# LEARNING PHYSICS-GROUNDED 4D DYNAMICS WITH NEURAL GAUSSIAN FORCE FIELDS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Predicting physical dynamics from raw visual data remains a major challenge in AI. While recent video generation models have achieved impressive visual quality, they still cannot consistently generate physically plausible videos due to a lack of modeling of physical laws. Recent approaches combining 3D Gaussian splatting and physics engine can produce physically plausible videos, but are hindered by high computational costs in both reconstruction and simulation, and often lack robustness in complex real-world scenarios. To address these issues, we introduce Neural Gaussian Force Field (NGFF), an **end-to-end neural framework** that integrates 3D Gaussian perception with physics-based dynamic modeling to generate **interactive, physically realistic 4D videos** from multi-view RGB inputs, achieving two orders of magnitude faster than prior Gaussian simulators. To support training, we also present GSCollision, a 4D Gaussian dataset featuring diverse materials, multi-object interactions, and complex scenes, totaling over 640k rendered physical videos (∼4 TB). Evaluations on synthetic and real 3D scenarios show NGFF's strong generalization and robustness in physical reasoning, advancing video prediction towards physics-grounded world models.

## 1 INTRODUCTION

From a very young age, infants can understand basic physical principles about the world (Spelke & Kinzler, 2007). When growing older, humans develop a robust and intuitive understanding of the 3D physical world, enabling them to rapidly infer an object's geometry and physical properties from complicated visual input and predict its future dynamics using humans' "intuitive physics engine" (Battaglia et al., 2013; Pramod et al., 2025).

However, current AI systems fail to achieve such robust and generalizable physical prediction. Recently, video generation models have made significant progress in producing visually realistic dynamics, highlighting their potential to serve as "world simulators" by predicting plausible future events (Ho et al., 2022; Yang et al., 2024; Yu et al., 2025). However, they often lack basic physical commonsense such as object permanence, solidity, and gravity, despite being trained on millions of videos (Kang et al., 2025; Motamed et al., 2025). This limitation hinders the ability of AI agents to effectively interact with real-world physical environments.

Achieving human-level physical reasoning in AI has two fundamental challenges. (i) *Learning effective object representations from RGB inputs for simulation and rendering.* Although numerous studies have achieved significant progress in predicting physical dynamics, these efforts depend heavily on precise and structured object-centric data (Rubanova et al., 2024; Ma et al., 2023; Li et al., 2025) or implicit volumetric encoding with Neural Radiance Fields (NeRF) that are hard to ground in physics (Driess et al., 2023; Xue et al., 2023b). Particle-based methods (Whitney et al., 2024) alleviate this problem but require an additional pretrained renderer to produce multi-view consistent rendering results, which limit their scalability to complex real-world objects. (ii) *Learning generalizable physical dynamics and laws.* Large pretrained video generation models, such as video diffusion models, tend to overfit superficial visual features like surface texture, ambient shadowing, or spatial occlusions within the training data distribution (Kang et al., 2025). In Out-of-distribution (OOD) scenarios, these models tend to retrieve patterns from training cases in a case-based manner (Li et al., 2022; Kang et al., 2025), rather than acquiring a robust capacity for physical reasoning (Shiri et al., 2024). Recent methods using Gaussian splatting for physical dynamics prediction represent scenes

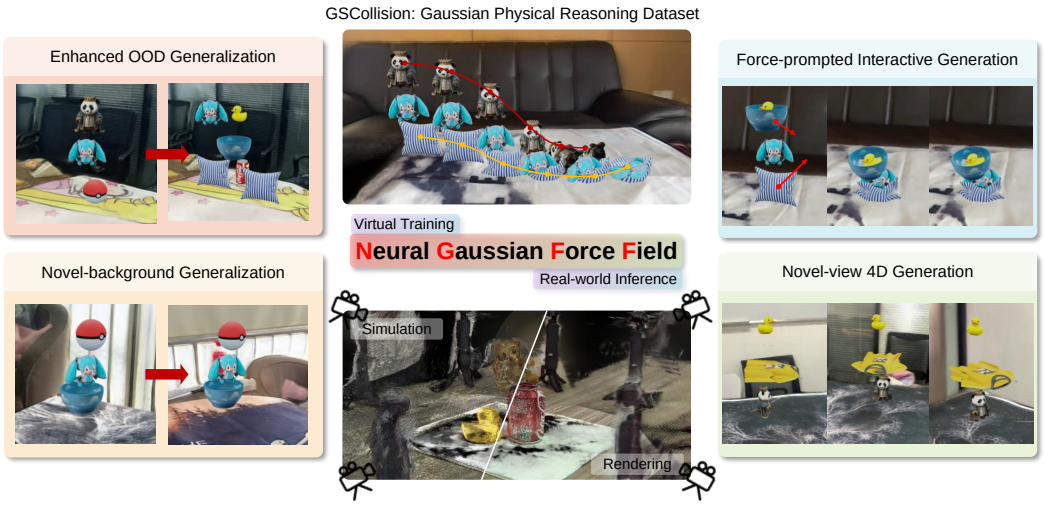


Figure 1: **Capabilities of NGFF.** NGFF is a physics-grounded video prediction framework that unifies perception and dynamics to model complex interactions and synthesize 4D videos. Built on Gaussian representations and force fields, it enables novel-view and novel-background synthesis as well as force-prompted interactive generation (Section 4.3). Moreover, NGFF achieves strong spatial and temporal generalization in dynamic prediction (Section 4.2) and can be effectively adapted to real-world scenarios (Section 4.4).

as sets of Gaussian points, capturing object properties like position and velocity (Xie et al., 2023; Lin et al., 2025b; Jiang et al., 2025a; Zhang et al., 2025). While effective for modeling simple interactions (Zhobro et al., 2025), these approaches face scalability issues and struggle with generalizing to complex environments with pre-defined simulators and intractable parameters (Xie et al., 2023).

To address these challenges, we propose the NGFF, a physics-grounded neural framework based on force modeling that learns generalized physical representations for multi-object interactions and generates interactive 4D scenes from pure multi-view RGB images. NGFF first encodes the input images into high-dimensional features and decodes them into a 3D scene of Gaussian points with object semantics through a feed-forward geometry transformer (Wang et al., 2025a;b). Then, a neural operator predicts object-centric force fields, which are integrated through an Ordinary Differential Equation (ODE) solver to simulate object movement and deformation. The predicted Gaussians are then rendered rapidly to produce multi-view videos that are congruent with physical reality. NGFF demonstrates three key capabilities, as illustrated in Figure 1. First, by reasoning force fields from object Gaussians and integrating them through ODE solvers, NGFF predicts physically grounded dynamics and exhibit **enhanced OOD generalization** across complex interactions. Second, it supports force-prompted interactive generation through the learned force field. Third, the object-aware 3D Gaussian representation enables efficient geometric reconstruction and **multi-view background-agnostic video generation**. Fourth, the neural field representations enable robust predictions when **transferred to real-world scenarios**.

To train and validate our framework, we construct GSCollision, a novel benchmark for 4D Gaussian-based physical reasoning with diverse rigid and soft body physics and rich visual complexities, covering a broad spectrum of physical phenomena such as falling, multi-body collision, rotation, sliding, and containment. To ground our simulations in realism, we utilize real-world scenes from the WildRGBD (Xia et al., 2024) dataset as backgrounds. We evaluate our method in three settings: **dynamic prediction, video generation, and real-world prediction**. The experimental results demonstrate that NGFF not only generates high-quality predictive videos but also achieves physically plausible simulations in unseen scenarios and supports rapid transfer to the real world, surpassing state-of-the-art (SOTA) particle-based dynamic prediction models such as Pointformer (Wu et al., 2024b), and video generation models such as Veo3 (DeepMind, 2025), NVIDIA Cosmos (NVIDIA et al., 2025), and PhysGen3D (Chen et al., 2025). Our work successfully bridges the gap in recent Gaussian-based simulation methods (Zhang et al., 2025; Jiang et al., 2025a; Zhobro et al., 2025) by simultaneously capturing both the high visual complexity of scenes and the complex physical interactions among multiple objects.

## 2 Related work

Physical reasoning and visual dynamics prediction are fundamental challenges in developing AI systems with human-like intuitive physics. Benchmarks based on the Violation-of-Expectation (VoE) paradigm and interactive environments have been proposed to test agents' ability to capture spatial structures and fundamental physical laws (Piloto et al., 2022; Dai et al., 2023; Bakhtin et al., 2019; Allen et al., 2020; Bear et al., 2021; Li et al., 2023a). Building on these benchmarks, visual dynamics prediction methods attempt to model the underlying scene dynamics. Neural simulator approaches typically rely on Graph Neural Networks (GNN)-based architectures with mesh, SDF, spring-mass, or particle-grid representations (Sanchez-Gonzalez et al., 2020; Bear et al., 2021; Allen et al., 2023; Rubanova et al., 2024; Jiang et al., 2025a; Zhang et al., 2025), which succeed in simulating various materials but often struggle to generalize to complex, out-of-distribution interactions. In parallel, differentiable physics simulators such as MPM-based Gaussian formulations (Xie et al., 2023; Lin et al., 2025b; Chen et al., 2025) achieve high physical fidelity, but they suffer from high computational cost and slow simulation speed, limiting their scalability. To support both simulation and rendering, scene representations have evolved from point clouds and NeRFs (Shi et al., 2024; Whitney et al., 2024; Driess et al., 2023; Xue et al., 2023a; Li et al., 2023b) to 3D Gaussian Splatting (Kerbl et al., 2023), which provides photorealistic quality and real-time performance. Recent advances accelerate 3D reconstruction via feed-forward prediction (Wang et al., 2025a; Jiang et al., 2025b; Wang et al., 2025b; Zhobro et al., 2025). Our work builds upon this trajectory by unifying feedforward Gaussian-based scene representations and neural dynamics modeling to enable generalizable physical reasoning across multi-object interaction in real-world scenarios. Detailed related work about physical reasoning, visual dynamics prediction, and scene representations for simulation and rendering can be found in Appendix D.

## 3 Method

In this section, we define our task as a 4D video prediction problem and introduce a two-stage approach. The first stage focuses on feed-forward object-level 3D Gaussian reconstruction, while the second stage employs an ODE-based neural dynamic simulator to predict 4D physics-grounded videos (Figure 2). We then show that our framework can scale effectively with large-scale world data and quickly adapt to real-world simulation scenarios.

### 3.1 Problem Formulation

Given a set of $N$ unposed RGB images of a static scene $\mathcal{I}_0 = \{I_0(p_k) \in \mathbb{R}^{H\times W\times 3} \mid k = 1, \dots, N\}$, captured from different views $p_k \in \mathbb{R}^6$, our goal is to model the temporal evolution of the scene's geometry, appearance, and physical dynamics, synthesizing future views $I_t(p)$ at arbitrary time steps $t \in T$ and camera poses $p \in \mathbb{R}^6$.

We represent the initial scene using $M$ 3D Gaussians $\mathcal{G}_0 = \{g_{0,i}\}_{i=1}^{M}$, extracted from the initial observations $\mathcal{I}_0$. A neural dynamics model $f_\theta$ is trained to predict the Gaussian state at the next time step $\mathcal{G}_{t+1} = f_\theta(\mathcal{G}_t)$. To render an image from any viewpoint at time $t$, we use Gaussian splatting $\hat{I}_{t,k} = \text{Render}(\mathcal{G}_t, p_k)$.

The overall learning objective jointly optimizes the dynamic model parameters $\theta$ and the initial scene representation $\mathcal{G}_0$:

$$\min_{\theta,\, \mathcal{G}_0} \sum_{t,k} \mathcal{L}(\hat{I}_{t,k}, I_{t,k}) + \mathcal{L}'(\hat{\mathcal{G}}_t, \mathcal{G}_t). \tag{1}$$

where $\mathcal{L}$ is the image reconstruction loss and $\mathcal{L}'$ regularizes the predicted Gaussians.

### 3.2 Neural Gaussian Force Field (NGFF) framework

#### 3.2.1 Feed-forward 3D reconstruction

**Geometric and appearance reconstruction.** Following VGGT (Wang et al., 2025a) and $\pi^3$ (Wang et al., 2025b), we adopt a pretrained transformer-based geometry encoder and three separate decoder heads for predicting camera poses $p$, Gaussian centers $\mu$ and Gaussian attributes $(\alpha, \mathbf{r}, \mathbf{s}, \mathbf{c})$.

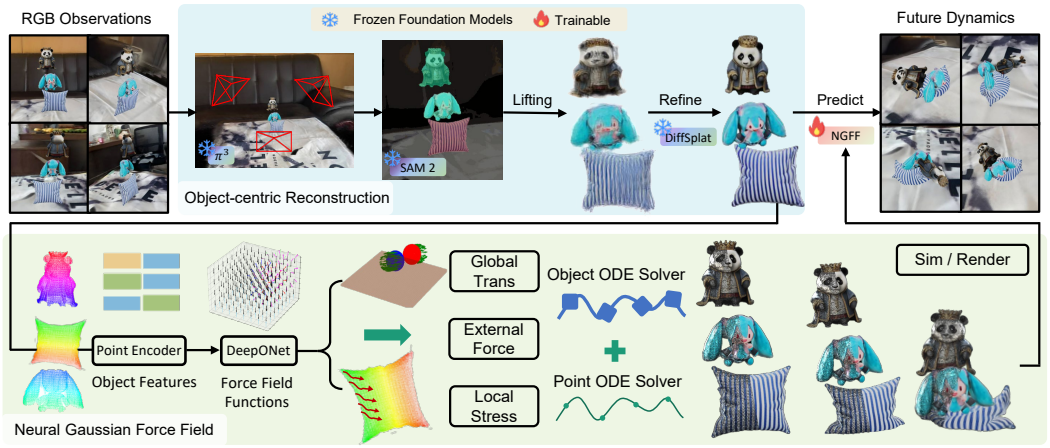


Figure 2: **Overall framework of NGFF.** Starting with unposed RGB inputs, a feed-forward model first reconstructs the scene as a set of 3D Gaussians. These Gaussians are then segmented into distinct objects and refined to mitigate noise and resolve occlusions. Subsequently, a point encoder transforms the Gaussians into feature representations. A DeepONet estimates a physics-grounded Gaussian force field, which is integrated with ODE solvers to predict the next-step dynamics. The Gaussians are predicted and rendered iteratively.

Each input image is first patchified into token sequences $t_k$ using DINOv2 (Oquab et al., 2023). The combined tokens $\{t_k\}_{k=1}^{N}$ from all $N$ views are processed by a $L$-layer Alternating-Attention Transformer to capture global geometric features. For the camera and point head, the feature tokens are then decoded by a pixel-shuffling decoder (Shi et al., 2016). For the splatter head, we simply incorporate a convolutional upsampling layer followed by a direct RGB shortcut (Ye et al., 2025) from the input image to preserve high-frequency details and enhance apperance reconstruction.

**Object-centric reconstruction.** To obtain the object-centric representations essential for physics simulation, we first employ a pretrained video segmentation model SAM2 (Ravi et al., 2024), to generate pixel-wise instance masks from the input images. These masks are then back-projected onto the Gaussian points via a voting scheme, partitioning the set of Gaussians $\mathcal{G}$ into $K$ instance groups, denoted as $\mathcal{G}_k = \{g \in \mathcal{G} \mid \text{label}(g) = k\}$. To address occlusions and invisible parts from inputs, we further adopt a refining module leveraging a 3D asset generation model DiffSplat (Lin et al., 2025a) and $\mathrm{Sim}(3)$ pose estimation to enhance the topological quality of the Gaussian object given single-view images. Details can be found in Appendix B.2

### 3.2.2 ODE-BASED NEURAL DYNAMICS SIMULATOR

We adopt an explicit **force field** representation to model the object's transformation, rotations, and local deformation under physical interactions and external forces. By learning physics-grounded, generalizable representations through explicit force field modeling, NGFF can model physical interactions between various *rigid and soft bodies* in a unified way and achieve *few-shot generalization*.

**Force field prediction.** The core of our Neural Force Field (NFF) framework is grounded in the physical principle of a *force field*—a vector field $\mathbf{F}(\cdot)$ that determines the force $\mathbf{F}(\mathbf{z}^q(t))$ acting on a query object $q$ based on its state $\mathbf{z}^q(t)$ at time $t$. We assume that the local point cloud is sufficiently expressive to encode physical properties for simulation. The state vector $\mathbf{z}^q(t) = \{\mathbf{h}^q, \mathbf{s}^q(t), \dot{\mathbf{s}}^q(t)\}$ encapsulates both geometric and dynamic attributes of the object, including: (1) **Semantic features** $\mathbf{h}^q$: object-level features encoded from current Gaussian centers using PointNet (Qi et al., 2017), (2) **Zeroth-order states** $\mathbf{s}^q(t)$: local point cloud $\mathbf{x}(t) \in \mathbb{R}^{M\times 3}$, center of mass $\mathbf{c}(t) \in \mathbb{R}^3$, and orientation (Euler angles) $\boldsymbol{\theta}(t) \in \mathbb{R}^3$, and (3) **First-order states** $\dot{\mathbf{s}}^q(t)$: local point cloud velocity $\dot{\mathbf{x}}(t) \in \mathbb{R}^{M\times 3}$, velocity of center of mass $\dot{\mathbf{c}}(t) \in \mathbb{R}^3$, and angular velocity $\dot{\boldsymbol{\theta}}(t) \in \mathbb{R}^3$.

We adopt a unified approach to model both *rigid* and *soft body* dynamics. For a scene with $K$ interacting objects, we represent the **global transformation force field** $\mathbf{F}^{\text{global}}(\cdot) \in \mathbb{R}^6$—encompassing both translational and rotational components—using a neural operator over a relational graph $\mathcal{N} = (V, E)$, where $V = \{\mathbf{z}^0(t), \ldots, \mathbf{z}^{K-1}(t)\}$ denotes object nodes and $E$ encodes physical contacts. Inspired by relational inductive biases in graph-based models (Battaglia et al., 2018) and

neural operator learning (Lu et al., 2021), we define the force field as:

$$\mathbf{F}^{\text{global}}(\mathbf{z}^q(t)) = \sum_{i \in \mathcal{N}(q)} \mathbf{W}\left(f_\eta(\mathbf{z}^i(t)) \odot f_\phi(\mathbf{z}^q(t))\right) + \mathbf{b}, \tag{2}$$

where $\mathcal{N}(q)$ denotes the set of neighboring objects in contact with $q$, $f_\eta$ and $f_\phi$ are neural encoders with learnable parameters, and $\odot$ is the element-wise product. The projection matrix $\mathbf{W} \in \mathbb{R}^{d_{\text{hidden}} \times d_{\text{force}}}$ and bias $\mathbf{b} \in \mathbb{R}^{d_{\text{force}}}$ map the hidden features to force vectors. This formulation captures a variety of interactions such as contact, sliding, and gravity.

To model local deformations in **soft bodies**, we introduce a neural network $\Phi$ that predicts the pointwise **local stress field** $\mathbf{F}^{\text{local}}(\cdot) \in \mathbb{R}^{M \times 3}$ based on a **Contact Area Mask** (CAM) that highlights the contact regions:

$$\mathbf{F}^{\text{local}}(\mathbf{z}^q(t)) = \Phi\left(\mathbf{F}^{\text{global}}(\mathbf{z}^q(t)), \text{CAM}, \mathbf{x}^q(t), \dot{\mathbf{x}}^q(t)\right). \tag{3}$$

The final force field unifies rigid and soft-body predictions as:

$$\mathbf{F}(\mathbf{z}^q(t)) = \left(\mathbf{F}^{\text{local}}, \mathbf{F}^{\text{global}}\right). \tag{4}$$

**Trajectory decoding via ODE integration.** To recover continuous and physically plausible trajectories, we integrate the learned force field using a second-order ordinary differential equation (ODE) solver. The trajectory of an object is computed as:

$$\mathbf{z}^q(t) = \text{ODESolve}\left(\mathbf{z}^q(0), \mathbf{F}, 0, t\right), \tag{5}$$

$$\mathbf{s}(t) = \mathbf{s}(0) + \int_0^t \dot{\mathbf{s}}(t)\, dt, \quad \dot{\mathbf{s}}(t) = \dot{\mathbf{s}}(0) + \int_0^t \mathbf{F}(\mathbf{z}^q(t))\, dt. \tag{6}$$

This formulation bridges learned neural force fields and physical dynamics simulation in a fully differentiable manner.

#### 3.2.3 Training

The feed-forward reconstruction module is initialized with pretrained $\pi^3$ parameters, where the feature encoder, point head, and camera head are frozen, and the splatter head is trained on WildRGBD using RGB and geometric consistency losses to align predicted and rendered depth maps. The neural dynamics simulator is trained on synthetic data with MSE loss to match Gaussian configurations and motion trajectories to MPM simulations.

### 3.3 Capabilities of NGFF

**Dynamic prediction as operator learning of force fields** NGFF formulates dynamics prediction as operator learning over explicit force fields, unifying rigid and deformable objects in a shared state space. Neural operators on relational graphs capture contact, collision, and deformation, enabling scalability to multi-body systems and generalization across spatial, temporal, and compositional shifts.

**Video generation as efficient rendering of physical trajectories** By combining feed-forward 3D Gaussian reconstruction with learned force fields, NGFF links perception and simulation. Differentiable Gaussian splatting renders trajectories that are both photorealistic and physically consistent, supporting viewpoint transfer, contextual variation, and interactive interventions.

**Real-world transfer** Gaussians provide a disentangled interface between noisy visual inputs and underlying dynamics, while force-field neural operators capture physics through robust representations. Together, these components facilitate sim-to-real transfer by adapting dynamics to real-world RGB data while maintaining physical consistency.

## 4 Experiments

### 4.1 Dataset

Building on prior physical reasoning benchmarks (Bakhtin et al., 2019; Bear et al., 2021; Greff et al., 2022; Li et al., 2023a), we introduce GSCollision, a 3D Gaussian-splats physical reasoning

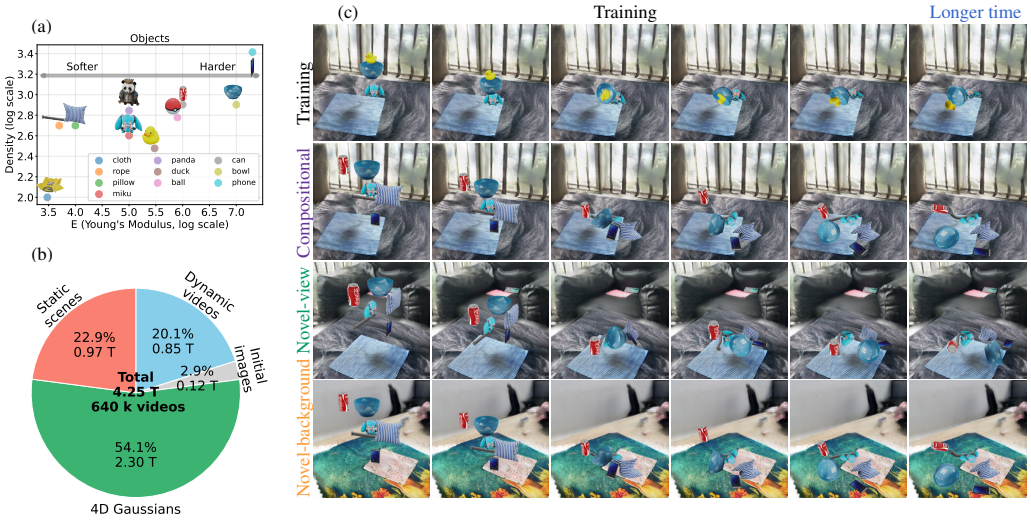


Figure 3: **GSCollision dataset.** (a) Distribution of 10 representative objects across density and material hardness, measured by Young's modulus on a log scale. Softer items (e.g., cloth, rope, pillow) occupy the lower-left region, while harder and denser objects (e.g., bowl, phone) lie on the upper-right. (b) Composition of the dataset, totaling 4.25 TB with 3200 scenes and 640 k videos. The pie chart illustrates the storage distribution across different simulation splits (train and test), with additional components such as multi-view images of initial scenes and other files. (c) Gallery of representative frames across training, longer time, compositional, novel-view, and novel-background splits.

dataset constructed with MPM simulators (Xie et al., 2023) that capture both rigid and deformable bodies. By adopting a Gaussian-based representation, GSCollision naturally bridges perception and reasoning: visual observations can be directly grounded in physically consistent predictions, while the predicted states can be rendered into videos with fast, differentiable pipelines.

GSCollision comprises 10 everyday objects with diverse material properties and densities, ranging from soft items like pillows and ropes to rigid objects such as balls and phones. Randomly sampling object compositions and positions within a 3D box yields 3,200 physical scenarios involving both object–object and object–ground interactions. Among these, 2,700 three-object scenes are used for training, while 500 are reserved for testing. The test set introduces distributional shifts, including 300 unseen three-object scenes, 100 four-object scenes, and 100 six-object scenes, thereby providing both compositional and scaling challenges.

The dataset spans a wide range of physical dynamics—falling, collisions, rotation, sliding, and containment—across scenarios such as stacked towers, container-based setups, and collision-driven interactions. Each sequence consists of 100 simulation steps, corresponding to approximately two seconds of real time. A statistical overview of the dataset is provided in Figure 3.

### 4.2 Dynamic Prediction

We define four splits of generalization to validate NGFF, including positional generalization, temporal generalization, compositional generalization, and external force generalization. (1) **Spatial generalization** tests whether the model can predict dynamics at unseen object positions, requiring accurate spatial extrapolation or interpolation of forces. (2) **Temporal generalization** assesses the ability to sustain stable, accurate predictions over longer rollouts than seen in training. (3) **Compositional generalization** (Lake & Baroni, 2023) evaluates performance on novel object combinations and larger numbers of interacting objects (4–6), requiring reasoning about unseen multi-body dynamics.

We benchmark NGFF against several SOTA particle-based prediction methods, including GCN (Kipf & Welling, 2017), and Pointformer (Wu et al., 2024b), as well as Material Point Method (MPM)-based simulators parameterized with estimates from Vision Language Models (VLM) (Chen et al., 2025). Baseline definitions are provided in Appendix C.1, and evaluation metrics are detailed in Appendix C.2. As shown in Table 1, NGFF consistently outperforms the baselines across all generalization splits, delivering significant improvements in modeling long-term and multi-object

Table 1: **Performance comparison of dynamic prediction across generalization splits.** Evaluation metrics consist of the Root Mean Squared Error (RMSE) between predicted and ground-truth trajectories, the Final Position Error (FPE), the correlation coefficient R, and the average inference time over 100 steps. Arrows indicate whether higher (↑) or lower (↓) values are better. NGFF w/o deform. indicates NGFF without modeling of soft body deformation.

| Model | Spatial | | | Temporal | | | Compositional | | | Time (s) ↓ |
|---|---|---|---|---|---|---|---|---|---|---|
| | RMSE ↓ | FPE ↓ | R ↑ | RMSE ↓ | FPE ↓ | R ↑ | RMSE ↓ | FPE ↓ | R ↑ | |
| VLM-MPM | 0.306 | 0.774 | 0.299 | 0.328 | 0.901 | 0.300 | 0.358 | 0.904 | 0.305 | 39.29 |
| GCN | 0.134 | 0.479 | 0.406 | 0.174 | 0.590 | 0.400 | 0.145 | 0.509 | 0.347 | 0.346 |
| Pointformer | 0.096 | 0.394 | 0.623 | 0.129 | 0.537 | 0.604 | 0.162 | 0.594 | 0.434 | **0.183** |
| NGFF w/o deform. | 0.110 | 0.459 | 0.595 | 0.144 | 0.600 | 0.578 | 0.131 | 0.546 | 0.515 | 0.303 |
| **Our NGFF** | **0.082** | **0.326** | **0.661** | **0.107** | **0.419** | **0.652** | **0.104** | **0.409** | **0.571** | 0.363 |

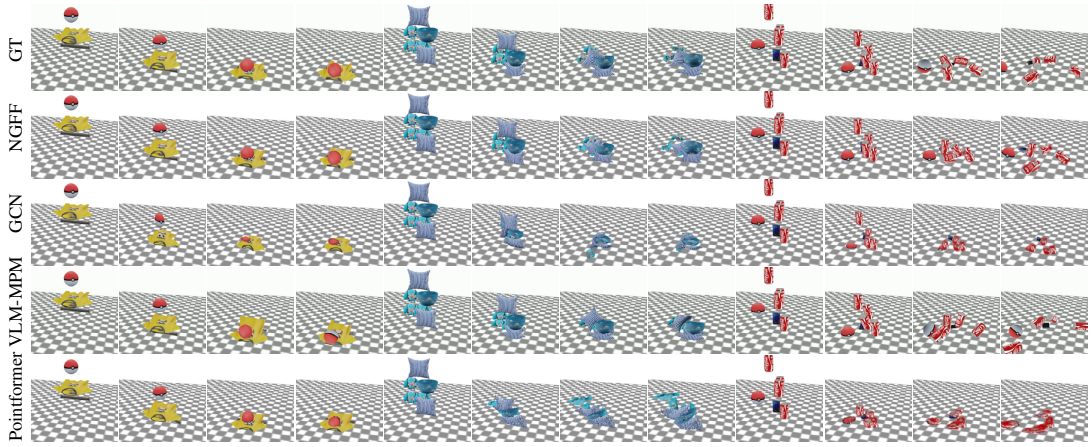


Figure 4: **Qualitative results of dynamic prediction.** NGFF more accurately matches ground-truth trajectories than graph-, transformer-, and MPM-based baselines when predicting unseen rigid–soft body interactions. See more visualization in Appendix F.1.

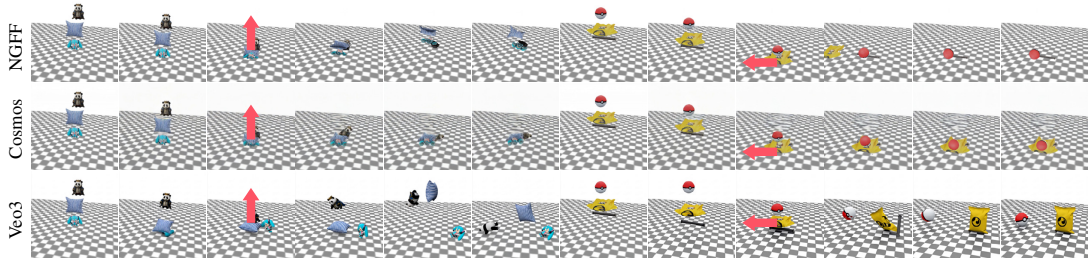


Figure 5: **Qualitative results of interactive generation.** Red arrows indicate external forces. The left scene illustrates how lifting the pillow influences its dynamics, while the right scene shows that pulling the cloth leftward causes the ball to rotate and slide. For Cosmos and Veo3, we use the following prompts, respectively: "modify the pillow in the video after it falls to the ground between 3.2 and 4s to show a significant, sudden external force stretching it upward into the air, with interactions with panda and miku", and "modify the clothing in the video between 3.2s and 4s to show a significant, sudden external force stretching it leftward".

dynamics. Notably, its inference speed is approximately two orders of magnitude faster than MPM-based approaches.

### 4.3 Video Generation

When evaluating from the perspective of video generation, we consider compositional, novel-background, novel-view, and interactive generation as they span the key dimensions of generalization required for robust video prediction. (1) **Compositional generation** tests generalization to novel object arrangements, including unseen positions and up to six objects not present during training. (2) **Novel-view generation** evaluates consistency and realism from unseen viewpoints, ensuring that models disentangle dynamics from appearance while maintaining coherent spatiotemporal representations. (3) **Novel-background genera-**

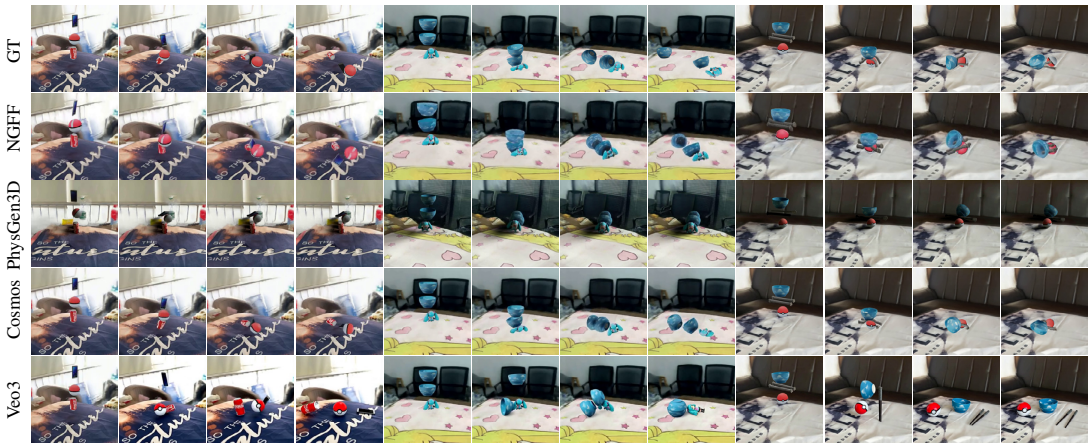


Figure 6: **Qualitative results of video generation.** The comparisons illustrate that NGFF better preserves coherent object shapes, physically plausible interactions, and consistent backgrounds in the generated videos, whereas other video generation models often produce distortions, unstable dynamics, or inconsistent scenes.

**tion** assesses robustness to previously unseen backgrounds, requiring models to preserve object dynamics and physical plausibility while adapting to new visual contexts. (4) **Interactive generation** probes adaptability under external perturbations (e.g., random forces), testing whether models capture causal physical interactions rather than merely memorizing trajectories.

We evaluate the generated videos with both VLMs and human annotators, focusing on physical realism and visual quality, measured by Physical Realism (PhysR) and Photo Realism (PhotoR), respectively. The evaluation metrics are detailed in Appendix C.2. As shown in Figure 6 and Table 2, NGFF learns generalizable physical representations from complex observations, surpassing prior SOTA video generation methods—including diffusion-based models such as NVIDIA Cosmos (NVIDIA et al., 2025) and Google Veo3 (DeepMind, 2025), as well as the physics-engine-based PhysGen3D (Chen et al., 2025)—in terms of physical accuracy in unseen scenarios, while achieving comparable, though slightly lower, visual quality due to 3D reconstruction error. In addition, the object-centric Gaussian representation enables NGFF to generate novel views with novel backgrounds (see Appendix E). Finally, Figure 5 demonstrates its capability for interactive video generation: through ODE-based force modeling, NGFF produces physically plausible interventions, whereas competing methods fail to preserve consistency under external perturbations.

Table 2: **Video generation across generalization splits.** Higher is better. Comp., NB, NV, and All represent compositional, novel-background, novel-view, and comprehensive split that considers all three aspects, respectively. Note that Cosmos performs generalization in the novel-view setting, while NGFF performs a harder novel-view-synthesis task.

| Model | Split | VLM Eval. | | Human Eval. | |
|---|---|---|---|---|---|
| | | PhysR | PhotoR | PhysR | PhotoR |
| Cosmos | Comp. | 0.34 | **0.42** | 0.29 | 0.43 |
| | NB | 0.26 | **0.46** | 0.30 | 0.41 |
| | NV | 0.39 | 0.42 | 0.26 | 0.39 |
| | All | 0.20 | 0.32 | 0.28 | 0.41 |
| Cosmos tuned | Comp. | 0.26 | 0.35 | **0.57** | **0.58** |
| | NB | 0.38 | 0.36 | 0.60 | 0.60 |
| | NV | **0.49** | **0.40** | **0.63** | **0.62** |
| | All | 0.24 | 0.36 | 0.59 | 0.58 |
| **NGFF-V** | Comp. | **0.47** | **0.42** | 0.56 | 0.55 |
| | NB | **0.56** | 0.42 | **0.63** | **0.61** |
| | NV | 0.44 | 0.38 | 0.55 | 0.54 |
| | All | **0.30** | 0.35 | 0.55 | 0.55 |
| Veo3 | All | 0.29 | **0.41** | 0.53 | **0.64** |
| PhysGen3D | All | 0.19 | 0.35 | **0.57** | 0.58 |

### 4.4 Real-world experiments

We aim to deploy the trained NGFF model in real-world applications. A key challenge in this transition is the sim-to-real gap, which arises due to uncertainties in both perception and physical properties. In Figure 7, we present the results of real-world predictions made by large video generation models—Veo3, Cosmos, and Cosmos after fine-tuning on our GSCollision dataset. While both Veo3 and Cosmos produce visually high-quality videos, they fail to accurately capture real-world gravity and object interactions. Moreover, after fine-tuning, Cosmos tends to overfit the training

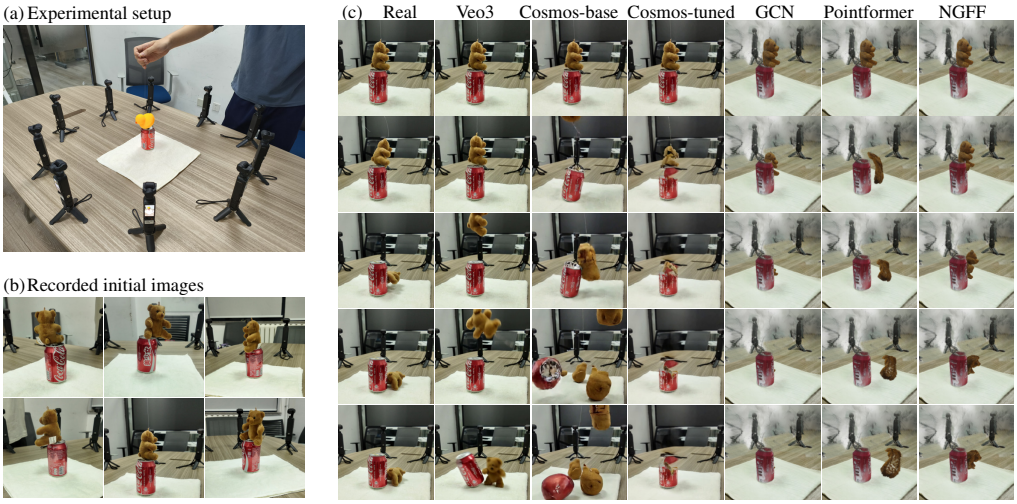


Figure 7: **Real-world results.** (a) Experimental setup in the real world, where 10 Pocket 3 cameras are used to capture multi-view dynamic videos of objects dropping onto a table. (b) Initial multi-view frames captured from the real-world setup. (c) Comparison of model predictions with ground truth. While large video generation models produce visually high-quality results, they fail to accurately simulate physical phenomena, such as the emergence of additional objects, unrealistic gravity, and incorrect collisions. In contrast, NGFF shows more robust and physically accurate predictions.

data, resulting in unreliable predictions. In contrast, our NGFF generates more accurate trajectories that closely match real-world behavior. Further details on our experimental setup can be found in Appendix C.3.

## 5 Discussion

**Predicting 4D physical dynamics from minimal observations.** Our method currently relies on multi-view inputs to reconstruct reliable 3D Gaussian, yet humans can often anticipate future dynamics from a single snapshot. Bridging this gap requires extending NGFF to extreme perceptual constraints such as monocular RGB or partial observations. Promising directions include integrating stronger generative priors on geometry and physics, or leveraging large-scale pretraining to reduce reliance on multi-view supervision, thereby moving closer to human-level physical reasoning.

**Scaling to diverse objects and complex scenes.** Our benchmark covers 10 representative objects across rigid and soft categories, yet real-world scenarios involve far more varied materials, articulated structures, and heterogeneous environments. Scaling NGFF to thousands of objects and compositions will require advances in both data efficiency and representation learning.

**Interpretable physics-grounded reasoning.** A central motivation of NGFF is to move beyond black-box video generation toward models that reveal interpretable intermediate states—such as explicit geometries and force fields. Future research could explore richer forms of interpretability, such as causal counterfactual reasoning ("what if" interventions) or explicit disentanglement of latent physical properties (e.g., mass, stiffness). Such capabilities would enhance the utility of NGFF in scientific discovery, robotics, and embodied AI.

## 6 Conclusion

We present NGFF, an efficient end-to-end neural framework that combines 3D Gaussian representations with physics-based modeling to generate interactive, physically realistic 4D videos from multi-view RGB inputs. Experiments on both synthetic and real data show that NGFF yields robust representations, generalizes to unseen scenarios, and outperforms SOTA video generation and physics simulation methods. Looking ahead, extending NGFF to broader object categories, noisy inputs, and interactive tasks may enable general-purpose world models that integrate physical consistency with visual realism for robust prediction, reasoning, and planning.

**Reproducibility statement** To facilitate reproducibility, we document the data generation process in Appendix A, provide implementation details of our model in Appendix B, describe the setup of baseline methods in Appendix C.1, outline the evaluation metrics in Appendix C.2, and detail the collection and processing of real-world data in Appendix C.3. Both the simulation and real-world datasets will be released on Hugging Face, while the code will be made publicly available on GitHub. In addition, we are developing an interactive website that allows users to directly experiment with our model.

## REFERENCES


Kelsey R Allen, Kevin A Smith, and Joshua B Tenenbaum. Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning. *Proceedings of the National Academy of Sciences (PNAS)*, 117(47):29302–29310, 2020. 3, 28

Kelsey R Allen, Yulia Rubanova, Tatiana Lopez-Guevara, William Whitney, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Tobias Pfaff. Learning rigid dynamics with face interaction graph networks. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023. 3, 28

Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. Phyre: A new benchmark for physical reasoning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3, 5, 28

Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences (PNAS)*, 2013. 1

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 4

Daniel M Bear, Elias Wang, Damian Mrowca, Felix J Binder, Hsiao-Yu Fish Tung, RT Pramod, Cameron Holdaway, Sirui Tao, Kevin Smith, Fan-Yun Sun, et al. Physion: Evaluating physical prediction from vision in humans and machines. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3, 5, 28

Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. Physgen3d: Crafting a miniature interactive world from a single image. *CVPR*, 2025. 2, 3, 6, 8, 22, 28

Bo Dai, Linge Wang, Baoxiong Jia, Zeyu Zhang, Song-Chun Zhu, Chi Zhang, and Yixin Zhu. X-voe: Measuring explanatory violation of expectation in physical events. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2023. 3, 28

Google DeepMind. Veo 3 technical report. Technical report, Google DeepMind, 2025. URL https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf. Accessed: 2025-08-30. 2, 8

Danny Driess, Zhiao Huang, Yunzhu Li, Russ Tedrake, and Marc Toussaint. Learning multi-object dynamics with compositional neural radiance fields. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1755–1768. PMLR, 14–18 Dec 2023. 1, 3, 28

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022. 5

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 28

Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. *ICCV*, 2025a. 2, 3, 28

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views, 2025b. URL https://arxiv.org/abs/2505.23716. 3, 28

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. 2025. 1

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 3, 28

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *Proceedings of International Conference on Learning Representations (ICLR)*, 2017. 6

Brenden M Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, 2023. 6

Shiqian Li, Kewen Wu, Chi Zhang, and Yixin Zhu. On the learning mechanisms in physical reasoning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

Shiqian Li, Kewen Wu, Chi Zhang, and Yixin Zhu. I-phyre: Interactive physical reasoning. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023a. 3, 5, 28

Shiqian Li, Ruihong Shen, Chi Zhang, and Yixin Zhu. Neural force field: Learning generalized physical representation from a few examples. *arXiv preprint arXiv:2502.08987*, 2025. 1

Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. Pac-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023b. 3

Chenguo Lin, Panwang Pan, Bangbang Yang, Zeming Li, and Yadong Mu. Diffsplat: Repurposing image diffusion models for scalable 3d gaussian splat generation. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2025a. 4, 18, 24

Yuchen Lin, Chenguo Lin, Jianjin Xu, and Yadong Mu. Omniphysgs: 3d constitutive gaussians for general physics-based dynamics generation. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2025b. 2, 3, 28

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. 5

Pingchuan Ma, Peter Yichen Chen, Bolei Deng, Joshua B Tenenbaum, Tao Du, Chuang Gan, and Wojciech Matusik. Learning neural constitutive laws from motion observations for generalizable pde dynamics. In *International Conference on Machine Learning*. PMLR, 2023. 1

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *Proceedings of International Conference on Computer Vision (ICCV)*, 2025. 1

NVIDIA, :, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai, 2025. URL https://arxiv.org/abs/2501.03575. 2, 8

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 4

Luis S Piloto, Ari Weinstein, Peter Battaglia, and Matthew Botvinick. Intuitive physics learning in a deep-learning model inspired by developmental psychology. *Nature human behaviour*, 6(9): 1257–1267, 2022. 3, 28

R. T. Pramod, Elizabeth Mieczkowski, Cyn X. Fang, Joshua B. Tenenbaum, and Nancy Kanwisher. Decoding predicted future states from the brain's "physics engine". *Science Advances*, 11(22): eadr7429, 2025. URL https://www.science.org/doi/abs/10.1126/sciadv.adr7429. 1

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714. 4

Yulia Rubanova, Tatiana Lopez-Guevara, Kelsey R. Allen, William F. Whitney, Kimberly Stachenfeld, and Tobias Pfaff. Learning rigid-body simulators over implicit shapes for large-scale scenes and vision. In *Advances in Neural Information Processing Systems*, 2024. Oral. 1, 3, 28

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks, 2020. 3, 28

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. URL https://arxiv.org/abs/1911.11763. 18

Haochen Shi, Huazhe Xu, Zhiao Huang, Yunzhu Li, and Jiajun Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects in 3d with graph networks. *The International Journal of Robotics Research*, 43(4):533–549, 2024. 3, 28

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

Fatemeh Shiri, Xiao-Yu Guo, Mona Golestan Far, Xin Yu, Reza Haf, and Yuan-Fang" Li. An empirical analysis on spatial reasoning capabilities of large multimodal models. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. 1, 28

Elizabeth S Spelke and Katherine D Kinzler. Core knowledge. *Developmental Science*, 10(1):89–96, 2007. 1

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a. 2, 3, 17, 28

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. $\pi^3$: Scalable permutation-equivariant visual geometry learning, 2025b. URL https://arxiv.org/abs/2507.13347. 2, 3, 17, 28

William F Whitney, Tatiana Lopez-Guevara, Tobias Pfaff, Yulia Rubanova, Thomas Kipf, Kimberly Stachenfeld, and Kelsey R Allen. Learning 3d particle-based simulators from rgb-d videos. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024. 1, 3, 28

Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. 18

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024b. 2, 6

Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22378–22389, June 2024. 2, 18

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. *arXiv preprint arXiv:2311.12198*, 2023. 2, 3, 6, 15, 28

Haotian Xue, Antonio Torralba, Joshua Tenenbaum, Daniel Yamins, Yunzhu Li, and Hsiao-Yu Tung. 3d-intphys: Towards more generalized 3d-grounded visual intuitive physics under challenging scenes. In *NIPS*, 2023a. 3, 28

Haotian Xue, Antonio Torralba, Joshua B. Tenenbaum, Daniel LK Yamins, Yunzhu Li, and Hsiao-Yu Tung. 3d-intphys: Towards more generalized 3d-grounded visual intuitive physics under challenging scenes, 2023b. 1

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024. 1, 28

Botao Ye, Sifei Liu, Haofei Xu, Li Xueting, Marc Pollefeys, Ming-Hsuan Yang, and Peng Songyou. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2025. 4, 18

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. In *SIGGRAPH Asia 2025 Conference Papers*, 2025. 1

Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Particle-grid neural dynamics for learning deformable object models from rgb-d videos. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. 2, 3, 28

Mikel Zhobro, Andreas René Geist, and Georg Martius. 3dgsim: Learning 3d-gaussian simulators from rgb videos. *arXiv*, 2025. URL https://arxiv.org/abs/2503.24009. 2, 3, 28

# CONTENTS OF SUPPLEMENTARY MATERIAL

# A DATASET CONFIGURATION

To support large-scale evaluation of physically grounded video prediction, we construct GSCollision, a dataset that couples Gaussian splatting with MPM-based simulation. The pipeline described in Appendix A.1 generates temporally consistent Gaussian trajectories from multi-object scenes under controlled physics. The dataset is organized into a modular structure (Appendix A.2) that includes backgrounds, object assets, scene configurations, simulated dynamics, and video recordings, providing a unified platform for reconstruction and prediction. Finally, we define systematic generalization splits (Appendix A.3) that cover spatial, temporal, and compositional variations, enabling rigorous testing of model robustness across diverse physical scenarios.

## A.1 DATA GENERATION

To construct our dataset, we employ a hybrid pipeline that integrates Gaussian splatting with a GPU-accelerated MPM engine implemented with Warp (Xie et al., 2023). First, pretrained Gaussian scene representations are loaded from checkpoints and pre-processed by removing low-opacity kernels, applying global rotations, and transforming particles to a normalized coordinate system. If available, segmentation masks are used to reorder particles by object identity, enabling per-object material assignments and stiffness parameters. Optionally, internal particle filling is performed to increase density for more accurate simulation.

The pre-processed particle states are then converted into initial conditions for the MPM solver, where particle volumes, covariance matrices, and object-specific material parameters (e.g., Young's modulus, density, boundary conditions) are configured. The simulation domain is defined as a cubic grid. The solver is initialized with either zero or user-specified velocities, and a box-shaped boundary of size 2 is enforced as defined by the scene configuration. Each particle stores position, velocity, deformation gradient, rotation, covariance, stress, mass, and density, which are dynamically updated at each step through the standard particle–grid–particle (P2G2P) pipeline.

During simulation, the solver advances dynamics through substeps, exporting per-frame particle attributes including positions, covariance matrices, and rotations. These outputs are saved as frame-wise datasets (e.g., in `.h5` format), which preserve all Gaussian attributes required for differentiable rendering. This process produces temporally consistent particle trajectories aligned with Gaussian splatting, yielding high-fidelity dynamic sequences that couple perception and physics.

## A.2 DATA STRUCTURE

GSCollision is organized into several components that together provide a complete pipeline from scene configuration to dynamic simulation:

- **backgrounds** stores environment-specific backgrounds (e.g., `table0`, `table1`). Each subdirectory contains camera parameters (`camera_2999.pt`) and Gaussian point cloud representations (`gaussians_feedforward.ply`), enabling consistent scene reconstruction and rendering.
- **objects** contains individual object assets. Each object (e.g., `ball`, `pillow`) includes camera calibration data (`cameras.json`) and its corresponding point cloud (`point_cloud`), serving as atomic units for scene composition and physical simulation.
- **scene_configs** provides scene-level configuration files (e.g., `3_0.json`, `3_1.json`) that specify object layouts and initialization conditions for simulation.
- **scenes** contains multi-object scene Gaussians grouped by index (e.g., `3_0`, `3_1`). Each scene contains different object combinations (e.g., `0_panda_ball_can`, `300_miku_miku_pillow`), representing diverse interaction setups.
- **mpm** stores dynamic Gaussian trajectories simulated with the *MPM*. Subdirectories mirror those in **scenes**, allowing direct correspondence between scene definitions and their physically grounded dynamics.
- **initial** contains the multi-view images of the initial scene prior to interaction, serving as the starting point for temporal evolution.
- **dynamic** records the dynamic videos of object interactions, aligned with **initial** and **mpm**, and used for training and evaluating video prediction models.

In summary, GSCollision integrates backgrounds, object assets, scene configurations, and both simulated (mpm) and recorded (initial, dynamic) trajectories. This structure enables systematic construction of complex multi-object environments and provides a unified platform for studying **scene reconstruction, physical simulation, and dynamic prediction**.

```
+-- backgrounds
|   +-- table0
|   |   +-- camera_2999.pt
|   |   \-- gaussians_feedforward.ply
|   +-- table1
|       +-- camera_2999.pt
|       \-- gaussians_feedforward.ply
+-- objects
|   +-- ball
|   |   +-- cameras.json
|   |   \-- point_cloud
|   \-- pillow
|       +-- cameras.json
|       \-- point_cloud
+-- scene_configs
|   +-- 3_0.json
|   \-- 3_1.json
\-- scenes
|   +-- 3_0
|   |   +-- 0_panda_ball_can
|   |   \-- 100_can_panda_phone
|   \-- 3_1
|       +-- 300_miku_miku_pillow
|       \-- 301_cloth_can_panda
+-- mpm
|   +-- 3_0
|   |   +-- 0_panda_ball_can
|   |   \-- 100_can_panda_phone
|   \-- 3_1
|       +-- 300_miku_miku_pillow
|       \-- 301_cloth_can_panda
+-- initial
|   +-- 3_0
|   |   +-- 0_panda_ball_can
|   |   \-- 100_can_panda_phone
|   \-- 3_1
|       +-- 300_miku_miku_pillow
|       \-- 301_cloth_can_panda
+-- dynamic
|   +-- 3_0
|   |   +-- 0_panda_ball_can
|   |   \-- 100_can_panda_phone
|   \-- 3_1
|       +-- 300_miku_miku_pillow
|       \-- 301_cloth_can_panda
```

The directory sizes of the dataset is shown in Table A1.

### A.3 Generalization splits

We partition the dataset into 12 groups. Among them, groups `3_0`–`3_8` serve as the training set, while group `3_9`, `4` and `6` are used to test generalization. Table A2 summarizes the dataset configuration and evaluation splits for both dynamics prediction and video generation. The training set is built from object triplets drawn from ten categories, across groups `3_0`–`3_8`, with trajectories spanning 80 simulation steps (1.6s), rendered from 20 viewpoints and 4 backgrounds. For dynamics prediction, we consider three generalization settings: **spatial** (novel object placements in group `3_9`), **temporal** (longer rollouts of 100 steps), and **compositional** (novel object combinations involving 4–6 objects in groups `4` and `6`). For video generation, we further define splits for **compositional**

Table A1: **Directory sizes of the GSCollision dataset.** Others contain objects, config files, reconstruction files, etc.

| Directory | Size (T) | Percentage |
|---|---|---|
| dynamic | 0.854 | 20.1% |
| initial | 0.122 | 2.9% |
| mpm | 2.300 | 54.1% |
| backgrounds | 0.032 | 0.8% |
| scenes | 0.061 | 1.4% |
| others | 0.881 | 20.7% |
| **Total** | **4.250** | **100%** |

(`3_9`, `4`, `6`), **novel-view** (5 unseen viewpoints), **novel-background** (held-out backgrounds), and a **comprehensive** split that jointly evaluates multiple factors with trajectories extended to 100 steps (2s). Green cells indicate aspects consistent with training, while blue cells denote novel conditions used for testing.

Table A2: **Statistics of different generalization splits.** Green indicates that the training and test data share the same configuration in certain aspects, whereas blue indicates they are different.

| | Objects | Groups | Time span | Viewpoints | Backgrounds |
|---|---|---|---|---|---|
| Training set | 3 from 10 kinds | 3_0 – 3_8 | 80 step / 1.6s | 20 | 4 |
| **Dynamics prediction** | | | | | |
| Spatial | 3 from 10 kinds | 3_9 | 80 step / 1.6s | / | / |
| Temporal | 3 from 10 kinds | 3_0 – 3_8 | 100 steps / 2s | / | / |
| Compositional | 4–6 from 10 kinds | 4, 6 | 80 step / 1.6s | / | / |
| **Video generation** | | | | | |
| Compositional | 3 from 10 kinds | 3_9, 4, 6 | 80 step / 1.6s | 20 | 4 |
| Novel-view | 3 from 10 kinds | 3_0 – 3_8 | 80 step / 1.6s | 5 | 4 |
| Novel-background | 3 from 10 kinds | 3_0 – 3_8 | 80 step / 1.6s | 20 | 4 |
| Comprehensive | 3–6 from 10 kinds | 3_9, 4, 6 | 100 steps / 2s | 5 | 4 |

# B Implementation Details

## B.1 Feed-forward Gaussian reconstruction

Starting from uncalibrated RGB images, the initial step is to recover the 3D point cloud structure of a scene. Traditional optimization-based methods, such as Structure-from-Motion and Multi-View Stereo, necessitate capturing tens or even hundreds of views, which are often impractical in real-world scenarios. Recently, feed-forward foundation reconstruction models (Wang et al., 2025a;b) have emerged as a powerful alternative. Pretrained on massive datasets, these models perform 3D reconstruction in a single forward pass, enabling lightning-speed scene reconstruction. This provides the foundation for subsequent neural simulation and planning within the reconstructed 3D representations.

In our experiments, we found that the permutation-equivariant architecture of $\pi^3$ achieves higher accuracy in object registration compared to VGGT, a model based on first reference frame reconstruction. Consequently, we selected $\pi^3$ as our backbone.

Building upon the $\pi^3$ model, we introduce modifications to create $\pi^3-$GS for feed-forward Gaussian scene reconstruction. To achieve stronger real-world generalization, we freeze the alternating attention encoder and the camera head of the pre-trained $\pi^3$ model. We directly use the predictions from its point head as the centers, $\mu$, for the Gaussians. Furthermore, we observed that MLP-based pixel-shuffling is prone to creating artifacts at patch boundaries. Since convolutional operations yield smoother results, we replaced this with a convolutional upsampling layer in the splatter head.

Specifically, we first refine the patch features from the transformer encoder with three convolutional blocks, followed by an upsampling layer and two additional convolutional blocks to eliminate artifacts. We also applied a direct RGB shortcut (Ye et al., 2025), composed of 3 Residual CNN blocks from the input image, to preserve high-frequency details and enhance appearance reconstruction.

We trained the splatter head of our $\pi^3-$GS model on the Wildrgbd (Xia et al., 2024) dataset, which contains approximately 22,000 scenes. The training was conducted on 8 NVIDIA H100 80G GPUs for 50 epochs with a global batch size of 24. Both mixed-precision training and gradient checkpointing were utilized.

### B.2 Single-view gaussian refinement

Feed-forward reconstruction models that lack a generative prior are inherently limited in handling challenges such as incomplete observations and occlusions. This deficiency can adversely affect the topological integrity of the object's 3D Gaussian representation and, consequently, the fidelity of subsequent neural simulations. To address this, we propose a pipeline that first completes the object's geometry using a 3D asset generation model, followed by a $\mathrm{Sim}(3)$ point cloud alignment to register it within the scene.

Initially, the segmented object image is processed through a super-resolution pipeline (Wu et al., 2024a) to enhance textural details. We then employ a pretrained 3D generative model, DiffSplat (Lin et al., 2025a), to infer a complete 3D Gaussian representation of the object, conditioned on the single input view.

The generated Gaussian asset resides in a normalized, object-centric coordinate system, which is inconsistent with the object's true scale and pose in the scene. To place the generated object accurately, we introduce a $\mathrm{Sim}(3)$ registration algorithm that combines visual feature matching with gradient-based optimization. First, we render a set of images $\{\mathcal{I}_k\}$ by orbiting the generated asset at multiple elevations. For each rendered image $\mathcal{I}_k$, we use SuperGlue (Sarlin et al., 2020) to establish matches with the original input image $\mathcal{I}_{in}$, and select the view that yields the maximum number of 2D correspondences, denoted as $\mathcal{C}_{2D} = \{(\mathbf{p}_i, \mathbf{p}'_i)\}_{i=1}^{N}$. These 2D matches are then lifted to 3D, $\mathcal{C}_{3D} = \{(\mathbf{P}_i, \mathbf{P}'_i)\}_{i=1}^{N}$, by identifying the 3D points in the respective point clouds, $\mathcal{P}_{gen}$ and $\mathcal{P}_{obs}$, that are closest to the corresponding camera rays. For initialization, we estimate the scale $s_{init}$ from the ratio of the point clouds' bounding box volumes and solve for an initial 6-DoF pose $[\mathbf{R}_{init}|\mathbf{t}_{init}] \in SE(3)$ using the Kabsch algorithm within a RANSAC framework. Subsequently, we jointly refine the similarity transformation $\mathbf{T} \in \mathrm{Sim}(3)$ by minimizing the Chamfer distance between the transformed generated point cloud and the observed point cloud via gradient descent:

$$(\mathbf{R}^*, \mathbf{t}^*, s^*) = \arg\min_{\mathbf{R}\in SO(3), \mathbf{t}\in\mathbb{R}^3, s\in\mathbb{R}^+} \mathcal{L}_{CD}(s\mathbf{R}\mathcal{P}_{gen} + \mathbf{t}, \mathcal{P}_{obs})$$

The entire registration process can be done within a few seconds.

### B.3 Neural Gaussian Force Field (NGFF)

Our framework builds upon a neural interaction–based dynamics predictor, which integrates object-level interaction modeling, boundary constraints, and stress field prediction into a differentiable ODE solver. The overall design couples four components: an Interaction Network (IN), a Stress Prediction Network (StressNet), boundary and collision modules, and a neural ODE–based temporal evolution module.

**Interaction Network (IN)** The IN module captures both geometric and state-dependent interactions among multiple objects. Each object is first encoded using a hierarchical PointNet backbone that extracts global geometric features from point clouds. State descriptors—including center of mass (CoM), orientation angles, linear and angular velocities—are embedded through multilayer perceptrons. Pairwise object relations are modeled via a branch–trunk structure: branch features encode relative states between objects, while trunk features preserve object-specific information. Their interaction is combined through element-wise multiplication and mapped to output forces and torques. To ensure physical consistency, the IN explicitly detects inter-object collisions and boundary contacts. Collision forces are masked by an intersection matrix, while boundary forces are predicted by a dedicated boundary network conditioned on both geometry and state features.

**Stress Prediction (StressNet)** Beyond rigid-body dynamics, the model accounts for distributed internal responses by predicting per-point stress fields. StressNet takes as input the local point coordinates, velocities, and the aggregated forces and torques from IN and boundary interactions. A shared MLP extracts local features, followed by a global max-pooling to capture object-level context. These are fused and projected to pointwise stress outputs. The design enforces rotation consistency by transforming predicted forces and stresses between global and local frames via differentiable Euler-angle rotation matrices.

**Boundary and Collision Modules** Physical validity is further maintained through two auxiliary functions: collision detection computes pairwise point distances between objects to construct overlap masks, which gate non-contact interactions; boundary detection evaluates the proximity of object points to the simulation domain limits, producing boundary masks to trigger repulsive boundary forces.

**Temporal Evolution with Neural ODE** To simulate motion, NGFFobj integrates the above predictors into a continuous-time dynamics system solved via the torchdiffeq ODE framework. The system state comprises point positions, point velocities, CoM and angular states, along with stress distributions. At each step, the IN outputs interaction forces and torques, and StressNet provides stress derivatives, which are combined with external forces (if any) and gravity. The resulting accelerations are integrated forward in time using either explicit Euler or adaptive-step solvers. This formulation enables stable long-horizon rollout while preserving differentiability for learning-based optimization.

**Training** The model is trained on 8 NVIDIA H100 80GB GPUs for 1001 epochs for 48 hours. The learning rate starts at $1 \times 10^{-5}$ and decays to a minimum of $1 \times 10^{-7}$. The architecture consists of 4 layers with a hidden dimension of 200. The batch size is set to 9 per node, and each epoch involves 80 steps with a chunk size of 80. The ODE method used is Euler with a step size of $2 \times 10^{-2}$, and a threshold of $5 \times 10^{-2}$ for collision detection is applied during training.

# C Experimental setup

## C.1 Baselines

### C.1.1 Graph Convolutional Neural Networks (GCN)

We adopt a Graph Convolutional Network (GCN) to model dynamics. Each object is represented by a set of keypoints, which serve as graph nodes, and edges are constructed using a radius-based neighbor search with a threshold. The node features are obtained by concatenating the 3D position and velocity of each keypoint.

The network consists of multiple GCNConv layers, where each layer performs message passing to aggregate information from neighboring nodes, followed by ReLU nonlinearities. A final fully connected layer predicts the residual update of position and velocity for each node. Prediction is performed in an autoregressive manner: at each step, the model updates the current state with the predicted residuals and rolls out the trajectory over multiple steps.

The GCN is trained on a single NVIDIA H100 80GB GPU with a learning rate starting at $1 \times 10^{-3}$, which decays to $1 \times 10^{-4}$. The model consists of 4 layers, each with a hidden dimension of 128. A batch size of 30 is used, with 80 steps per epoch, and the training runs for 500 epochs. At each step, the model processes 3000 samples, with data processed in chunks of 80 to ensure efficient memory usage. The dynamic model used in this setup is GCN, which is specifically designed to handle graph-structured data and learn complex relationships.

### C.1.2 Pointformer

Pointformer directly models interactions across all object keypoints. Each keypoint is embedded using a positional encoding derived from its 3D coordinates, followed by a linear projection into a high-dimensional latent space. The set of embedded keypoints from all objects is then processed by

a stack of multi-head self-attention layers, allowing each point to attend to and aggregate information from all others in the scene.

To handle variable numbers of objects and keypoints, a padding mask is applied to prevent attention from propagating through invalid nodes. The transformer output is normalized and projected back into the point space via a feedforward head to predict residual updates for each keypoint's position. As in the GCN baseline, prediction proceeds autoregressively over multiple rollout steps, generating a sequence of future trajectories.

Unlike GCNs, which rely on local neighborhood graphs, PointFormer captures global interactions across all keypoints through self-attention. This enables the model to represent long-range dependencies and complex multi-object dynamics, but at the cost of higher computational complexity due to quadratic attention scaling.

The Pointformer is trained on 4 NVIDIA H100 80GB GPUs for 60 hours. The model is trained with a learning rate starting at $5 \times 10^{-4}$, decaying to a minimum of $5 \times 10^{-6}$. The architecture consists of 3 layers and a hidden dimension of 128, with dropout 0.1. The batch size is set to 8 per node, with a total of 2001 epochs, and each epoch involves 80 steps with a chunk size of 80.

#### C.1.3 VLM-MPM

We employ Gemini-2.5-flash to infer the Young's modulus and density from 20 training videos. The estimated parameters are subsequently normalized to align with the value ranges required by the MPM simulator. The prompt used is:

```
For each object in the videos, estimate the object's density in kilograms
     per cubic meter and its Young's modulus in Pa. Return an json array
    of objects in JSON where each object has fields: name, density,
    youngs_modulus. Do not include extra text, only valid JSON that
    matches the schema. The objects you need to estimate are: {objects}.
```

The following simulations are identical to those employed in data generation.

#### C.1.4 COSMOS-PREDICT2

Cosmos-Predict2 is a World Foundation Model trained by NVIDIA, designed to simulate and predict the future state of the world as video. It can serve as a foundation for training physical AI systems in digital environments. The model balances both visual quality and physics awareness and is capable of generalizing to downstream tasks with a small amount of post-training.

We performed full-parameter fine-tuning on the **Cosmos-Predict2-2B-Video2World-480P-16FPS** model. For this process, we utilized a total of 216K video clips from the GSCollision dataset, which amounts to 17.28 million frames, each with a resolution of $448 \times 448$.

For text conditioning, we used the following prompt for all video clips:

```
A photorealistic video. Simulate the future dynamics of the foreground
    objects falling from the air onto the table. The simulation should
    realistically model various physical interactions including
    deformation, gravity, collisions between the objects, and their
    impact with the surface. Capture the subsequent motions until the
    objects come to a complete rest.
```

For image (video) conditioning, we randomly used 3-5 latent frames (corresponding to 9-17 actual frames) during training. During testing, we conditioned on the first 13 frames of the video.

The training was conducted on 8 NVIDIA H100 80G GPUs. We trained for 20,000 iterations until convergence, using an initial learning rate of $2.5 \times 10^{-4}$ and a global batch size of 24.

#### C.1.5 PHYSGEN3D

PhysGen3D transforms a single static image into an interactive, amodal 3D scene capable of simulating physically plausible future outcomes. The framework first reconstructs a complete 3D world

by leveraging a suite of pretrained vision models to infer geometry, semantics, materials, and lighting properties from the input image. This reconstructed scene is then passed to a physics-based simulator, which uses the MPM to generate object dynamics in response to LLM-inferred physical parameters. Finally, a physics-based rendering module seamlessly composites the simulated dynamic objects and their corresponding shadows back into the original scene, producing a coherent and controllable video. PhysGen3D enables fine-grained control over object interactions and generates motions that adhere to physical laws.

However, the framework's reliance on single-view reconstruction makes it susceptible to errors in complex scenarios. The method is primarily designed for scenes with simple geometry and can fail when dealing with heavy occlusions and multiple objects. The ill-posed nature of inferring 3D properties from a 2D image can lead to perception failures and parameter estimation errors under challenging situations. Besides, reliance on MPM simulators makes it slower than neural simulation methods on modern GPUs.

#### C.1.6 VEO3

Veo3 is a SOTA diffusion-based video generation model developed by Google DeepMind. It can interpret complex text prompts, capable of generating smooth and consistent dynamics for people and objects. It avoids the uncanny or jarring artifacts common in earlier models, producing motion that is both believable and visually pleasing.

However, during testing, we observed that while Veo3 maintains excellent temporal consistency during non-strenuous motion, the model still frequently generates outputs that violate fundamental Newtonian physics principles or object permanence during strenuous events, such as collisions.

### C.2 EVALUATION METRICS

In this study, we adopt different metrics for evaluating NGFF. For the accuracy of the predicted dynamics, we choose RMSE, FPE, and R as our primary metrics. For assessing the video generation correctness, we employ PhysR and PhotoR.

**Root Mean Squared Error (RMSE)** The RMSE, defined as the square root of the Mean Squared Error (MSE), retains the property of penalizing larger deviations but expresses the error in the same units as the original data. This makes it easier to interpret in physical contexts, as it reflects the average magnitude of prediction errors relative to true trajectories:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{z}_t - z_t)^2}. \tag{A1}$$

**Final Position Error (FPE)** The FPE evaluates the difference between the predicted and ground-truth final positions of an object. This metric is particularly important for goal-oriented physical reasoning, where accuracy at the endpoint is critical. By focusing on the final state, FPE complements trajectory-based metrics and ensures that models not only capture motion dynamics but also predict the ultimate destination correctly:

$$\text{FPE} = \left|\hat{z}\text{final} - z\text{final}\right|. \tag{A2}$$

**Position Change Error (PCE)** The PCE measures the discrepancy between the predicted and actual changes in position over time. This metric can be interpreted as an indicator of how accurately the model captures the object's velocity throughout its motion:

$$\text{PCE} = \left|\Delta\hat{z}_t - \Delta z_t\right|. \tag{A3}$$

**Pearson Correlation Coefficient (R)** The R coefficient captures the linear correlation between predicted and actual trajectories. Rather than measuring absolute error, it reflects how well the model aligns with the overall trajectory pattern. A high value indicates strong agreement in motion trends, while a low value suggests that the model fails to capture the underlying trajectory structure:

$$\text{R} = \frac{\sum_{t=1}^{n}(\hat{z}t - \bar{\hat{z}})(z_t - \bar{z})}{\sqrt{\sum t=1^{n}(\hat{z}t - \bar{\hat{z}})^2 \sum t=1^{n}(z_t - \bar{z})^2}}. \tag{A4}$$

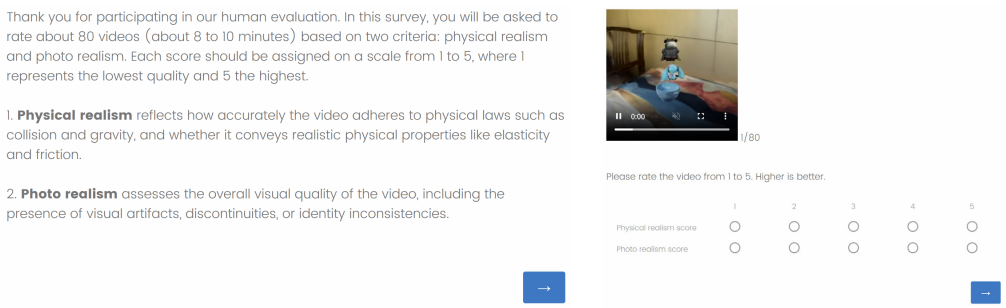


Figure A1: **An example of human study questionaire.**

Given that video-generation models like Veo3 and PhysGen3D are closed-source or untrainable, for which a direct comparison with ground truth would be inequitable, we adopted the qualitative evaluation framework established by PhysGen3D (Chen et al., 2025) to quantitatively evaluate video generation quality. This involves leveraging a Vision-Language Model, Gemini-2.5-flash to assess two key criteria: PhysR and PhotoR.

**Physical Realism (PhysR)** The PhysR measures how realistically the video follows the physical rules like collision and gravity and whether the video represents real physical properties like elasticity and friction.

**Photo Realism (PhotoR)** The PhotoR measures the overall visual quality of the video, including the visual artifacts, discontinuities, and id-inconsistency.

The prompt is as follows:

```
# [video inputs]
I would like you to evaluate the quality of generated videos above based
    on the following criteria: physical realism and photorealism. The
    evaluation will be based on 10 evenly sampled frames from each video.
     Given the original image and the above instructions , please
    evaluate the quality of each video on the two criteria mentioned
    above. Note that: Physical Realism measures how realistically the
    video follows the physical rules and whether the video represents
    real physical properties like elasticity and friction. To discourage
    completely stable video generation, we instruct respondents to
    penalize such cases. Photorealism assesses the overall visual quality
     of the video, including the presence of visual artifacts,
    discontinuities, and how accurately the video replicates details of
    light, shadow, texture, and materials. Please provide the following
    details for each video in an json array of videos where each video
    object has fields: physical_realism score, photorealism score and
    content. The content should be a sentence summarizing the video,
    scores should be ranging from 0-1, with 1 to be the best, round to 2
    decimal places:
```

**Human evaluation** We designed a questionnaire to conduct human evaluation on video generation quality across different models, as illustrated in Figure A1. A total of 61 participants were recruited to complete an 80-page questionnaire. At the beginning, we provided a detailed explanation of two metrics. Each page of the questionnaire contains a 2–3 second video randomly chosen from all models and generalization splits. Participants are instructed to assess each video based on the two dimensions above: PhysR and PhotoR. This human study design, accompanied by results from VLMs, ensures a fair, consistent, and comprehensive evaluation. Detailed distributions of human evaluation results can be found in Figure A2 and Figure A3

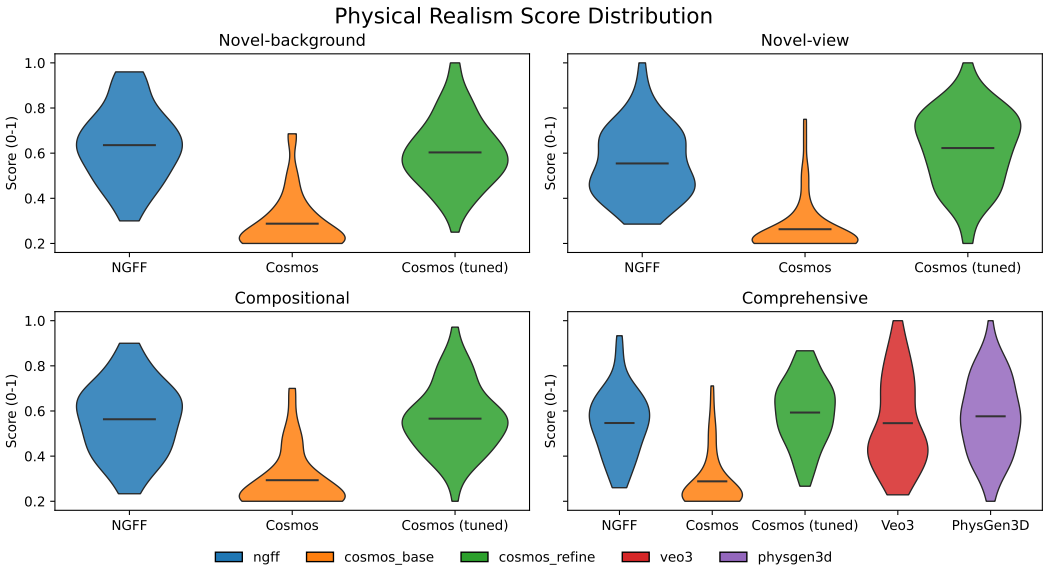


Figure A2: **Detailed distribution of human evaluation results on Physical Realism (PhysR).**

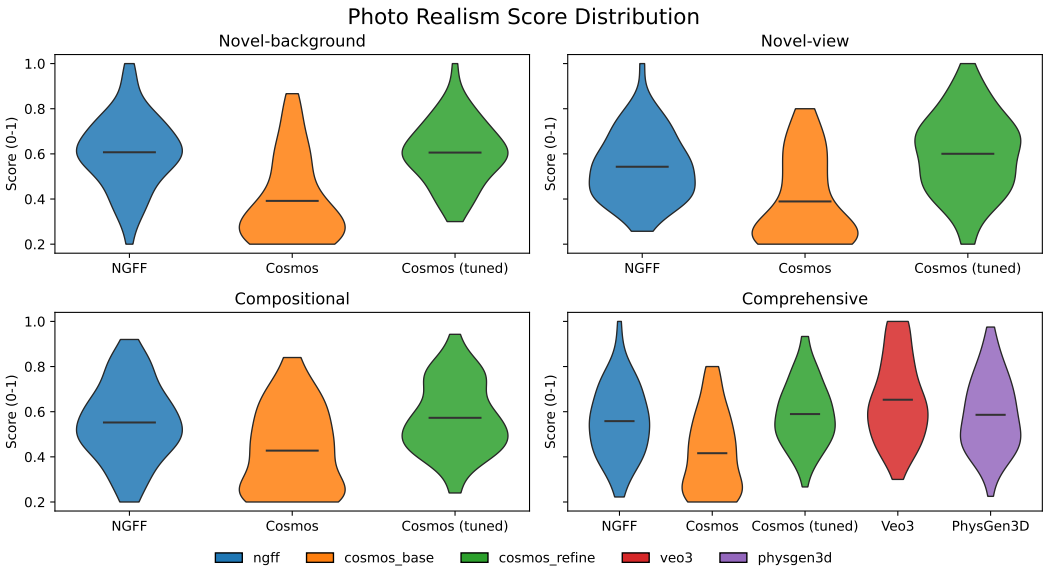


Figure A3: **Detailed distribution of human evaluation results on Photo Realism (PhotoR).**

### C.3 Real world environments

#### C.3.1 Data collection

We collected real-world interaction sequences using a multi-view setup of ten DJI Pocket 3 cameras arranged around a table in a standard office environment. All cameras were calibrated to share identical intrinsic parameters, ensuring geometric consistency across views. To induce controlled dynamics, objects were lifted and released with transparent fishing line, creating falling and collision events while guaranteeing that each object started from a static state. In total, we recorded 40 dynamic sequences at 50 FPS and 3K resolution. The object set included a cola can, a teddy bear, and a rubber duck, allowing us to generate diverse two-object and three-object interaction scenarios with varying mass and material properties.

### C.3.2 VIDEO PROCESSING

For each sequence, we temporally trimmed the videos from the instant of release until all objects came to rest, typically spanning 50–60 frames. Each frame from every camera view was annotated with axis-aligned bounding boxes, obtained semi-automatically using SAM2 and refined by manual correction where necessary to ensure pixel-level accuracy. Object identities were explicitly labeled to support subsequent use in multimodal learning tasks. To enable 3D reconstruction, all frames were synchronized across views and processed using a feed-forward pretrained Gaussian-splatting model, with further refinement using DiffSplat (Lin et al., 2025a), producing multi-view-consistent 3D Gaussian representations. This pipeline ensured both high-quality geometry recovery and consistent object-level alignment, establishing a reliable benchmark for evaluating dynamic prediction models under real-world conditions. See the representative recorded videos in Figure A4, Figure A5, and Figure A6.


Figure A4: **Recorded multi-view dynamic interaction in the real world.** A teddy bear is released above a cola can and falls onto the table.

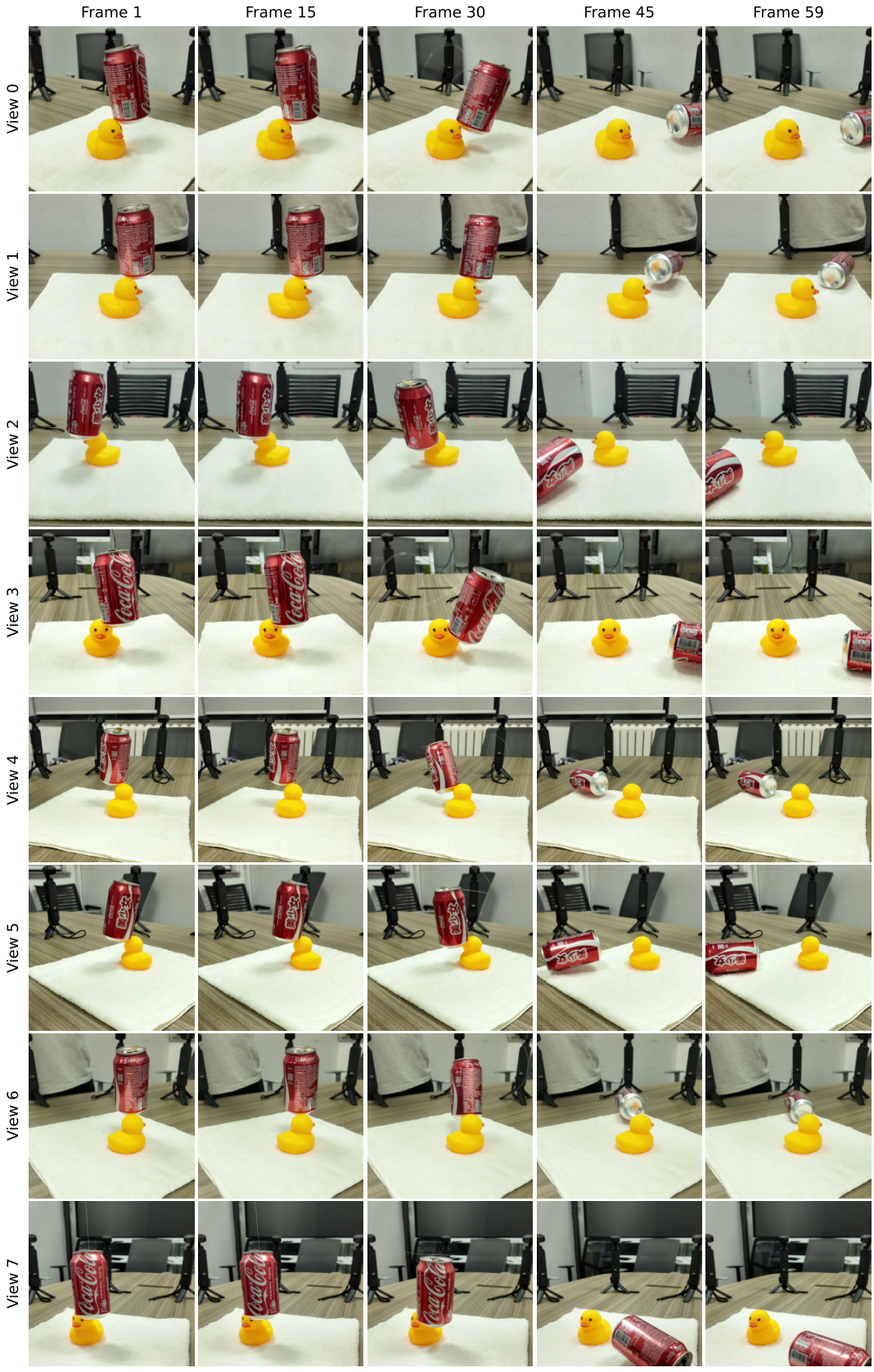


Figure A5: **Recorded multi-view dynamic interaction in the real world.** A cola can is released above a duck and falls onto the table.


Figure A6: **Recorded multi-view dynamic interaction in the real world.** A cola can is released above a duck, collides with the teddy bear, and falls onto the table.

# D MORE RELATED WORKS

## D.1 PHYSICAL REASONING

Physical reasoning is a core human ability to understand and interact with the physical world. Besides generating continuous and high-fidelity videos (Ho et al., 2022; Yang et al., 2024), physical reasoning tackles the challenges to comprehend and reason about the governing physical dynamics of visual scenes, representing a core capability required for AI systems to achieve human-level intuitive physics abilities. This core skill encompasses two critical domains: First, it involves spatial reasoning (Shiri et al., 2024) from video inputs, the ability to reconstruct and understand three-dimensional scenes, including object relationships, spatial configurations, and perspective. Second, it requires an understanding of fundamental physical laws governing object interactions and generalizing it to OOD scenarios.

Various benchmarks have been proposed to assess the physical reasoning capabilities of both humans and machines. Previous studies build datasets based on the VoE paradigm to examine agents' understanding of basic physical concepts (Piloto et al., 2022; Dai et al., 2023). Recent studies extend the passive observation paradigm to interactive environments, which require the agents to apply actions to finish tasks (Bakhtin et al., 2019; Allen et al., 2020; Bear et al., 2021; Li et al., 2023a). Our work builds upon the interactive physical environment to demonstrate the reasoning capability of our model.

## D.2 VISUAL DYNAMICS PREDICTION

Visual dynamics prediction, the task of predicting future frames from visual inputs, has been addressed through diverse approaches. Neural simulator-based methods commonly employ GNN as their dynamics backbone due to their relational inductive bias. Early approaches, while capable of simulating various physical phenomena (Sanchez-Gonzalez et al., 2020; Bear et al., 2021), often fail on complex materials and physical interactions. More recent approaches inject physics inductive bias into simulation such as mesh (Allen et al., 2023) or SDF (Rubanova et al., 2024) representation for rigid bodies and spring-mass models (Jiang et al., 2025a) or particle-grid representations (Zhang et al., 2025) for deformable objects. Despite their advancements, these methods often struggle with complex multi-object interaction scenarios and exhibit limited generalization abilities. While our method adopts a unified representation for different object materials and physical interactions by predicting force fields.

In contrast, physics simulator-based methods explicitly model scene dynamics using differentiable simulators. For example, techniques that render scenes into particles via 3D Gaussian rendering and simulate their evolution with Material Point Method (MPM)-based simulators (Xie et al., 2023; Lin et al., 2025b; Chen et al., 2025) produce realistic outcomes but rely heavily on strong physics priors or case-specific optimization, which may not be available in intuitive physics scenarios.

## D.3 SCENE REPRESENTATIONS FOR SIMULATION AND RENDERING

Early methods extracted geometry, such as point clouds, directly from RGB-D inputs for simulation and planning (Shi et al., 2024) and trained a separate module for rendering (Whitney et al., 2024). Later, NeRF enables differentiable rendering and can be jointly optimized for simulation (Driess et al., 2023; Xue et al., 2023a) at the cost of degraded flexibility due to implicit encoders. 3D Gaussian Splatting has emerged as a powerful alternative, offering photorealistic quality and real-time performance (Kerbl et al., 2023). The utility of 3D Gaussians extends beyond static rendering; works like PhysGaussian (Xie et al., 2023) have integrated them with Newtonian dynamics for high-quality motion synthesis. Advances in feed-forward reconstruction significantly accelerate the reconstruction process by directly inferring Gaussian attributes from unposed multi-view images (Wang et al., 2025a; Jiang et al., 2025b; Wang et al., 2025b), enabling fast, end-to-end scene creation suitable for downstream simulations. Our concurrent work 3DGSIM (Zhobro et al., 2025) also employs feed-forward Gaussian reconstruction and a transformer for prediction. They primarily focus on single-object dynamics, having limited generalization to multi-object interactions and planning capability.

Table A3: **Ablation results of NGFF and NGFF without deformation across different generalization settings.** Arrows indicate whether higher (↑) or lower (↓) values are better.

| Setting | Method | MSE (↓) | RMSE (↓) | PCE (↓) | FPE (↓) | PCC (↑) |
|---|---|---|---|---|---|---|
| Spatial | NGFF w/o deform. | 0.01466 | 0.10971 | 0.01386 | 0.45927 | 0.59506 |
| | NGFF | 0.00835 | 0.08199 | 0.01165 | 0.32576 | 0.66111 |
| Temporal | NGFF w/o deform. | 0.02605 | 0.14403 | 0.01421 | 0.59975 | 0.57836 |
| | NGFF | 0.01471 | 0.10711 | 0.01167 | 0.41933 | 0.65238 |
| Compositional-4 | NGFF w/o deform. | 0.02092 | 0.13031 | 0.01487 | 0.54689 | 0.52474 |
| | NGFF | 0.01052 | 0.09533 | 0.01210 | 0.37274 | 0.59444 |
| Compositional-6 | NGFF w/o deform. | 0.01910 | 0.13249 | 0.01527 | 0.54564 | 0.50577 |
| | NGFF | 0.01379 | 0.11268 | 0.01358 | 0.44583 | 0.54707 |

Table A4: **Inference time for different video generation methods** Times are measured on a single NVIDIA H100 80G GPU.

| Model | Time |
|---|---|
| **NGFF-V** | 37s (3 objects) / 72s (6 objects) |
| **NGFF-V (w/o refine)** | 12s (3 objects) / 19s (6 objects) |
| **Pointformer-V** | 37s (3 objects) / 72s (6 objects) |
| **GCN-V** | 37s (3 objects) / 72s (6 objects) |
| **PhysGen3D** | 400s (3 objects) / 590s (6 objects) |
| **Cosmos-predict2-2B** | 20s |
| **Veo3** | 11–360s (via API) |

# E Ablations and More Results

In this section, we provide supplementary results to further analyze the effectiveness and efficiency of our proposed framework. First, we report an ablation study in Table A3, which compares NGFF against its variant without deformation modeling across different generalization settings. The results demonstrate that explicitly modeling deformation consistently improves predictive accuracy, yielding lower errors (MSE, RMSE, PCE, and FPE) and higher correlations (PCC).

We also benchmark the inference speed of our method against alternative approaches in Table A4. NGFF-V attains efficient inference (2.5s per sequence on a single H100 GPU), significantly outperforming computationally expensive physics-based simulators (e.g., PhysGen3D) while remaining competitive with large-scale generative models (e.g., Cosmos-predict2-2B and Veo3). Together, these results highlight that our framework achieves a favorable balance between accuracy, realism, and efficiency.

# F More Visualizations

## F.1 Dynamic Prediction

We present more visualizations of dynamic prediction in Figures A7 to A11.

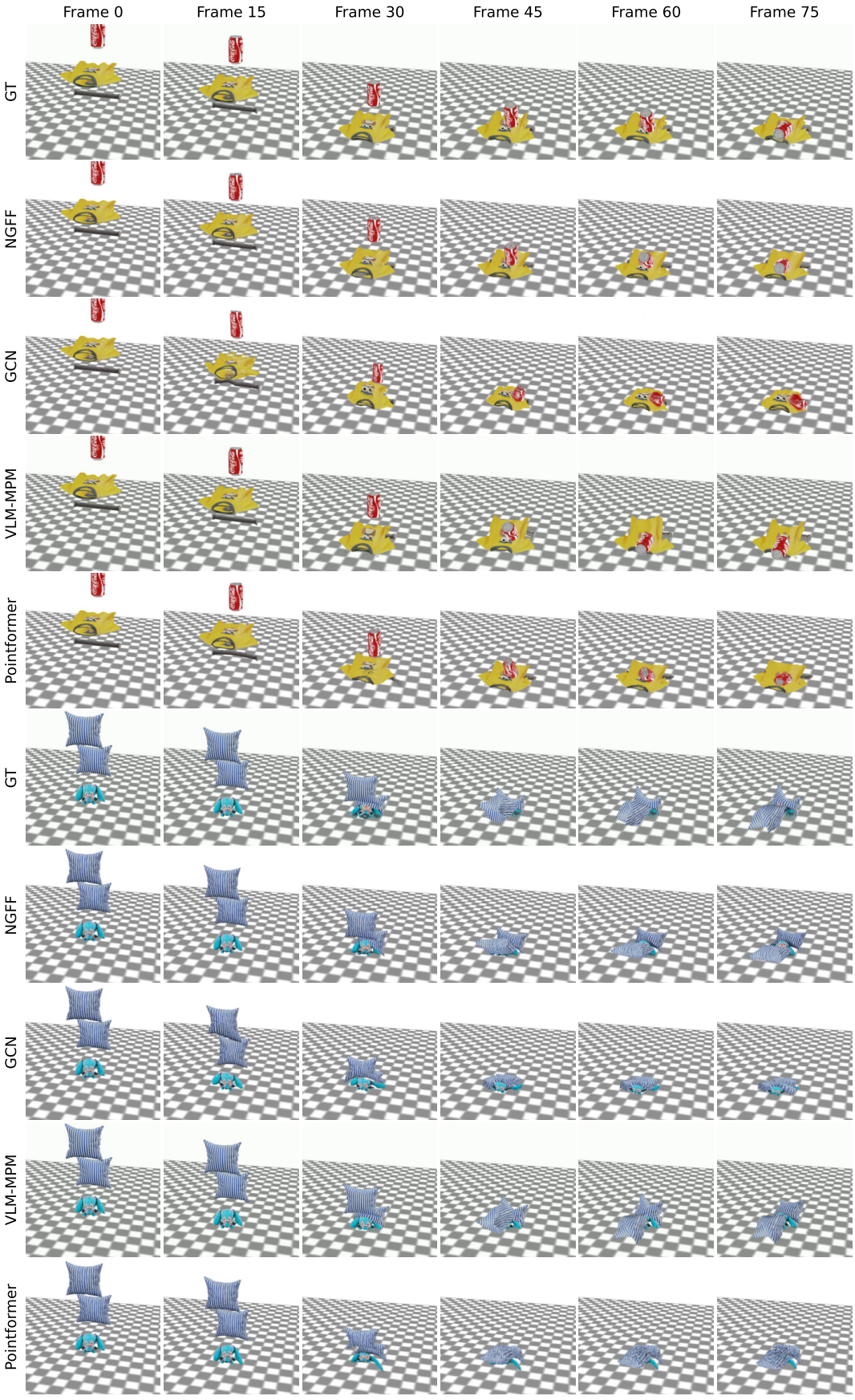


Figure A7: **Dynamic prediction results.**

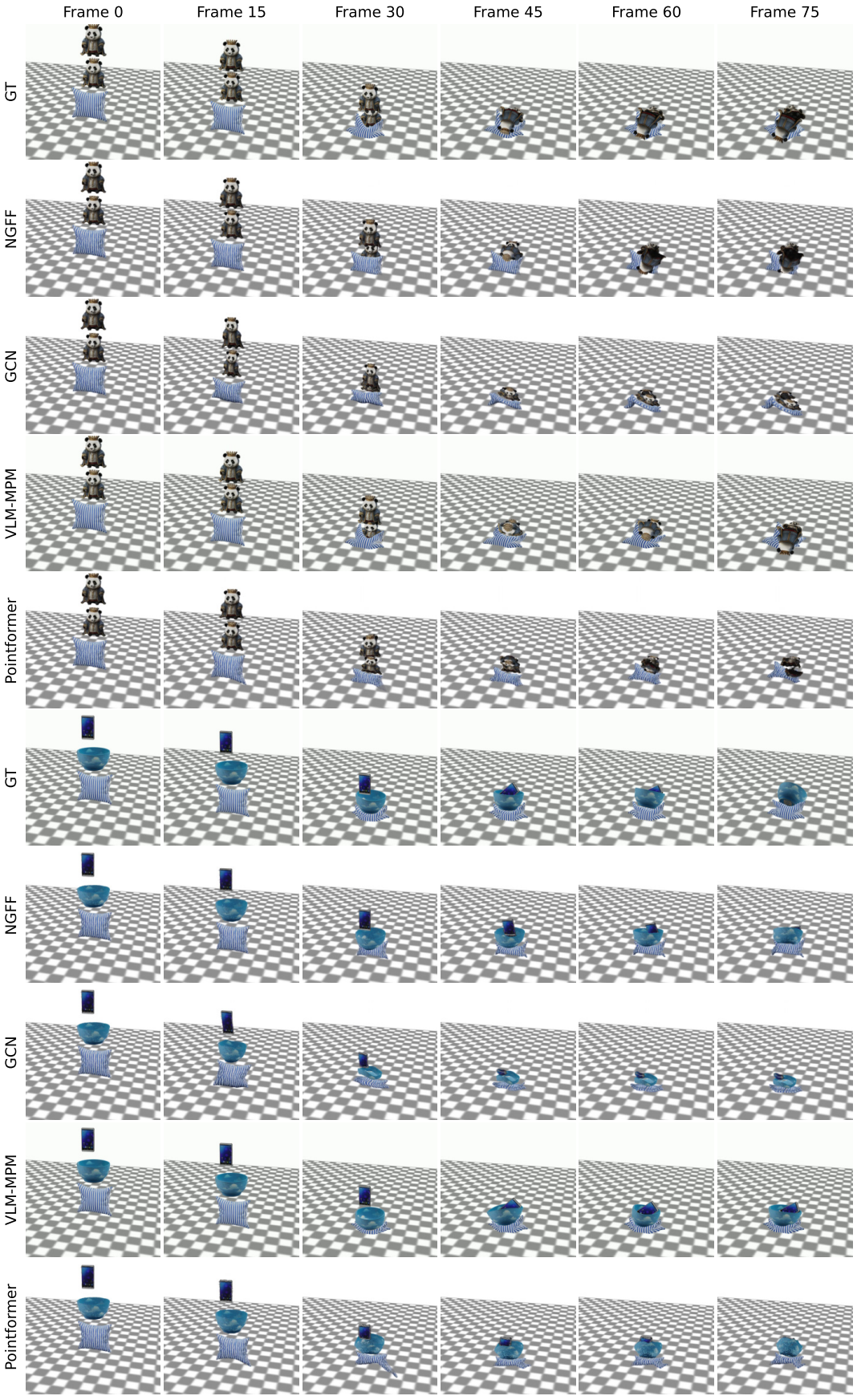


Figure A8: **Dynamic prediction results.**

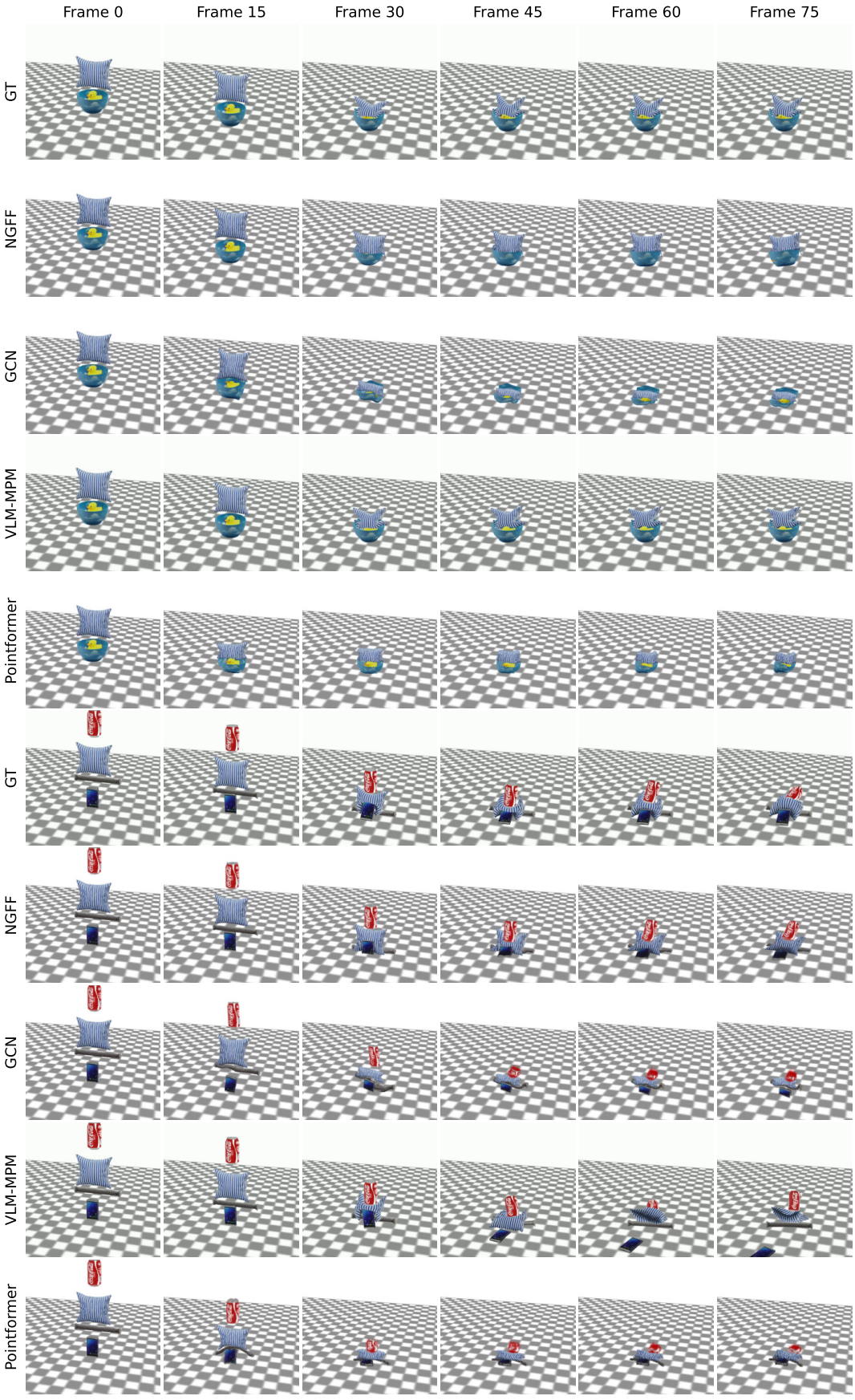


Figure A9: **Dynamic prediction results.**

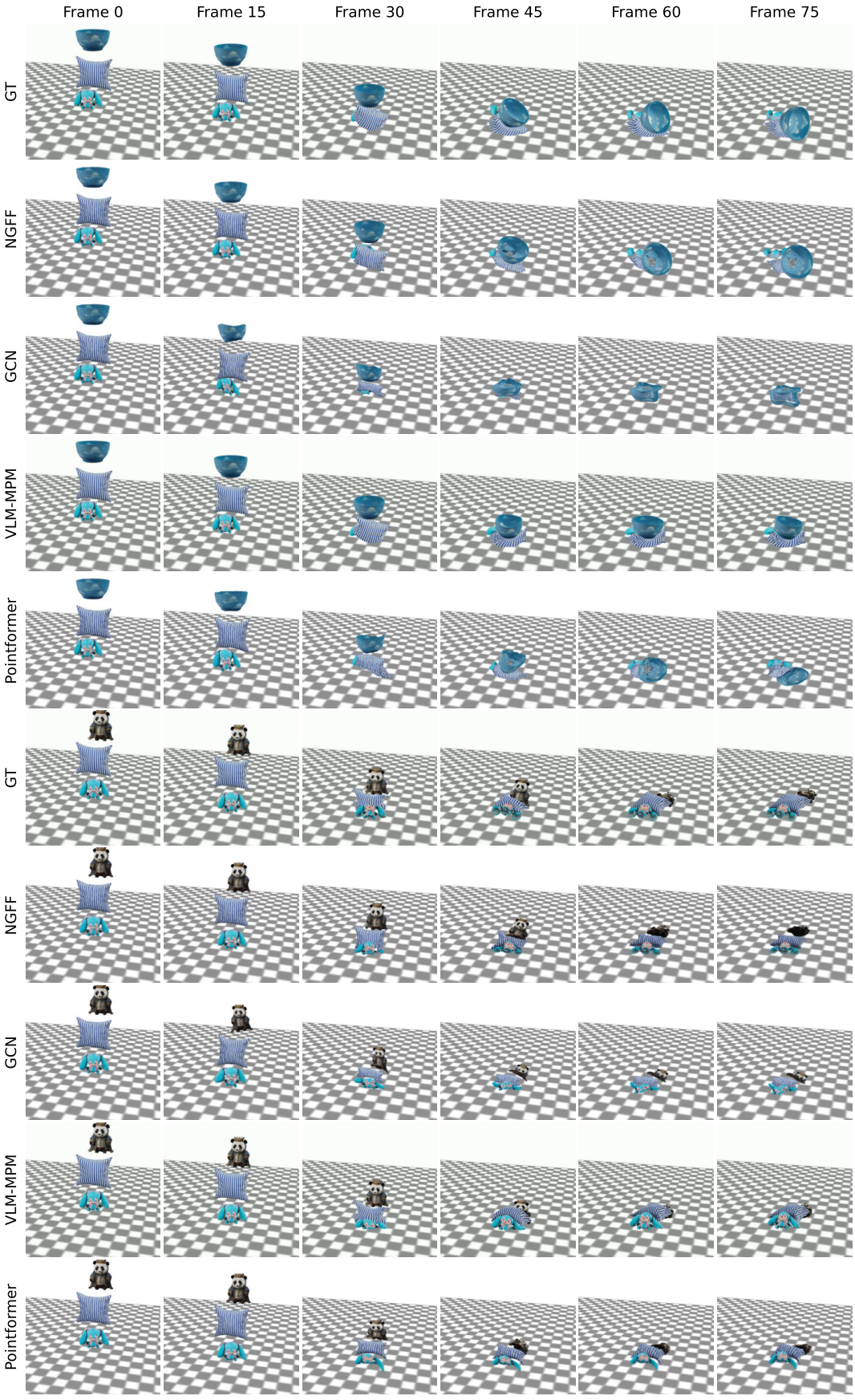


Figure A10: **Dynamic prediction results.**

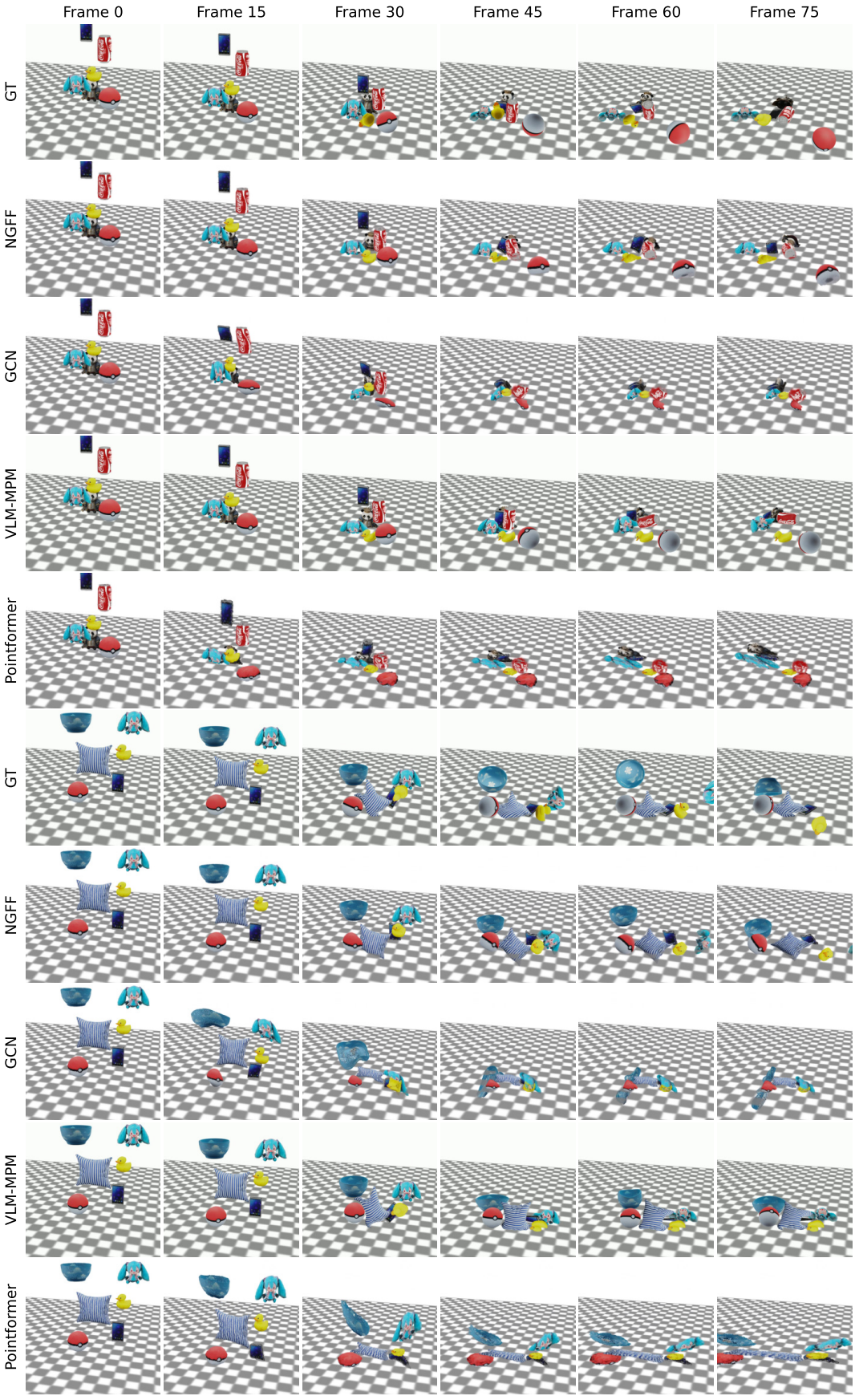


Figure A11: **Dynamic prediction results.**

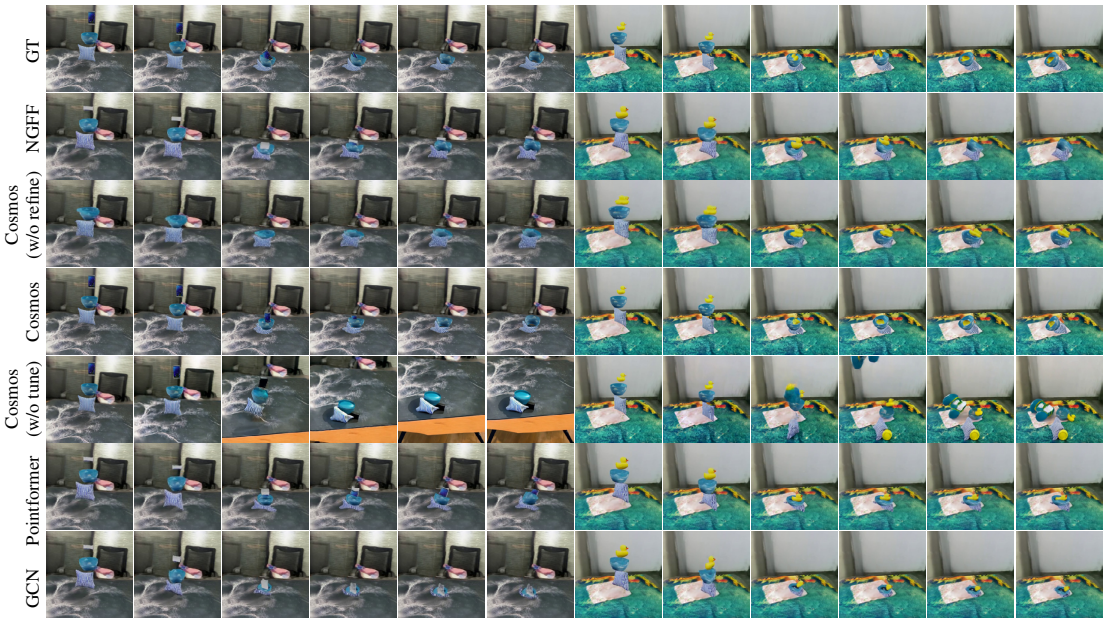


Figure A12: **Video generation results from compositional split.**

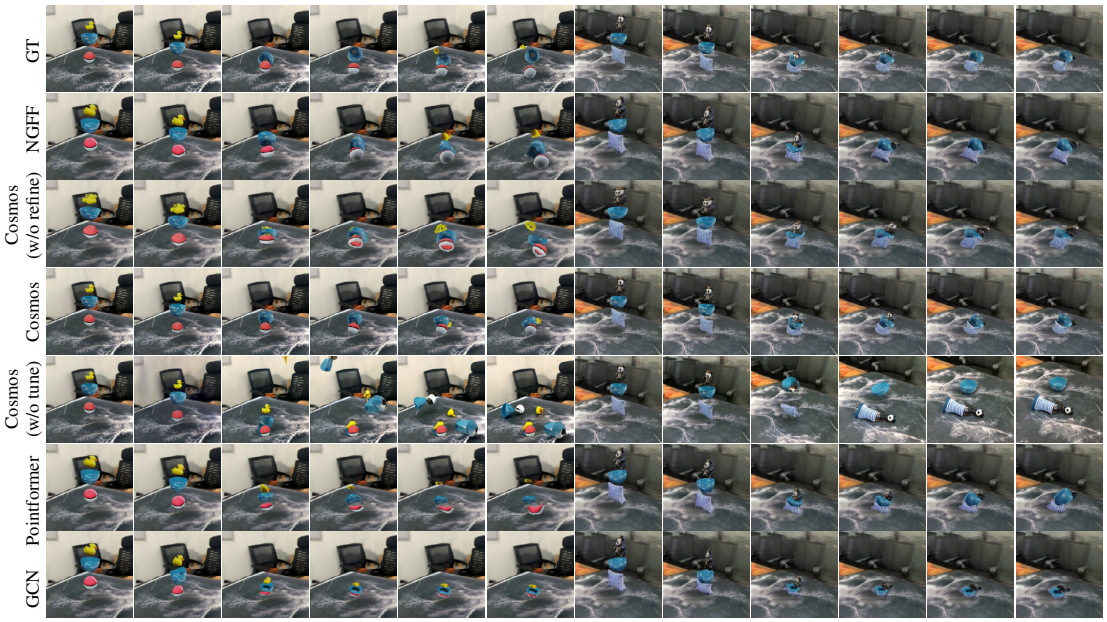


Figure A13: **Video generation results from compositional split.**

### F.2 VIDEO GENERATION

We present additional visualizations of video generation in Figures A12 to A19.

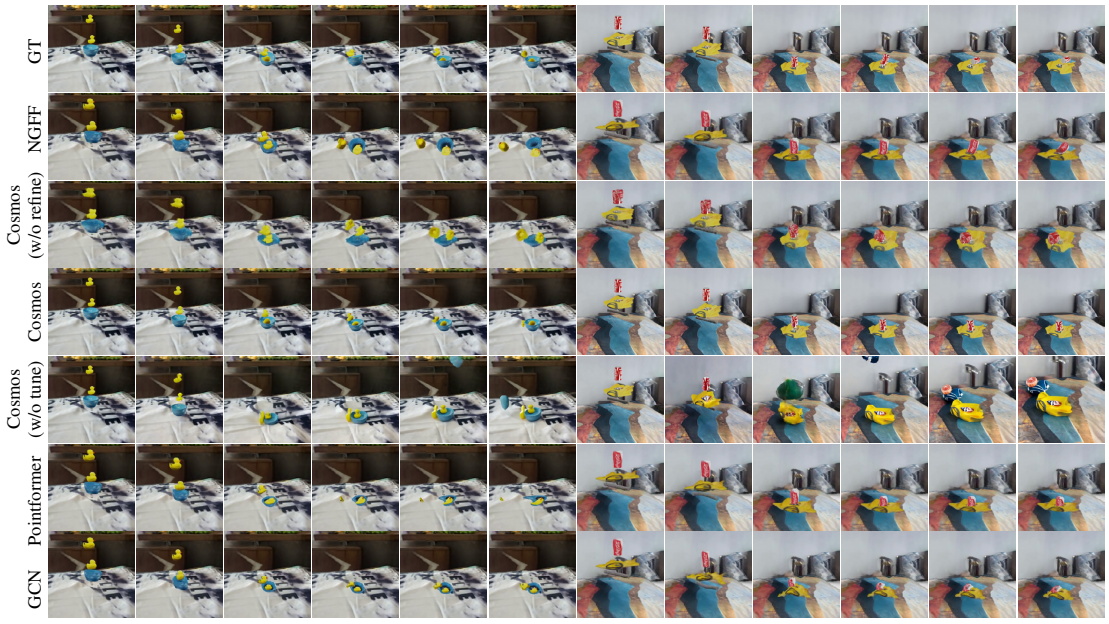


Figure A14: **Video generation results from novel-view split.**

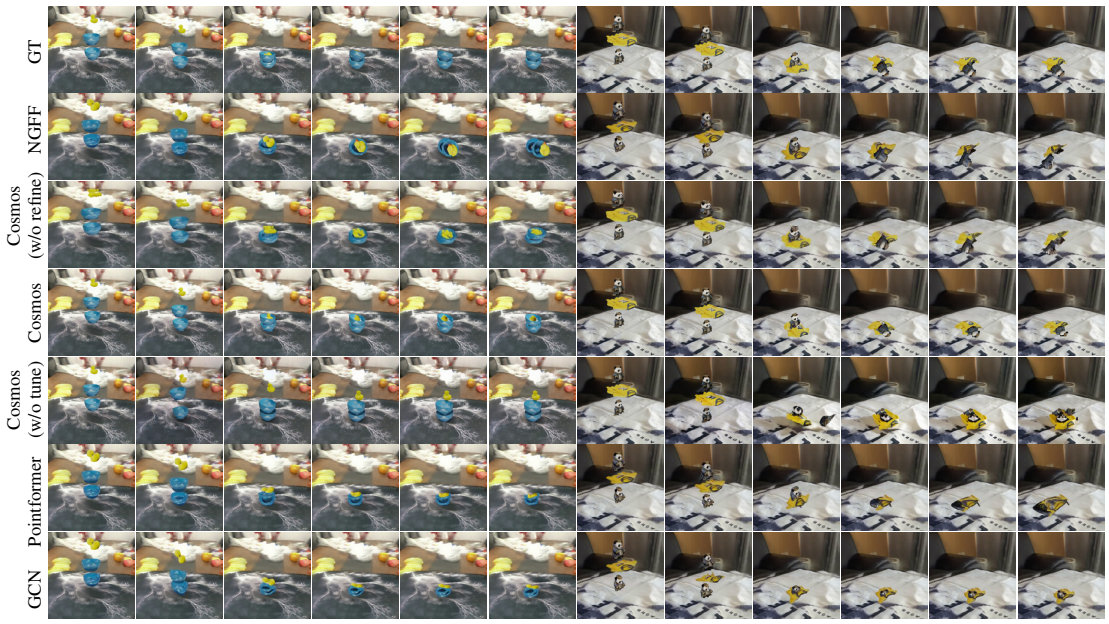


Figure A15: **Video generation results from novel-view split.**

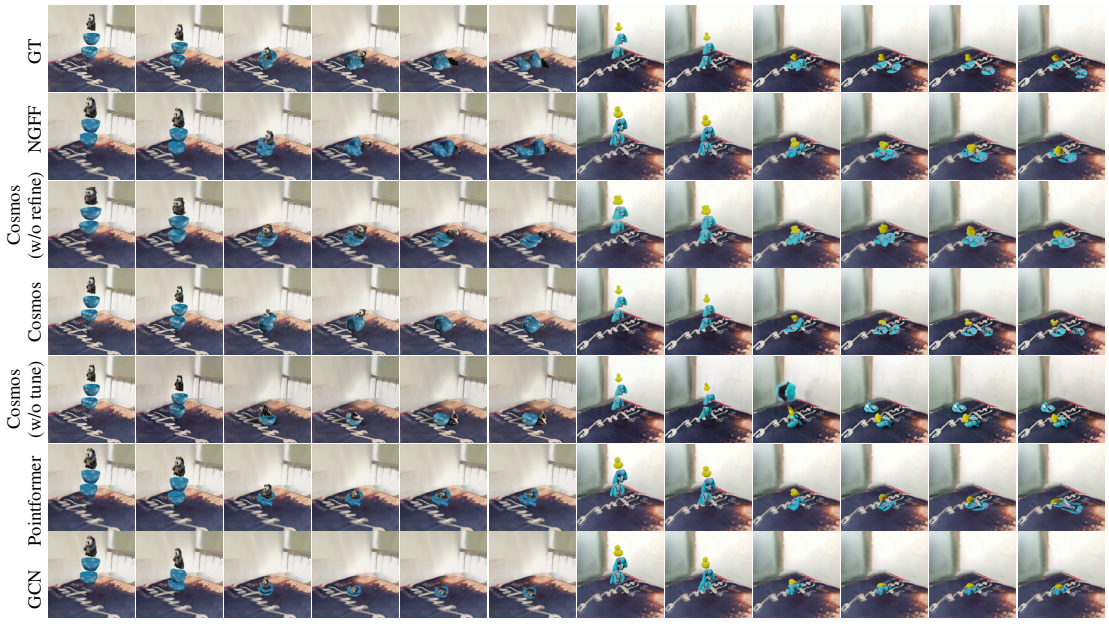


Figure A16: **Video generation results from novel-background split.**


Figure A17: **Video generation results from novel-background split.**

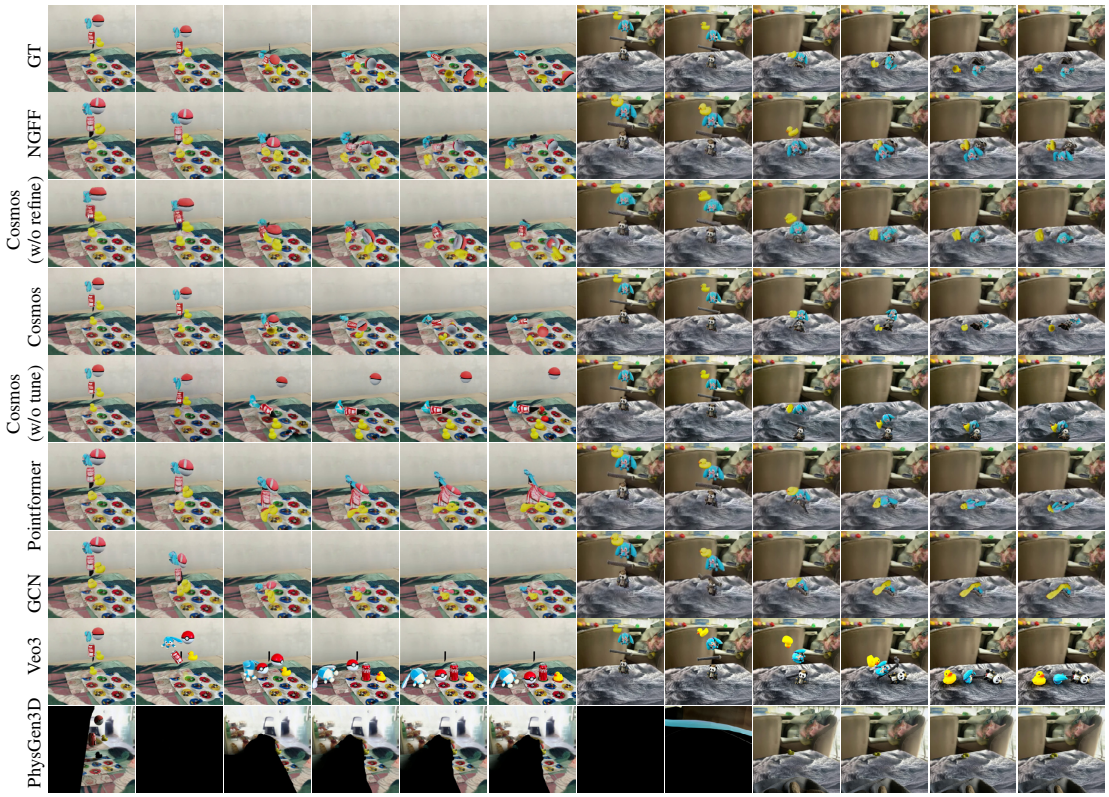


Figure A18: **Video generation results from comprehensive split.**

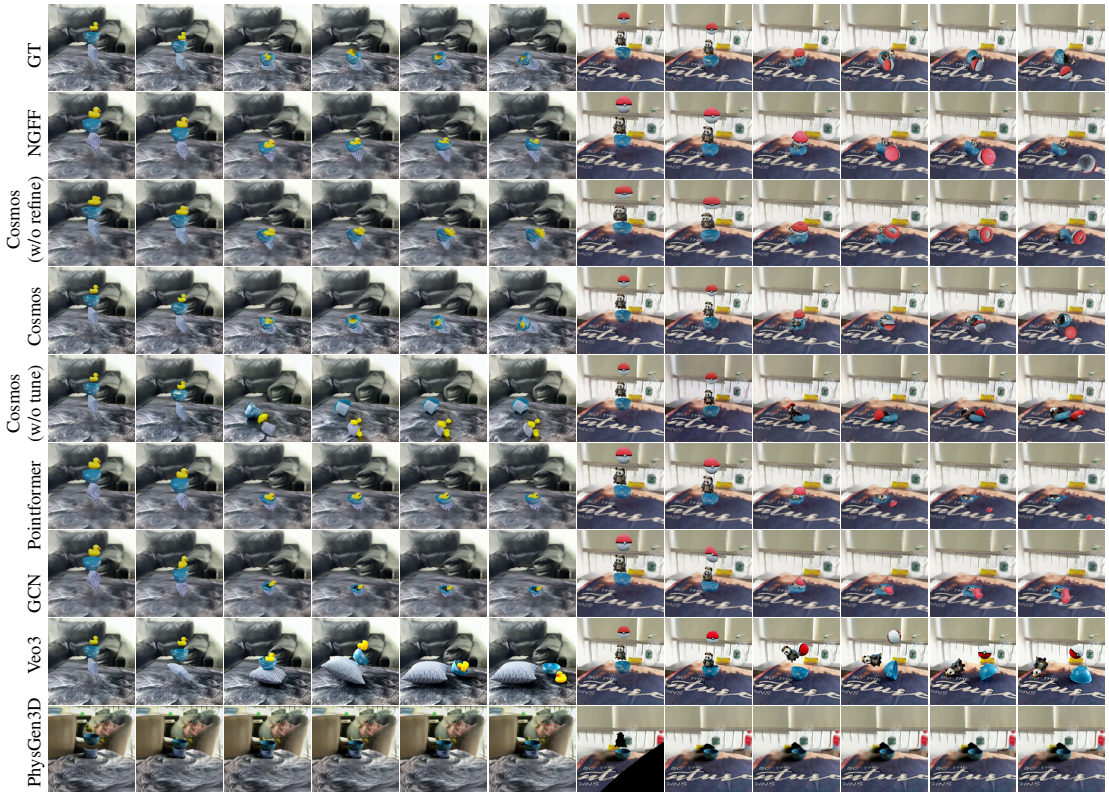


Figure A19: **Video generation results from comprehensive split.**