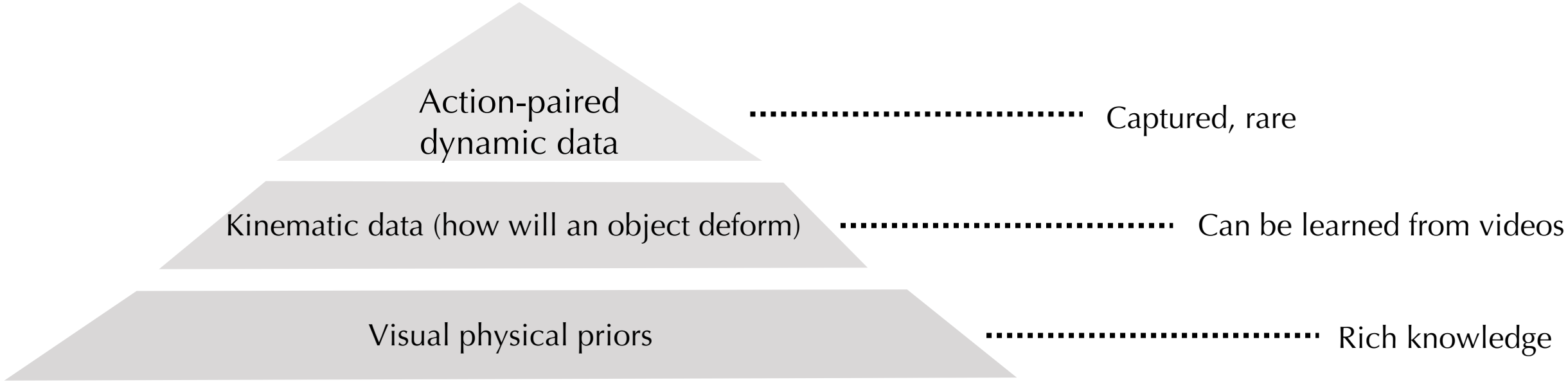


Pixie



OmniphysGS

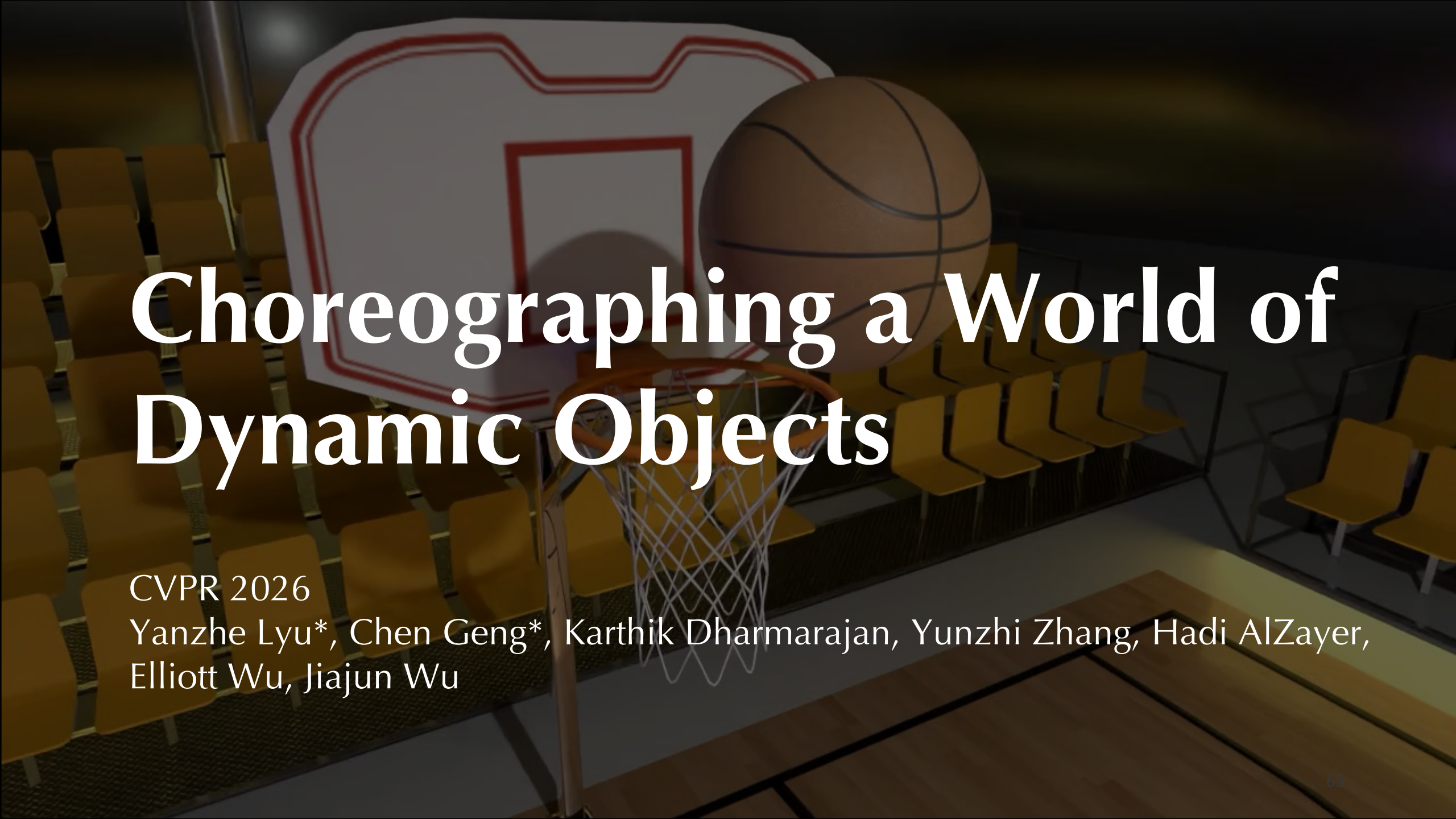
NeuROK tells us: inverse simulation can be framed as a generative modeling problem to be learned from data



But how can we get the 4D data needed to train NeuROK?

We introduce CHORD.

Lyu*, Geng*, et al., Choreographing a World of Dynamic Objects, CVPR 2026.



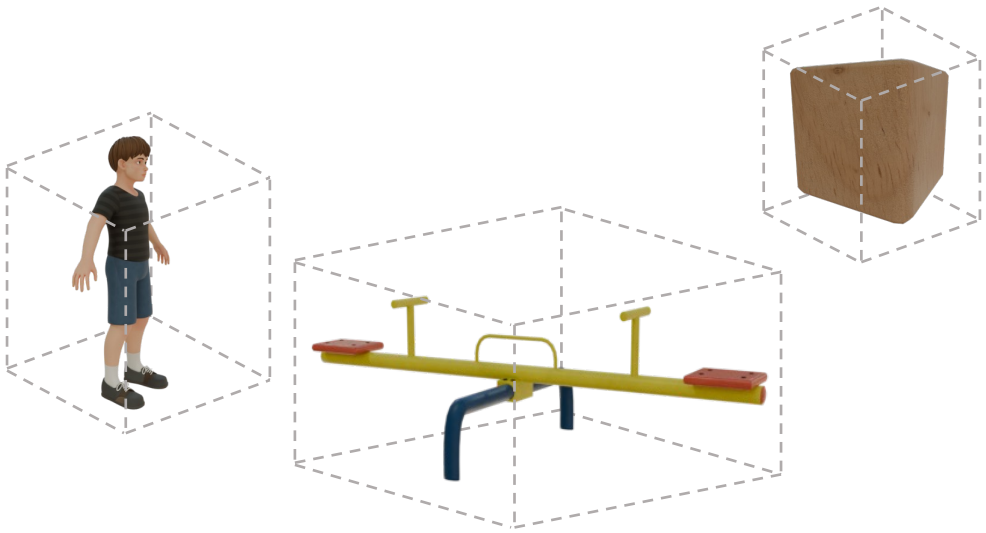
Choreographing a World of Dynamic Objects

CVPR 2026

Yanzhe Lyu*, Chen Geng*, Karthik Dharmarajan, Yunzhi Zhang, Hadi AlZayer, Elliott Wu, Jiajun Wu

Problem setting

- **Input:** 3D scene + text prompt



Static objects

Compose

Static 3D scene

Prompt:
"A child jumps on a seesaw, sending a brick flying."

Problem setting

- **Input:** 3D scene + text prompt
- **Output:** an animated 3D scene that follows the text prompt

Static 3D scene

Prompt:
"A child jumps on a seesaw, sending a brick flying."

Motion generation

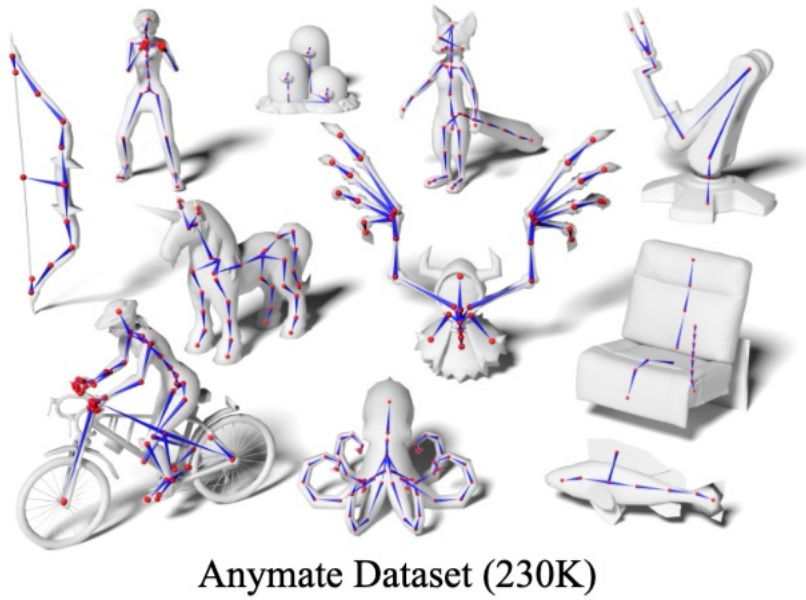
4D scene


Input meshes
Text prompt: “A robot arm picking up a brick.”


Input meshes
Text prompt: *“A basketball hits the rim and bounces away.”*


Input meshes
Text prompt: *“A cat jumps on a cushion.”*

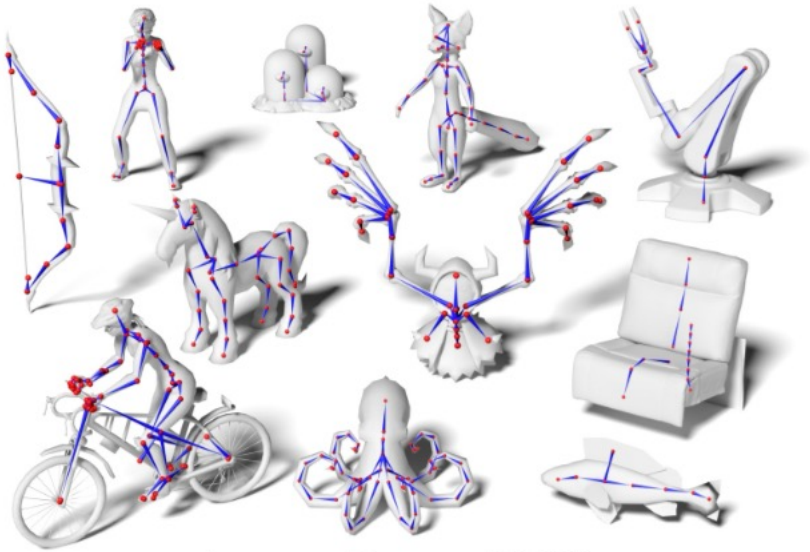
Generating these 4D scenes has been challenging due to the lack of data.



Anymate Dataset (230K)

Existing 4D datasets are small-scale and object-level.

Video datasets are much larger-scale and diverse...



Existing 4D datasets are small-scale and object-level.

Video datasets typically consist of millions of diverse videos.

… and were used to train video generative models.

Video datasets typically consist of millions of diverse videos.

Pretrain

Wan

State-of-the-Art video generative models.

Can we use them to generate 4D scene-level object motion?

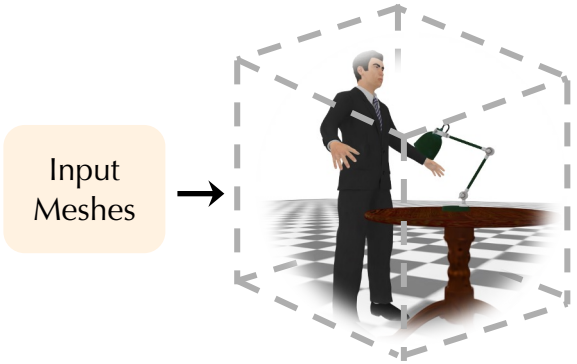
4D worlds of dynamic objects

Distill

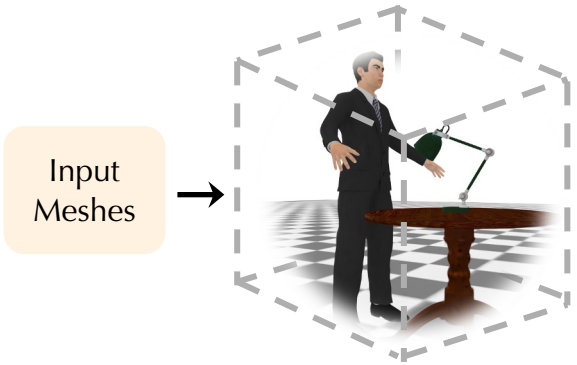
Wan

State-of-the-Art video generative models.

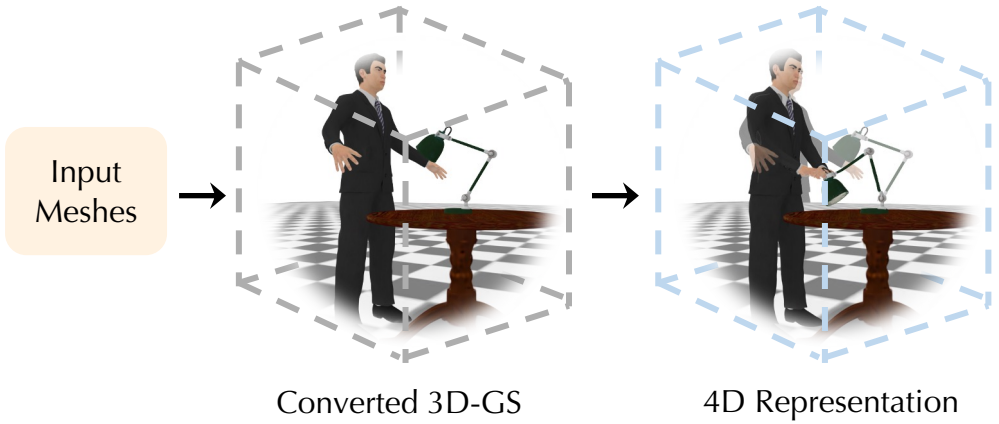
We propose a robust and versatile pipeline following this idea.



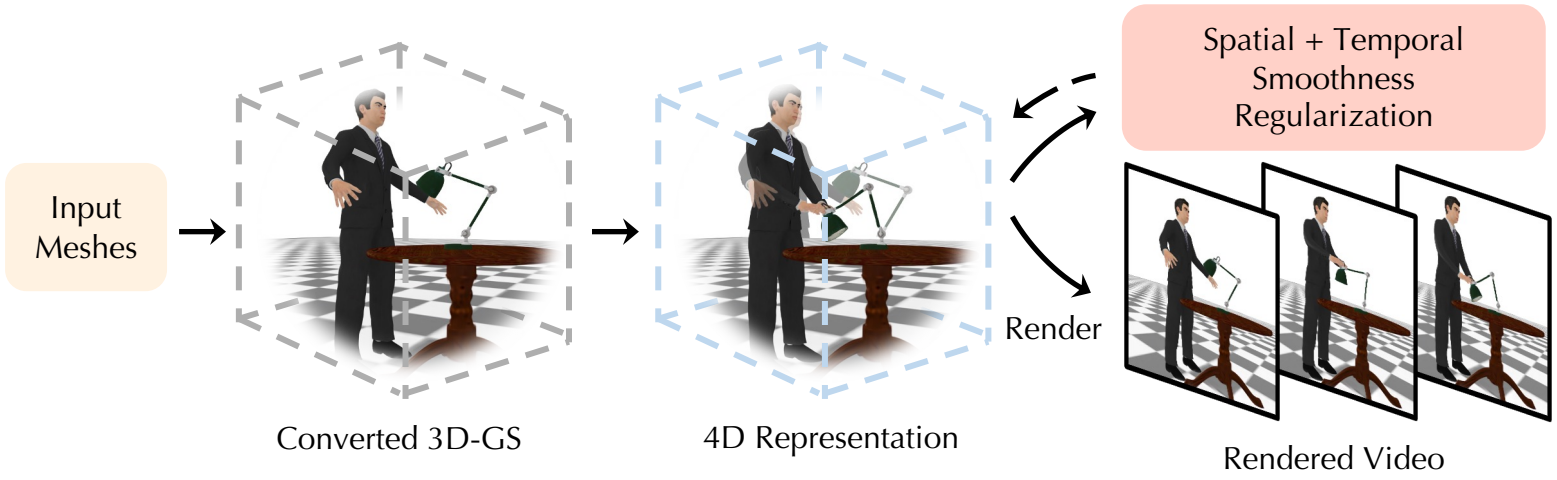
The input to our pipeline is static 3D objects in a scene.



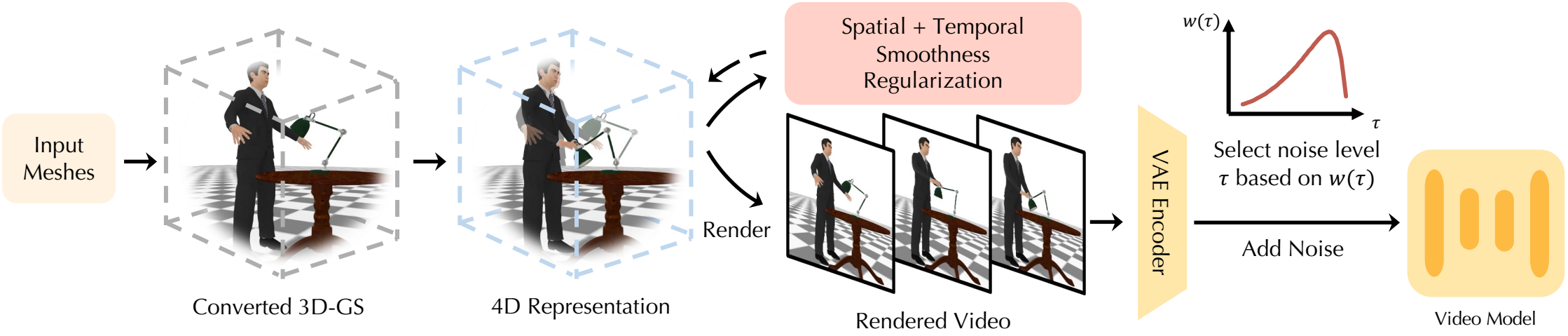
They are used to construct a hierarchical 4D representation.



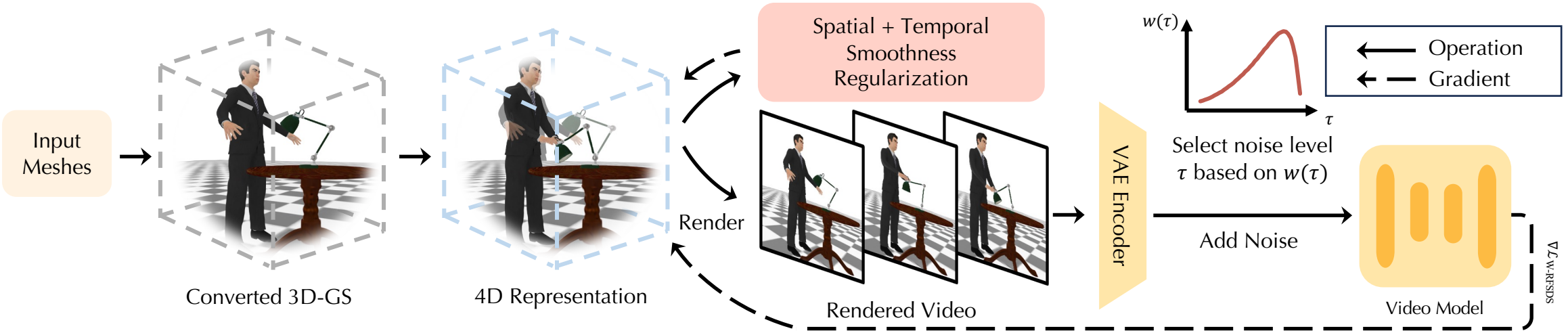
The 4D representation is rendered into a video.



We pass the rendered video to the video generative model with a noise-level selection strategy.



And compute update gradients based on our SDS target for flow-based models.

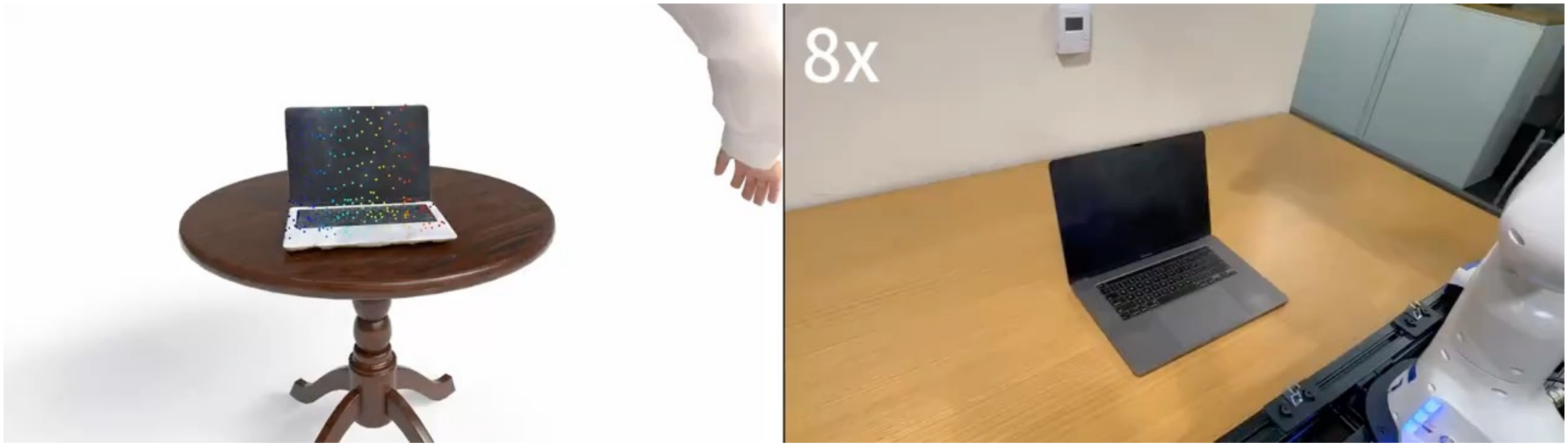


Transferring generated motions to real robots.

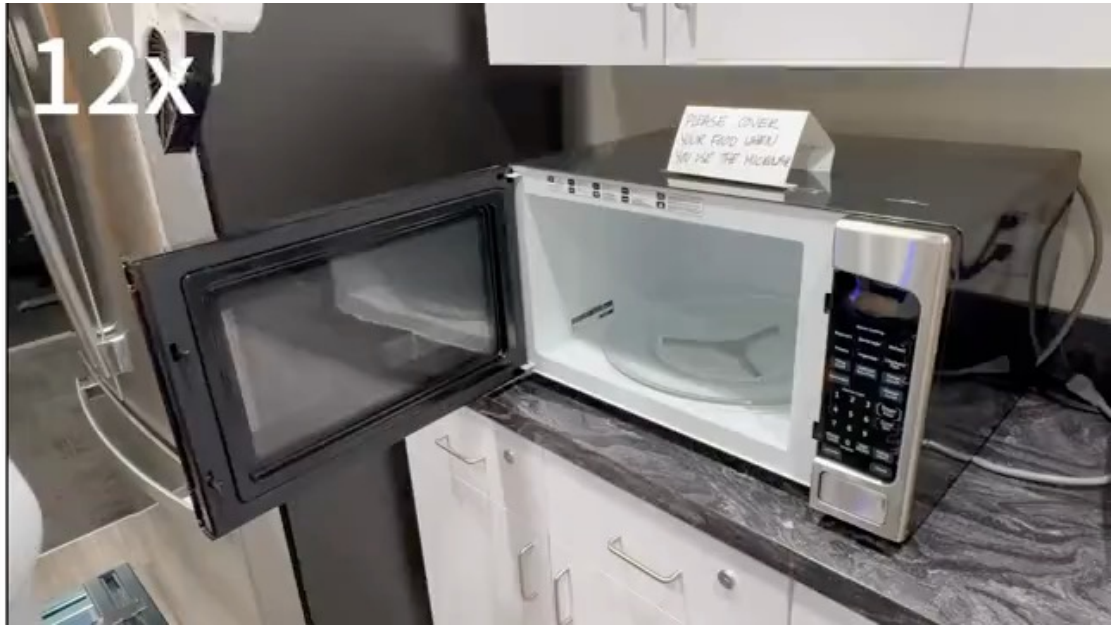
Transferring generated motions to real robots.

Transferring generated motions to real robots.

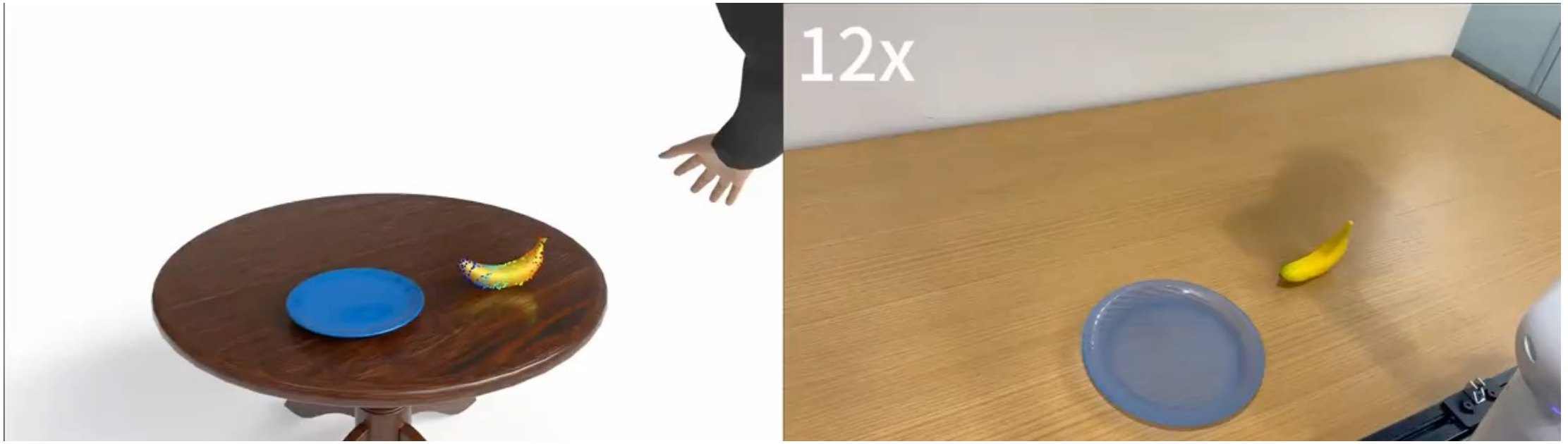
Transferring generated motions to real robots.



Transferring generated motions to real robots.



Transferring generated motions to real robots.



1. To build a scalable neural simulator, we should avoid category-specific priors.

2. By keeping Lagrangian mechanics fixed and learning generalized coordinates from cross-category data, inverse simulation becomes a generative modeling problem.

3. A scalable neural simulator should learn kinematic priors from abundant visual data and use physics to bridge them to interactive dynamics, reducing its dependence on scarce action-paired data.

4. Generalized coordinates can be learned from observational 4D data (NeuROK), and advances in video generation may make such data available at scale (CHORD).

Scalable Neural Simulation with Lagrangian Mechanics

Thanks / Q&A

Chen Geng
Stanford
08/11/2026