ANALYTIC DYNAMICS: LEARNING PHYSICS-GROUNDED REPRESENTATION FOR FAST INTRINSIC DYNAMICS INFERENCE FROM MONOCULAR VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inferring intrinsic dynamics from observed motion is essential for intelligent agents to reason about and interact with the physical world, yet remains challenging due to the fundamental gap between visual evidence and intrinsic dynamics. Existing methods either rely on costly per-scene optimization, limiting efficiency and scalability, or directly map visual evidence to intrinsic dynamics, making them prone to learning shortcuts that hinder generalization. To address this challenge, we propose Analytic Dynamics, a feed-forward framework that introduces a physics-grounded dynamics representation to bridge monocular videos and intrinsic dynamics. Specifically, we leverage privileged physical states available in simulation, including positions, displacements, and deformation gradients, which explicitly characterize object dynamics, to learn a structured dynamics representation that is difficult to discover from visual observations alone. By aligning video representations with this learned space, we equip video models with a physics-grounded inductive bias, guiding them to capture dynamics-relevant patterns for intrinsic dynamics inference. To support this research, we develop a dynamics data generation pipeline and benchmark comprising paired physical state trajectories, rendered videos, and ground-truth material labels. Extensive experiments demonstrate that Analytic Dynamics outperforms existing baselines, achieves strong generalization, and requires only 3.83 ms per inference.

1 INTRODUCTION

Embodied agents Duan et al. (2022b); Long et al. (2025) need to reason about how objects respond to physical interactions to make informed decisions. This capability relies on understanding the intrinsic dynamics of objects, including the underlying material models and material parameters. Humans intuitively infer such dynamics from observed motion, enabling them to anticipate future behaviors and interact effectively with their surroundings. Endowing machines with similar intuitive physics Duan et al. (2022a) is a fundamental goal of embodied intelligence Sun & Lu (2025).

With advances in differentiable rendering Mildenhall et al. (2021); Kerbl et al. (2023) and simulation Jiang et al. (2016), a line of work Li et al. (2023); Cai et al. (2024); Lin et al. (2026) formulates intrinsic dynamics inference as an inverse physics problem, optimizing material parameters to match simulated behaviors with multi-view videos. While achieving accurate estimation, these methods require extensive simulation–optimization iterations from scratch for each scene, resulting in high computational cost and limited scalability to unseen objects. Alternatively, another line of work Zhai et al. (2024); Shuai et al. (2025); Chopra et al. (2025) explores prompting large vision-language models (VLMs) to infer intrinsic dynamics from visual observations. While offering efficient and versatile inference, these approaches primarily rely on commonsense knowledge acquired from large-scale corpora, resulting in noisy and coarse-grained predictions.

To explore a better balance between inference efficiency and accuracy, recent works Le et al. (2025); Huang et al. (2026); Das & Manocha (2026) leverage large-scale synthetic 3D asset–dynamics pairs to train feed-forward models. However, static 3D assets only provide appearance and geometry, which are inherently ambiguous for inferring intrinsic dynamics: visually identical objects may correspond to different material models or parameters Huang et al. (2026). In contrast, monocu-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

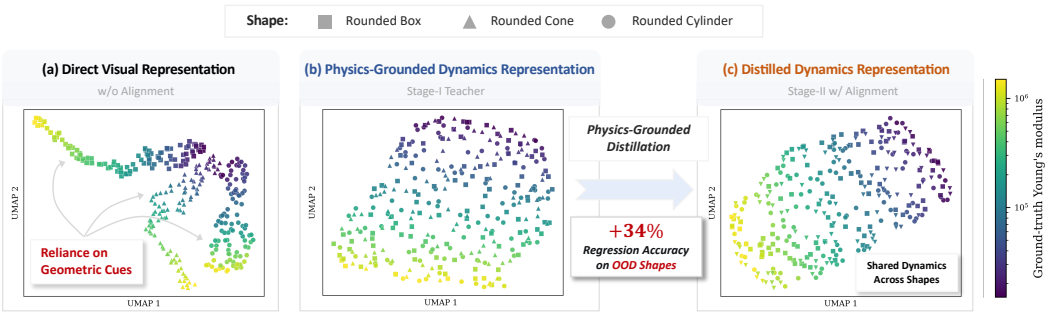

Figure 1: **Physics-grounded distillation reduces shape dependence.** We use UMAP to visualize representations of jelly samples from three training shapes: Box, Cone, and Cylinder. Marker types denote object shapes, while colors indicate the ground-truth Young's modulus E. Without alignment, direct visual representations form three shape-specific branches, revealing strong reliance on geometric cues. In contrast, the Stage I teacher organizes samples into a smooth continuum according to E, forming a shared dynamics structure across geometries. Distillation transfers this organization to Stage II, enabling the visual model to capture dynamics-relevant patterns across shapes and improving E regression accuracy by 34% on unseen shapes.

lar videos capture how objects move and deform under physical interactions, providing behavioral evidence of their intrinsic dynamics. However, this evidence is indirect, as videos capture only 2D projections of physical responses, losing critical 3D information and entangling dynamics cues with factors such as viewpoint, geometry, and appearance. Consequently, without physics-grounded guidance, models can easily exploit spurious correlations rather than learning dynamics-relevant patterns, leading to poor generalization.

To address the aforementioned challenges, we propose *Analytic Dynamics*, a feed-forward framework for efficiently inferring constitutive material models and parameters from monocular videos. Our key insight is to introduce a physics-grounded dynamics representation between visual observations and intrinsic dynamics, providing visual models with a physically meaningful learning target. Specifically, we develop a two-stage privileged dynamics distillation framework. In Stage I, we leverage privileged physical states available in simulation, including position, displacement, and deformation gradient fields, to learn a structured dynamics representation through material model classification and parameter regression. Since these states directly describe physical responses governed by intrinsic dynamics, the learned representation is encouraged to capture dynamics-relevant information, thereby establishing a physics-grounded representation space. In Stage II, we align monocular video representations with the learned space. This alignment transfers the physics-grounded inductive bias derived from privileged physical states to the visual model, enabling more accurate and generalizable inference of material models and parameters. To support this research, we construct an MPM-based benchmark comprising 10,800 dynamics instances across 60 object geometries, covering diverse material models, parameters, and viewpoints. Each instance pairs monocular videos with corresponding physical state trajectories and ground-truth material models and parameters.

Our contributions are summarized as follows:

- We propose a feed-forward framework that bridges visual observations and intrinsic dynamics through an intermediate physics-grounded representation, enabling efficient and effective inference of constitutive material models and parameters from monocular videos.
- We introduce a two-stage privileged dynamics distillation framework that learns a structured representation from privileged physical states and transfers it to the visual model, encouraging dynamics-relevant feature learning for improved generalization.
- We develop a dynamics data generation pipeline and benchmark for intrinsic dynamics inference. Extensive experiments demonstrate superior inference accuracy with a latency of only 3.83 ms, and robust generalization to unseen objects and varying observation lengths. We further conduct a real-world experiment, providing preliminary evidence of the method's applicability to real-world scenarios.

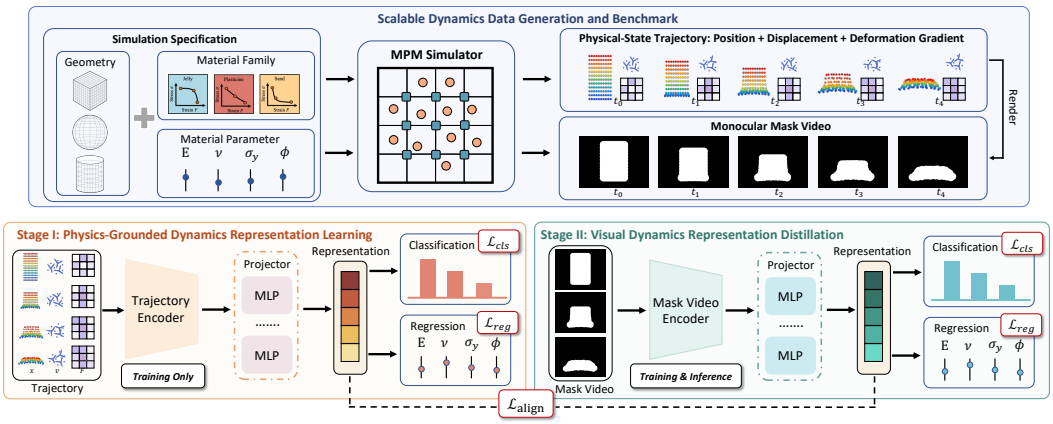

Figure 2: **Overview of *Analytic Dynamics*.** Given sampled geometry, material family, material parameters, an MPM simulator generates paired tracked particle trajectories and synchronized monocular mask videos. The trajectories contain particle positions, displacements, and deformation gradients. In Stage I, a trajectory teacher learns a dynamics-relevant representation from these simulator-only states using classification and parameter-regression supervision. In Stage II, a video student learns the same inference task from mask videos while aligning its representation with the frozen teacher. At test time, only the video student is retained to predict intrinsic dynamics from readily captured monocular videos.

2 Related Work

2.1 Per-Scene Dynamics Optimization

Per-scene inverse-physics methods identify object dynamics from videos by optimizing physical parameters through differentiable rendering and simulation. PAC-NeRF, GIC, Spring-Gaus, and M-PhyGs Li et al. (2023); Cai et al. (2024); Zhong et al. (2024); Wada et al. (2025) operate under prescribed material models, progressively supporting explicit geometry, heterogeneous elasticity, and multi-material objects. NeuMA and MASIV Cao et al. (2024); Zhao et al. (2025b) relax fixed constitutive assumptions through neural constitutive modeling, whereas VisionLaw Lin et al. (2026) searches for interpretable symbolic laws. PhysDreamer, DreamPhysics, Physics3D, PhysFlow, and OmniPhysGS Zhang et al. (2024); Huang et al. (2024); Liu et al. (2024; 2025); Lin et al. (2025b) instead use video diffusion priors to optimize physical properties without captured motion. Vid2Sim combines feed-forward prediction with lightweight per-scene refinement Chen et al. (2025). Despite these advances, optimization-based methods remain computationally expensive and require scene-specific fitting, limiting their scalability.

2.2 Feed-Forward Dynamics Inference

To avoid expensive scene-specific system identification, feed-forward methods directly predict physical properties from visual observations. NeRF2Physics, GaussianProperty, PUGS, and PhysGS Zhai et al. (2024); Xu et al. (2025); Shuai et al. (2025) transfer material semantics and physical priors from pretrained vision-language models (VLMs) to 3D scene representations for zero-shot property estimation; among them, PUGS improves property propagation over 3D Gaussians, while PhysGS Chopra et al. (2025) explicitly models prediction uncertainty. Phys4DGen and PhysSplat Lin et al. (2025a); Zhao et al. (2025a) extend this strategy to physics-driven 4D generation through explicit simulation. While efficient and broadly applicable, these methods are constrained by VLM commonsense priors and may produce coarse estimates. Another line, exemplified by PIXIE Le et al. (2025), learns feed-forward material inference from large-scale annotated 3D assets. Building on this paradigm, SLAT-Phys Das & Manocha (2026) enables efficient single-image prediction, while UniPixie Huang et al. (2026) extends deterministic estimation to controllable material distributions. However, these methods may exploit appearance and geometry shortcuts rather than learn generalizable physical regularities.

3 METHODOLOGY

3.1 PROBLEM FORMULATION AND OVERVIEW

For a dynamics instance i, let $\mathcal{V}_i^{(r)}$ denote a monocular mask video of a moving object observed from viewpoint r. Our objective is to learn a feed-forward video model $\mathcal{M}_\Phi$ that predicts the material family $y_i \in \mathcal{C}$ and its corresponding material parameters $\boldsymbol{\theta}_i$:

$$\left(\hat{y}_i, \hat{\boldsymbol{\theta}}_i\right) = \mathcal{M}_\Phi\left(\mathcal{V}_i^{(r)}\right). \tag{1}$$

Directly learning this mapping from videos is challenging, as a monocular video provides only an indirect 2D observation of the physical response. In simulation, each dynamics instance additionally provides a tracked physical state trajectory $\mathcal{S}_i$, offering a more direct observation of the underlying physical response. We therefore use these training-only physical states to guide the learning of $\mathcal{M}_\Phi$, while relying only on video inputs at inference.

To this end, we propose a two-stage privileged dynamics distillation framework. We first develop a data generation pipeline that produces paired physical state trajectories and monocular mask videos for each simulated dynamics instance. In Stage I, a trajectory teacher learns a physics-grounded representation predictive of the material family and its parameters. In Stage II, this representation is distilled into a video student through representation alignment and direct task supervision. The overall framework of *Analytic Dynamics* is illustrated in Fig. 2.

3.2 DYNAMICS DATA GENERATION AND BENCHMARK

To support privileged dynamics distillation, we construct paired physical state trajectories and monocular videos from the same simulated dynamics instance. Each instance is defined by an object geometry, a material family, and material parameters. Given these configurations, we use the Material Point Method (MPM) Jiang et al. (2016); Ma et al. (2023) to simulate the object response, producing temporally evolving particle states together with a synchronized deforming surface.

To construct the physical state trajectory, we perform farthest-point sampling (FPS) only on the initial particle set and track the same particles throughout the simulation. This preserves particle identity over time and converts independent state snapshots into Lagrangian trajectories. For the k-th tracked particle, we compute its frame displacement as

$$\Delta\mathbf{x}_{i,t,k} = \begin{cases} 0, & t = 0, \\ \mathbf{x}_{i,t,k} - \mathbf{x}_{i,t-1,k}, & t \geq 1. \end{cases} \tag{2}$$

To encourage the model to focus on relative motion patterns rather than absolute displacement magnitudes, we normalize the displacements using their median magnitude over all tracked particles and non-initial frames:

$$s_i = \text{median}_{t=1,\ldots,T-1}\left(\|\Delta\mathbf{x}_{i,t,k}\|_2\right), \tag{3}$$

$$\mathbf{v}_{i,t,k} = \frac{\Delta\mathbf{x}_{i,t,k}}{s_i}. \tag{4}$$

The resulting physical state trajectory $\mathcal{S}_i$ consists of tracked particle positions $\mathbf{x}_i$, normalized frame displacements $\mathbf{v}_i$, and deformation gradients $\mathbf{F}_i$.

To generate the corresponding visual observations, we bind each surface-mesh vertex to nearby material particles in the initial configuration. During simulation, the mesh vertices are advected by interpolating the motion of their associated particles, yielding a deforming surface synchronized with the particle trajectory. We render this surface from multiple viewpoints into a monocular mask video $\{\mathcal{V}_i^r\}_{r=1}^R$.

Each simulated instance therefore provides one physical state trajectory, multiple synchronized monocular videos, and their shared material annotations:

$$\mathcal{D}_i = \left(\mathcal{S}_i, \{\mathcal{V}_i^{(r)}\}_{r=1}^R, y_i, \theta_i\right). \tag{5}$$

The trajectory is used to train the Stage-I teacher, while the paired videos support Stage-II representation distillation and task supervision.

3.3 PRIVILEGED DYNAMICS REPRESENTATION LEARNING

Stage I aims to learn a teacher representation from privileged physical state trajectories. Unlike monocular videos, these trajectories directly capture 3D particle motion and deformation, encouraging the model to learn dynamics-relevant patterns.

Given a physical state trajectory $\mathcal{S}_i$, P4Transformer Fan et al. (2021) models local and long-range spatiotemporal interactions among the tracked particles. A projector then maps the pooled trajectory feature to a teacher representation $\mathbf{z}_i^T \in \mathbb{R}^d$, which defines the representation space used for subsequent visual distillation.

To make this representation predictive of intrinsic dynamics, we train the teacher with material-family classification and parameter regression. Classification distinguishes discrete material families, while regression preserves continuous parameter variations within each family. Since different material families involve distinct parameter sets and numerical scales, we normalize each parameter and compute RegL1 as the mean L1 error over the parameters defined for the ground-truth family:

$$\mathcal{L}_{\mathrm{RegL1}} = \frac{\sum_{i=1}^{B} \sum_{j=1}^{d_{\max}} m_{i,j} \left| \hat{\overline{\theta}}_{i,j} - \overline{\theta}_{i,j} \right|}{\sum_{i=1}^{B} \sum_{j=1}^{d_{\max}} m_{i,j}}. \tag{6}$$

Here, $\overline{\theta}_{i,j}$ and $\hat{\overline{\theta}}_{i,j}$ denote the normalized ground-truth and predicted parameters, respectively, and $m_{i,j}$ indicates whether the j-th parameter is defined for sample i. Parameter normalization places different physical quantities on a comparable scale, while the validity mask enables unified optimization across material families. The classification branch is trained with standard cross-entropy $\mathcal{L}_{cls}$. The overall Stage-I objective is:

$$\mathcal{L}_{\mathrm{Stage-I}} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{reg}\mathcal{L}_{\mathrm{RegL1}} \tag{7}$$

By combining direct 3D physical observations with material supervision, Stage I learns a physics-grounded representation predictive of the material family and its parameters. After training, the trajectory encoder and projector are frozen, and $\mathbf{z}^T$ serves as the representation target for the paired monocular videos in Stage II.

3.4 VISUAL DYNAMICS REPRESENTATION DISTILLATION

Stage II distills the physics-grounded representation learned in Stage I into a deployable video model. Given a monocular mask video $\mathcal{V}_i^{(r)}$, a UniFormer encoder Li et al. (2022) followed by a projector produces a visual representation $\mathbf{z}_{i,r}^V \in \mathbb{R}^d$. We use mask videos to retain silhouette motion and deformation while removing variations in texture and color. Meanwhile, the frozen teacher extracts the target representation $\mathbf{z}_i^T$ from the paired physical state trajectory $\mathcal{S}_i$. Different views of the same dynamics instance share the same teacher target, encouraging view-consistent visual representations.

We align the visual and teacher representations using cosine and ℓ_1 distances:

$$\mathcal{L}_{\mathrm{align}} = \frac{1}{B} \sum_{i=1}^{B} \left[1 - \left(\widetilde{\mathbf{z}}_{i,r}^V \right)^\top \widetilde{\mathbf{z}}_i^T + \alpha \left\| \mathbf{z}_{i,r}^V - \mathbf{z}_i^T \right\|_1 \right], \tag{8}$$

where $\widetilde{\mathbf{z}} = \mathbf{z}/(\|\mathbf{z}\|_2 + \varepsilon)$, and α balances the two terms. The cosine term aligns the overall feature direction, while the ℓ_1 term reduces their element-wise discrepancy. The teacher remains frozen throughout Stage II, so the alignment loss updates only the video branch. The video model uses the same material-family classification and parameter-regression design as the teacher, supervised by cross-entropy and RegL1, respectively. The overall Stage-II objective is

$$\mathcal{L}_{\mathrm{Stage\text{-}II}} = \lambda_{\mathrm{align}}\mathcal{L}_{\mathrm{align}} + \lambda_{\mathrm{cls}}\mathcal{L}_{\mathrm{cls}} + \lambda_{\mathrm{reg}}\mathcal{L}_{\mathrm{RegL1}}. \tag{9}$$

Representation alignment guides the video model toward features learned from direct 3D motion and deformation, while task supervision keeps the visual representation predictive of material families and parameters. At inference, the teacher is discarded, and the video model directly predicts intrinsic dynamics from a monocular mask video.

Method	Jelly			Plasticine				Sand	
	Acc. ↑	E ↓	ν ↓	*Acc.* ↑	E ↓	ν ↓	$Yield$ ↓	*Acc.* ↑	ϕ ↓
Per-scene Optimization	-	0.179	0.104	-	0.752	0.101	**0.065**	-	3.209
VLM baseline	0.633	2.163	0.134	0.367	2.007	0.052	1.115	0.000	-
Video-LLM baseline	0.847	1.376	0.118	0.833	0.756	0.045	0.785	0.327	7.803
Ours (w/o alignment)	0.997	0.265	0.043	1.000	0.223	0.024	0.115	0.943	1.991
Ours (alignment only)	1.000	0.188	**0.036**	1.000	0.146	0.021	0.106	**0.973**	1.927
Ours (Full)	**1.000**	**0.175**	0.037	**1.000**	**0.141**	**0.021**	0.101	0.970	**1.777**

Table 1: **Quantitative comparison on objects unseen during training.** We report material classification accuracy (↑) and material-specific parameter errors (↓) for elasticity, plasticity, and sand. Young's modulus E and yield stress σ_y are evaluated using log-MAE, while Poisson's ratio ν and friction angle ϕ are evaluated using MAE.

4 Experiments

4.1 Experimental Setup

4.1.1 Implementation Details

In Stage I, the teacher takes 25-frame trajectories of 1,024 tracked particles, comprising positions $\mathbf{x}$, displacements $\mathbf{v}$, and deformation gradients $\mathbf{F}$. It employs a P4Transformer Fan et al. (2021) encoder, followed by a projector, a material classifier, and material-specific regression heads. In Stage II, the student processes 25-frame, 128×128, single-channel mask videos using a UniFormer-XXS Li et al. (2022) encoder, followed by a corresponding projector, classifier, and material-specific regression heads. Both stages use a 256-dimensional latent representation. The Stage I teacher is trained for 300 epochs and subsequently frozen during Stage II, where the student is trained for 20 epochs. Both stages use AdamW with an initial learning rate of 10^{-4}, weight decay of 10^{-4}, and cosine learning rate decay to 10^{-6}. The batch sizes are 16 and 32 for Stage I and Stage II, respectively. The Stage II loss weights are set to $\lambda_{\text{cls}} = 1.0$, $\lambda_{\text{reg}} = 0.2$, and $\lambda_{\text{align}} = 0.1$. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

4.1.2 Baselines

We compare against three representative baselines spanning two paradigms. First, following recent studies Zhai et al. (2024); Shuai et al. (2025) that leverage multimodal large language models for physical property reasoning, we construct a VLM baseline and a Video-LLM baseline. They perform feed-forward inference of material models and parameters from an input image and video, respectively, using the material commonsense priors encoded in pretrained models. The prompts are provided in the Appendix F. Second, we consider per-scene optimization Cai et al. (2024); Li et al. (2023), which iteratively estimates the material parameters by matching the simulated dynamics to the observed video. We adopt the implementation of VisionLaw Lin et al. (2026) for this baseline.

4.1.3 Datasets and Metrics

We construct an MPM-based dataset comprising 10,800 dynamics instances across three material families: `jelly`, `plasticine`, and `sand`, with their constitutive formulations detailed in Appendix D. We vary Young's modulus (E) and Poisson's ratio (ν) for `jelly`, additionally varying yield stress (σ_y) for `plasticine`, and friction angle (ϕ) for `sand`. The benchmark covers 60 object shapes, including hand-crafted primitives and common real-world objects from OmniObject3D Wu et al. (2023), evenly split into 30 training and 30 test shapes. For each dynamics instance, we sample an object shape and material parameters, and generate synchronized physical state trajectories and monocular mask videos from multiple views. To mitigate viewpoint bias, we randomly sample a fixed number of views per instance during training.

We report material classification accuracy and material-specific parameter regression errors. Young's modulus E and yield stress σ_y are evaluated using log-MAE, while Poisson's ratio ν and the friction angle ϕ are evaluated using MAE. For an aggregate evaluation across material families and parameters, we report RegL1 (Eq. 6), where lower values indicate more accurate parameter

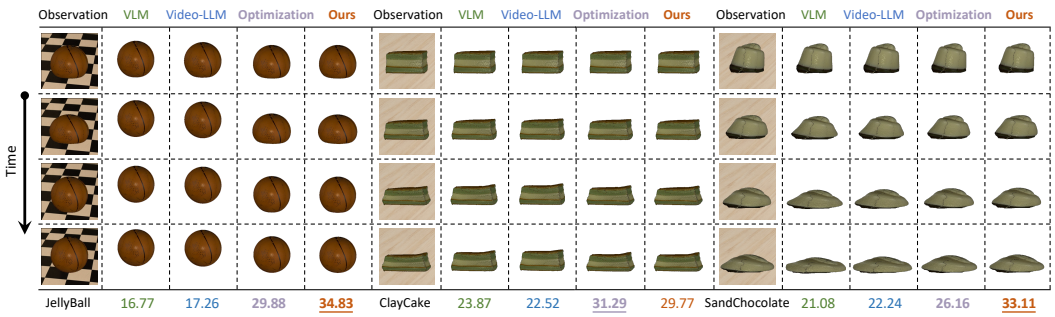

Figure 3: **Visual dynamics reconstruction on unseen objects.** We re-simulate each object using the intrinsic dynamics inferred by different methods. PSNR (↑) between the reconstructed and observed frames is reported below each sequence.

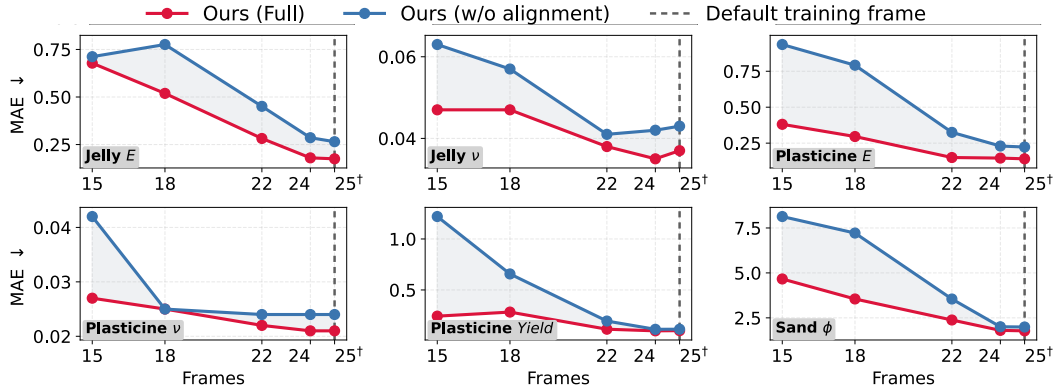

Figure 4: **Temporal generalization across varying observation lengths.** Parameter estimation errors of Analytic Dynamics with and without privileged-state alignment across six material parameters. Both models are trained on 25-frame sequences and evaluated using 15–25 input frames. We report log-MAE for Young's modulus and yield stress, and MAE for Poisson's ratio and friction angle. The vertical dashed line indicates the default training length. Our full model maintains lower estimation errors across nearly all settings.

estimation. Finally, we re-simulate each object using its inferred material parameters and evaluate visual dynamics reconstruction using PSNR between the rendered and observed frames.

4.2 Comparison on Unseen Objects

We first evaluate material classification and parameter estimation on unseen objects. As shown in Tab. 1, *Analytic Dynamics* achieves the best overall performance. The VLM baseline fails to classify sand, highlighting the ambiguity of inferring intrinsic dynamics from static images. Although Video-LLM incorporates temporal information, it still exhibits substantial parameter errors. Per-scene optimization remains competitive on several metrics but incurs high computational costs. Its rendering-based objective also entangles errors from simulation, 3D representation, and rendering, making accurate parameter estimation challenging. Meanwhile, under large deformations, the degradation and rendering artifacts of 3DGS further amplify this issue.

The results across model variants further validate privileged dynamics distillation. Alignment alone outperforms direct label supervision, indicating that the dynamics representation learned in Stage I provides the video model with a more physically meaningful learning target than material labels alone. Adding task supervision produces further gains: alignment captures dynamics-relevant information, while task supervision makes it predictive of material classes and parameters.

Fig. 3 further evaluates whether the inferred dynamics reproduce the observations. Using the predicted material models and parameters, we re-simulate each object with fixed geometry and initial conditions. Our method more accurately reproduces the deformation of JellyBall and the flow and

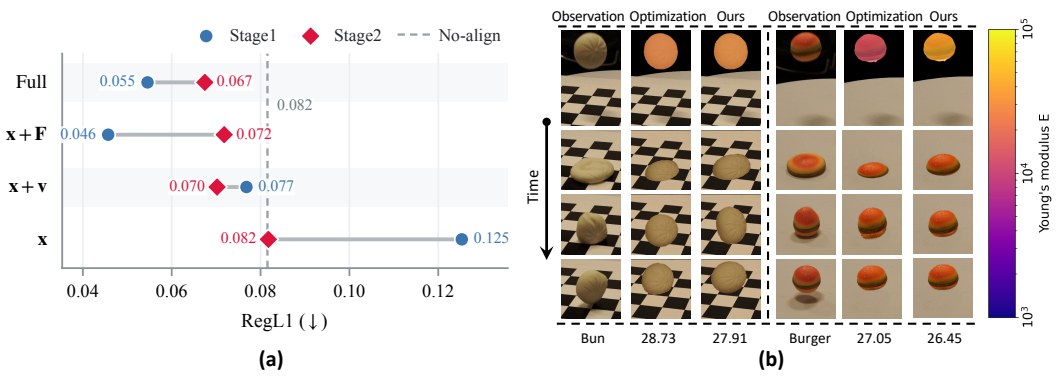

Figure 5: **(a) Ablation of privileged physical states.** We compare position x, displacement v, and deformation gradient $\mathbf{F}$ as inputs to the Stage I teacher, and report the RegL1 (↓) errors of both the teacher and its corresponding Stage II student. Gray lines connect each teacher–student pair, while the dashed line denotes the Stage II baseline without distillation. **(b) Real-world dynamics reconstruction.** The mask color denotes the predicted Young's modulus (E), following the right color bar. Without any fine-tuning, our method achieves performance comparable to per-scene optimization and reconstructs dynamics close to the observations.

accumulation of SandChocolate. Although per-scene optimization achieves higher reconstruction accuracy on ClayCake, our method produces visually comparable results without test-time optimization. Overall, these results demonstrate that *Analytic Dynamics* effectively infers material models and parameters from monocular videos in 3.83 ms and generalizes to unseen objects.

4.3 Temporal Generalization

We further evaluate the temporal generalization of our model across varying observation lengths. Both models are trained on 25-frame sequences and evaluated using 15–25 input frames. As shown in Fig. 4, estimation errors generally increase for both models as the observation length decreases, due to the reduced temporal information. However, the model without alignment exhibits greater performance degradation, whereas our method consistently achieves lower errors across nearly all settings. These results demonstrate that *Analytic Dynamics* not only improves estimation accuracy with full observations but also enhances robustness to shorter sequences, suggesting its ability to capture more stable dynamics-relevant cues from limited temporal observations.

4.4 Effect of Privileged Physical States

To study the effects of privileged physical states on representation learning and cross-modal distillation, we train Stage I teachers with different combinations of position ($\mathbf{x}$), displacement ($\mathbf{v}$), and deformation gradient ($\mathbf{F}$), and distill their representations into Stage II video models. Fig. 5(a) reports the parameter regression errors (RegL1) for both stages. The results reveal that dynamics discriminability does not necessarily translate into visual transferability. Using position alone yields the highest Stage I error, as position trajectories primarily preserve object geometry. Adding the deformation gradient reduces the Stage I error to 0.046, yielding the most accurate teacher. However, the full state ($\mathbf{x} + \mathbf{v} + \mathbf{F}$) achieves better Stage II performance than $\mathbf{x} + \mathbf{F}$, despite its higher Stage I error. This suggests that effective distillation requires representations that are both dynamics-discriminative and visually transferable. Displacement provides observable motion cues, while the deformation gradient captures fine-grained material responses. Combining them produces the best Stage II performance. We therefore adopt the full state in all experiments.

4.5 Real-World Generalization

We further test whether a model trained only on synthetic data can directly generalize to real-world scenes. We evaluate on two real scenes, Bun and Burger, from Spring-Gaus Zhong et al. (2024). Since the ground-truth material parameters are unknown, we rely on PSNR as an indirect measure of the accuracy of material parameter inference. As shown in Fig. 5(b), our model predicts material

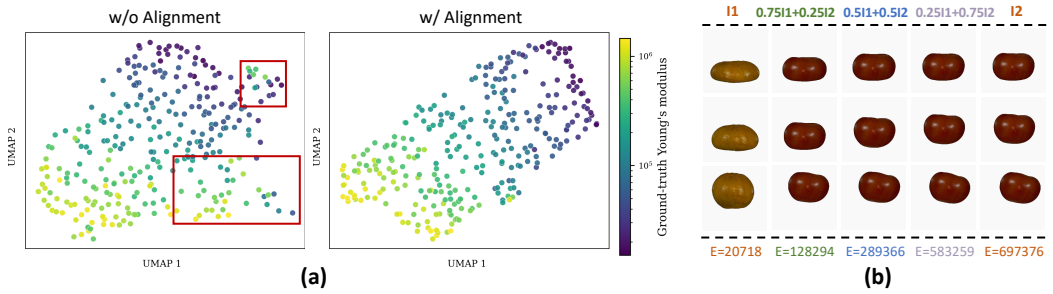

Figure 6: **Analysis of learned dynamics representations on unseen objects.** (a) UMAP visualizations of Stage II representations without and with alignment, colored by the ground-truth Young's modulus. (b) Linear interpolation between two jelly representations from different objects with different stiffnesses. The top row shows the interpolation coefficients, and the bottom row shows the Young's moduli decoded from the interpolated features.

parameters without any fine-tuning on real videos, achieving PSNR values only 0.82 dB and 0.60 dB below per-scene optimization on Bun and Burger, respectively. The simulated deformations also closely match the real observations. These results show that the dynamics learned from synthetic data can transfer to real observations, providing initial evidence of the framework's applicability to real-world scenes. The remaining performance gap may stem from discrepancies between simulated and real-world physics. Our simulator employs a limited set of idealized constitutive models Chaves (2013), whereas real objects may exhibit damping, viscoelasticity, material heterogeneity, and complex contact behavior. Per-scene optimization can partially compensate for these unmodeled effects by iteratively fitting parameters to each video, although the resulting parameters may not reflect the true material properties. In contrast, our model performs feed-forward inference without scene-specific refinement. Incorporating richer constitutive models and more accurate physical modeling may further improve real-world generalization.

4.6 Analysis of Learned Dynamics Representations

To examine whether distillation reduces geometric dependence and promotes dynamics-aware representations, we visualize the representation space on unseen objects. As shown in Fig. 6(a), the unaligned representation exhibits local mixing, with samples of similar Young's modulus scattered across the space and samples of different stiffness appearing nearby. In contrast, alignment produces a smoother organization by Young's modulus across unseen objects, suggesting reduced reliance on object-specific geometric cues. Fig. 6(b) further examines the continuity of the learned representation. We linearly interpolate between the representations of two unseen jelly samples with different stiffnesses. As the interpolation coefficient changes, the predicted Young's modulus increases monotonically from $20,718$ to $697,376$, while the simulated deformation transitions from large to small. Together, these results suggest that privileged dynamics distillation encourages a continuous, dynamics-organized visual representation that generalizes across unseen shapes.

5 Conclusion

We presented *Analytic Dynamics*, a feed-forward framework for efficiently inferring constitutive material models and parameters from monocular videos. Our key idea is to exploit privileged physical states available in simulation to learn a physics-grounded dynamics representation and then distill it into a video model. By providing the visual branch with a learning target derived from direct physical responses, our framework encourages video representations to capture dynamics-relevant patterns. To enable systematic training and evaluation, we further developed an MPM-based data generation pipeline and benchmark that pair monocular mask videos with tracked physical state trajectories and material annotations. Extensive experiments demonstrate superior inference accuracy, robust generalization to unseen objects and varying observation lengths, and effective transfer to real-world settings, with an inference time of only 3.83 ms. Nevertheless, the current framework is trained on a limited set of idealized constitutive models. Extending it to richer material behaviors and interactions is a promising direction for future work.

AI USE STATEMENT

Large language models (LLMs) were utilized in this work to improve the fluency and clarity of the manuscript. Their application was specifically focused on detailed proofreading to correct spelling errors and ensure grammatical accuracy, as well as refining sentence structures to enhance the readability and logical flow of the paper. It is crucial to note that all scientific contributions, including the core concepts, experimental design, data analysis, and conclusions, were entirely conceived and written by the authors. The LLMs were employed solely as a writing assistance tool and did not contribute to the conceptualization or analysis of the study.

ETHICS STATEMENT

This research adheres to the ethical guidelines outlined by ICLR. We confirm that no human subjects were involved in this study, and all datasets used have been properly sourced and are publicly available. Our methods have been designed with fairness and transparency in mind, ensuring no biases are introduced in the analysis. Privacy and security of data have been prioritized throughout the research, and we comply with all applicable legal regulations. No conflicts of interest or sponsorships have influenced the research outcomes. We are committed to upholding research integrity and have followed appropriate ethical practices throughout the study.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our work. We provide the source code and implementation details in the supplementary material, with an anonymized repository available at https://anonymous.4open.science/r/AnalyticDynamics-8336. Additionally, a detailed description of the experimental setup and datasets is provided in the appendix. We encourage reviewers and readers to refer to these materials for complete reproducibility.

REFERENCES

Junhao Cai, Yuji Yang, Weihao Yuan, Yisheng He, Zilong Dong, Liefeng Bo, Hui Cheng, and Qifeng Chen. Gic: Gaussian-informed continuum for physical property identification and simulation. *Advances in Neural Information Processing Systems*, 37:75035–75063, 2024.

Junyi Cao, Shanyan Guan, Yanhao Ge, Wei Li, Xiaokang Yang, and Chao Ma. Neuma: Neural material adaptor for visual grounding of intrinsic dynamics. *Advances in Neural Information Processing Systems*, 37:65643–65669, 2024.

Eduardo WV Chaves. *Notes on continuum mechanics*. Springer Science & Business Media, 2013.

Chuhao Chen, Zhiyang Dou, Chen Wang, Yiming Huang, Anjun Chen, Qiao Feng, Jiatao Gu, and Lingjie Liu. Vid2sim: Generalizable, video-based reconstruction of appearance, geometry and physics for mesh-free simulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26545–26555, 2025.

Samarth Chopra, Jing Liang, Gershom Seneviratne, and Dinesh Manocha. Physgs: Bayesian-inferred gaussian splatting for physical property estimation. *arXiv preprint arXiv:2511.18570*, 2025.

Rocktim Jyoti Das and Dinesh Manocha. Slat-phys: Fast material property field prediction from structured 3d latents. *arXiv preprint arXiv:2603.23973*, 2026.

Jiafei Duan, Arijit Dasgupta, Jason Fischer, and Cheston Tan. A survey on machine learning approaches for modelling intuitive physics. *arXiv preprint arXiv:2202.06481*, 2022a.

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022b.

Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14204–14213, 2021.

Qilin Huang, Quynh Anh Huynh, Long Le, Chen Wang, Chuhao Chen, Ryan Lucas, Eric Eaton, and Lingjie Liu. Unipixie: Unified and probabilistic 3d physics learning via flow matching. *arXiv preprint arXiv:2606.05399*, 2026.

Tianyu Huang, Haoze Zhang, Yihan Zeng, Zhilu Zhang, Hui Li, Wangmeng Zuo, and Rynson WH Lau. Dreamphysics: Learning physics-based 3d dynamics with video diffusion priors. *arXiv preprint arXiv:2406.01476*, 2024.

Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. The material point method for simulating continuum materials. In *Acm siggraph 2016 courses*, pp. 1–52. 2016.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):139–1, 2023.

Long Le, Ryan Lucas, Chen Wang, Chuhao Chen, Dinesh Jayaraman, Eric Eaton, and Lingjie Liu. Pixie: Fast and generalizable supervised learning of 3d physics from pixels. *arXiv preprint arXiv:2508.17437*, 2025.

Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676*, 2022.

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. *arXiv preprint arXiv:2303.05512*, 2023.

Jiajing Lin, Zhenzhong Wang, Dejun Xu, Shu Jiang, Yunpeng Gong, and Min Jiang. Phys4dgen: Physics-compliant 4d generation with multi-material composition perception. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10398–10407, 2025a.

Jiajing Lin, Shu Jiang, Qingyuan Zeng, Zhenzhong Wang, and Min Jiang. Visionlaw: Inferring interpretable intrinsic dynamics from visual observations via bilevel optimization. In *International Conference on Learning Representations*, volume 2026, pp. 96128–96165, 2026.

Yuchen Lin, Chenguo Lin, Jianjin Xu, and Yadong Mu. Omniphysgs: 3d constitutive gaussians for general physics-based dynamics generation. *arXiv preprint arXiv:2501.18982*, 2025b.

Fangfu Liu, Hanyang Wang, Shunyu Yao, Shengjun Zhang, Jie Zhou, and Yueqi Duan. Physics3d: Learning physical properties of 3d gaussians via video diffusion. *arXiv preprint arXiv:2406.04338*, 2024.

Zhuoman Liu, Weicai Ye, Yan Luximon, Pengfei Wan, and Di Zhang. Unleashing the potential of multi-modal foundation models and video diffusion for 4d dynamic physical scene simulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11016–11025, 2025.

Xiaoxiao Long, Qingrui Zhao, Kaiwen Zhang, Zihao Zhang, Dingrui Wang, Yumeng Liu, Zhengjie Shu, Yi Lu, Shouzheng Wang, Xinzhe Wei, et al. A survey: Learning embodied intelligence from physical simulators and world models. *arXiv preprint arXiv:2507.00917*, 2025.

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*, pp. 23279–23300. PMLR, 2023.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Yinghao Shuai, Ran Yu, Yuantao Chen, Zijian Jiang, Xiaowei Song, Nan Wang, Jv Zheng, Jianzhu Ma, Meng Yang, Zhicheng Wang, et al. Pugs: Zero-shot physical understanding with gaussian splatting. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4478–4485. IEEE, 2025.

Jianhua Sun and Cewu Lu. Digital gene: Learning about the physical world through analytic concepts. *arXiv preprint arXiv:2504.04170*, 2025.

Norika Wada, Kohei Yamashita, Ryo Kawahara, and Ko Nishino. M-phygs: Multi-material object dynamics from video. *arXiv preprint arXiv:2512.16885*, 2025.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 803–814, 2023.

Xinli Xu, Wenhang Ge, Dicong Qiu, ZhiFei Chen, Dongyu Yan, Zhuoyun Liu, Haoyu Zhao, Hanfeng Zhao, Shunsi Zhang, Junwei Liang, et al. Gaussianproperty: Integrating physical properties to 3d gaussians with lmms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7231–7240, 2025.

Albert J Zhai, Yuan Shen, Emily Y Chen, Gloria X Wang, Xinlei Wang, Sheng Wang, Kaiyu Guan, and Shenlong Wang. Physical property understanding from language-embedded feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28296–28305, 2024.

Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. In *European Conference on Computer Vision*, pp. 388–406. Springer, 2024.

Haoyu Zhao, Hao Wang, Xingyue Zhao, Hao Fei, Hongqiu Wang, Chengjiang Long, and Hua Zou. Physsplat: Efficient physics simulation for 3d scenes via mllm-guided gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5242–5252, 2025a.

Yizhou Zhao, Haoyu Chen, Chunjiang Liu, Zhenyang Li, Charles Herrmann, Junhwa Hur, Yinxiao Li, Ming-Hsuan Yang, Bhiksha Raj, and Min Xu. Masiv: Toward material-agnostic system identification from videos. *arXiv preprint arXiv:2508.01112*, 2025b.

Licheng Zhong, Hong-Xing Yu, Jiajun Wu, and Yunzhu Li. Reconstruction and simulation of elastic objects with spring-mass 3d gaussians. In *European Conference on Computer Vision*, pp. 407–423. Springer, 2024.

A Appendix

Overview

This appendix provides: i) additional details on dataset generation implementation; ii) extended experimental results and analyses; iii) the details of the MPM algorithm; iv) the prompts used for vision-language baselines; and v) a discussion of limitations and future work. The supplementary material also includes our **source code**, covering both **data generation and model training**, as well as additional **video results**. Further code and implementation details are also available through our anonymous repository https://anonymous.4open.science/r/AnalyticDynamics-8336. These resources will be publicly released upon publication.

B Additional Dataset and Benchmark Details

B.1 Simulation and Rendering Settings

We use the Material Point Method (MPM) Jiang et al. (2016) to generate dynamics trajectories. Each normalized mesh is first converted into volumetric particles at a sampling resolution of 20. During volume sampling, we establish correspondences between the sampled particles and the mesh vertices, allowing the simulated particle motion to be transferred back to the surface mesh for rendering. The sampled particles are then translated into the unit simulation domain $[0, 1]^3$.

The simulation domain is discretized using an MPM grid of resolution 32. We use a time step of 5×10^{-4} and apply gravity $\mathbf{g} = 9.8m/s^2$. All instances share the same ground boundary and contact settings. Each trajectory is simulated for 1,200 steps. We record the particle positions $\mathbf{x}$, displacements $\mathbf{v}$, and deformation gradients $\mathbf{F}$ throughout the simulation.

We render one frame every 50 simulation steps, including the initial state, resulting in 25 frames per sequence. All observations are rendered as monocular mask videos at a resolution of 512×512. During training, we randomly sample four views for each dynamics instance to reduce viewpoint bias. These views share the same underlying trajectory, physical states, and material annotations. During evaluation, we randomly select one view as input.

B.2 Object Geometries and Data Splits

Our benchmark contains 60 object geometries, including 25 synthetic shapes and 35 everyday objects selected from OmniObject3D Wu et al. (2023). The synthetic set consists of basic primitives and manually constructed shapes, while the OmniObject3D objects provide more diverse and geometrically complex structures. We consider three material families: `jelly`, `plasticine`, and `sand`. Their constitutive formulations, varied parameters, and sampling ranges are summarized in Tab. 4. Before particle sampling, each mesh is centered and uniformly scaled to fit within the normalized space $[-0.5, 0.5]^3$.

The training set contains 30 geometries, comprising 10 synthetic shapes and 20 OmniObject3D objects. For each geometry and each of the three material families, we sample 100 material-parameter configurations. Young's modulus and yield stress are sampled uniformly in the logarithmic domain, while Poisson's ratio and friction angle are sampled uniformly in their original domains. This produces $30 \times 3 \times 100 = 9{,}000$ training instances in total.

We construct two evaluation splits. The *seen-object* split uses the same 30 geometries as the training set but contains newly sampled material parameters that are not used during training. We sample 10 parameter configurations for each geometry and each material family, resulting in $30 \times 3 \times 10 = 900$ instances.

The *unseen-object* split contains the remaining 30 geometries, comprising 15 synthetic shapes and 15 OmniObject3D objects. None of these geometries appears during training. We again sample 10 parameter configurations for each geometry and each material family, resulting in $30 \times 3 \times 10 = 900$ instances.

Method	Jelly			Plasticine				Sand	
	Acc. ↑	E ↓	ν ↓	*Acc.* ↑	E ↓	ν ↓	*Yield* ↓	*Acc.* ↑	ϕ ↓
Per-scene Optimization	-	0.184	0.089	-	0.501	0.092	0.069	-	3.086
Ours (w/o alignment)	1.000	0.058	0.017	1.000	0.105	0.025	**0.032**	1.000	**0.203**
Ours (alignment only)	1.000	0.073	0.009	1.000	0.071	0.019	0.049	1.000	0.728
Ours (Full)	**1.000**	**0.053**	**0.008**	**1.000**	**0.062**	**0.018**	0.035	**1.000**	0.327

Table 2: **Quantitative comparison on seen geometries with newly sampled material parameters.** The test split uses the same 30 geometries as training but contains disjoint material-parameter configurations. We report material classification accuracy (↑) and material-specific parameter errors (↓). Young's modulus E and yield stress σ_y are evaluated using log-MAE, while Poisson's ratio ν and friction angle ϕ are evaluated using MAE.

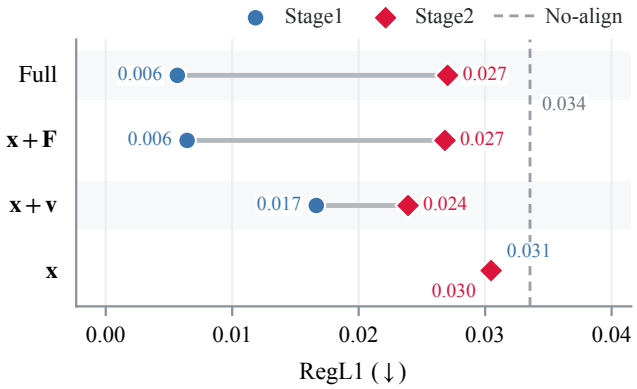

Figure 7: **Ablation of privileged physical states on seen objects with newly sampled material parameters.** We compare position x, displacement v, and deformation gradient F as inputs to the Stage I teacher, and report the RegL1 (↓) errors of both the teacher and its corresponding Stage II student. Gray lines connect each teacher–student pair, while the dashed line denotes the Stage II baseline without distillation.

C Additional Experimental Results

C.1 Evaluation on Seen Objects

We further isolate material-parameter inference from geometric generalization. Specifically, we evaluate on the same 30 geometries used for training, while sampling new material-parameter configurations that are not observed during training. As shown in Table 2, all variants achieve nearly perfect material classification, indicating that material families can be reliably distinguished when the geometry is familiar. Our full model provides the best overall regression performance, achieving the lowest errors for E and ν on both jelly and plasticine, while substantially outperforming per-scene optimization across all three material families.

The gains from alignment are smaller than those observed on unseen objects, and the unaligned model remains slightly better for plastic yield stress and sand friction angle. This is expected: when test geometries have already been observed during training, direct supervision can partially exploit geometry-correlated cues. Under cross-shape generalization, however, these shortcuts become unreliable, making the shape-independent dynamics structure introduced by alignment more important. These results show that our method preserves strong parameter interpolation on seen geometries, while it also improves generalization to unseen objects.

C.2 Privileged-State Ablation on Seen Objects

The main ablation evaluates privileged physical states on unseen geometries, where the full state achieves the best Stage-II performance. To determine whether this advantage mainly arises from

Variant	Frames	Jelly			Plasticine				Sand		Overall RegL1↓
		Acc. ↑	E ↓	ν ↓	*Acc.* ↑	E ↓	ν ↓	*Yield* ↓	*Acc.* ↑	ϕ ↓	
Ours (w/o alignment)	25†	0.997	0.265	0.043	1.000	0.223	0.024	0.115	0.943	1.991	0.082
	24	0.997	0.286	0.042	1.000	0.230	0.024	0.114	0.943	2.004	0.082
	22	1.000	0.451	0.041	1.000	0.325	0.024	0.195	0.963	3.548	0.109
	18	1.000	0.776	0.057	0.993	0.792	0.025	0.657	0.867	7.217	0.185
	15	0.940	0.712	0.063	0.870	0.936	0.042	1.220	0.037	8.136	0.224
Ours (Full)	25†	1.000	0.175	0.037	1.000	0.141	0.021	0.101	0.970	1.777	0.067
	24	1.000	0.180	0.035	1.000	0.146	0.021	0.099	0.970	1.803	0.068
	22	1.000	0.282	0.038	0.997	0.150	0.022	0.115	0.987	2.380	0.080
	18	0.993	0.519	0.047	0.973	0.296	0.025	0.283	0.963	3.550	0.117
	15	0.787	0.678	0.047	0.940	0.381	0.027	0.243	0.550	4.663	0.138

Table 3: **Temporal OOD generalization on unseen objects under varying observation lengths.** We evaluate the models using different numbers of observed frames and report material classification accuracy (↑), material-specific parameter errors (↓), and the overall RegL1 error (↓). The 25-frame setting, marked by † and shaded in gray, is used during training, while the remaining observation lengths are unseen during training.

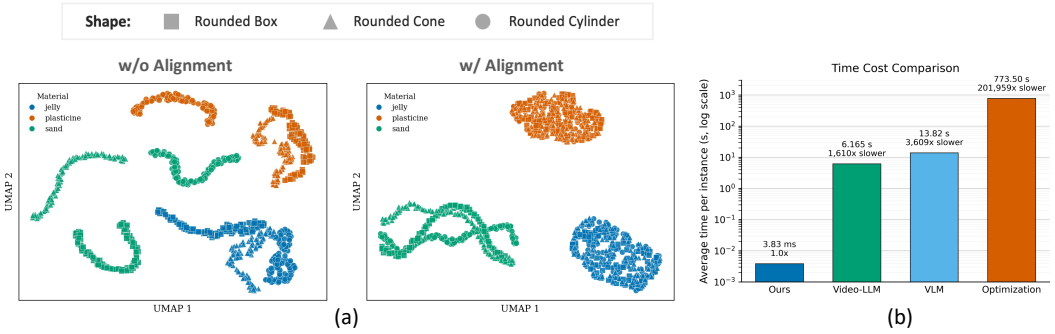

Figure 8: (a) UMAP visualizations of Stage-II representations without and with alignment. Colors denote material families, while markers denote object geometries. Without alignment, samples form geometry-dependent branches; alignment brings different geometries of the same material closer, producing more compact and material-discriminative clusters. (b) Average inference time per instance on a logarithmic scale. Our feed-forward method requires only 3.83 ms, achieving orders-of-magnitude faster inference than Video-LLM, VLM, and per-scene optimization while maintaining accurate dynamics prediction.

cross-geometry generalization, we repeat the same ablation on seen geometries with newly sampled material parameters.

As shown in Fig. 7, all distilled students outperform the Stage-II baseline without alignment, confirming that privileged physical states provide a more informative learning target than material supervision alone. For Stage I, adding the deformation gradient reduces RegL1 from 0.031 with position alone to 0.006, demonstrating its strong dynamics discriminability. For Stage II, however, $\mathbf{x} + \mathbf{v}$ achieves the lowest error of 0.024, slightly outperforming $\mathbf{x} + \mathbf{F}$ and the full state, both at 0.027. When geometry is familiar, displacement provides directly observable motion cues that the video model can readily match, reducing the benefit of additional deformation information. In contrast, the full state performs best on unseen geometries, where combining visible motion with fine-grained deformation responses is more important for reducing shape dependence. These results further show that privileged-state design must balance dynamics discriminability with visual transferability, and that the full state primarily benefits cross-shape generalization.

C.3 Temporal OOD Generalization on Unseen Objects

We further examine whether the model remains reliable when only shorter, unseen temporal observations are available. Keeping the unseen-object split fixed, we evaluate the models trained with

Material	Elastic model	Plastic model	Sampled parameters and ranges
`Jelly`	Corotated	Identity	$E \in [1.5 \times 10^4,\ 1.5 \times 10^6]$, $\nu \in [0.20,\ 0.40]$
`Plasticine`	Sigma	von Mises	$E \in [5 \times 10^4,\ 5 \times 10^5]$, $\nu \in [0.35,\ 0.48]$, $\sigma_y \in [500,\ 25000]$
`Sand`	Sigma	Drucker–Prager	$E = 1 \times 10^5$, $\nu = 0.30$, $\phi_f \in [10^\circ,\ 45^\circ]$

Table 4: Constitutive models and parameter ranges used for each material family. The selected parameter ranges span a broad spectrum of mechanical behaviors commonly observed within these material families. Here, E and ν denote Young's modulus and Poisson's ratio, respectively; ϕ_f denotes the friction angle; and σ_y is the yield stress. For `sand`, only ϕ_f is varied because it is the dominant parameter governing the observed frictional dynamics, while the remaining elastic parameters are fixed.

25-frame videos using the first $K \in \{25, 24, 22, 18, 15\}$ consecutive frames, and directly feed the resulting variable-length sequence to the video encoder. We compare the full model with its unaligned counterpart.

As shown in Table 3, performance remains nearly unchanged with 25 frames but gradually degrades as the observation becomes shorter, since less motion and deformation evidence is available for dynamics inference. The full model consistently outperforms the unaligned baseline, with a larger advantage under shorter observations. At 15 frames, it reduces the overall RegL1 from 0.224 to 0.138. This suggests that physics-grounded alignment helps the video model extract more stable dynamics cues from incomplete temporal evidence. The remaining degradation under very short observations also indicates that sufficiently informative motion is still necessary for accurate material inference.

C.4 Effect of Alignment and Inference Efficiency

Figure 8(a) provides additional visual evidence for the central finding in the main paper: physics-grounded alignment reduces the dependence of visual representations on object geometry. Without alignment, samples of the same material remain separated by geometry, forming shape-specific branches. After alignment, representations from different geometries cluster more compactly according to material family. This complementary visualization further confirms that distillation shifts the organization of the visual space from geometric cues toward intrinsic dynamics, consistent with the improved Shape-OOD performance reported in the main experiments.

Figure 8(b) further compares the average inference time per instance. Our method requires only 3.83 ms, achieving orders-of-magnitude faster inference than VLM, Video-LLM, and per-scene optimization. This efficiency comes from predicting the material model and parameters in a single forward pass, without iterative simulation or test-time optimization. Together, these results show that our method learns a more dynamics-centered visual representation while retaining the efficiency of feed-forward inference.

D Constitutive Material Families and Parameters

Within the MPM framework, each material family is defined by an elastic constitutive law and a plastic projection. Given a trial deformation gradient $\mathbf{F}^{\text{trial}}$, the plastic model φ_P first projects it onto the admissible elastic domain, after which the elastic model φ_E computes the Kirchhoff stress $\boldsymbol{\tau}$:

$$\mathbf{F} = \varphi_P\big(\mathbf{F}^{\text{trial}}\big), \qquad \boldsymbol{\tau} = \varphi_E(\mathbf{F}). \tag{10}$$

For a deformation gradient $\mathbf{F} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, we define

$$J = \det(\mathbf{F}), \qquad \boldsymbol{\epsilon} = \log(\boldsymbol{\Sigma}), \qquad \hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon} - \frac{\operatorname{tr}(\boldsymbol{\epsilon})}{d}\mathbf{I}, \tag{11}$$

where $d = 3$ is the spatial dimension. The Lamé parameters are computed from Young's modulus E and Poisson's ratio ν as

$$\mu = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}. \tag{12}$$

D.1 JELLY MATERIAL

We model `jelly` using `CorotatedElasticity` together with `IdentityPlasticity`. This combination describes a purely elastic material without irreversible deformation.

Identity plasticity. The identity projection leaves the trial deformation gradient unchanged:

$$\mathbf{F} = \mathbf{F}^{\text{trial}}. \tag{13}$$

Corotated elasticity. Given the polar rotation

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T, \tag{14}$$

the Kirchhoff stress is

$$\boldsymbol{\tau} = 2\mu\,(\mathbf{F} - \mathbf{R})\,\mathbf{F}^T + \lambda J(J-1)\mathbf{I}. \tag{15}$$

The mechanical behavior of `jelly` is therefore controlled by Young's modulus E and Poisson's ratio ν. Young's modulus determines material stiffness, while Poisson's ratio controls volumetric deformation.

D.2 PLASTICINE MATERIAL

We model `plasticine` using `SigmaElasticity` together with `VonMisesPlasticity`. The elastic model describes the recoverable response, while the von Mises projection introduces irreversible deformation once the yield limit is exceeded.

Von Mises plasticity. Let

$$\mathbf{F}^{\text{trial}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T, \qquad \boldsymbol{\epsilon} = \log(\boldsymbol{\Sigma}). \tag{16}$$

The plastic multiplier is defined as

$$\delta\gamma = \|\hat{\boldsymbol{\epsilon}}\| - \frac{\sigma_y}{2\mu}, \tag{17}$$

where σ_y is the yield stress. The corrected singular values are

$$\mathcal{Z}_{\text{VM}}(\boldsymbol{\Sigma}) = \begin{cases} \boldsymbol{\Sigma}, & \delta\gamma \le 0, \\ \exp\left(\boldsymbol{\epsilon} - \delta\gamma \dfrac{\hat{\boldsymbol{\epsilon}}}{\|\hat{\boldsymbol{\epsilon}}\|}\right), & \delta\gamma > 0. \end{cases} \tag{18}$$

The corrected deformation gradient is then

$$\mathbf{F} = \mathbf{U}\mathcal{Z}_{\text{VM}}(\boldsymbol{\Sigma})\mathbf{V}^T. \tag{19}$$

Sigma elasticity. For the corrected deformation gradient $\mathbf{F} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, the Kirchhoff stress is computed in the logarithmic principal-strain space:

$$\boldsymbol{\tau} = \mathbf{U}\,(2\mu\boldsymbol{\epsilon} + \lambda\,\mathrm{tr}(\boldsymbol{\epsilon})\mathbf{I})\,\mathbf{U}^T. \tag{20}$$

The behavior of `plasticine` is controlled by Young's modulus E, Poisson's ratio ν, and yield stress σ_y. The first two parameters determine its elastic response, while σ_y determines the onset of permanent deformation.

D.3 SAND MATERIAL

We model `sand` using `SigmaElasticity` together with `DruckerPragerPlasticity`. The elastic response follows Eq. equation 20, while the Drucker–Prager projection models pressure-dependent yielding and frictional flow.

Drucker–Prager plasticity. Let

$$\mathbf{F}^{\text{trial}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T, \qquad \boldsymbol{\epsilon} = \log(\boldsymbol{\Sigma}). \tag{21}$$

We define

$$\alpha = \sqrt{\frac{2}{3}}\,\frac{2\sin\phi_f}{3-\sin\phi_f}, \tag{22}$$

where ϕ_f is the friction angle. The plastic multiplier is

$$\delta\gamma = \|\hat{\boldsymbol{\epsilon}}\| + \alpha\frac{(d\lambda+2\mu)\,\mathrm{tr}(\boldsymbol{\epsilon})}{2\mu}. \tag{23}$$

The corrected singular values are defined as

$$\mathcal{Z}_{\text{DP}}(\boldsymbol{\Sigma}) = \begin{cases} \mathbf{I}, & \mathrm{tr}(\boldsymbol{\epsilon}) > 0, \\ \boldsymbol{\Sigma}, & \delta\gamma \le 0 \text{ and } \mathrm{tr}(\boldsymbol{\epsilon}) \le 0, \\ \exp\left(\boldsymbol{\epsilon} - \delta\gamma\dfrac{\hat{\boldsymbol{\epsilon}}}{\|\hat{\boldsymbol{\epsilon}}\|}\right), & \text{otherwise.} \end{cases} \tag{24}$$

The corrected deformation gradient is

$$\mathbf{F} = \mathbf{U}\mathcal{Z}_{\text{DP}}(\boldsymbol{\Sigma})\mathbf{V}^T. \tag{25}$$

The friction angle ϕ_f controls the shear resistance and flow behavior of `sand`. In our benchmark, we vary only ϕ_f, which is the dominant factor governing the observed sand dynamics, while keeping E and ν fixed.

E Material Point Method Formulation

In continuum mechanics Chaves (2013), material motion is governed by the conservation of mass and linear momentum:

$$\frac{D\rho}{Dt} + \rho\nabla\cdot\mathbf{v} = 0, \qquad \rho\frac{D\mathbf{v}}{Dt} = \nabla\cdot\boldsymbol{\sigma} + \rho\mathbf{g}, \tag{26}$$

where ρ is the density, $\mathbf{v}$ is the velocity field, $\mathbf{g}$ is the gravitational acceleration, and $\boldsymbol{\sigma}$ is the Cauchy stress. The stress is determined by the constitutive law, which characterizes the material response to deformation.

We solve these equations using an explicit Material Point Method (MPM) solver with affine particle-in-cell transfers Jiang et al. (2016). MPM represents the material using Lagrangian particles and evaluates physical interactions on a background Eulerian grid. Each particle p stores its position $\mathbf{x}_p$, mass m_p, reference volume V_p^0, velocity $\mathbf{v}_p$, deformation gradient $\mathbf{F}_p$, and affine velocity matrix $\mathbf{C}_p$. Each simulation step consists of particle-to-grid transfer, grid update, and grid-to-particle transfer.

Let

$$w_{ip}^t = N_i(\mathbf{x}_p^t), \qquad \nabla w_{ip}^t = \nabla N_i(\mathbf{x}_p^t), \tag{27}$$

denote the interpolation weight and its spatial gradient between particle p and grid node i. During particle-to-grid transfer, particle mass and momentum are accumulated on the grid:

$$m_i^t = \sum_p w_{ip}^t m_p, \tag{28}$$

$$\mathbf{p}_i^t = \sum_p w_{ip}^t m_p\left[\mathbf{v}_p^t + \mathbf{C}_p^t\left(\mathbf{x}_i - \mathbf{x}_p^t\right)\right]. \tag{29}$$

The internal force at each grid node is computed as

$$\mathbf{f}_i^{\text{int},t} = -\sum_p V_p^0\boldsymbol{\tau}_p^t\nabla w_{ip}^t, \tag{30}$$

where $\boldsymbol{\tau}_p^t$ denotes the particle stress measure used by the discrete MPM solver.

For grid nodes with nonzero mass, the velocity is updated according to

$$\mathbf{v}_i^t = \frac{\mathbf{p}_i^t}{m_i^t}, \tag{31}$$

$$\widetilde{\mathbf{v}}_i^{t+1} = \mathbf{v}_i^t + \Delta t \left(\frac{\mathbf{f}_i^{\mathrm{int},t}}{m_i^t} + \mathbf{g} \right), \tag{32}$$

$$\mathbf{v}_i^{t+1} = \mathcal{B}\left(\widetilde{\mathbf{v}}_i^{t+1}\right), \tag{33}$$

where $\mathcal{B}$ applies the prescribed boundary conditions and collision responses.

The updated grid velocities are then interpolated back to the particles:

$$\mathbf{v}_p^{t+1} = \sum_i w_{ip}^t \mathbf{v}_i^{t+1}, \tag{34}$$

$$\mathbf{x}_p^{t+1} = \mathbf{x}_p^t + \Delta t\, \mathbf{v}_p^{t+1}. \tag{35}$$

Using quadratic B-spline interpolation, the affine velocity matrix is updated as

$$\mathbf{C}_p^{t+1} = \frac{4}{\Delta x^2} \sum_i w_{ip}^t \mathbf{v}_i^{t+1} \left(\mathbf{x}_i - \mathbf{x}_p^t\right)^T, \tag{36}$$

where Δx is the grid spacing.

The trial deformation gradient is subsequently computed by

$$\mathbf{F}_p^{\mathrm{tr}} = \left(\mathbf{I} + \Delta t\, \mathbf{C}_p^{t+1}\right) \mathbf{F}_p^t. \tag{37}$$

We then apply the plastic projection and elastic stress mapping:

$$\mathbf{F}_p^{t+1} = \varphi_P\left(\mathbf{F}_p^{\mathrm{tr}}; \boldsymbol{\theta}_P\right), \tag{38}$$

$$\boldsymbol{\tau}_p^{t+1} = \varphi_E\left(\mathbf{F}_p^{t+1}; \boldsymbol{\theta}_E\right), \tag{39}$$

where φ_P denotes the plastic projection, φ_E denotes the elastic constitutive model, and $\boldsymbol{\theta}_P$ and $\boldsymbol{\theta}_E$ are their corresponding material parameters. Here, $\mathbf{F}_p^{t+1}$ denotes the projected deformation gradient used for stress evaluation. For purely elastic materials, φ_P reduces to the identity mapping.

All material families share the same MPM update procedure and differ only in their elastic model, plastic model, and associated material parameters.

F Prompts for Vision-Language Baselines

We use GPT-5.5 as the VLM baseline and Gemini 3.5 Flash as the Video-LLM baseline. GPT-5.5 receives multi-view RGB images of the target object, whereas Gemini 3.5 Flash receives a monocular RGB video captured from a single view.

The system prompt for both models is set as:

```
You are an expert assistant in continuum mechanics and Material Point Method (MPM) simulation.
```

For the VLM, the user prompt is designed for static spatial analysis:

```
### TASK OVERVIEW
Given multi-view RGB images of a single 3D object, your task is to infer its physically plausible
material model and material parameters for MPM simulation.

1. First, infer the most appropriate global material model for the whole object based on its
visual appearance and expected physical behavior.
2. Then, infer the most likely continuous material parameter values required by the selected
material model.

Treat the object as a single material body. Do not decompose it into parts.

### INPUTS
* Multi-view RGB images of the target object.

### Material Model Candidates
```

```
The material model must be selected from the following list:
- elastic
- plasticine
- sand

### Material Parameters
The required material parameters depend on the selected material model:

- elastic:
  - Young's Modulus
  - Poisson's Ratio

- plasticine:
  - Young's Modulus
  - Poisson's Ratio
  - yield stress

- sand:
  - friction angle

### Parameter meaning
Young's Modulus controls stiffness. A larger value means the object is harder to deform, while a
smaller value means it is softer or more flexible.

Poisson's Ratio controls lateral expansion under compression. For most solid materials, it should
be between 0 and 0.5.

Yield stress controls when a plastic material starts to undergo irreversible deformation. A larger
value means the material resists plastic flow more strongly.

Friction angle controls the internal friction of granular material. A larger value means the
material forms steeper piles and resists flow more strongly.

### REQUIRED OUTPUT (JSON)
You MUST output a JSON object containing:
{
  "material": {
    "material_model": "elastic | plasticine | sand",
    "material_parameters": {
      "...": value
    }
  },
  "reasoning": "Brief explanation of the visual cues and physical intuition used to select the
  material model and parameter values."
}

### Important Rules
1. Select exactly one material model.
2. Only output the parameters required by the selected material model.
3. Do not output parameters for unselected material models.
4. All continuous parameters must be output as single scalar values, not ranges.
5. The parameter values should be physically plausible for MPM simulation.
6. If the visual evidence is ambiguous, choose the most likely material model and output
conservative parameter values.
7. Output only valid JSON. Do not include markdown, comments, or any extra text outside the JSON
object.
```

For the Video-LLM, the user prompt is designed for temporal evolution analysis:

```
### TASK OVERVIEW
Given a motion video of a single object, your task is to infer its physically plausible material
model and material parameters for MPM simulation.

You should perform the inference in two steps:

1. First, infer the most appropriate global material model for the whole object based on its
observed motion, deformation behavior, and visual appearance.
2. Then, infer the most likely continuous material parameter values required by the selected
material model.

Treat the object as a single material body. Do not decompose it into parts.

The video contain object motion caused by gravity, collision, compression. You should use the
dynamic cues in the video as the primary evidence, and use visual appearance as secondary evidence
when the motion is ambiguous.

### INPUTS
* A motion video of the target object.

### MATERIAL MODEL CANDIDATES
The material model must be selected from the following list:
- elastic
- plasticine
- sand

### MATERIAL PARAMETERS
The required material parameters depend on the selected material model:

- elastic:
  - Young's Modulus
  - Poisson's Ratio
```

```
- plasticine:
  - Young's Modulus
  - Poisson's Ratio
  - yield stress

- sand:
  - friction angle

### PARAMETER MEANING
Young's Modulus controls stiffness. A larger value means the object is harder to deform, while a
smaller value means it is softer or more flexible.

Poisson's Ratio controls lateral expansion under compression. For most solid materials, it should
be between 0 and 0.5.

Yield stress controls when a plastic material starts to undergo irreversible deformation. A larger
value means the material resists plastic flow more strongly.

Friction angle controls the internal friction of granular material. A larger value means the
material forms steeper piles and resists flow more strongly.

### DYNAMIC CUES TO CONSIDER
Use the object's motion and deformation behavior in the video to guide your inference:

- If the object deforms under force but recovers its original shape, bounces, oscillates, or shows
reversible deformation, it is likely elastic.
- If the object deforms and does not fully recover, keeps a dent, flattens, spreads, or undergoes
permanent shape change, it is likely plasticine.
- If the object consists of many small grains, flows, piles up, collapses, avalanches, or forms a
slope under gravity, it is likely sand.

For elastic materials:
- Larger Young's Modulus should be assigned when the object appears stiff, deforms only slightly,
rebounds quickly, or preserves its shape.
- Smaller Young's Modulus should be assigned when the object appears soft, bends strongly,
compresses visibly, or oscillates slowly.
- Poisson's Ratio should be higher for nearly incompressible soft materials such as rubber-like
objects.

For plasticine materials:
- Larger Young's Modulus should be assigned when the object resists deformation before yielding.
- Smaller Young's Modulus should be assigned when the object is easily compressed or reshaped.
- Larger yield stress should be assigned when permanent deformation starts only under strong force.
- Smaller yield stress should be assigned when the object yields easily and retains deformation
under weak force.

For sand materials:
- Larger friction angle should be assigned when the granular material forms a steep stable pile
and resists flow.
- Smaller friction angle should be assigned when it flows easily, spreads widely, or forms a
shallow pile.

### REQUIRED OUTPUT (JSON)
You MUST output a JSON object containing:
{
  "material": {
    "material_model": "elastic | plasticine | sand",
    "material_parameters": {
      "...": value
    }
  },
  "reasoning": "Brief explanation of the observed motion/deformation cues and physical intuition
  used to select the material model and parameter values."
}

### IMPORTANT RULES
1. Select exactly one material model.
2. Only output the parameters required by the selected material model.
3. Do not output parameters for unselected material models.
4. All continuous parameters must be output as single scalar values, not ranges.
5. The parameter values should be physically plausible for MPM simulation.
6. Use dynamic cues from the video as the primary evidence.
7. Use visual appearance only as secondary evidence when motion cues are insufficient or ambiguous.
8. If the video does not clearly show deformation or interaction, choose the most likely material
model based on appearance and output conservative parameter values.
9. Output only valid JSON. Do not include markdown, comments, or any extra text outside the JSON
object.
```

G Limitations and Future Work

First, our inference space is limited to a finite set of predefined constitutive models, which cannot fully capture the diversity of real-world material behaviors. Unseen materials are therefore forced into the closest known family, leading to biased dynamics estimates. A promising direction is to close the loop between constitutive discovery and data generation: identify candidate constitutive laws from real-world motion using methods such as VisionLaw, generate targeted simulations, and

update the inference model to progressively expand the supported material space. Second, a sim-to-real gap remains. Mask videos reduce appearance discrepancies caused by texture, lighting, and background, but cannot eliminate physical mismatches arising from simulation time steps, rendering intervals, contact and damping assumptions, and imperfect simulation. A practical direction is to combine feed-forward inference with lightweight scene-specific optimization. Our model can efficiently provide reliable initial estimates of the material model and parameters, which can then be refined with a few optimization steps using methods such as VisionLaw or NeuMA. This hybrid strategy preserves the efficiency of feed-forward inference while improving scene-level accuracy.