Inferring Dynamic Physical Properties from Video Foundation Models

Anonymous CVPR submission

Paper ID 4

Abstract

We study the task of predicting dynamic physical properties from videos. More specifically, we consider physical properties that require temporal *information to be inferred: elasticity of a bouncing object, viscosity of a flowing liquid, and dynamic friction of an object sliding on a surface. To this end, we make the following contributions: (i) We collect a new video dataset for each physical property, consisting of synthetic training and testing splits, as well as a real split for real world evaluation. (ii) We explore three ways to infer the physical property from videos: (a) an oracle method where we supply the visual cues that intrinsically reflect the property using classical computer vision techniques; (b) a simple read out mechanism using a visual prompt and trainable prompt vector for cross-attention on pre-trained video generative and self-supervised models; and (c) prompt strategies for Multi-modal Large Language Models (MLLMs). (iii) We show that video foundation models trained in a generative or self-supervised manner achieve a similar performance, though behind that of the oracle, and MLLMs are currently inferior to the other models, though their performance can be improved through suitable prompting. The datasets will be publicly released.*

1. Introduction

Humans are remarkably adept at intuitively estimating physical properties from visual observations. Without direct interaction, people can often estimate how bouncy a ball is, how thick a liquid seems, or how slippery a surface might be—simply by watching how objects move. While these estimations are not precise in a scientific sense, they are sufficiently accurate for guiding perception, prediction, and action. Bringing this capability to machines is an important step towards building more general and physically grounded artificial intelligence. In particular, visual systems that can infer dynamic physical properties from raw video could enhance robotic manipulation, embodied agents, and video understanding tasks in ways that go beyond the traditional perception tasks of recognition, detection, and segmentation.

Recent progress in video foundation models, including generative models [26, 52], self-supervised models [3, 5] and multi-modal large language models (MLLMs) [9, 18, 19], have shown impressive capability in synthesizing realistic dynamics, learning general-purpose video representations, and tackling semantic understanding tasks, for example, video question answering. However, a question that remains underexplored is: **do these models acquire an understanding of *dynamic* physical properties from videos ?**

In this paper, we address this question by focusing on several representative physical properties that are not directly observable in static frames but instead emerge through temporal dynamics: the elasticity of a bouncing object, the viscosity of a flowing liquid, and the dynamic friction between a surface and a sliding object. These properties are especially compelling because their inference requires temporal reasoning and sensitivity to subtle visual cues—such as deformation, deceleration, spreading, or oscillation. By examining how well current video foundation models capture these dynamic attributes, we aim to assess their physical understanding beyond static appearance.

To support this investigation, we introduce a new dataset, *PhysVid*, specifically designed to evaluate the dynamic physical properties from video. Existing datasets lack ground-truth annotations for such properties, so we construct *PhysVid* using a combination of synthetic videos—rendered via a physics simulator—and real-world videos sourced from the internet or captured in-house. Each video is annotated with physical property values, either derived from simulation parameters or estimated manually. The dataset is designed to facilitate the study of out-of-domain generalization, both within the synthetic domain and from synthetic to real-world data. To establish an upper bound on what is inferable from visual input alone, we implement an oracle method for each property. These oracles leverage privileged access to the visual cues that directly reflect the corresponding property.

We evaluate three categories of video foundation models: generative models, self-supervised models, and multi-

modal large language models (MLLMs). For the generative and self-supervised models, we propose a simple yet effective readout mechanism that extracts dynamic physical properties from pre-trained, frozen representations. Our method introduces a learnable query vector that attends to internal representation tokens via cross-attention, enabling the selective extraction of relevant information. This approach is both lightweight and training-efficient. For MLLMs, we explore various prompting strategies to elicit predictions of dynamic physical properties directly from video input. These strategies include few-shot prompting to provide task context, as well as procedural prompting that guides the model through the oracle estimation steps—helping it focus on the intrinsic visual cues that reveal the target properties.

2. Related Work

Physics Prediction from Images and Videos. Inferring physical properties from visual observations remains a core challenge in computer vision. Early methods estimate latent physical parameters (e.g., mass, friction, stiffness) via differentiable physics engines or learning-based simulators [10, 20, 24, 43, 44, 50], while later works infer salient attributes like viscosity or elasticity from task-specific visual cues [2, 21, 22, 28–31], yet both rely heavily on simulation supervision, domain priors, or handcrafted heuristics. More recently, unsupervised learning of intuitive physics has emerged via next-frame prediction from large-scale everyday physical scenes [1, 4, 11, 12, 15, 16, 27, 42], capturing latent dynamics without explicit physical supervision. However, the resulting representations are usually implicit and lack interpretability in terms of concrete physical quantities. In contrast, we infer physical properties by directly prompting pre-trained video foundation models, enabling explicit estimation without reliance on task-specific heuristics, or end-to-end prediction pipelines from scratch.

Physics Datasets and Benchmarks. An increasing number of physics-related datasets have been collected in recent years to provide ground truth annotations for different physical properties, including material [14, 36], shadow [45, 46], support relations [39],occlusion [53, 54], mass and volume [51]. Another line of work [6–8, 35, 37, 41] proposes broad benchmarks with video-image-text QA tasks to assess physical understanding in vision-language models, but the questions are typically qualitative and categorical. More recently, Zhang et al. [56] introduces a benchmark consisting of 130 real-world videos capturing physical phenomena guided by conservation laws, to evaluate the physics plausibility of video generative models by assessing the trajectory of objects in their generated videos. In contrast, our datasets consist of both *synthetic* and *real-world* videos annotated with the *quantitative value* for the associated physical parameter of the coefficient of friction, elasticity, and viscosity.

3. Problem Scenario and The *PhysVid* Datasets

In this paper, we address the problem of estimating physical properties from videos. Specifically, we focus on three properties: **elasticity** of a bouncing object, **viscosity** of a flowing liquid, and the **dynamic friction coefficient** between a surface and a sliding object. Given a video $v \in \mathbb{R}^{T\times H\times W\times 3}$, we consider two formulations, the first is **absolute value prediction**, where the input is a single video and the model is tasked with predicting the numerical value of the physical property, *i.e.*, $y_{\text{abs}} = \Phi(v;\theta_1)$. The second is **relative value comparison**, where the input is a pair of videos captured from the same viewpoint, and the model must determine whether the first video exhibits a higher physical property value than the second, *i.e.*, $y_{\text{rel}} = \Phi(v_1, v_2;\theta_2)$, and y_{rel} is binary.

Each scenario is parameterized by a set of variables, including the value of the target *physical property* (*e.g.,* elasticity, viscosity, or friction), and a set of *nuisance parameters* (including camera viewpoint, object appearance, lighting, *etc.*). While the model must be sensitive to changes in the physical property, it should be robust (ideally invariant) to variations in nuisance parameters.

To assess generalization, we define two domains of nuisance parameters, denoted as $\mathcal{A}_1$ and $\mathcal{A}_2$, which differ in their distributions. For instance, $\mathcal{A}_2$ may have different camera viewpoints or different lighting conditions to $\mathcal{A}_1$ (full details of these differences are given in Supplementary Section B). We generate a dataset using a physics-based simulator, consisting of one training split and two test splits. The models are only trained on the training split from the simulator for all the evaluations. The training and `test-1` splits are sampled from $\mathcal{A}_1$, sharing the same nuisance distribution; `test-2` is drawn from $\mathcal{A}_2$, introducing a distribution shift. The target property values are sampled from a shared range across all splits to ensure consistency. Finally, `test-3` consists of real-world videos, used to evaluate generalization beyond simulation.

3.1. The *PhysVid* Datasets

To study the dynamic physical properties of elasticity, viscosity, and friction, we construct a dataset for each, containing both synthetic and real-world videos. Synthetic ones are generated with the Genesis simulator [58], and real ones are captured with an iPhone in slow-motion mode or downloaded from the Internet. For each property we have: $10,000$ videos for `train`; 1000 videos for each of `test-1` and `test-2`; and 100 videos for `test-3`. Sample frames are shown in Figure 1. In the following we describe how each property is realized in the video. Please refer to Supplementary Section B for more details of the datasets.

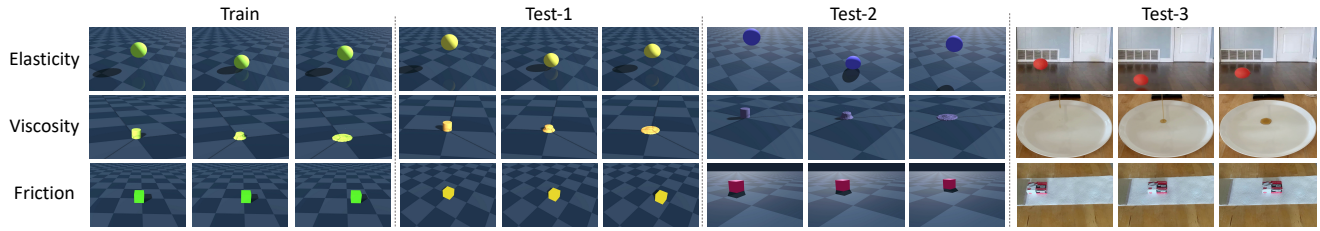

Figure 1. **Examples of the *PhysVid* dataset.** Each row shows a different property, and each column shows three frames from video samples in the synthetic sets (`train`, `test-1`, and `test-2`) and the real `test-3` set. The `train` and `test-1` sets are from the same distribution. In `test-2` parameters, such as lighting, viewpoint and color, differ from those in `test-1`.

3.1.1. Elasticity

We study an object's elasticity by analyzing the motion of a ball dropped onto the ground and its subsequent bounces. In physics, elasticity e is quantified as the ratio of the rebound velocity $v_{\text{after impact}}$ to the impact velocity $v_{\text{before impact}}$, and also equals $\sqrt{h_{\text{bounce}}/h_{\text{drop}}}$, where h_{drop} is the dropping height and h_{bounce} is the bouncing height. Here and for the following properties, please refer to Supplementary Section C for the detailed derivations. These expressions are used for the oracle estimation in Section 4.1.

Synthetic Dataset. All synthetic videos are generated using Genesis [58], with object's elasticity as the target property. Nuisance factors include drop height, camera viewpoint, object appearance, and lighting conditions. The object is of the same size in all videos. Note, here and for the following properties, the ground truth property value is obtained directly from the simulator.

Real-World Dataset. The real-world videos are collected from YouTube using the search term "ball bouncing experiments". Each clip is manually trimmed to include the drop-and-bounce sequence of a single ball. The dataset includes a wide range of materials (*e.g.*, rubber balls, tennis balls, basketballs, balloons, *etc*), resulting in diverse elasticity values. The ground truth elasticity values for the real sequences are estimated by computing $\sqrt{h_{\text{bounce}}/h_{\text{drop}}}$: the videos are chosen such that the balls bounce in a fronto-parallel plane, which means that ratios of image heights (differences in y-coordinates) are approximately equal to the ratio of heights in 3D. These image differences are obtained by manual annotation.

3.1.2. Viscosity

We study the viscosity by observing a liquid column dropping and spreading on the ground. The viscosity can be reflected by the growth rate of the liquid area on the ground. The viscosity μ is negatively correlated to the liquid area growth rate $\frac{d(A(t))}{dt}$, given the controlled liquid density ρ, controlled liquid column diameter D, and controlled dropping velocity v of the liquid column when it reaches the ground.

Synthetic Dataset. The synthetic videos are generated using Genesis [58], where the target property is the viscosity of liquid. Nuisance factors include camera viewpoint, object appearance, and lighting conditions. The liquid column is of the same size in all videos.

Real-World Dataset. Since it is challenging to find real-world videos online that provide ground-truth viscosity values while controlling for other relevant physical parameters—such as ρ, D and v, we collected real videos under controlled conditions. We use a funnel with a fixed nozzle diameter to produce a consistent liquid column. A funnel holder allows us to fix the height from which the liquid is poured, thereby controlling the initial velocity v. Ground-truth viscosity values for each liquid are obtained from standard physics reference tables. The selected liquids span a wide range of viscosities, from 1.2 (*e.g.*, coffee) to 225 (*e.g.*, maple syrup), allowing for a diverse and comprehensive evaluation.

3.1.3. Friction

We study friction between an object and a surface by observing how the object slows down as it slides with an initial velocity. The dynamic friction coefficient μ_k is proportional to the (negative) acceleration of the object a.

Synthetic Dataset. The synthetic videos are generated using Genesis [58], where the target property is the dynamic friction coefficient at the contacting surface of the object and the ground. Nuisance factors include initial location and initial velocity of the object, camera viewpoint, object appearance, and lighting conditions. The object is of the same size in all videos.

Real-World Dataset. While many online videos depict objects sliding on surfaces, they lack ground-truth annotations for friction coefficients. We therefore collect a real video dataset featuring 5 different objects and 6 surface materials, spanning a wide range of dynamic friction values. Each object is given an initial velocity by sliding it down from a slope and it then slides on a horizontal plane. To obtain ground-truth friction coefficients, we use a spring dynamometer to measure the friction force F for each object-surface pair (by dragging the object at constant speed), and record the object's weight G. The dynamic friction coeffi-

cient is then computed as: $\mu_k = F/G$.

4. Inferring Physical Properties

This section presents the three different ways for inferring dynamic physical properties: an oracle method via classical computer vision techniques (Section 4.1); a visual prompt mechanism for video generative and self-supervised models (Section 4.2); and prompts for MLLMs (Section 4.3).

4.1. Oracle Estimation

4.1.1. Elasticity

We aim to estimate elasticity from both synthetic and real-world videos. The key visual cue is the relative height of the ball during its drop and subsequent bounce, observed in 3D. As noted earlier, the ratio in 3D can be approximated from their corresponding image-space measurements. This approximation is exact when the motion occurs in a fronto-parallel plane, and remains reasonably accurate otherwise—since the ratio of lengths between parallel line segments is invariant under affine transformations [17]. Given that perspective effects are minimal in our videos, the affine approximation provides a reliable estimate for elasticity. To automate this process, we extract the ball's trajectory $y(t)$ from the video and input the sequence of ratios into a GRU network to regress the elasticity. In detail, we segment the ball in each frame and use their centroids as the y-coordinate. From this trajectory, we identify key points: the initial drop position, the first ground contact, and the peak of the first bounce. The resulting trajectory is normalized to the range $[0, 1]$, by subtracting the y-coordinate of the first ground contact and dividing by the initial drop height. This normalization not only ensures invariance to viewpoint and scale, but also simplifies learning for the GRU by standardizing the input distribution. We train a GRU, as it is noisy to directly obtain h_{drop} and h_{bounce} using heuristics (*e.g.,* determining the maximum and minimum points), and in practice a GRU provides a good estimate. The full pipeline is illustrated in Figure 2 (top row). For the **absolute prediction**, the normalized trajectory is fed into a GRU network, which directly regresses the elasticity value. For the **relative comparison**, the binary decision score between two videos v_1 and v_2 is calculated as:

$$\texttt{score} = \sigma(\log(\frac{e_1}{e_2})), \tag{1}$$

where e_1 and e_2 are the estimated elasticities based on height ratios, and $\sigma(\cdot)$ denotes the sigmoid function.

4.1.2. Viscosity

The key visual cue for estimating viscosity is the rate at which the liquid spreads on the ground-plane, measured as an area ratio normalized by the initial area of the liquid column. As with elasticity, we approximate perspective using an affine transformation – here of the ground-plane. Since area ratios are invariant under affine transformations [17], the liquid's normalized image-space area growth approximates its true normalized ground-plane expansion (in our setup the liquid spreads only within a limited area around the release point, and the camera is distant; consequently an affine viewing approximation is adequate). Specifically, we extract segmentation masks for each frame and compute the liquid's area over time. This area sequence is normalized by the area in the first frame where the liquid contacts the surface, ensuring invariance to viewpoint and scale. The process is illustrated in Figure 2 (middle row). For **absolute prediction**, we calculate the slope k of $A(t)$ and use $1/k$ to represent the viscosity value; For **relative comparison**, the binary decision score between two videos v_1 and v_2 is calculated as in Equation 1, where e_1 and e_2 are the estimated viscosities based on area growth rate.

4.1.3. Friction

The key visual cue for estimating dynamic friction is the acceleration of the sliding object—*i.e.*, how quickly its velocity decreases due to friction—which can be inferred from its position over time. Since the object moves significantly in the video, we do not use an affine approximation, but instead take account of the projective geometry by mapping the object's motion to a bird's-eye view, allowing for consistent trajectory analysis. This is achieved by estimating a homography between the image and bird's eye view (normal to the plane) from the four corners of the object's top surface (see Figure 2, bottom row). We fit a parabola $x = \alpha t^2 + \beta t + c$ to the transformed top surface trajectory to estimate the acceleration a from the coefficient α, and the coefficient of friction $\mu_k = 2\alpha/g$. For **absolute prediction**, we use the estimated μ_k to represent the friction coefficient value; For **relative comparison**, the binary decision score between two videos v_1 and v_2 is calculated as in Equation 1, where e_1 and e_2 are the estimated friction coefficients based on the transformed object trajectory.

4.2. Video Generative and Self-Supervised Models

4.2.1. Video Feature Extraction

Given a video $v \in \mathbb{R}^{T\times H\times W\times 3}$, we extract features with a pre-trained video backbone, that can either be generative or self-supervised, resulting into spatiotemporal feature representations, *i.e.*, $r = \psi(v) \in \mathbb{R}^{t\times h\times w\times c}$, which can be detailed as follows.

Generative Model as Backbone. We adopt a pre-trained video diffusion model (Figure 3, left), namely DynamiCrafter [52], to compute the visual features. Specifically, given an input video, we add noise to the latent representations after the pre-trained VAE encoder, and replace the text prompt with a learnable embedding. We extract multi-scale features from all U-Net layers at diffusion time step 50,

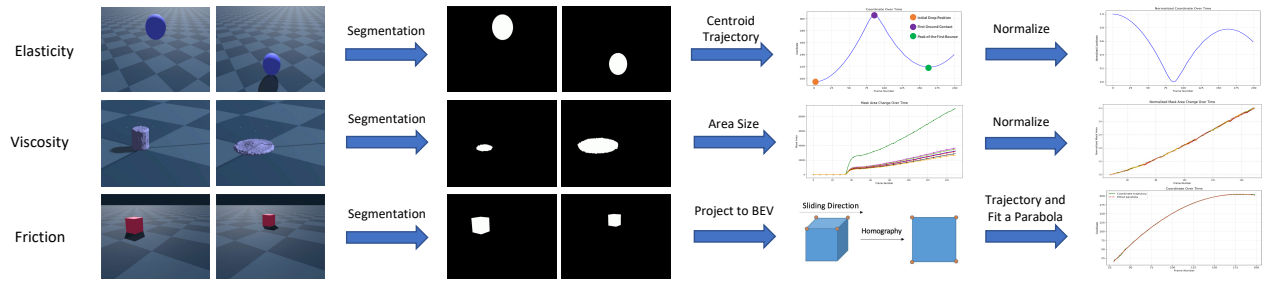

Figure 2. **Oracle methods for physical properties.** The objective in each case is to extract a measurement from the sequence that can directly be used to predict the property. For elasticity, we extract the centroid trajectory from segmentation masks, and then normalize the y-coordinates into 0-1; the ratio of bouncing to dropping height over the sequence indicates the elasticity. For viscosity, we calculate the area size in the image via segmentation masks, and then normalize the area sizes by the area in the frame when the liquid first touches the ground; the slope of the normalized area size sequence reflects the viscosity. For friction, we transform to a bird's eye view (using a homography transformation based on 4 corner points of the top surface of the sliding object), and fit a parabola $x = \alpha t^2 + \beta t + c$ to the transformed trajectory; the parabola coefficient α predicts the friction coefficient. For each video, we show the segmentation for two frames (left $\rightarrow$ right).

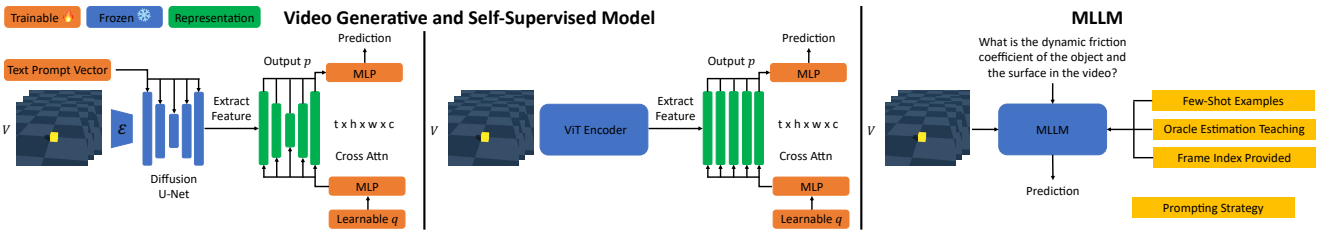

Figure 3. **Architectures for dynamic physical property prediction. Left**: video generative model as backbone; **Middle**: video self-supervised model as backbone; **Right**: multimodal large language model (MLLM). For the pre-trained video diffusion model (U-Net, left) and the pre-trained self-supervised model (ViT, middle), the representations are kept frozen, and a 'visual prompt' learns to infer the physical properties. For the MLLMs, the physical properties are inferred using a language prompt (right).

which was shown to be effective for capturing 3D physics in prior work [40, 55]. To aggregate the features, we introduce a learnable query vector q, which is first mapped to the different dimensions of the multi-scale features (see Supplementary Section A.2 for details), and then attends to the diffusion tokens (r_i) via cross-attention:

$$p = \sum_{i=1}^{t \times h \times w} \texttt{softmax}(q \cdot r_i) \cdot r_i \qquad (2)$$

The resulting vectors p from different layers are then mapped by another MLP network to a common dimension and average pooled to generate the final video feature representation P. To predict the physical properties, we train the text token of the generative model, together with the 'visual prompt' architecture that includes the query q and the MLPs.

Self-Supervised Model as Backbone. Here, we adopt a pre-trained self-supervised model (Figure 3, middle), namely V-JEPA-2 [3], as the visual backbone. The input video is passed through the model, and we extract feature tokens from all layers of the ViT encoder. Similar to the generative setting, we introduce a learnable query vector q to extract the video feature representation P from the ViT tokens via attentive pooling. Although the feature dimension at each ViT layer is the same, we still use a MLP network to map q to generate the query vector of each layer (keeping it similar to the generative setting in terms of MLP network architecture), and use another MLP network to map the output vectors p to a same dimension as the generative setting before average pooling them to get P. Please see Supplementary Section A.2 for more details.

4.2.2. Physical Property Prediction

Given the computed feature P from video foundation models, we train a MLP network to predict the physical properties using the synthetic video dataset training split. The network for each property is trained separately.

Absolute Value Prediction. Given the resulting video feature (P), we pass it through a MLP network γ to predict the absolute value χ of the physical property:

$$\chi = \gamma(P) \qquad (3)$$

For elasticity and friction, the absolute value prediction is supervised with L1 loss with the ground truth value; For

viscosity, as the ground truth values may have very different scales, *i.e.,* from $1e^{-5}$ to $1e^{-2}$, the absolute value prediction is trained with Log L1 loss, which calculates L1 loss between the log of the predicted value and the log of the ground truth value.

Relative Value Prediction. Given the resulting features for a pair of videos, P_1 and P_2, we concatenate them and formulate a binary classification problem, indicating which video has a larger physical property value via a MLP network γ:

$$\xi = \gamma([P_1, P_2]) \tag{4}$$

The binary prediction for all three tasks is trained with binary cross entropy loss with the binary ground truth.

Bridging the Sim2real Gap. Since our models are trained on synthetic datasets, they may not generalize well to real-world test videos due to the domain gap. To mitigate this sim-to-real gap, for both synthetic training and real test, we draw a red circle on each video frame, enclosing the full trajectory of the target object or liquid, as illustrated in Figure 4 (middle). The red circle is obtained automatically as a bounding ellipse enclosing the merged masks of the target object or liquid across all frames. This visual cue directs the model's attention to the relevant region [38], effectively signaling which object to focus on for physical reasoning. The red circle serves as a lightweight yet effective form of weak annotation that helps the model localize and interpret the dynamics of interest. Please refer to Supplementary Section G for the quantitative results demonstrating the effectiveness of drawing such red circles to mitigate the sim-to-real gap.

4.3. Multimodal Large Language Models

This section studies off-the-shelf multimodal large language models (MLLMs) for understanding dynamic physical properties from video. We explore various prompting strategies on state-of-the-art MLLMs, including Qwen2.5-VL-Max [18], GPT-4o [19], and Gemini 2.5 Pro [9], as illustrated in Figure 3 (right). Examples of the prompting strategies are provided in Supplementary Section E.

4.3.1. Preliminary

The MLLM receives video frames as visual input. The text prompt includes (1) a brief description of the target property—for example: "we are studying the viscosity of the liquid, where water is 1.0 and honey is 5000.0." This is followed by (2) a query, such as: "what is the viscosity value of the liquid in the video?" (absolute) or "which video shows a liquid with higher viscosity? please output a decision score between 0 and 1, indicating the likelihood that the first video exhibits a higher property value." (relative). All the following prompt strategies provide (1) and (2) by default.

4.3.2. Baseline Prompt

For *relative* tasks, we specify that the first n frames belong to the first video and the last n to the second. For *absolute* tasks, the default prompt is used. Supplementary Figure 8 and Figure 13 provide an example of *baseline prompt* for the absolute formulation and the relative formulation, respectively.

4.3.3. Oracle Estimation Teaching

For both *relative* and *absolute* settings, we provide the key cue to concentrate on from the Section 4.1 description to teach the MLLM how to estimate the properties step by step. Supplementary Figure 9 and Figure 14 provide an example of *oracle estimation teaching* for the absolute formulation and the relative formulation, respectively.

4.3.4. Few-Shot Examples

For both *relative* and *absolute* settings, we provide several examples, including the video input and desired ground truth. For fair comparison with visual prompting, we use examples in the synthetic training split. Supplementary Figure 10 and Figure 15 provide an example of *few-shot examples* for the absolute formulation and the relative formulation, respectively.

4.3.5. Frame Index Provided

For both *relative* and *absolute* settings, we input the text of the index of each frame along with the frames. In this way the MLLMs may have a better understanding about the temporal relations between the input video frames. Supplementary Figure 11 and Figure 16 provide an example of *frame index provided* for the absolute formulation and the relative formulation, respectively.

4.3.6. Black Frames in Between

This strategy is only used for the *relative* setting. We insert black frames between the two video segments to clearly separate them. In the prompt, we refer to the videos as the frames before and after the black frames, rather than as the first and last n frames. Supplementary Figure 17 provides an example of *black frames in between* for the relative formulation.

5. Experiments

Implementation Details. During oracle estimation, we train the GRU network with a learning rate of $1e^{-3}$ and the batch size is 128. For the generative and self-supervised video models, the backbones are frozen, the trainable parameters are optimised with a learning rate of $1e^{-5}$ and the batch size 16. For MLLMs, we perform promp selection, and use the best strategy that we find for each of the absolute and relative settings for the experiments. *Few-shot examples* and *oracle estimation teaching* work best for the absolute and relative settings, respectively, as they directly

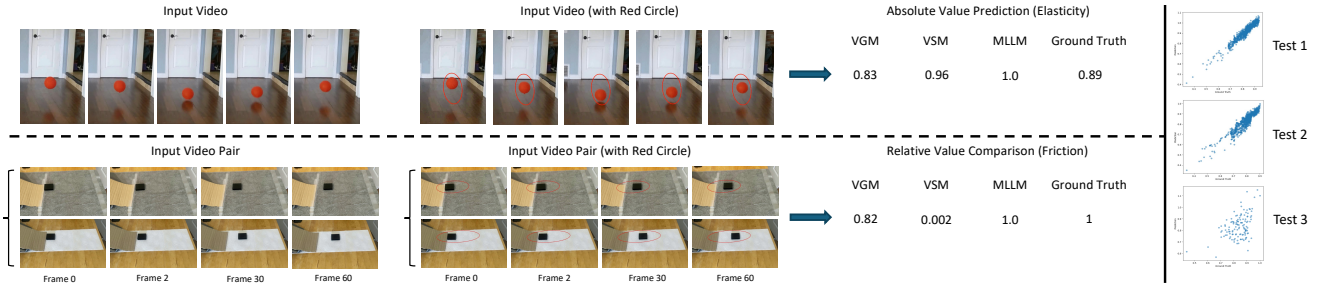

Figure 4. **Qualitative results. Top Left**: An example for elasticity absolute value prediction; **Bottom Left**: An example for friction relative value comparison. For each example, the original input video is shown on the left. A static red circle is overlaid in the center to highlight the full trajectory of the object on every frame, shown in the middle. Model predictions are shown on the right, including results from the Video Generative Model (VGM), Video Self-Supervised Model (VSM), and a MLLM (Gemini in this case). For the relative formulation, the ground truth value of '1' indicates that the first (top) video has larger dynamic friction coefficient than the second video. In this example, the initial velocity of the lego brick in the two videos is similar (note the same displacement from frame 0 to 2), but the velocity reduces to 0 at frame 30 in the first video, while the object is still moving in frame 30 to 60 in the second video. **Right**: Scatter plots of prediction vs ground truth for the elasticity property from the V-JEPA-2 model.

provide the model with more context information about the properties. Please refer to Supplementary Section D for the comparison results and analysis. All models are trained on H100/A6000/A40 GPUs. Please refer to Supplementary Section A for more implementation details.

Evaluation Metrics. For *relative value comparison*, we report the ROC AUC score; for *absolute value prediction*, we use the Pearson Correlation Coefficient between the prediction and ground truth as this automatically calibrates the predictions to the scale of the ground truth. Please refer to Supplementary Section A.4 for more details and motivations on the evaluation metrics.

5.1. Results for Relative Value Comparison

Table 1 (left) shows relative value comparison results across physical properties and model types. The oracle estimator performs nearly perfectly on `test-1` and `test-2`, and strongly on `test-3`, indicating that the task is largely solvable using visual cues, geometry, and physics. Both generative and self-supervised video models achieve strong results on synthetic splits (`test-1` and `test-2`). Notably, they can also generalize well to the real-world split (`test-3`) for viscosity and elasticity, which rely on simple height ratios and expansion. However, friction proves more challenging. Models trained on synthetic data struggle to generalize, likely due to the fact that reliance on visual references (*e.g.*, ground plane grids) is absent in real videos, and due to friction's inherent complexity involving higher-order motion and projective geometry of the viewpoint. To further confirm, we introduce an additional real-world training split for friction videos with disjoint objects and surfaces from the test set (see Supplementary Section B.2 for more details). Fine-tuning the visual prompting architecture on this data improves performance on the real test split, as shown by the * values in Table 1. Multimodal large language models (MLLMs), though not working very well with *Baseline Prompt* (see Supplementary Section D), when prompted properly, also perform well, especially on real videos, which are more *in-distribution* for them – while on synthetic splits, their performance drops significantly. This is likely due to the fact that the models tend to leverage semantic cues, *e.g.,* the type of liquid or the category of object and surface, rather than visual motion.

5.2. Results for Absolute Value Prediction

Table 1 (right) shows results for absolute value prediction across physical properties and methods. This task is more challenging than relative comparison, as models must regress quantitative physical values rather than compare video pairs from the same viewpoint. Similar to the relative setting, the oracle estimator achieves near-perfect performance on `test-1` and `test-2`, and strong performance on `test-3`, confirming that the task is largely solvable through visual cues, multi-view geometry, and physical laws. We highlight several key observations: (i) **comparable performance across backbones**. Despite being trained for generative tasks, video generative models perform on par with self-supervised models when predicting dynamic physical properties. (ii) **friction remains challenging**. Similar to the relative setting, both generative and self-supervised models struggle with friction estimation. Performance again improves with domain adaptation. (iii) **MLLMs better on real test split than synthetic**. MLLMs continue to perform better on the real test split than synthetic test splits, benefiting from their familiarity with real-world visual semantics. (iv) **greater gap from oracle**. The performance gap between video foundation models and the oracle is more pronounced here than in the relative setting, indicating that accurate physical value regression remains a significant challenge for current video models. (v) **more difficult to generalise to real situations.** Compared to the relative setting, video generative model and video self-

Table 1. **Results for relative value comparison and absolute value prediction.** Left: ROC AUC scores for relative comparisons (range $[0, 1]$). Right: Pearson correlation coefficients for absolute predictions (range $[-1, 1]$). * indicates results after domain adaptation using a disjoint real training set. `test-1` is the synthetic in-distribution test split; `test-2` is the synthetic out-of-distribution test split; `test-3` is the real-world test split.

Property	Method	Relative – ROC AUC			Absolute – Pearson Corr.		
		Test-1	Test-2	Test-3	Test-1	Test-2	Test-3
Elasticity	**Oracle**	1.00	1.00	1.00	0.99	0.98	0.87
	Video Generative Model	1.00	0.98	0.84	0.92	0.82	0.07
	Video Self-Supervised Model	0.89	0.96	0.77	0.96	0.93	0.47
	Qwen2.5VL-max	0.59	0.50	0.54	-0.05	0.11	0.16
	GPT-4o	0.51	0.66	0.62	0.19	0.11	0.30
	Gemini-2.5-pro	0.64	0.80	0.47	0.04	0.15	0.24
Viscosity	**Oracle**	0.99	1.00	1.00	0.99	0.98	0.80
	Video Generative Model	1.00	1.00	1.00	0.99	0.95	0.76
	Video Self-Supervised Model	1.00	1.00	0.99	1.00	0.97	0.79
	Qwen2.5VL-max	0.64	0.61	0.86	0.16	0.06	0.02
	GPT-4o	0.63	0.59	0.99	0.18	0.08	0.55
	Gemini-2.5-pro	0.48	0.69	0.95	-0.06	-0.05	0.60
Friction	**Oracle**	1.00	1.00	0.87	0.99	1.00	0.83
	Video Generative Model	0.98	0.89	0.47	0.95	0.78	0.21
	+ Domain Adaptation	–	–	0.74*	–	–	0.82*
	Video Self-Supervised Model	1.00	0.97	0.58	0.71	0.58	0.28
	+ Domain Adaptation	–	–	0.63*	–	–	0.71*
	Qwen2.5VL-max	0.50	0.62	0.80	0.03	0.14	0.06
	GPT-4o	0.34	0.42	0.67	-0.10	0.03	0.38
	Gemini-2.5-pro	0.54	0.59	0.97	-0.03	-0.05	0.12

supervised model exhibit a significantly larger performance drop from synthetic to real test splits (especially for elasticity and viscosity). This demonstrates the absolute setting is challenging.

5.3. Qualitative Results

Figure 4 (left) shows qualitative examples comparing model predictions across different tasks. In the **first row**, we illustrate an example from the elasticity absolute value prediction task. The video generative model, self-supervised model, and MLLMs predict values of 0.83, 0.96, and 1.0, respectively—all reasonably close to the ground-truth value of 0.89. In the **second row**, we present a friction relative value comparison task. The input consists of two videos, where the first exhibits a higher dynamic friction coefficient than the second. Both the video generative model and the MLLM correctly assign high likelihoods to this relationship (0.82 and 1.0, respectively), aligning with the ground truth. In contrast, the self-supervised model incorrectly predicts the reverse and does so with high confidence. Figure 4 (right) shows examples of the scatter plots for the absolute value prediction. More specifically, we show the scatter plots of video self-supervised model on the three test splits. It can be observed that the performance degrades from `test-1` to `test-3`, as `test-1` is of the same distribution as the synthetic training split, while `test-2` is out-of-distribution synthetic test and `test-3` is for real evaluation. We provide more scatter plots in Supplementary Section F.

6. Conclusion

We investigate the task of inferring dynamic physical properties—elasticity, viscosity, and friction—from videos. To support this, we introduce a benchmark dataset with ground-truth annotations and evaluate a range of video foundation models under both absolute prediction and relative comparison settings. We adopt a simple architecture to extract physical cues from off-the-shelf generative and self-supervised video models, and explore prompting strategies to elicit predictions from MLLMs. Experiments show that generative and self-supervised models have similar and reasonable performance. This confirms the findings in previous works, where the prior of video generative models can be used to estimate the material field of objects [57], and the video self-supervised models can be used for robots to interact with the physical world [3]. MLLMs perform worse overall but improve with more informative prompting, especially on real-world data. The worse performance of MLLMs is consistent with previous work [13], where it is observed that the visual information is not properly fused in the language model. However, all models fall short of the oracle, particularly in absolute value prediction. These results highlight the need to enhance physical reasoning in video models—a key direction for future research.

References

[1] Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Adcances on Neural Information Processing Systems (NeurIPS)*, 2016. 2

[2] Jan Assen, Pascal Barla, and Roland Fleming. Visual features in the perception of liquids. *Current Biology*, 2018. 2

[3] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. 1, 5, 8

[4] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021. 2

[5] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. *OpenReview*, 2023. 1

[6] 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. *arXiv preprint arXiv:2106.08261*, 2021. 2

[7] Florian Bordes, Quentin Garrido, Justine T Kao, Adina Williams, Michael Rabbat, and Emmanuel Dupoux. Intphys 2: Benchmarking intuitive physics understanding in complex synthetic environments. *arXiv preprint arXiv:2506.09849*, 2025.

[8] Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. Physbench: Benchmarking and enhancing vision-language models for physical world understanding. *International Conference on Learning Representation (ICLR)*, 2025. 2

[9] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 1, 6

[10] Mingyu Ding, Zhenfang Chen, Tao Du, Ping Luo, Josh Tenenbaum, and Chuang Gan. Dynamic visual reasoning by learning differentiable physics models from video and language. *Advances In Neural Information Processing Systems (NeurIPS)*, 2021. 2

[11] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 2

[12] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[13] Stephanie Fu, Tyler Bonnen, Devin Guillory, and Trevor Darrell. Hidden in plain sight: Vlms overlook their visual representations. *arXiv preprint arXiv:2506.08008*, 2025. 8

[14] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2024. 2

[15] Alejandro Castañeda Garcia, Jan Warchocki, Jan van Gemert, Daan Brinks, and Nergis Tomen. Learning physics from video: Unsupervised physical parameter estimation for continuous dynamical systems. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025. 2

[16] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning (ICML)*, 2019. 2

[17] Richard I. Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 4

[18] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024. 1, 6

[19] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1, 6

[20] Krishna Murthy Jatavallabhula, Miles Macklin, Florian Golemo, Vikram Voleti, Linda Petrini, Martin Weiss, Breandan Considine, Jerome Parent-Levesque, Kevin Xie, Kenny Erleben, Liam Paull, Florian Shkurti, Derek Nowrouzezahrai, and Sanja Fidler. gradsim: Differentiable simulation for system identification and visuomotor control. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[21] Takahiro Kawabe and Shin'ya Nishida. Seeing jelly: Judging elasticity of a transparent object. In *Proceedings of the ACM Symposium on Applied Perception*, 2016. 2

[22] Takahiro Kawabe, Kazushi Maruya, Roland Fleming, and Shin'ya Nishida. Seeing liquids from visual motion. *Vision Research*, 2014. 2

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 12

[24] Yunzhu Li, Toru Lin, Kexin Yi, Daniel Bear, Daniel L.K. Yamins, Jiajun Wu, Joshua B. Tenenbaum, and Antonio Torralba. Visual grounding of learned physical models. In *International Conference on Machine Learning (ICML)*, 2020. 2

[25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 2024. 12

[26] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024. 1

[27] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. Vdt: An empirical study on video diffusion with transformers. *arXiv preprint arXiv:2305.13311*, 2023. 2

[28] J Norman, Elizabeth Wiesemann, Hideko Norman, M Taylor, and Warren Craft. The visual discrimination of bending. *Perception*, 2007. 2

[29] Vivian Paulun, Takahiro Kawabe, Shin'ya Nishida, and Roland Fleming. Seeing liquids from static snapshots. *Vision research*, 2015.

[30] Vivian Paulun, Filipp Schmidt, Jan Assen, and Roland Fleming. Shape, motion, and optical cues to stiffness of elastic objects. *Journal of Vision*, 2017.

[31] Vivian C. Paulun and Roland W. Fleming. Visually inferring elasticity from the motion trajectory of bouncing cubes. *Journal of Vision*, 2020. 2

[32] 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. In *International Conference on Learning Representations (ICLR)*, 2025. 12

[33] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, Yuda Xiong, Hao Zhang, Feng Li, Peijun Tang, Kent Yu, and Lei Zhang. Grounding dino 1.5: Advance the "edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024.

[34] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 12

[35] Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. Intphys: A framework and benchmark for visual intuitive physics reasoning. *arXiv preprint arXiv:1803.07616*, 2018. 2

[36] Prafull Sharma, Julien Philip, Michaël Gharbi, Bill Freeman, Fredo Durand, and Valentin Deschaintre. Materialistic: Selecting similar materials in images. *ACM Transactions on Graphics (TOG)*, 2023. 2

[37] Hui Shen, Taiqiang Wu, Qi Han, Yunta Hsieh, Jizhou Wang, Yuyue Zhang, Yuxin Cheng, Zijian Hao, Yuansheng Ni, Xin Wang, et al. Phyx: Does your model have the" wits" for physical reasoning? *arXiv preprint arXiv:2505.15929*, 2025. 2

[38] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 6

[39] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, 2012. 2

[40] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 5

[41] Hsiao-Yu Tung, Mingyu Ding, Zhenfang Chen, Daniel Bear, Chuang Gan, Josh Tenenbaum, Dan Yamins, Judith Fan, and Kevin Smith. Physion++: Evaluating physical scene understanding that requires online inference of different physical properties. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2

[42] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. Masked conditional video diffusion for prediction, generation, and interpolation. *arXiv preprint arXiv:2205.09853*, 2022. 2

[43] Bin Wang, Paul Kry, Yuanmin Deng, Uri Ascher, Hui Huang, and Baoquan Chen. Neural material: Learning elastic constitutive material and damping models from sparse data. *arXiv preprint arXiv:1808.04931*, 2018. 2

[44] Kun Wang, Mridul Aanjaneya, and Kostas Bekris. A first principles approach for data-efficient system identification of spring-rod systems via differentiable physics engines. In *Learning for Dynamics and Control*, 2020. 2

[45] Tianyu Wang, Xiaowei Hu, Qiong Wang, Pheng-Ann Heng, and Chi-Wing Fu. Instance shadow detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[46] Tianyu Wang, Xiaowei Hu, Chi-Wing Fu, and Pheng-Ann Heng. Single-stage instance shadow detection with bidirectional relation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[47] Wikipedia contributors. Coefficient of restitution — wikipedia, the free encyclopedia, 2025. 17

[48] Wikipedia contributors. Viscosity — wikipedia, the free encyclopedia, 2025. 17

[49] Wikipedia contributors. Wetting — wikipedia, the free encyclopedia, 2025. 17

[50] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems (NeurIPS)*, 2015. 2

[51] Jiajun Wu, Joseph J Lim, Hongyi Zhang, Joshua B Tenenbaum, and William T Freeman. Physics 101: Learning physical object properties from unlabeled videos. In *British Machine Vision Conference (BMVC)*, 2016. 2

[52] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision (ECCV)*, 2024. 1, 4

[53] Guanqi Zhan, Weidi Xie, and Andrew Zisserman. A tri-layer plugin to improve occluded detection. *British Machine Vision Conference (BMVC)*, 2022. 2

[54] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. Amodal ground truth and completion in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[55] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. A general protocol to probe large vision models for 3d physical understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 5

[56] Chenyu Zhang, Daniil Cherniavskii, Antonios Tragoudaras, Antonios Vozikis, Thijmen Nijdam, Derck WE Prinzhorn, Mark Bodracska, Nicu Sebe, Andrii Zadaianchuk, and Efstratios Gavves. Morpheus: Benchmarking physical reasoning of video generative models with real physical experiments. *arXiv preprint arXiv:2504.02918*, 2025. 2

[57] 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 (ECCV)*, 2024. 8

[58] Xian Zhou, Yiling Qiao, Zhenjia Xu, Tsun-Hsuan Wang, Zhehuan Chen, Juntian Zheng, Ziyan Xiong, Yian Wang, Mingrui Zhang, Pingchuan Ma, Yufei Wang, Zhiyang Dou, Byungchul Kim, Yunsheng Tian, Yipu Chen, Xiaowen Qiu, Chunru Lin, Tairan He, Zilin Si, Yunchu Zhang, Zhanlue Yang, Tiantian Liu, Tianyu Li, Kashu Yamazaki, Hongxin Zhang, Huy Ha, Yu Zhang, Michael Liu, Shaokun Zheng, Zipeng Fu, Qi Wu, Yiran Geng, Feng Chen, Milky, Yuanming Hu, Guanya Shi, Lingjie Liu, Taku Komura, Zackory Erickson, David Held, Minchen Li, Linxi "Jim" Fan, Yuke Zhu, Wojciech Matusik, Dan Gutfreund, Shuran Song, Daniela Rus, Ming Lin, Bo Zhu, Katerina Fragkiadaki, and Chuang Gan. Genesis: A universal and generative physics engine for robotics and beyond, 2024. 2, 3