Supplementary

A. Additional Implementation Details

As mentioned in Section 5 of the main paper, in this section we provide more implementation details. Please refer to our code in the .zip file for more details.

A.1. Computing Infrastructure

The video generative and video self-supervised models are trained on a single H100/A6000/A40 GPU. The models are trained much faster on H100, and much slower on A6000 and A40, but all of them can be used to train and inference our models. The GPU memory of H100 GPU is 96GB, and 48GB for A6000/A40 GPU. The CPU memory associated with the GPU is 120GB for the H100 GPU, and 90GB for A6000/A40 GPU. The experiments are conducted on a Linux cluster. We provide the names and versions of the relevant software libraries and frameworks in the code.

A.2. Dimension of Video Feature Extraction Vectors and Details of MLPs

We set the dimension of the learnable query vector q to be 2560 for DynamiCrafter or V-JEPA-2. For DynamiCrafter, q is mapped to the dimensions of different layers for cross-attention with different layers of pre-trained feature tokens, by a set of MLPs $f_1^G : \mathbb{R}^{2560} \rightarrow \mathbb{R}^{(320^{\times 4}, 640^{\times 3}, 1280^{\times 12}, 640^{\times 3}, 320^{\times 3})}$; For V-JEPA-2, q is mapped to the dimensions of different layers for cross-attention with different layers of pre-trained feature tokens, by a set of MLPs $f_1^S : \mathbb{R}^{2560} \rightarrow \mathbb{R}^{(1024^{\times 16})}$. The resulting vectors p therefore can be represented as $\mathbb{R}^{(320^{\times 4}, 640^{\times 3}, 1280^{\times 12}, 640^{\times 3}, 320^{\times 3})}$ for DynamiCrafter and $\mathbb{R}^{(1024^{\times 16})}$ for V-JEPA-2. The p vectors are then mapped by another set of MLPs $f_2^G : \mathbb{R}^{(320^{\times 4}, 640^{\times 3}, 1280^{\times 12}, 640^{\times 3}, 320^{\times 3})} \rightarrow \mathbb{R}^{(2560^{\times 25})}$ for DynamiCrafter, and $f_2^S : \mathbb{R}^{(1024^{\times 16})} \rightarrow \mathbb{R}^{(2560^{\times 16})}$ for V-JEPA-2 and then average pooled to get $P \in \mathbb{R}^{2560}$.

A.3. Details of Video Input

We uniformly sample 16 frames per video as input to all the models for fair comparison. The 16 frames are uniformly sampled so that the physics process we want to study is properly reflected with the sampled 16 frames, *e.g.,* the dropping and bouncing of the ball, the expansion of the liquid, and the slowing-down sliding process of the object. For construction of relative pairs of videos, we randomly get a list of different viewpoints first, and then for each viewpoint, we randomly generate m videos and then randomly sample pairs from $m \times (m-1)$ possible video pairs. The binary ground truth for the pair is obtained via comparing the property values of the two videos.

A.4. Motivation and Details of Evaluation Metrics

We use the ROC AUC score for the relative formulation, as it is a binary classification problem, and AUC is a good evaluation metric to reflect the model's performance over different decision thresholds. ROC AUC is computed as the area under the receiver-operating-characteristic curve, *i.e.*, the integral of the true-positive rate versus the false-positive rate over all possible classification thresholds. We use the Pearson Correlation Coefficient for the absolute formulation, as it can reflect how the model's predicted values correlate to the ground truth values – and our goal here is the *value ordering*, rather than their absolute prediction, as accurately determining the ground truth values is difficult, particularly for viscosity. Also, the correlation is more forgiving than absolute error for out of distribution prediction (between the train and test sets).

A.5. Segmentation Masks and Corner Detections for Oracle Estimation

In the oracle estimation, we need to segment the target object or liquid. For the synthetic videos, the segmentation masks can be directly obtained from the simulator. For the real videos, we obtain the segmentations by using the pre-trained Grounded SAM 2 [23, 25, 32–34], with text prompts such as 'falling ball' (for elasticity), and 'sliding' + the names of different sliding objects (for friction). For viscosity, we only need the mask of the liquid on the plate surface, but the presence of the liquid column interferes – and directly using the liquid's name as a prompt for Grounded SAM 2 makes segmentation difficult. Therefore, we first segment the plate, and then apply morphological processing (taking the enclosed central region of the plate and using a closing operation to remove the liquid columns) to obtain the mask of the liquid on the plate surface. After getting the automatically predicted masks, we manually filter and adjust them to make sure they are of good quality. Apart from segmentation masks, for the friction oracle estimation, we also need to detect the corner points of the sliding cube. For the synthetic videos, we annotate the corners with a different color so we can easily detect them. For the real videos, we annotated the corner positions manually as they are difficult to obtain automatically.

B. Dataset Details

As mentioned in Section 3 of the main paper, in this section we provide further details regarding the collection of synthetic and real datasets.

B.1. Synthetic Datasets

As described in Section 3 of the main paper, the simulator uses two distinct domains of nuisance parameters: $\mathcal{A}_1$ and $\mathcal{A}_2$. The `train` and `test-1` splits are generated by sampling from $\mathcal{A}_1$, while `test-2` is generated from $\mathcal{A}_2$. Below, we detail the differences between these domains for each dynamic physical property.

The Genesis simulator operates in a world coordinate system where gravity points in the $-z$ direction, and physical processes are centered around the origin $(0, 0, 0)$. The camera position is defined by three parameters: height h (controls the z coordinate), radius R (distance from the $(0, 0)$ point in the xy-plane), and angle α (deviation from the $+x$ direction). Camera orientation is further specified by the 3D point (x_l, y_l, z_l) that the camera looks at. Object and liquid colors are defined by RGB values (r, g, b). Lighting remains fixed from the $+x$ direction, meaning changes in camera viewpoint also affect lighting conditions on the object.

Table 2, Table 3 and Table 4 detail the parameter settings for each physical property. All parameter ranges are chosen to ensure the visibility of the studied phenomena—e.g., the drop-and-bounce motion of a ball—in the synthetic videos.

Table 2. **Parameter Ranges for Elasticity**. Values are randomly sampled per domain if it is a range. Top: Nuisance parameters; Bottom: The target dynamic physical property we study.

Parameter	$\mathcal{A}_1$	$\mathcal{A}_2$
R	1.5	1.5
h	$(0.5, 1.5)$	$(0.25, 0.5)$
α	$(0, \frac{1}{2}\pi)$	$(\frac{1}{2}\pi, 2\pi)$
x_l	$(-0.1, 0.1)$	$(0.1, 0.2)$
y_l	$(-0.1, 0.1)$	$(-0.2, -0.1)$
z_l	$(0.05, 0.27)$	$(-0.05, 0.05)$
r	$(0, 1)$	0
g	$(0, 1)$	0
b	0	$(0, 1)$
Drop height	$(0.25, 0.4)$	$(0.4, 0.5)$
Ball radius	0.1	
Elasticity	$(0, 1)$	

B.2. Real Datasets Details

Elasticity Dataset. This dataset contains video clips sourced from the Internet, capturing a variety of ball types being dropped—*e.g.*, basketball, tennis ball, soccer ball, rubber ball, balloon (air-filled), exercise ball, medicine ball, marble, and tomato. Ground truth elasticity values range from 0.44 (tomato) to 0.98 (tennis ball).

Viscosity Dataset. We include 12 different liquids: coffee, vinegar, cola, wine, cooking wine, whole milk, hot chocolate, dark soy sauce, smoothie, sesame oil, cream, and maple syrup. Ground truth viscosity values are obtained from online sources. For cases where the value is reported as a range, we use the midpoint as the ground truth. Values range from 1.2 (coffee) to 225 (maple syrup).

Friction Dataset. This dataset contains 5 objects—plastic disk, plastic LEGO brick, paper box, metal pencil box, and wooden box—and 6 surfaces: gray towel, kitchen paper, tablecloth, red towel, wooden table, and cardboard.

The training split includes:

- **Objects:** paper box, metal pencil box, wooden box
- **Surfaces:** red towel, wooden table, cardboard

The testing split includes:

- **Objects:** plastic disk, LEGO brick

Table 3. **Parameter Ranges for Viscosity**. Values are randomly sampled per domain if it is a range. Top: Nuisance parameters; Bottom: The target dynamic physical property we study.

Parameter	$\mathcal{A}_1$	$\mathcal{A}_2$
R	1.5	1.5
h	$(0.5, 1.5)$	$(0.25, 0.5)$
α	$(0, \frac{1}{2}\pi)$	$(\frac{1}{2}\pi, 2\pi)$
x_l	$(-0.1, 0.1)$	$(0.1, 0.2)$
y_l	$(-0.1, 0.1)$	$(-0.2, -0.1)$
z_l	$(0.05, 0.27)$	$(-0.05, 0.05)$
r	$(0, 1)$	0
g	$(0, 1)$	0
b	0	$(0, 1)$
Drop height	0.056	
Liquid column height	0.1	
Liquid column radius	0.05	
Viscosity	$(5\text{e}{-5}, 1\text{e}{-2})$	

Table 4. **Parameter Ranges for Friction**. Values are randomly sampled per domain if it is a range. Top: Nuisance parameters; Bottom: The target dynamic physical property we study. (x_0, y_0): initial position of the sliding cube; (v_0^x, v_0^y): initial velocity of the sliding cube.

Parameter	$\mathcal{A}_1$	$\mathcal{A}_2$
R	1.5	1.5
h	$(0.5, 1.5)$	$(0.25, 0.5)$
α	$(0, \frac{1}{2}\pi)$	$(\frac{1}{2}\pi, 2\pi)$
x_l	$(-0.1, 0.1)$	$(0.1, 0.2)$
y_l	$(-0.1, 0.1)$	$(-0.2, -0.1)$
z_l	$(-0.1, 0.12)$	$(-0.14, -0.1)$
r	$(0, 1)$	$(0, 1)$
g	$(0, 1)$	$(0, 1)$
b	0	$(0, 1)$
x_0	$(-0.1, 0.1)$	$(-0.15, -0.1)$
y_0	$(-0.1, 0.1)$	$(0.1, 0.15)$
v_0^x	$(0.6, 1.0)$	$(1.0, 1.2)$
v_0^y	$(0.6, 1.0)$	$(1.0, 1.2)$
Motion direction	v_0^x or v_0^y with prob. 0.5	
Cube size	0.1	
Friction coeff.	$(0, 0.2)$	

- **Surfaces:** gray towel, kitchen paper, tablecloth

Figure 5 shows close-up images of all objects and surfaces. Ground truth friction values range from 0.105 (pencil box on plastic paperboard) to 0.544 (LEGO on gray towel). The ground truth dynamic friction coefficient values are measured using a spring dynamometer by dragging the object at constant speed, as mentioned in the main paper. All datasets will be made publicly available upon paper acceptance under the CC-BY-4.0 license.

B.3. Devices for Real Dataset Collection

As mentioned in Section 3.1 of the main paper, we show the devices that we used to collect our real datasets in Figure 6. The funnel (left) and the funnel holder (middle-left) are used to collect the viscosity real dataset. The spring dynamometer (middle-right) is used to measure the ground truth dynamic friction coefficient in the collection of the friction real dataset. The slope (right) is used in the friction experiment, where we slides the object down the slope to give it an initial velocity on

CVPR #4

Figure 5. **Objects and surfaces in the *friction* real dataset.** Top: Objects used for friction real dataset collection; Bottom: Surfaces used for friction real dataset collection.

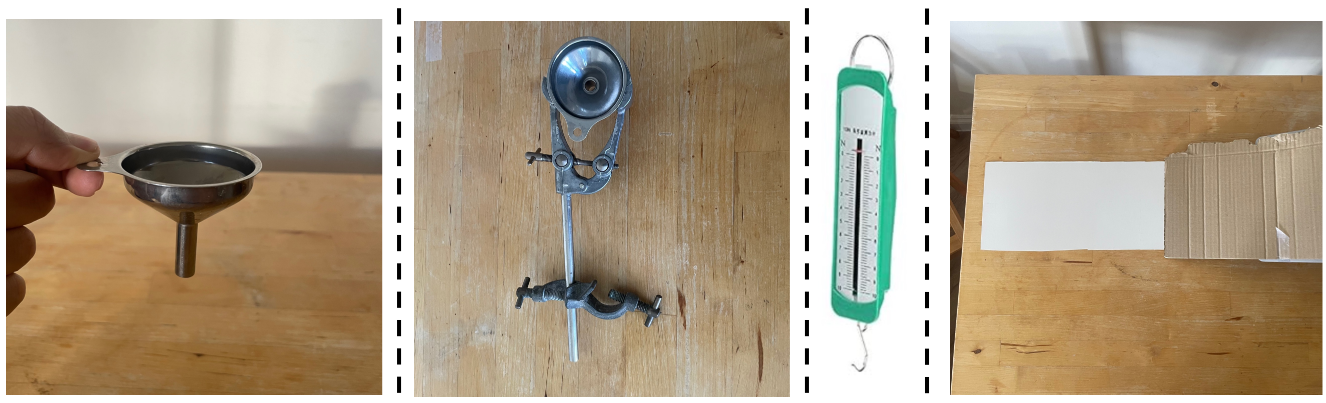

Figure 6. **Devices used to collect real datasets. Left**: The funnel used in the collection of the viscosity real dataset; **Middle-left**: The funnel holder used in the collection of the viscosity real dataset; **Middle-right**: The spring dynamometer used to measure the ground truth dynamic friction coefficient in the collection of the friction real dataset; **Right**: The slope used to give the objects an initial velocity on the horizontal surface in the collection of the friction real dataset.

the horizontal surface.

C. Derivation of Oracle Estimations Equations

As mentioned in Section 3.1 of the main paper, in this section, we give the derivation for the oracle estimations in the main paper, based on standard definitions for the properties and the laws of physics. Note, we only require the oracle values for each property to be determined up to a common scale, since the correlation used to evaluate performance is unaffected by the overall scale.

Elasticity. The *coefficient of elasiticity* or *coefficient of restitution* e is defined as

$$e = \frac{v_{\text{before impact}}}{v_{\text{after impact}}}$$

where v is the magnitude of the velocity. In our case the ball is dropped from rest at a height h_{drop} onto a horizontal surface, and bounces in the vertical direction to the height h_{bounce}. It can be shown (see [47]) that in this case:

$$e = \sqrt{\frac{h_{\text{bounce}}}{h_{\text{drop}}}} \tag{5}$$

Viscosity. *Viscosity* is a physical property that characterizes a fluid's internal resistance to flow [48]. In our case, we study the case that the liquid is expanding on the ground plane. According to the spreading dynamics [49] of liquid, the radius (thus the area) of the liquid is an inverse function of the viscosity, given other parameters controlled, such as the density of the liquid ρ, the diameter (thus the volume) of the liquid column D, and the dropping velocity v of the liquid column when it reaches the ground. In our case, we control D as we use a funnel with a fixed nozzle diameter to produce a consistent liquid column and we always pour the same volume of liquid into the funnel; we control v as we use a funnel holder that allows us to fix the height from which the liquid is poured; ρ is roughly controlled as the liquids are of similar density, ranging from 0.92 (*e.g.,* sesame oil) to 1.05 (*e.g.,* smoothie), except for maple syrup which is 1.32, but as the ground truth viscosity of maple syrup is much higher than other liquids, this variation will not influence much when we calculate the Pearson Correlation Coefficient between predictions and ground truth values. Therefore, we assume

$$\mu \propto \frac{1}{(d(A(t))/dt)^{\alpha}} \tag{6}$$

where μ is the viscosity of the liquid, and $A(t)$ is the liquid area size as a function of time t. In practice, we try with $\alpha = 1$, *i.e.,* $\mu \propto \frac{1}{d(A(t))/dt}$ and gets reasonable oracle test results, so we set $\alpha = 1$ for our oracle estimations.

Friction. If F is the dynamic friction force acting on the object, then the dynamic friction coefficient μ_k is defined by the equation $F = \mu_k \times$ normal force on the object. In our case, the object moves on a horizontal surface, and the normal force is the weight of the object, so $F = \mu_k mg$, where m is the mass of the object, and g is the gravity acceleration. From Newton's Second Law $F = ma$, we therefore have $a = \mu_k g$, *i.e.,*

$$\mu_k = \frac{a}{g} \tag{7}$$

where a is the acceleration of the object.

CVPR
#4

D. Ablation for Different Strategies of MLLM prompting

As mentioned in Section 5 of the main paper, we conduct an ablation study on the elasticity task to identify the most effective prompting strategy for MLLMs, using **Gemini 2.5 Pro** due to its strong performance in video understanding and visual reasoning. Results for the absolute and relative formulations are shown in Table 5 and Table 6, respectively. Due to the high computational cost of MLLM inference, we perform the ablation on a randomly selected subset of 20 samples per test split. The results show that the **Few-Shot Examples** strategy performs best for the absolute formulation, while **Oracle Estimation Teaching** is most effective for the relative formulation.

In Section E, we provide examples for each of the prompting strategies and detailed analysis regarding the influence of each strategy to the final performance.

Table 5. **Absolute prediction results for different MLLM prompting strategies**. We conduct the study on Gemini 2.5 Pro for the elasticity task.

Strategy	Test-1	Test-2	Test-3	Avg
Baseline	-0.03	0.26	0.06	0.10
+ Frame Index Provided	0.06	0.35	0.55	0.32
+ Few-Shot Examples	0.39	0.34	0.24	**0.33**
+ Oracle Estimation Teaching	0.19	0.14	0.24	0.19

Table 6. **Relative comparison results for different MLLM prompting strategies**. We conduct the study on Gemini 2.5 Pro for the elasticity task.

Strategy	Test-1	Test-2	Test-3	Avg
Baseline	0.51	0.74	0.55	0.60
+ Black Frames in Between	0.56	0.72	0.63	0.64
+ Frame Index Provided	0.54	0.80	0.52	0.62
+ Few-Shot Examples	0.43	0.65	0.52	0.54
+ Oracle Estimation Teaching	0.63	0.79	0.54	**0.65**

E. Examples of Different Prompting Strategies

As mentioned in Section 4.3 of the main paper, in this section we provide examples of different prompting strategies for both the **absolute formulation** and the **relative formulation**.

The examples of **absolute formulation** are provided in Figure 7 to Figure 11. More specifically, Figure 7 shows the visual input to the MLLM; Figure 8 shows the prompt and model output of *baseline prompt*; Figure 9 shows the prompt and model output of *oracle estimation teaching*; Figure 10 shows the prompt and model output of *few-shot examples*; Figure 11 shows the prompt and model output of *frame index provided*.

It can be observed that:

- **Baseline Prompt.** The initial state of object motion is incorrectly recognized from the beginning.
- **Oracle Estimation Teaching.** Although the model strictly follows the oracle's step-by-step guidance, an incorrect identification of the peak in the third step leads to a significantly inaccurate final prediction.
- **Few-Shot Examples.** The ground-truth examples provided in the few-shot setting serve as effective calibration signals, leading to notably improved performance.
- **Frame Index Provided.** Providing frame indices helps the model better interpret the motion process. However, estimating the final value based solely on this information remains challenging.

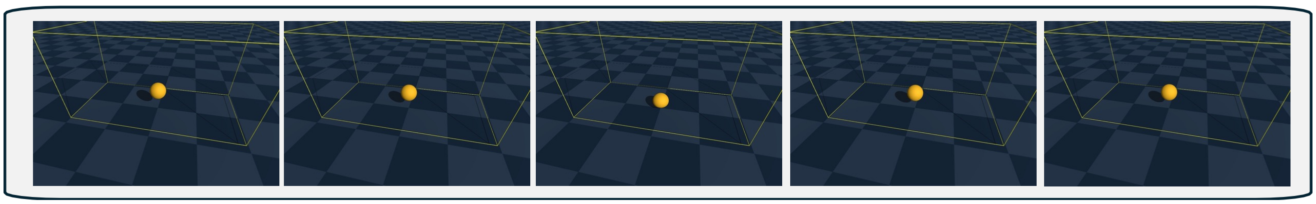

Figure 7. **Visual input for the *absolute formulation*.** The example is on Gemini for the elasticity property. Five representative frames are shown for the input video.

```
system_prompt = {

'role': 'system', 'content': 'You are an expert in elasticity analysis with the ability to analyze
video frames and determine elasticity values. You MUST provide a specific elasticity estimate.
Never say you cannot analyze the images.'

'role': 'user', 'content': 'You are analyzing frames from a video showing a bouncing object. Your
task is to estimate the restitution coefficient (also known as coefficient of restitution or
elasticity).

The restitution coefficient is the ratio of separation speed to approach speed during collision:
- Perfectly elastic collision: 1.0 (ball bounces back with same speed)
- Perfectly inelastic collision: 0.0 (ball sticks to surface, no bounce)
- Typical values: 0.1-0.9 for most materials

If the object bounces back significantly → restitution coefficient closer to 1.0
If the object barely bounces or sticks → restitution coefficient closer to 0.0

YOUR RESPONSE MUST END WITH A SPECIFIC RESTITUTION COEFFICIENT VALUE IN THIS FORMAT:
"Estimated restitution coefficient value: X "

Where X is a number between 0 and 1 (e.g., 0.7, 0.3, 0.85).' }

visual_input = {
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame2}"}
 ...
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"}
}

[Output]

Based on my analysis of the provided frames, I will now estimate the restitution coefficient.

The frames show a yellow sphere suspended or resting within a defined space. There is no visible
downward motion or approach towards the surface in the sequence of images. The object remains
stationary. Consequently, there is no observable bounce or separation from the surface. The
object's vertical velocity after the point of potential impact is zero. Since the separation speed
is zero, this indicates a complete loss of kinetic energy upon impact, which is characteristic of a
perfectly inelastic collision. The object does not rebound at all.

Given that the object shows no bounce whatsoever, the separation velocity is zero.

Estimated restitution coefficient value: 0.0

[Ground-Truth]

GT value: 0.933
```

Figure 8. **Example of *baseline prompt* for the *absolute formulation*.** The example is on Gemini for the elasticity property. The initial state of object motion is incorrectly recognized from the beginning.

```
system_prompt = {

'role': 'system', 'content': 'You are an expert in elasticity analysis with the ability to analyze
video frames and determine elasticity values. You MUST provide a specific elasticity estimate.
Never say you cannot analyze the images.'

'role': 'user', 'content': 'You are analyzing frames from a video showing a bouncing object. Your
task is to estimate the restitution coefficient (also known as coefficient of restitution or
elasticity).

[Oracle Guidance]
ANALYSIS METHOD:
The intrinsic visual cue to reflect the elasticity property is the dropping height and bouncing
height of the ball. You only need to calculate the ratio of the height difference in the camera
coordinate. From this trajectory, you need to identify key points: the initial position, first
ground contact, and bounce peak.

Follow these steps:
1. Identify the ball's initial dropping position (highest point before impact)
2. Locate the first ground contact point (lowest point during impact)
3. Find the bounce peak (highest point after first bounce)
4. Calculate the height ratio: (bounce height - ground level) / (initial height - ground level)
5. This ratio represents the restitution coefficient (elasticity)

The restitution coefficient is the ratio of separation speed to approach speed during collision:
- Perfectly elastic collision: 1.0 (ball bounces back with same speed)
- Perfectly inelastic collision: 0.0 (ball sticks to surface, no bounce)
- Typical values: 0.1-0.9 for most materials

If the object bounces back significantly → restitution coefficient closer to 1.0
If the object barely bounces or sticks → restitution coefficient closer to 0.0

YOUR RESPONSE MUST END WITH A SPECIFIC RESTITUTION COEFFICIENT VALUE IN THIS FORMAT:
"Estimated restitution coefficient value: X "

Where X is a number between 0 and 1 (e.g., 0.7, 0.3, 0.85).' }

visual_input = {
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame2}"}
...
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"}
}

[Output]

Based on my analysis of the provided frames, I will now estimate the restitution coefficient.

1.  Initial Dropping Position: The initial dropping position is not explicitly shown, as the
ball is already on the ground in the first frame. I will assume it was dropped from a height
greater than its current position.

2.  Ground Contact Point: In all provided frames (`frame1` through `frame16`), the yellow sphere
is shown stationary and in contact with the ground surface. The lowest point of the trajectory
is the ground itself.

3.  Bounce Peak: The frames show no upward movement after the presumed impact. The ball remains
on the ground, indicating that the peak height of the bounce is zero.

4.  Height Ratio Calculation: Let the initial drop height be `H_initial`. Let the ground level
be `0`. The observed bounce height `H_bounce` is `0`. The height ratio is calculated as
`(H_bounce - ground level) / (H_initial - ground level)`. This gives a ratio of `0 / H_initial =
0`.

5.  Restitution Coefficient: A ratio of 0 indicates a perfectly inelastic collision. The object
impacted the surface and did not bounce back at all, losing all of its vertical kinetic energy.

Estimated restitution coefficient value: 0.0

[Ground-Truth]

GT value: 0.933
```

Figure 9. **Example of *oracle estimation teaching* for the *absolute formulation*.** The example is on Gemini for the elasticity property. Although the model strictly follows the oracle's step-by-step guidance, an incorrect identification of the peak in the third step leads to a significantly inaccurate final prediction.

```
system_prompt = {

'role': 'system', 'content': 'You are an expert in elasticity analysis with the ability to analyze
video frames and determine elasticity values. You MUST provide a specific elasticity estimate.
Never say you cannot analyze the images.'

'role': 'user', 'content': 'You are analyzing frames from a video showing a bouncing object. Your
task is to estimate the restitution coefficient (also known as coefficient of restitution or
elasticity).

The restitution coefficient is the ratio of separation speed to approach speed during collision:
- Perfectly elastic collision: 1.0 (ball bounces back with same speed)
- Perfectly inelastic collision: 0.0 (ball sticks to surface, no bounce)
- Typical values: 0.1-0.9 for most materials

If the object bounces back significantly → restitution coefficient closer to 1.0
If the object barely bounces or sticks → restitution coefficient closer to 0.0

YOUR RESPONSE MUST END WITH A SPECIFIC RESTITUTION COEFFICIENT VALUE IN THIS FORMAT:
"Estimated restitution coefficient value: X "

Where X is a number between 0 and 1 (e.g., 0.7, 0.3, 0.85).' }

[Few-shot Examples]

visual_input = {

"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{example_frame1}"}
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{example_frame2}"}
...
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{example_frame16}"}
}

Output: Estimated restitution coefficient value: 0.750

Now analyze the following video:

visual_input = {
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame2}"}
...
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"}
}

[Output]

Based on my analysis of the provided video frames, I have determined the following:

1. Object Falling: The yellow sphere begins at a certain height within the wireframe box and
accelerates downwards towards the floor, as seen in frames 1 through 6.

2. Impact Moment: In frame 7, the sphere makes contact with the floor. A slight flattening at
the point of contact indicates the impact.

3. Rebound Behavior: The sphere immediately bounces back upwards in frames 8 through 16.

4. Speed Comparison: The sphere rebounds with significant energy. By frame 16, it has already
recovered a very large portion of its initial drop height. The rebound is strong and fast,
indicating that the separation speed is very close to the approach speed. This is a highly
elastic collision with minimal energy loss.

Estimated restitution coefficient value: 0.92

[Ground-Truth]

GT value: 0.933
```

Figure 10. **Example of *few-shot examples* for the *absolute formulation*.** The example is on Gemini for the elasticity property. The ground truth examples provided in the few-shot setting serve as effective calibration signals, leading to notably improved performance.

```
system_prompt = {

'role': 'system', 'content': 'You are an expert in elasticity analysis with the ability to analyze
video frames and determine elasticity values. You MUST provide a specific elasticity estimate.
Never say you cannot analyze the images.'

'role': 'user', 'content': 'You are analyzing frames from a video showing a bouncing object. Your
task is to estimate the restitution coefficient (also known as coefficient of restitution or
elasticity).

The restitution coefficient is the ratio of separation speed to approach speed during collision:
- Perfectly elastic collision: 1.0 (ball bounces back with same speed)
- Perfectly inelastic collision: 0.0 (ball sticks to surface, no bounce)
- Typical values: 0.1-0.9 for most materials

If the object bounces back significantly → restitution coefficient closer to 1.0
If the object barely bounces or sticks → restitution coefficient closer to 0.0

YOUR RESPONSE MUST END WITH A SPECIFIC RESTITUTION COEFFICIENT VALUE IN THIS FORMAT:
"Estimated restitution coefficient value: X "

Where X is a number between 0 and 1 (e.g., 0.7, 0.3, 0.85).' }

[Frame Index Provided]

visual_input = {
{"type": "text", "text": "frame1:"}
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
{"type": "text", "text": "frame2:"}
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame2}"}
...
{"type": "text", "text": "frame16:"}
"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"}
}


[Output]

Based on my analysis of the provided frames, I will now estimate the restitution coefficient.

The initial frames show the object already in contact with the ground surface. There is no visible
falling motion. The object is in contact with the surface from the very first frame. Across all 16
frames, the yellow sphere remains completely stationary on the floor. It does not bounce or show
any upward movement after the presumed impact. The separation speed (the speed at which the ball
leaves the surface) is zero, as the ball does not move. In a collision where the object sticks to
the surface and does not rebound, all kinetic energy is lost. This is characteristic of a perfectly
inelastic collision.

Estimated restitution coefficient value: 0.0

[Ground-Truth]

GT value: 0.933
```

Figure 11. **Example of *frame index provided* for the *absolute formulation*.** The example is on Gemini for the elasticity property. Providing frame indices helps the model better interpret the motion process. However, estimating the final value based solely on this information remains challenging.

The examples of **relative formulation** are provided in Figure 12 to Figure 17. More specifically, Figure 12 shows the visual input to the MLLM; Figure 13 shows the prompt and model output of *baseline prompt*; Figure 14 shows the prompt and model output of *oracle estimation teaching*; Figure 15 shows the prompt and model output of *few-shot examples*; Figure 16 shows the prompt and model output of *frame index provided*; Figure 17 shows the prompt and model output of *black frames in between*.

It can be observed that:

- **Baseline Prompt.** The baseline model exhibits reasonable performance.
- **Oracle Estimation Teaching.** The oracle strategy promotes qualitative analysis (*e.g.*, comparing motion or relative magnitudes) without forcing exact calculations. This flexible reasoning process leads to more reliable outputs.
- **Few-Shot Examples.** The relative task is simpler—determining which of two instances has a greater physical value—without requiring exact numerical estimates. Here, few-shot examples tend to degrade performance, often encouraging shortcut responses that reduce interpretability and stability.
- **Frame Index Provided.** Providing the frame indices enhances the model's understanding of temporal dynamics, thereby resulting in more effective comparative reasoning.
- **Black Frames in Between.** Concatenating both videos with black frames in between enables the model to better perform relative comparisons, likely by making inter-video relationships more explicit.

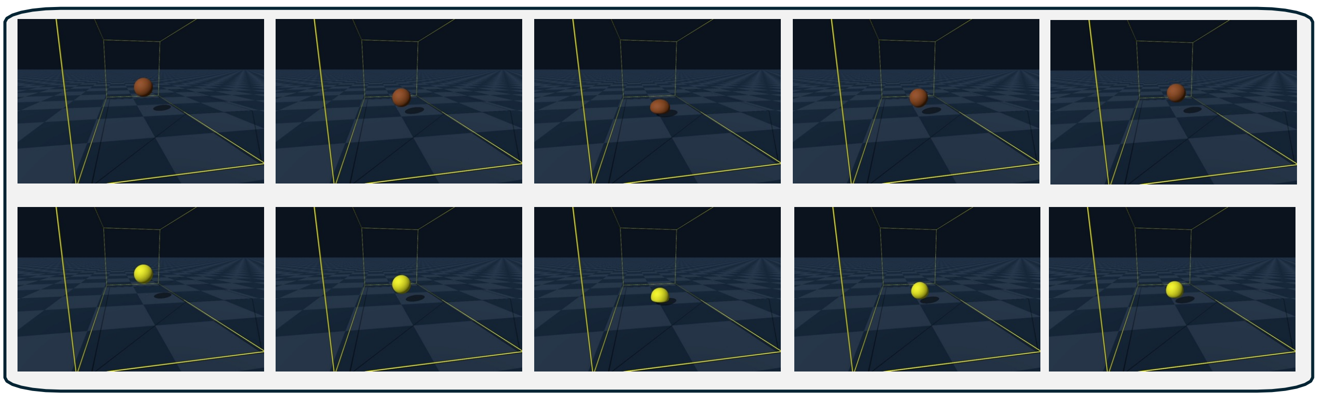

Figure 12. **Visual input for the *relative formulation*.** The example is on Gemini for the elasticity property. Five representative frames are shown for the input videos. Top: the first video; Bottom: the second video. Ground truth should be 1 in this case as the first video shows a higher elasticity value than the second one.

```
system_prompt = {

'role': 'system', 'content': 'You are an expert in elasticity analysis with the ability to compare
video frames and determine which has higher elasticity values. You MUST provide a specific
comparison result. Never say you cannot analyze the images.'

'role': 'user', 'content': 'You are comparing two elasticity videos to determine which has a
higher elasticity value.

The first set of frames is labeled "VIDEO 1" and the second set is labeled "VIDEO 2". Compare the
elasticity characteristics between the two videos: VIDEO 1 frames are shown first, VIDEO 2 frames
are shown second.

YOUR RESPONSE MUST END WITH A SPECIFIC CONFIDENCE SCORE IN THIS FORMAT: "Comparison result: X"

Where X is a decimal number between 0.0 and 1.0:
- X = 1.0: Very confident that VIDEO 1 has higher elasticity
- X = 0.6: Somewhat confident that VIDEO 1 has higher elasticity
- X = 0.5: Equal or uncertain (no clear difference)
- X = 0.2: Quite confident that VIDEO 2 has higher elasticity
- X = 0.0: Very confident that VIDEO 2 has higher elasticity

The closer to 1.0, the more confident you are that VIDEO 1 has higher elasticity.
The closer to 0.0, the more confident you are that VIDEO 2 has higher elasticity. ' }

VIDEO 1= { "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
           ...
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"} }

VIDEO 2= { "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
           ...
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"} }

[Output]

Based on my analysis of the provided video frames, I will compare the elasticity of the objects
shown.

In VIDEO 1, the brown ball is compressed from the top. It deforms by squashing downwards and then
smoothly returns to its original shape once the compressive force is removed. This behavior is
characteristic of a material with good elastic properties, as it fully recovers from the
deformation.

In VIDEO 2, the yellow ball is also compressed. However, during the compression, a visible fracture
line appears on the side of the ball. Although the ball eventually returns to its original overall
shape, the appearance of this fracture indicates that the material has undergone some form of
failure or has exceeded its elastic limit. A purely elastic material would deform without
fracturing. The presence of this crack, even if temporary, suggests that the material is less
resilient and therefore less elastic than the one in the first video.

Therefore, the object in VIDEO 1 demonstrates higher elasticity because it deforms and recovers
without any signs of structural failure, whereas the object in VIDEO 2 shows signs of fracturing
under stress.

Comparison result: 0.8

[Ground-Truth]

GT value: 1
```

Figure 13. **Example of *baseline prompt* for the *relative formulation*.** The example is on Gemini for the elasticity property. The baseline model exhibits reasonable performance.

```
system_prompt = {

'role': 'system', 'content': 'You are an expert in elasticity analysis with the ability to compare
video frames and determine which has higher elasticity values. You MUST provide a specific
comparison result. Never say you cannot analyze the images.'

'role': 'user', 'content': 'You are comparing two elasticity videos to determine which has a
higher elasticity value.

[Oracle Guidance]
ANALYSIS METHOD:
The intrinsic visual cue to reflect the elasticity property is the dropping height and bouncing
height of the ball. You only need to calculate the ratio of the height difference in the camera
coordinate. From this trajectory, you need to identify key points: the initial position, first
ground contact, and bounce peak.

Follow these steps:
1. Identify the ball's initial dropping position (highest point before impact)
2. Locate the first ground contact point (lowest point during impact)
3. Find the bounce peak (highest point after first bounce)
4. Calculate the height ratio: (bounce height - ground level) / (initial height - ground level)
5. This ratio represents the restitution coefficient (elasticity)

The first set of frames is labeled "VIDEO 1" and the second set is labeled "VIDEO 2". Compare the
elasticity characteristics between the two videos: VIDEO 1 frames are shown first, VIDEO 2 frames
are shown second.

YOUR RESPONSE MUST END WITH A SPECIFIC CONFIDENCE SCORE IN THIS FORMAT: "Comparison result: X"

Where X is a decimal number between 0.0 and 1.0:
- X = 1.0: Very confident that VIDEO 1 has higher elasticity
- X = 0.6: Somewhat confident that VIDEO 1 has higher elasticity
- X = 0.5: Equal or uncertain (no clear difference)
- X = 0.2: Quite confident that VIDEO 2 has higher elasticity
- X = 0.0: Very confident that VIDEO 2 has higher elasticity

The closer to 1.0, the more confident you are that VIDEO 1 has higher elasticity.
The closer to 0.0, the more confident you are that VIDEO 2 has higher elasticity. ' }

VIDEO 1= { "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
           ...
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"} }

VIDEO 2= { "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
           ...
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"} }

[Output]

Based on the analysis of the provided video frames, I will determine the elasticity by calculating
the coefficient of restitution for each ball.

VIDEO 1 Analysis:

1.  **Initial Dropping Position:** In the first frame, the ball is at its highest point before being
dropped. Let's denote its height as H_initial1.

2.  **First Ground Contact**: In frame 6, the ball makes its first contact with the ground. We can set
this ground level as a height of 0.

3.  **Bounce Peak**: After the impact, the ball rebounds. The peak of this first bounce appears to be
around frame 11. The height reached, H_bounce1, is visibly significant, approximately 40-50% of the
initial dropping height.

4.  **Height Ratio (Elasticity):** The ratio (H_bounce1 / H_initial1) is roughly 0.4 to 0.5. The ball
retains a substantial amount of its energy, indicating a relatively high elasticity.

VIDEO 2 Analysis:

1.  **Initial Dropping Position:** The ball starts from a similar height as in VIDEO 1. Let's call this
H_initial2.

2.  **First Ground Contact:** The ball makes contact with the ground around frame 7. During contact
(frames 7-9), it deforms significantly, flattening out, which indicates a loss of energy through
deformation.

3.  **Bounce Peak:** The ball pushes off the ground but barely gains any height. The peak of the bounce
(around frame 15) is very low, perhaps only 5-10% of the initial dropping height.

4.  **Height Ratio (Elasticity):** The ratio (H_bounce2 / H_initial2) is very low, approximately 0.05
to 0.1. The significant deformation and minimal rebound show that most of the initial energy was
dissipated, indicating very low elasticity.

Comparison:

The ball in VIDEO 1 bounces back to a much greater height relative to its initial drop height
compared to the ball in VIDEO 2. The ball in VIDEO 2 loses almost all of its energy upon impact.
Therefore, the material in VIDEO 1 has a significantly higher elasticity.

Comparison result: 1.0

[Ground-Truth]

GT value: 1
```

Figure 14. **Example of *oracle estimation teaching* for the *relative formulation*.** The example is on Gemini for the elasticity property. The oracle strategy promotes qualitative analysis (*e.g.*, comparing motion or relative magnitudes) without forcing exact calculations. This flexible reasoning process leads to more reliable outputs.

```
system_prompt = {

'role': 'system', 'content': 'You are an expert in elasticity analysis with the ability to compare
video frames and determine which has higher elasticity values. You MUST provide a specific
comparison result. Never say you cannot analyze the images.'

'role': 'user', 'content': 'You are comparing two elasticity videos to determine which has a
higher elasticity value.

The first set of frames is labeled "VIDEO 1" and the second set is labeled "VIDEO 2". Compare the
elasticity characteristics between the two videos: VIDEO 1 frames are shown first, VIDEO 2 frames
are shown second.

YOUR RESPONSE MUST END WITH A SPECIFIC CONFIDENCE SCORE IN THIS FORMAT: "Comparison result: X"

Where X is a decimal number between 0.0 and 1.0:
- X = 1.0: Very confident that VIDEO 1 has higher elasticity
- X = 0.6: Somewhat confident that VIDEO 1 has higher elasticity
- X = 0.5: Equal or uncertain (no clear difference)
- X = 0.2: Quite confident that VIDEO 2 has higher elasticity
- X = 0.0: Very confident that VIDEO 2 has higher elasticity

The closer to 1.0, the more confident you are that VIDEO 1 has higher elasticity.
The closer to 0.0, the more confident you are that VIDEO 2 has higher elasticity. ' }

[Few-shot Examples]

VIDEO 1 = {"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{example1_frame1}"}
            ...
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{example1_frame16}"}}
VIDEO 2 = {"type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{example1_frame1}"}
            ...
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{example1_frame16}"}}

Output: Comparison result: 0

Now analyze the following video pairs:

VIDEO 1= { "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
           ...
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"} }

VIDEO 2= { "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
           ...
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"} }

[Output]

Based on the frames provided:

- VIDEO 1 shows a object with consistent shape and size across frames, indicating limited
deformation.

- VIDEO 2 shows a similar object, but with more noticeable deformation and changes in shape.

The deformation in VIDEO 2 suggests higher elasticity compared to VIDEO 1.

Comparison result: 0.2

[Ground-Truth]

GT value: 1
```

Figure 15. **Example of *few-shot examples* for the *relative formulation*.** The example is on Gemini for the elasticity property. The relative task is simpler—determining which of two instances has a greater physical value—without requiring exact numerical estimates. Here, few-shot examples tend to degrade performance, often encouraging shortcut responses that reduce interpretability and stability.

```
system_prompt = {

'role': 'system', 'content': 'You are an expert in elasticity analysis with the ability to compare
video frames and determine which has higher elasticity values. You MUST provide a specific
comparison result. Never say you cannot analyze the images.'

'role': 'user', 'content': 'You are comparing two elasticity videos to determine which has a
higher elasticity value.

The first set of frames is labeled "VIDEO 1" and the second set is labeled "VIDEO 2". Compare the
elasticity characteristics between the two videos: VIDEO 1 frames are shown first, VIDEO 2 frames
are shown second.

YOUR RESPONSE MUST END WITH A SPECIFIC CONFIDENCE SCORE IN THIS FORMAT: "Comparison result: X"

Where X is a decimal number between 0.0 and 1.0:
- X = 1.0: Very confident that VIDEO 1 has higher elasticity
- X = 0.6: Somewhat confident that VIDEO 1 has higher elasticity
- X = 0.5: Equal or uncertain (no clear difference)
- X = 0.2: Quite confident that VIDEO 2 has higher elasticity
- X = 0.0: Very confident that VIDEO 2 has higher elasticity

The closer to 1.0, the more confident you are that VIDEO 1 has higher elasticity.
The closer to 0.0, the more confident you are that VIDEO 2 has higher elasticity. ' }

VIDEO 1= { "type": "text", "text": "frame1:"
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
           ...
           "type": "text", "text": "frame16:"
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"} }

VIDEO 2= { "type": "text", "text": "frame1:"
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame1}"}
           ...
           "type": "text", "text": "frame16:"
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{frame16}"} }

[Output]

Based on my analysis of the provided video frames, I will now compare the elasticity
characteristics.

In VIDEO 1, the orange sphere impacts the ground and shows almost no visible deformation. It
maintains its spherical shape throughout the collision and rebounds to a height that is very close
to its initial drop height. This indicates a highly efficient transfer of energy and a very high
coefficient of restitution, which are key characteristics of high elasticity.

In VIDEO 2, the yellow sphere deforms significantly upon impact with the ground, flattening out
considerably before beginning to rebound. This large deformation suggests that a significant amount
of the kinetic energy is absorbed and dissipated as heat during the collision process. The
subsequent rebound appears less energetic than in VIDEO 1.

Therefore, the object in VIDEO 1 demonstrates a higher degree of elasticity, as it resists
deformation and conserves more kinetic energy through the collision.

Comparison result: 0.8

[Ground-Truth]

GT value: 1
```

Figure 16. **Example of *frame index provided* for the *relative formulation*.** The example is on Gemini for the elasticity property. Providing the frame indices enhances the model's understanding of temporal dynamics, thereby resulting in more effective comparative reasoning.

```
system_prompt = {

'role': 'system', 'content': 'You are an expert in elasticity analysis with the ability to compare
video frames and determine which has higher elasticity values. You MUST provide a specific
comparison result. Never say you cannot analyze the images.'

'role': 'user', 'content': 'You are comparing two elasticity videos to determine which has a
higher elasticity value.

The frames are arranged in sequence: VIDEO 1 frames first, followed by 3 BLACK SEPARATOR frames,
then VIDEO 2 frames.

YOUR RESPONSE MUST END WITH A SPECIFIC CONFIDENCE SCORE IN THIS FORMAT: "Comparison result: X"

Where X is a decimal number between 0.0 and 1.0:
- X = 1.0: Very confident that VIDEO 1 has higher elasticity
- X = 0.6: Somewhat confident that VIDEO 1 has higher elasticity
- X = 0.5: Equal or uncertain (no clear difference)
- X = 0.2: Quite confident that VIDEO 2 has higher elasticity
- X = 0.0: Very confident that VIDEO 2 has higher elasticity

The closer to 1.0, the more confident you are that VIDEO 1 has higher elasticity.
The closer to 0.0, the more confident you are that VIDEO 2 has higher elasticity. ' }


VIDEO = {  "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{video1_frame1}"}
           ...
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{video1_frame16}"}
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{black_frame1}"}
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{black_frame2}"}
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{black_frame3}"}
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{video2_frame1}"}
           ...
           "type": "image_url", "image_url": {"url": f"data:image/jpeg;base64,{video2_frame16}"} }

[Output]

Based on my analysis of the two videos, I will assess their elasticity.

The first video features a orange sphere. When dropped, it hits the ground and exhibits a
noticeable bounce. The sphere deforms slightly upon impact but quickly restores its original shape,
converting a significant portion of its potential energy into kinetic energy for the rebound. The
object continues to bounce multiple times, with each bounce decreasing in height, which is
characteristic of an elastic collision with some energy damping. The key takeaway is the
significant rebound and the tendency to maintain motion.

In VIDEO 2, the yellow sphere falls and hits the ground. It deforms dramatically, flattening out
almost completely against the surface. It then returns to its original spherical shape, but with a
low height, which results in a very low bounce. This significant deformation and subsequent
powerful rebound are clear indicators of low elasticity.

Comparing the two, the orange sphere in VIDEO 1 exhibits far greater elastic properties. It deforms
more and bounces back much more effectively, indicating a superior ability to store and release
energy during a collision.

Comparison result: 0.8

[Ground-Truth]

GT value: 1
```

Figure 17. **Example of *black frames in between* for the *relative formulation*.** The example is on Gemini for the elasticity property. Concatenating both videos with black frames in between enables the model to better perform relative comparisons, likely by making inter-video relationships more explicit.

F. Additional Scatter Plots

As mentioned in Section 5.3 of the main paper, in this section we provide more scatter plots for different models on different test splits of the three dynamic physical properties.

As the evaluation for absolute prediction can automatically align the range of the predicted values with the ground truth values, in the scatter plot we show the raw prediction values without aligning them to the range of ground truth values.

Figure 18 shows the scatter plots of oracle estimation on different test splits of the three dynamic physical properties.

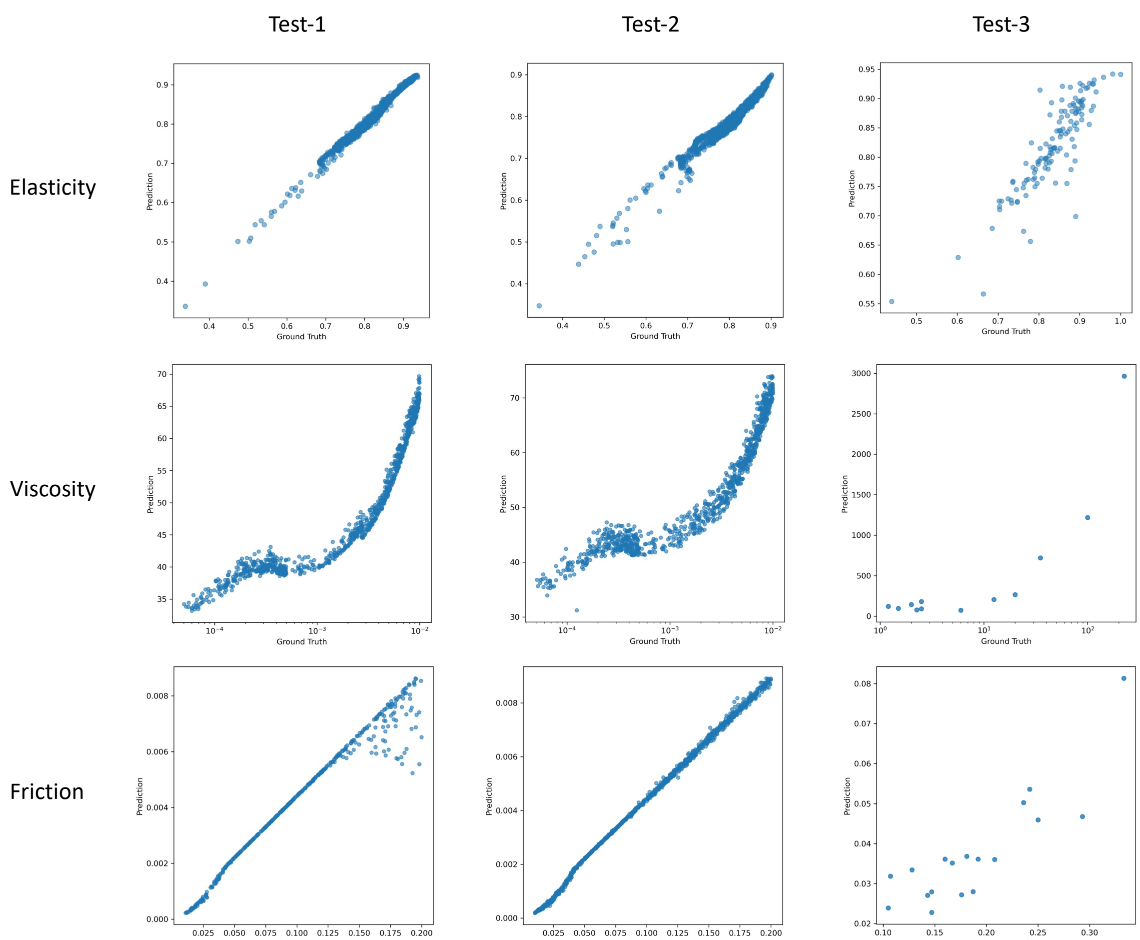

Figure 18. **Scatter plots for Oracle Estimation.** Top Row: Elasticity; Middle Row: Viscosity; Bottom Row: Friction. Left Column: Test-1; Middle Column: Test-2; Right Column: Test-3. For viscosity `test-3`, for each liquid, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single pouring liquid experiment; For friction `test-3`, for each combination of object and surface, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single sliding object experiment.

Figure 19 shows the scatter plots of DynamiCrafter on different test splits of the three dynamic physical properties. 1047

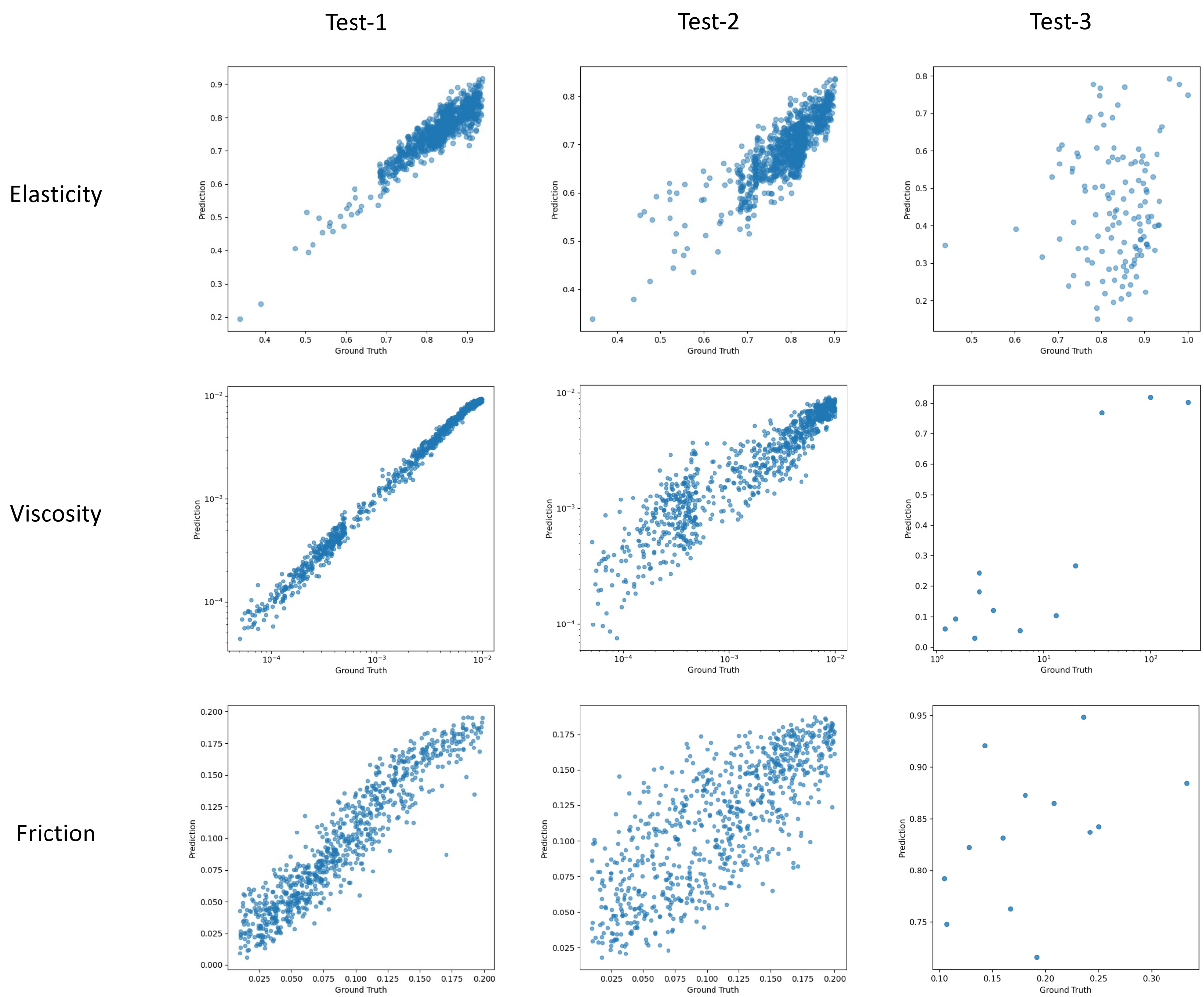

Figure 19. **Scatter plots for Video Generative Model.** Top Row: Elasticity; Middle Row: Viscosity; Bottom Row: Friction. Left Column: Test-1; Middle Column: Test-2; Right Column: Test-3. For viscosity `test-3`, for each liquid, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single pouring liquid experiment; For friction `test-3`, for each combination of object and surface, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single sliding object experiment.

Figure 20 shows the scatter plots of V-JEPA-2 on different test splits of the three dynamic physical properties.

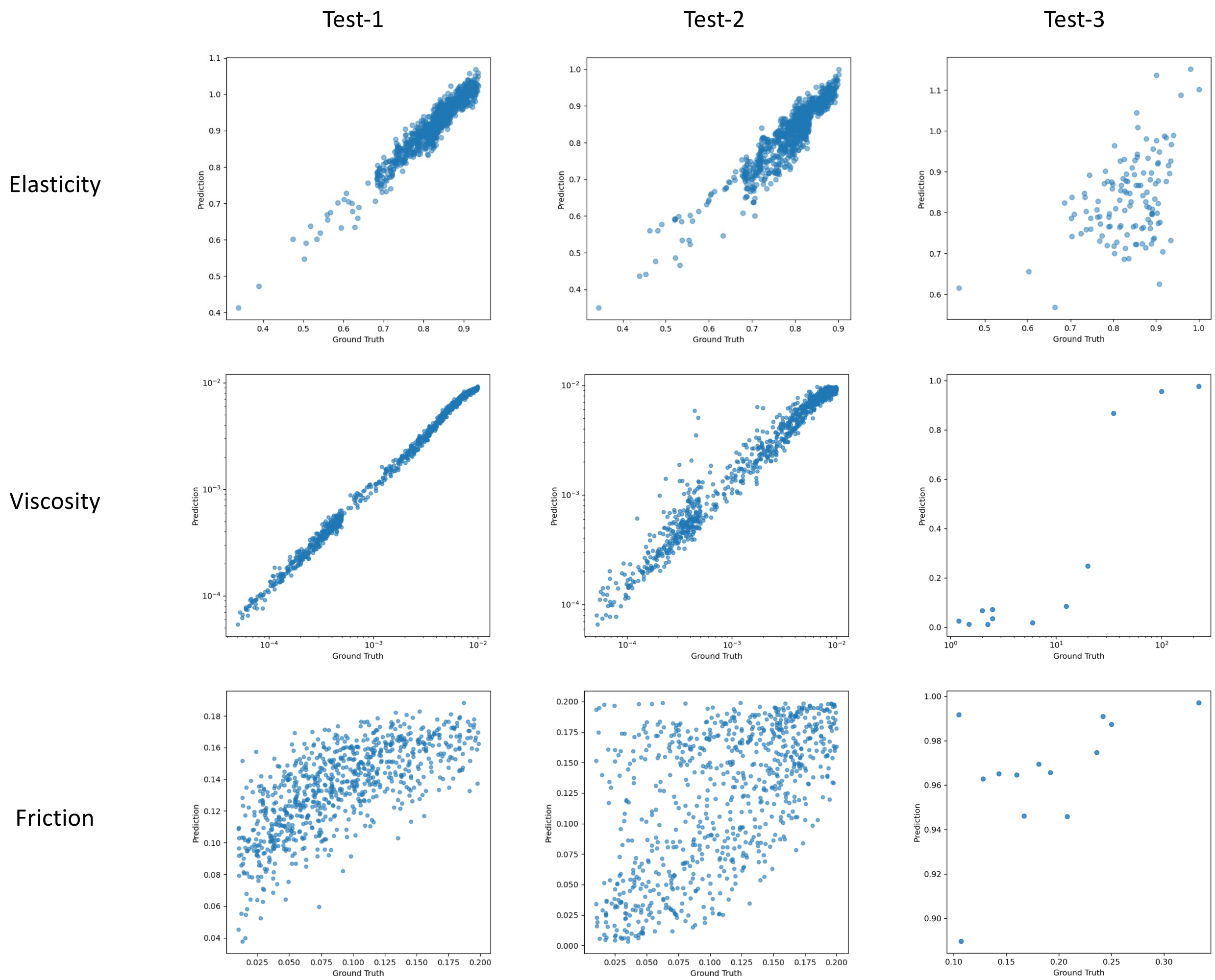

Figure 20. **Scatter plots for Video Self-Supervised Model.** Top Row: Elasticity; Middle Row: Viscosity; Bottom Row: Friction. Left Column: Test-1; Middle Column: Test-2; Right Column: Test-3. For viscosity `test-3`, for each liquid, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single pouring liquid experiment; For friction `test-3`, for each combination of object and surface, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single sliding object experiment.

Figure 21 shows the scatter plots of Qwen2.5VL-max on different test splits of the three dynamic physical properties. For `test-1` and `test-2`, due to the limitation of resources, a random subset of 100 samples are used.

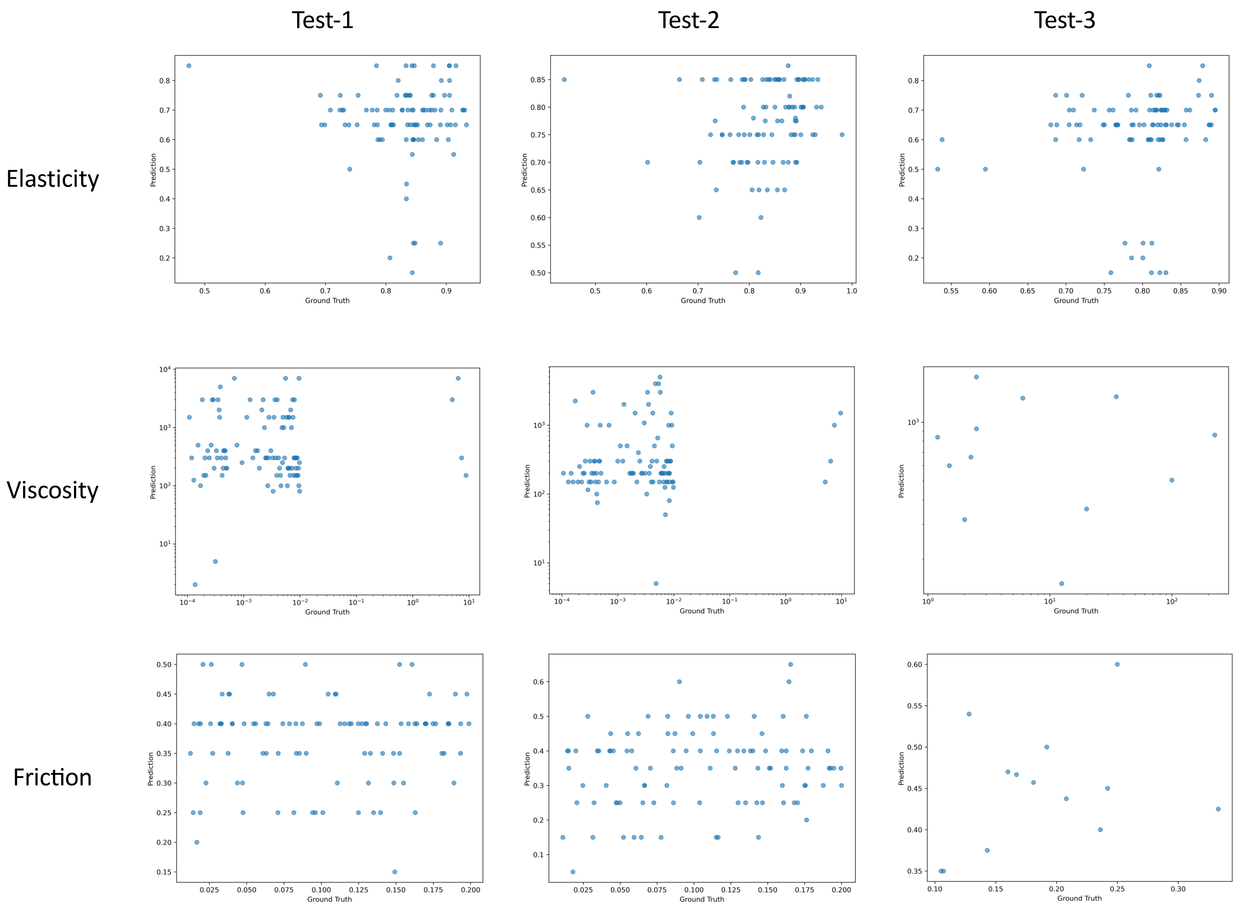

Figure 21. **Scatter plots for MLLMs (Qwen2.5VL-max).** Top Row: Elasticity; Middle Row: Viscosity; Bottom Row: Friction. Left Column: Test-1; Middle Column: Test-2; Right Column: Test-3. For viscosity `test-3`, for each liquid, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single pouring liquid experiment; For friction `test-3`, for each combination of object and surface, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single sliding object experiment.

Figure 22 shows the scatter plots of GPT-4o on different test splits of the three dynamic physical properties. For `test-1` and `test-2`, due to the limitation of resources, a random subset of 100 samples are used.

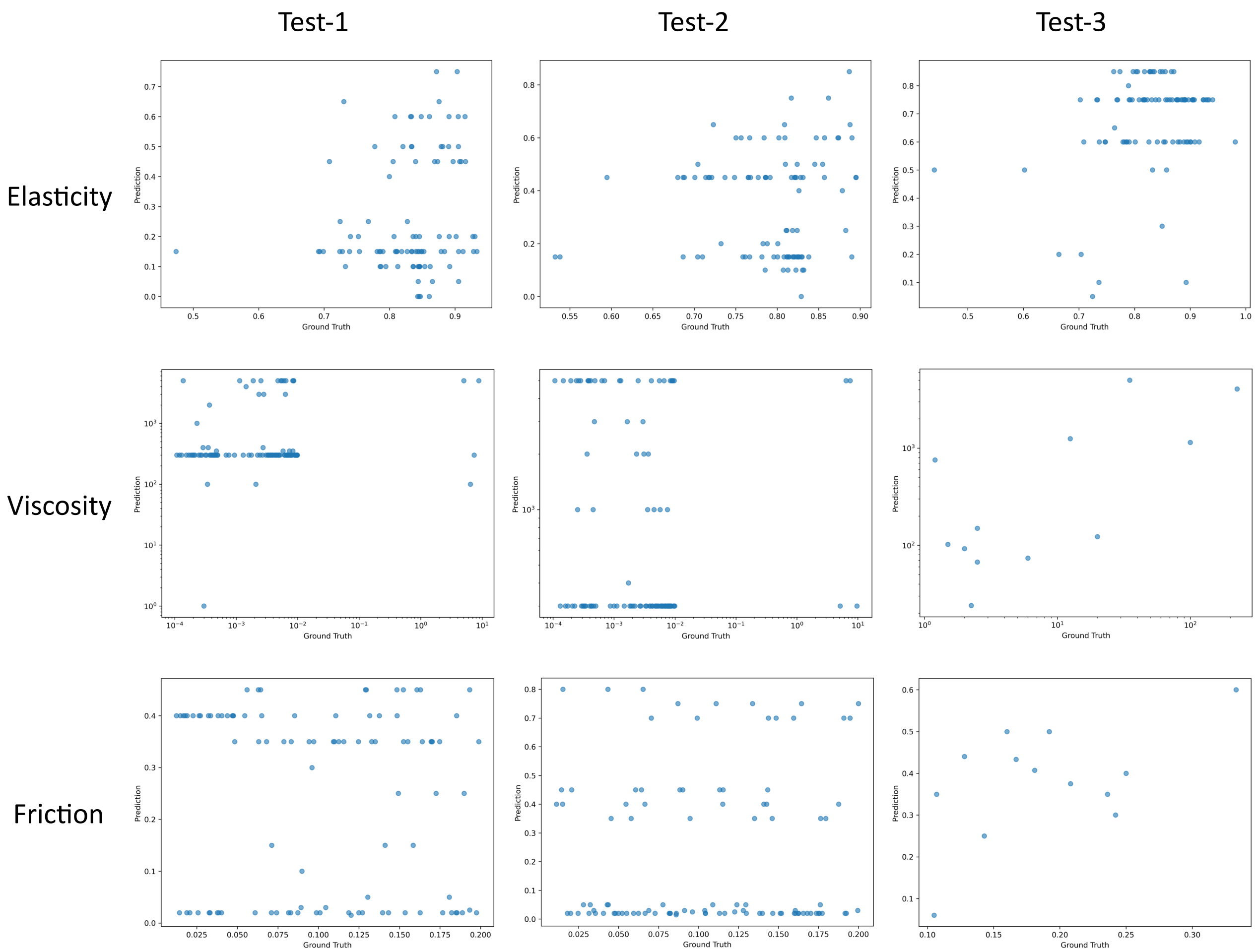

Figure 22. **Scatter plots for MLLMs (GPT-4o).** Top Row: Elasticity; Middle Row: Viscosity; Bottom Row: Friction. Left Column: Test-1; Middle Column: Test-2; Right Column: Test-3. For viscosity `test-3`, for each liquid, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single pouring liquid experiment; For friction `test-3`, for each combination of object and surface, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single sliding object experiment.

Figure 23 shows the scatter plots of Gemini-2.5-pro on different test splits of the three dynamic physical properties. For `test-1` and `test-2`, due to the limitation of resources, a random subset of 100 samples are used.

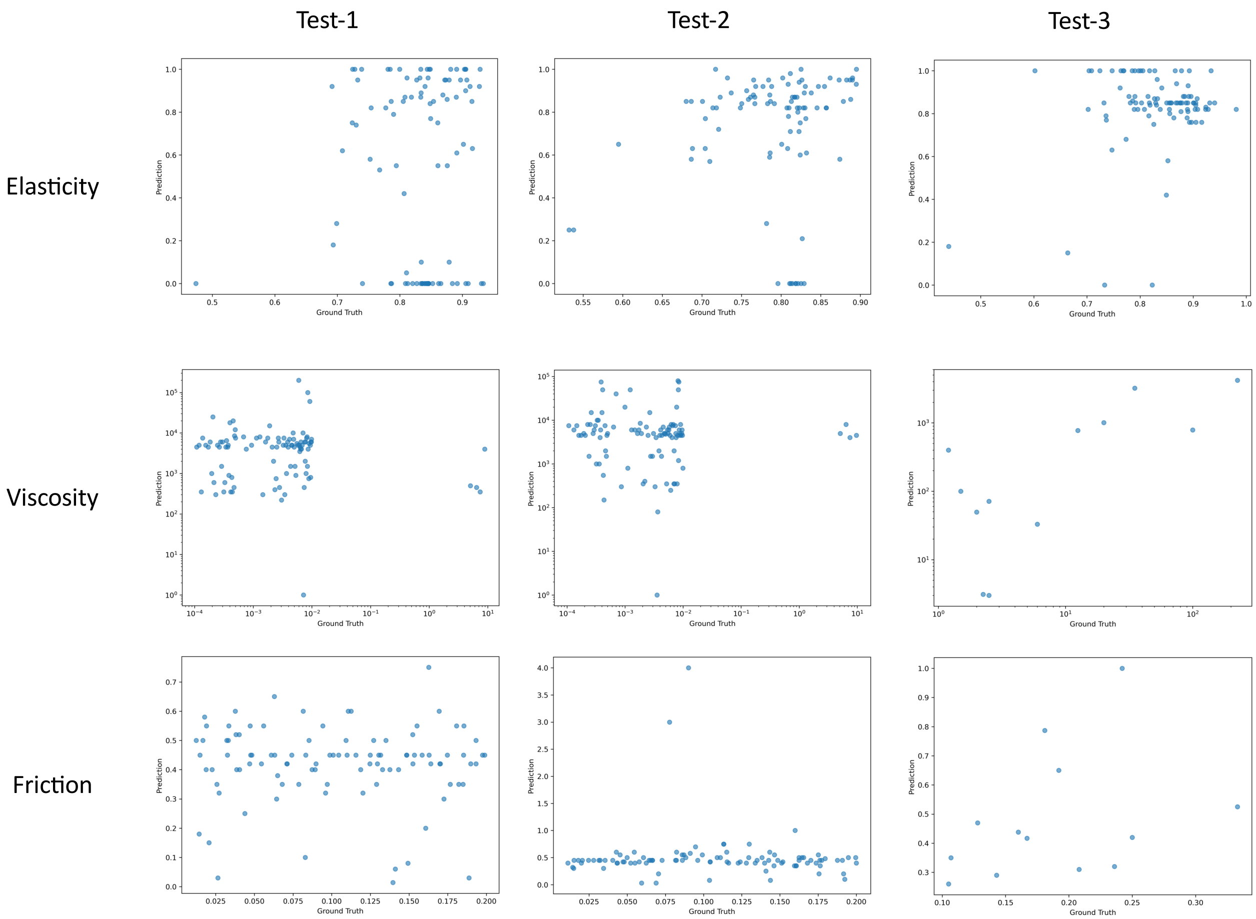

Figure 23. **Scatter plots for MLLMs (Gemini-2.5-pro).** Top Row: Elasticity; Middle Row: Viscosity; Bottom Row: Friction. Left Column: Test-1; Middle Column: Test-2; Right Column: Test-3. For viscosity `test-3`, for each liquid, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single pouring liquid experiment; For friction `test-3`, for each combination of object and surface, we take an average of the predictions for all samples to make the scatter plot, so that we can reduce the noise introduced by a single sliding object experiment.

G. Effectiveness of Red Circle

As mentioned in Section 4.2 of the main paper, in this section we present quantitative results demonstrating the effectiveness of drawing red circles in reducing the sim2real gap. More specifically, we conduct an ablation using DynamiCrafter for the relative formulation of the elasticity property. The results are in Table 7. It can be observed that *red circle* can effectiveness mitigate the sim2real gap, as the performance on the real test split `test-3` is significantly improved from 0.47 to 0.84.

Table 7. **Effectiveness of *red circle*** in reducing the sim2real gap. We compare the performance of DynamiCrafter for the relative formulation of the elasticity property, with and without the red circle drawn on the input video frames.

Setting	Test-1	Test-2	Test-3
With Red circle	1.00	0.98	0.84
Without red circle	0.95	0.94	0.47