

Action Without Interaction: Probing the Physical Foundations of Video LMMs via Contact-Release Detection

Supplementary Material

1. Ablation Experiments

1.1. Action and object recognition

This section provides extended descriptions of the experimental setup of the evaluation on the tasks related to action and object recognition, mentioned in Section 4 of the main manuscript. All experiments were conducted using the extracted video frames of the full length videos from the original SSv2 dataset [1]. The models were given, as input, the sequence of video frames and prompted with the task-specific textual information detailed below. Five models including, GPT-4o, GPT-5.2, Gemini-2.5-Flash, Gemini-3-Pro and Qwen-2.5-VL-72B, were evaluated under the zero-shot regime (without any examples), and without any of the modifying conditions (*Grounding* and *Reasoning*).

Task-specific textual information was provided only when required:

- **Action-template recognition:** The prompts included the list of candidate template labels and their corresponding template IDs.
- **Object-placeholder extraction:** The prompts included the template ID, the template sentence, and the number of placeholder slots required by the template.
- **Event bounding-box detection:** No additional textual information was provided beyond the sequence of video frames.

Action and object recognition. Listing S1 and Listing S3 presents the prompts used in the experiments for testing the models performance in the tasks of object and action recognition, respectively. In the listed prompts, the strings `template_sentence`, `template_id`, and `n_slots` denote variables that were automatically replaced during execution for each experiment, according to the specific interaction template. The prompt structure itself was fixed, while these fields changed to reflect the corresponding template text, its numeric ID, and the number of placeholder slots.

Event bounding-box detection. Listing S2 present the prompt used in the experiment for testing the models performance in the task of detecting the spatial location where the event occurs in the predicted frame.

1.2. Tow-Shot with Feedback

Inspired by recent work on boosting LLMs reasoning with verification feedback [2–4], we designed a variant to the common TS regime, in which the label of the second example was presented to the model indirectly, through an iterative feedback session. In this session, the model performed the detection task on the second example video and then prompted a numerical verification feedback by the user side. The feedback indicate a metric on the gap between the predicted frame (f_p) and the true frame (f_t) of the event. We defined the error in prediction, $\epsilon(f_t, f_p)$, via a sigmoid function, as shown in Eq. (1). The error function is shifted to the middle of the frame range ($\frac{N}{2}$) (as we constrained the error to be in the range of $[0, 1]$, and required it to be $\epsilon(\frac{N}{2}) = 0.5$). The score was defined as shown in Eq. (2). The iterative feedback session ended when the model predicted the correct frame, or after it exceeded a limit of allowed trials T_{th} , which in our experiments was set to 10 (equivalent to the maximal number of iterations in the naive case where the model simply scans all the frames in turn until it gets to the right frame). After the iterative session ended, the model was instructed to perform the main detection task on the test video. In our experiment, we included examples only of other events from the same full video from which the test event was cropped, thus providing the model context with familiar context from the test video. The protocol followed the algorithm in Algorithm 1. The instructing prompt is presented in Listing S4.

$$\epsilon(f_t, f_p) = \sigma(|f_t - f_p| + \frac{N}{2}) = \frac{1}{1 + e^{-|f_t - f_p| + \frac{N}{2}}} \quad (1)$$

$$s(f_t, f_p) = 1 - \epsilon(f_t, f_p) \quad (2)$$

1.3. Chain-of-Thought (CoT) Tuning

In the main experiment, examples included only videos with the true frame, but without further description of the dynamic event. In contrast, in this experiment we provided the agent with detailed steps for detecting the frame where the event occurs, for each example video. Taking inspiration from [3], in which the authors showed that an LMM can improve in the task of solving mathematical problems by presenting it examples of solutions to similar mathematical problems with a detailed, step-by-step solutions.

Algorithm 1 Iterative feedback algorithm

```
Ensure:  $fbScr = 0.5$ 
Ensure:  $prvPred_0 = 0$ 
Ensure:  $prvPred_1 = 0$ 
while  $0.98 > |fbScr|$  do
  if  $(sgn(fbScr) > 0) \ \& \ (prvPred_1 \geq prvPred_0)$  then
     $curPred \leftarrow [prvPred_1, 10]$ 
  else if  $(sgn(fbScr) > 0) \ \& \ (prvPred_1 \leq prvPred_0)$  then
     $curPred \leftarrow [1, prvPred_1]$ 
  else if  $(sgn(fbScr) < 0) \ \& \ (prvPred_1 \geq prvPred_0)$  then
     $curPred \leftarrow [prvPred_0, prvPred_1]$ 
  else if  $(sgn(fbScr) < 0) \ \& \ (prvPred_1 \leq prvPred_0)$  then
     $curPred \leftarrow [prvPred_1, prvPred_0]$ 
  end if
   $prvPred_0 \leftarrow prvPred_1$ 
   $prvPred_1 \leftarrow curPred$ 
   $fbScr \leftarrow MeasureScore(prvPred_0, prvPred_1, trueFrame)$ 
end while
```

In our experiments we fed the agent a prompt that included three parts, viz. 1) a general explanation of the goal the agent was required to achieve, 2) a set of 1 — 8 examples with a full solution, and 3) the final test set, for which the final accuracy was computed (see Listing S5). For this matter we manually formulated 18 CoT prompts (9 for contact and 9 for release events), each with a detailed explanation of the scene and the interactions between a hand and objects located in it.

As before, at each iteration we requested the LMM agent to detect the exact frame in which the interaction (i.e., contact or release) happens, but this time, we provided it also

Listing S1. Instruction prompt for object detection.

```
System:
You are an expert video-interaction classifier.
User:
The uploaded images are consecutive frames from a video.
  ↳ The numbers in the file name indicate the order
  ↳ of the frames in the sequence, so frame_0.jpg
  ↳ is the first frame, followed by frame_1.jpg
  ↳ which is the second frame, etc.
You will see frames from a short video and one template
  ↳ sentence.
Template (with placeholders):
{template_sentence}
Return JSON ONLY (no extra text, no markdown fences):
{
  "template_id": {template_id},
  "placeholders": [
    "<slot1>",
    "<slot2>",
    ...
  ]
}
Rules:
- Provide exactly {n_slots} placeholders, ordered left-
  ↳ to-right as they appear in the template.
- Use short, concrete noun phrases for visible objects (
  ↳ e.g., "potato", "vicks vaporub bottle").
- Avoid generic words such as "object", "thing", or "
  ↳ item".
- Do not include explanations, labels, confidence scores
  ↳ , or any additional fields.
```

Listing S2. Instruction prompt for event bounding-box detection.

```
System:
You are a useful assistant and an expert in video
  ↳ understanding.
User:
The uploaded images are consecutive frames from a video.
  ↳ The numbers in the file name
  ↳ indicate the order of the frames in the sequence, so
  ↳ frame_1.jpg is the first frame,
  ↳ followed by frame_2.jpg which is the second frame, etc.
  ↳ The sequence shows an interaction between a hand
  ↳ and an object. An interaction usually begins
  ↳ when an object starts to move with the hand. An
  ↳ interaction usually ends when the hand starts to
  ↳ move without the object.
Q: In which frame does the interaction end?
Answer briefly with: 'Predicted frame: <frame number
  ↳ only>'
Q: Where in the predicted frame does the interaction
  ↳ occur?
Answer with bounding box coordinates in the original
  ↳ frame's pixel scale:
'Predicted BBox: [<left_x>, <top_y>, <width>, <height>]'
Bounding box requirements:
- Use the SAME coordinate scale as the original video
  ↳ frame (pixels, not normalized).
- The box should tightly enclose the MAIN CONTACT REGION
  ↳ between the interacting objects (e.g., where a
  ↳ hand touches an object), not the entire objects.
- Make the bounding box as SMALL as possible while still
  ↳ fully containing this contact region.
- The bounding box MUST stay within the frame boundaries
  ↳ ; never extend beyond the image edges.
- If uncertain, err on slightly smaller rather than
  ↳ larger, as long as the contact area is included.
```

with a subset of examples, with a varying length between 1 to 8, similarly to the experiments in [3]. In each iteration a random set of examples (of the same type as the test video) were chosen and their videos together with the corresponding CoT were sent as a prefix to the actual question. It is also worth to note that the examples were checked to be different from the test video in each iteration.

Text-based ablation. LMMs strongly rely on the text context to answer questions. To establish a lower-bound for the LMMs' expected performance, we conducted text-based ablation on blind model performance. We tried removing completely the images from the input prompt, as well as replacing the video images with blank images. However, in both cases the models detected the missing visual input and did not provide a prediction. The models' response was something like: "Actual images are not provided, therefore, this is a hypothetical analysis... Prediction: None".

Listing S3. Instruction prompt for action recognition.

System:
You are an expert video-interaction classifier.

User:
The uploaded images are consecutive frames from a video. The numbers in the file name indicate the order of the
↳ frames in the sequence, so frame_0.jpg is the first frame, followed by frame_1.jpg which is the second
↳ frame, etc.
You will see frames from a short video.
Choose the FIVE best-matching interaction templates (ranked by confidence).
Your goal:
Pick the template ID corresponding to the template label that best describes what happens in the sequence of
↳ frames. Focus on the physical interaction between visible objects.

Illustrative Examples (for clarity):

- If the frames show a human hand placing several books one after another on a shelf, the correct template is "
↳ Putting number of something onto something" - because the action repeats multiple times and involves a
↳ series of objects being placed on another object (the shelf).
- If the frames show a person putting three distinct objects on a table, the correct template is "Putting
↳ something, something and something on the table"
- because exactly three objects are placed on the surface at once.
- If the frames show one object being placed next to another, the correct template is "Putting something next to
↳ something".
- If the frames show an object being placed inside another, the correct template is "Putting something into
↳ something".

Return JSON ONLY (no text or markdown fences):

```
[
  { "template_id": <int> },
  { "template_id": <int> },
  { "template_id": <int> },
  { "template_id": <int> },
  { "template_id": <int> }
]
```

Rules:

- Return exactly 5 objects, ranked most->least confident.
- Each object MUST have only one field: "template_id" (integer from the list below).
- Do NOT include any text, explanations, or reasoning.
- Choose IDs based purely on what the video depicts.
- Prefer the main **physical interaction** over camera motion.

Disambiguation:

- Return exactly 5 objects, ranked most->least confident.
- "into" -> containment / inside relation.
- "onto" -> on top of.
- "next to" -> lateral adjacency without contact stacking.
- "slanted surface" -> object accelerates along a plane.
- "on a flat surface w/o rolling" -> stable placement without motion.

Scoring & Specificity Rules (very important):

- Prefer the template whose action AND number of involved objects best match the scene.
- If THREE distinct objects interact, prefer a 3-object template over any 2-object option, if the action fits.
- Tie-breakers: object count > verb precision > physical outcome > surface/slant qualifiers.
- Don't pick a broader template if a more specific one fits.

Templates (id: label):

- 1: Attaching something to something
- 33: Moving part of something
- 48: Piling something up
- 54: Poking a stack of something without the stack collapsing
- 57: Poking something so that it falls over
- 58: Poking something so that it spins around
- 88: Pulling something onto something
- 97: Pushing something so it spins
- 98: Pushing something so that it almost falls off but doesn't
- 99: Pushing something so that it falls off the table
- 102: Putting number of something onto something
- 120: Putting something, something and something on the table
- 122: Rolling something on a flat surface
- 144: Stacking number of something
- 148: Taking something out of something

Listing S4. Instruction prompt for TS with feedback experiment.

User:
In the next procedure follow these rules:
1) Your output should ALWAYS be the word "Prediction"
↳ followed by the frame number.
2) You will be provided a score with absolute values in
↳ the range [0, 1], representing the correctness
↳ of your prediction.
3) Score of 0 means that your prediction is incorrect,
↳ while an absolute value of 1 means you have
↳ found the correct frame.
4) The absolute values in the range of [0,1] reflect the
↳ proximity of the predicted frame to the true
↳ target frame, where the higher value is better.
5) The sign of the score signifies the direction for
↳ your next prediction. If the sign is positive,
↳ your next prediction should be in the same
↳ direction as the previous prediction. A negative
↳ sign means you should change the direction of
↳ your next prediction. For example, if your last
↳ prediction was frame 5, your current prediction
↳ is frame 6 and the score is negative - your next
↳ prediction should be smaller than 5. On the
↳ other hand, if the last prediction was frame 7,
↳ the current is frame 4 and your score is
↳ negative - you should predict values greater
↳ than 4.
6) You should never predict the same frame twice.
7) Stop your predictions only when the score is above
↳ 0.98.
Now we will perform an iterative session, during which
↳ you will need to find the frame in the provided
↳ image sequence. Follow the instructions above
↳ when prompted with the feedback score.

Listing S5. The prompt template provided to the LMM agent in course of the CoT prompt tuning experiment. The prompt included an introduction describing a general goal the agent was required to achieve; a set of 1 to 8 examples, each including a detailed CoT, which described the objects in the scene and the steps towards the correct solution, and the corresponding frames of the examples; finally, the test sequence was included as the last part of the prompt.

User:
You are an expert in video understanding and motion
↳ analysis.

We are interested in detecting and interpreting
↳ interactions in video sequences. We focus on
↳ interactions between hands and objects.
An interaction usually begins with a contact between a
↳ hand and a target object, when the static target
↳ object starts to move with a hand holding it.
An interaction usually ends when a hand releases a
↳ target object and starts to move without the
↳ object after it ungrasps the object.

You will be provided with consecutive frames from video
↳ sequences, where the order of the frames in each
↳ sequence is determined by the upload ordering
↳ and the ascending index numbers in the file names
↳ .
You are asked to detect the exact frame where the
↳ interaction occurs. Your final answer should be
↳ brief in the format: "Prediction: <frame number
↳ >". Before your final answer, provide a step by
↳ step detailed description of all intermediate
↳ reasoning grounded in the video frames, in
↳ support of your final answer.

Example1:

<example video frame #1>
...
<example video frame #10>

STEPS:
1. Identify the Event:
 < example text >
2. Frames analysis:
 < example text >
3. Conclusion for prediction:
 < example text >.
4. Final answer:
 Prediction: < example true frame >

Example2:
...

Example8:
...

Test:
The following is the test video sequence.
<test video frame #1>
...
<test video frame #10>

Please detect to exact frame where the interaction
↳ occurs. Similar to the examples, provide a
↳ detailed reasoning before your final answer.

2. Detailed results

Event bounding-box detection. We first examine a basic property: whether the model-predicted bounding box contains the true event location point. Tab. S1 reports, for each model, the fraction of ground-truth event points that fall inside the predicted bounding box.

Model	Total label points in bbox	Percentage
GPT-4o	56 / 99	56.57%
GPT-5.2	86 / 99	86.87%
Gemini-2.5-Flash	25 / 99	25.25%
Gemini-3-Pro	92 / 99	92.93 %
Qwen-2.5VL	99 / 99	100%

Table S1. Ground-truth event point locations contained inside the model-predicted bounding boxes.

Although the true event point often lies inside the predicted box (Tab. S1), the Intersection-over-Union (IoU) between the predicted box and a 120×120 pixels box around the true location point are extremely low (Tab. S2, Fig. S4). This discrepancy indicates that while the models often detect the hand or the target object, they tend to ignore important image regions around the target objects and hands, which contain critical information about the interactions. In some cases the predicted bounding box extended beyond the frame boundaries despite explicit instructions to preserve the original spatial scale (see Listing S2). Overall, the models struggle to localize the interaction event regions reliably.

Model	Mean IoU (%)	#IoU \geq 50% / 99
GPT-4o	1.48%	0 / 99
GPT-5.2	8.62%	0 / 99
Gemini-2.5-Flash	0.21%	0 / 99
Gemini-3-Pro	14.74%	2 / 99
Qwen-2.5VL	9.55%	1 / 99

Table S2. Per-model IoU statistics for the event localization task.

Reasoning and Grounding conditions. Tab. S3 and Tab. S4 complements the results for the models Qwen-2.5VL-72B and GPT-4o from our experiments on the two modifying conditions: *Reasoning* and *Grounding*.

Example predictions and associated CoT. Fig. S2 and Fig. S3 show additional examples of false and correct frame predictions. The examples include the models’ Chain-of-Thought, which seemingly presents a logical reasoning text for detecting an interaction event, but the grounding to the video frames is often very loose.

Table S3. Reasoning effect on LMMs’ performance. Models are instructed to describe step-by-step the reasoning behind their prediction of the frame where the interaction occurs. Mean accuracy is measured for the detection of the frame where the test event occurs, within an allowed error tolerance (here the exact or one-frame off the true frame). Results are reported as per 3 ICL regimes (ZS, OS, TS) for the models: (i) Qwen-2.5VL-72B and (ii) GPT-4o.

ICL	RSN	Mean Accuracy Percentage (SD)			
		Exact		1-off	
		Qwen	GPT-4o	Qwen	GPT-4o
ZS	W/O	7.07 (1.83)	9.60 (2.10)	31.31 (3.30)	28.79 (3.23)
	W	10.61 (2.19)	9.09 (2.10)	28.79 (3.23)	31.82 (3.32)
OS	W/O	6.21 (1.38)	12.34 (1.55)	29.81 (2.81)	36.97 (2.63)
	W	11.53 (1.66)	11.29 (1.20)	36.12 (2.58)	36.62 (2.36)
TS	W/O	10.39 (1.72)	14.72 (1.63)	36.44 (3.05)	38.61 (2.86)
	W	14.34 (1.90)	13.44 (1.29)	39.31 (2.76)	40.52 (2.44)

Table S4. Grounding effect on LMMs’ performance. Prior to the main interaction detection task, models are instructed to name the target object and specify the length of the video sequence, to improve their perceptual grounding. Mean accuracy is measured for the detection of the frame where the test event occurs, within an allowed error tolerance (here the exact true frame or up to one-frame off). Results are reported as per 3 ICL regimes (ZS, OS, TS) for the models: (i) Qwen-2.5VL-72B and (ii) GPT-4o.

ICL	GRND	Mean Accuracy Percentage (SD)			
		Exact		1-off	
		Qwen	GPT-4o	Qwen	GPT-4o
ZS	W/O	9.60 (2.10)	10.61 (2.19)	30.30 (3.27)	32.32 (3.33)
	W	8.08 (1.94)	8.08 (1.94)	29.80 (3.26)	28.28 (3.21)
OS	W/O	9.40 (1.58)	12.13 (1.39)	34.59 (2.77)	37.29 (1.39)
	W	8.34 (1.50)	11.49 (1.39)	31.34 (2.63)	36.49 (3.31)
TS	W/O	13.11 (1.72)	14.04 (1.44)	38.92 (2.94)	42.06 (2.57)
	W	11.63 (1.77)	14.11 (1.49)	36.83 (2.89)	37.07 (2.73)

Two-shot with feedback. The results of this experiment showed no improvement in the model’s performance on the test task of predicting the frame where an event occurs in the test video. The mean accuracy is reported in Tab. S5. The results suggest that the feedback session may even interfere with the main prediction task, by shifting away the model from the visual input to the number of the frame, while trying to maximize the feedback score, which is a metric on the prediction error of the frame number.

Nevertheless, an analysis of the model’s weighted success rate and its test error shown in Fig. S6, indicate that when the model is required to provide reasoning which lead to its predictions, the test error remains low up until the 6th feedback iterations, suggesting that some learning may occur. However, as discussed in the main text, the loose grounding to the visual input, becomes even worst with this

Table S5. **Effect of feedback on the model’s performance.** The true label of the second example is provided through an iterative session with feedback, indicating the gap between the predicted and the true frame. Mean accuracy is measured for the detection of the frame where the test event occurs, within an allowed error tolerance (here the exact true frame or up to one-frame off). The evaluation is performed under the two modifying conditions: *Reasoning* and *Grounding*. The evaluated model is Qwen-2.5VL-72B.

Reasoning	Grounding	Mean Accuracy (%)	
		Exact	1-off
W/O	W/O	2.55 ± 1.12	18.37 ± 3.43
W/O	W	5.61 ± 1.76	21.94 ± 3.70
W	W/O	12.59 ± 2.63	41.84 ± 4.15
W	W	16.33 ± 3.06	39.80 ± 4.10

feedback approach since the focus of the model is drawn away from the image contents, trying to satisfy the feedback score around the frame number, rather than grounded visual cues.

The weighted success rate of a test trial (see Fig. S6a,b) was calculated with the conditioned probability as follows

$$p(\text{success} = n | \text{iterations} = k) = \frac{n}{N} \frac{n}{N_k} = \frac{n^2}{NN_k} \quad (3)$$

where N is the total number of successful test trials (i.e., where the agent predicted the correct frame in the test task), and N_k is the number of successful test predictions conditioned on training session having k iterations.

It should be noted that we also tested a setting in where the images were fed to the model in an arbitrary order. This change in image ordering alone resulted in a significant drop in performance, as seen in Tab. S5, despite an explicit instruction that was given to the model that the frames should be treated in consecutive order. From this experiment we conclude that in the functioning of LMMs there is no notion of ordering of frames, unless they are fed in as an ordered sequence.

Chain-of-Thought (CoT) Tuning. In this ICL few-shot experiment, examples included specific CoT descriptions for detecting the frame where the contact/release events occurred. The results in Tab. S6 show that the accuracy does not improve beyond 2 examples. However, introducing an explicit CoT description in the examples yielded enhanced accuracy compared to the experiment without CoT (see Tab. S3).

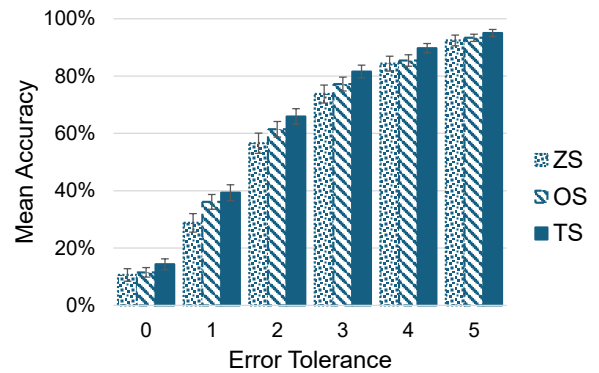
3. Experimental Dataset

Our experimental dataset included 33 videos from SSv2. For each video, we cropped short 10-frame video clips

Table S6. Results for the CoT experiment expressed in mean accuracy within an error tolerance of one-frame off from the true frame in each test sample. The evaluated model is Qwen-2.5VL-72B.

Number of Examples	Mean Accuracy (%)
1	51.52 ± 5.02
2	55.56 ± 4.99
3	45.45 ± 5.00
4	44.44 ± 4.99
5	39.39 ± 4.91
6	43.43 ± 4.98
7	38.38 ± 4.89
8	48.48 ± 5.02

(a) Qwen-2.5VL-72B



(b) GPT-4o

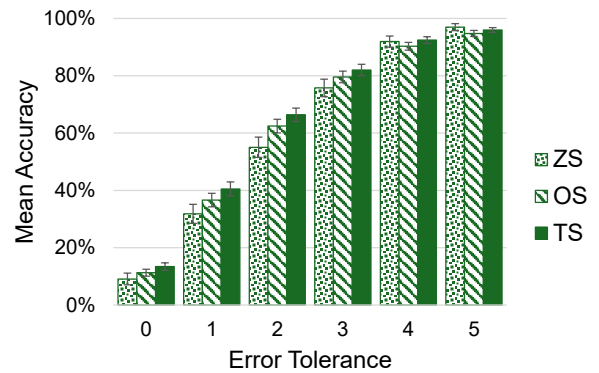




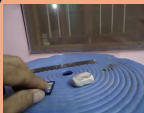





Figure S1. **Mean accuracy vs. detection error tolerance.** A correct detection of the models represents a predicted frame within the allowed error tolerance, where an error tolerance of zero means the exact true frame was predicted. Results of Qwen-2.5VL-72B (a) and GPT-4o (b) are shown for the difference ICL regimes under the "with reasoning" condition. Note, that the length of all videos in the experimental dataset is 10 frames.

around three temporally separated core interaction events, resulting with 99 event clips. By construction, event frames are evenly distributed over the 10-frame window to avoid any bias. In addition, we have ensured that only a single event appears in the time window. Tab. S7 includes the video ID, action template and object placeholders from the original SSv2 dataset. In addition, for each short video clip, the table includes the crop start frame, the frame where the event occurs and the type of the event, i.e., 'contact' or 'release'. All video clips in this set are at 12 fps. Fig. S5 presents a few examples of the 10-frames clips and annotated event frames used in the evaluation. The full annotations are available online at: [ssv2-contact-release-interaction-dataset](#).

<p>Frame 4</p> 	<p>Frame 5</p> 	<p>Frame 6</p> 	<p>Frame 3</p> 	<p>Chain of Thought</p> <p>To determine the frame number that represents the start of an event, let's analyze the sequence of actions depicted in the frames:</p> <ol style="list-style-type: none"> 1. Frame 1**: The hand is positioned above the white circular object, indicating an intention to interact with it. 2. Frame 2**: The hand moves closer to the white circular object. 3. Frame 3**: The hand makes contact with the white circular object. 4. Frame 4**: The hand lifts the white circular object. 5. Frame 5**: The hand places the white circular object back on the surface. 6. Frame 6**: The hand moves away from the white circular object. 7. Frame 7**: The hand returns to the white circular object. 8. Frame 8**: The hand lifts the white circular object again. 9. Frame 9**: The hand places the white circular object back on the surface. 10. Frame 10**: The hand moves away from the white circular object. <p>### Chain of Thought</p> <ul style="list-style-type: none"> - The event of interest here is the interaction between the hand and the white circular object. - The interaction begins when the hand first makes contact with the white circular object. - This contact occurs in Frame 3. <p>### Prediction</p> <p>Prediction 3</p>
<p>Event type: Release</p>				<p>False Prediction</p>

<p>Frame 5</p> 	<p>Frame 6</p> 	<p>Frame 7</p> 	<p>Frame 1</p> 	<p>Chain of Thought</p> <p>To determine the frame number that represents the start of an event, let's analyze the sequence of images step by step:</p> <p>### Chain of Thought:</p> <ol style="list-style-type: none"> 1. Frame 1**: A hand is holding a small black object near a white object on a blue circular surface. The hand is positioned as if it is about to place the black object down. 2. Frame 2**: The hand is still holding the black object, and it appears to be in the process of placing it on the blue surface. 3. Frame 3**: The hand is still holding the black object, and it appears to be in the process of placing it on the blue surface. 4. Frame 4**: The hand is still holding the black object, and it appears to be in the process of placing it on the blue surface. 5. Frame 5**: The hand is no longer visible, and the black object is now resting on the blue surface. The white object remains in place. 6. Frame 6**: The hand reappears, holding the black object again, suggesting a new interaction. 7. Frame 7**: The hand is no longer visible, and the black object is still resting on the blue surface. The white object remains in place. 8. Frame 8**: The hand is no longer visible, and the black object is still resting on the blue surface. The white object remains in place. 9. Frame 9**: The hand is no longer visible, and the black object is still resting on the blue surface. The white object remains in place. 10. Frame 10**: The hand reappears, holding the black object again, suggesting a new interaction. <p>From the analysis, the start of an event occurs when the hand first appears holding the black object and moves it towards the blue surface. This happens in Frame 1, Frame 6, and Frame 10. Since we need to find the earliest start of an event, we focus on Frame 1.</p> <p>### Prediction:</p> <p>Prediction 1</p>
<p>Event type: Release</p>				<p>False Prediction</p>





<p>Frame 8</p> 	<p>Frame 9</p> 	<p>Frame 10</p> 	<p>Frame 1</p> 	<p>Chain of Thought</p> <p>### Chain of Thought:</p> <ol style="list-style-type: none"> 1. Identify the Event**: <ul style="list-style-type: none"> - The event in question is the hand interacting with the container. Specifically, the hand is pointing at or touching the container. 2. Temporal Analysis**: <ul style="list-style-type: none"> - In Frame 1, the hand is already in contact with the container. - From Frame 2 onwards, the hand continues to interact with the container in a similar manner. 3. Determine the Start of the Event**: <ul style="list-style-type: none"> - Since the hand is already in contact with the container in Frame 1, this indicates that the interaction has already started. - There is no clear indication of a prior state where the hand was not interacting with the container. 4. Conclusion**: <ul style="list-style-type: none"> - The start of the event, which is the hand beginning to interact with the container, is in Frame 1. <p>### Prediction:</p> <p>Prediction 1</p>
<p>Event type: Contact</p>				<p>False Prediction</p>

Figure S2. **Example false predictions.** The model (here Qwen-2.5VL-72B) provides the presented chain-of-thought under the "WITH" Reasoning condition. The examples show that the reasoning text seems logical and realistic, but the relation to the actual video frames is often very loose. A red box marks a false prediction, while the orange box marks the true frame.

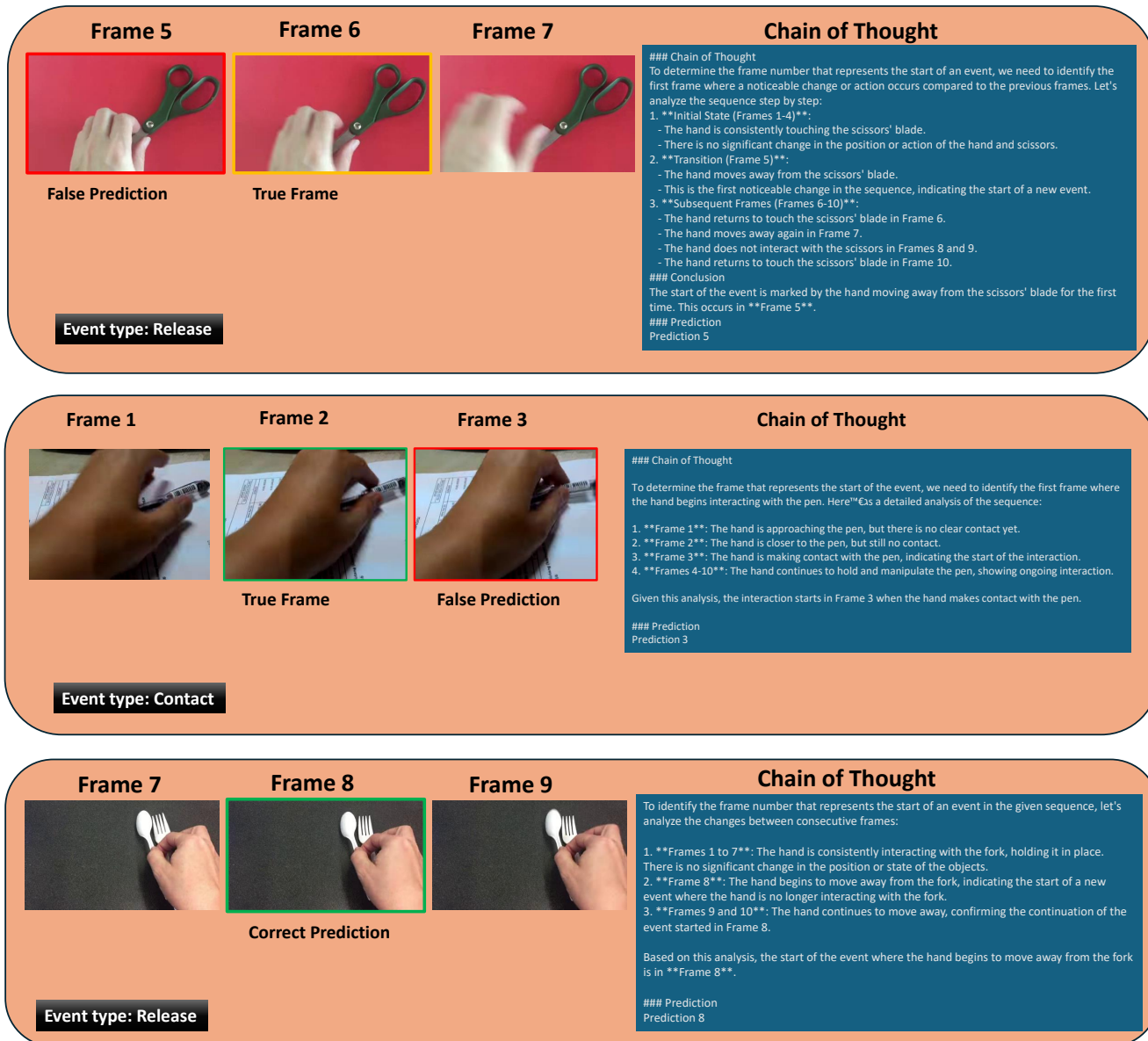


Figure S3. **Additional example predictions.** The top two examples show false predictions of the model (here Qwen-2.5VL-72B), in which the predicted frame was one frame before or after the true frame, where the event occurs. Humans can see clearly the moment of release or contact in the true frames, but the visual cues are too subtle for the models to detect. The Chain-of-Thought text lists the correct reasoning flow, but is not well grounded in the video frames. Orange and green boxes mark the true frame. Red boxes mark false predictions.

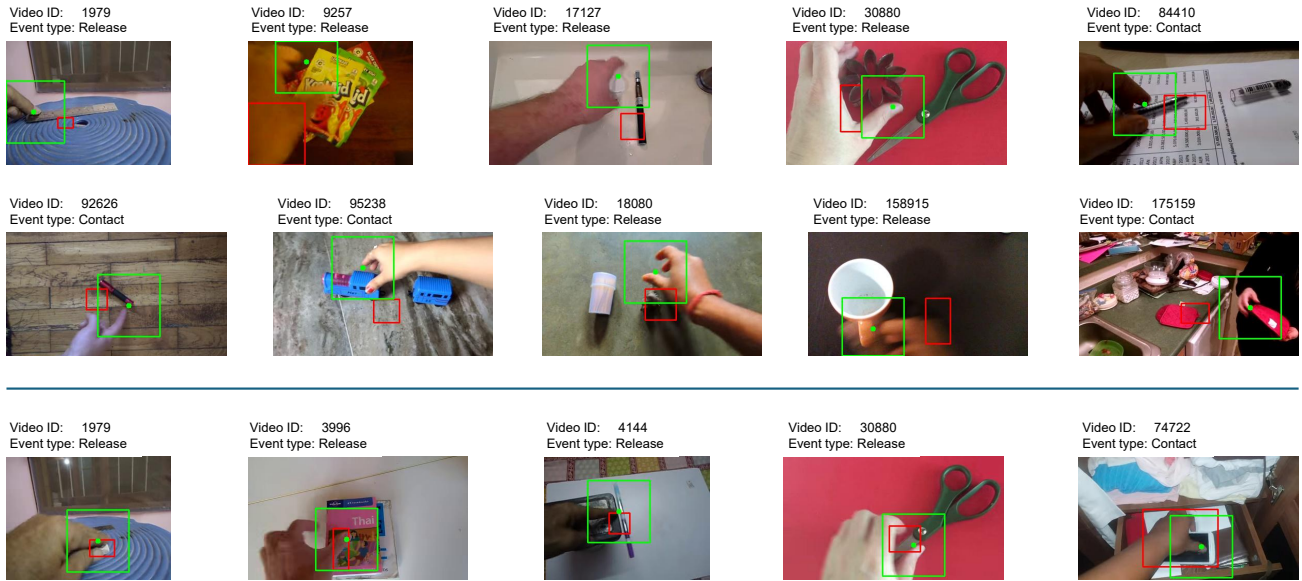


Figure S4. **Bounding box event detection examples of Gemini-3-Pro model.** Upper two rows: Failure detection examples. Bottom row: Successful detection examples. Green dot and bbox indicate the ground truth event location and a 120 × 120 pixels bboux around it. Red bbox indicates the model's prediction.

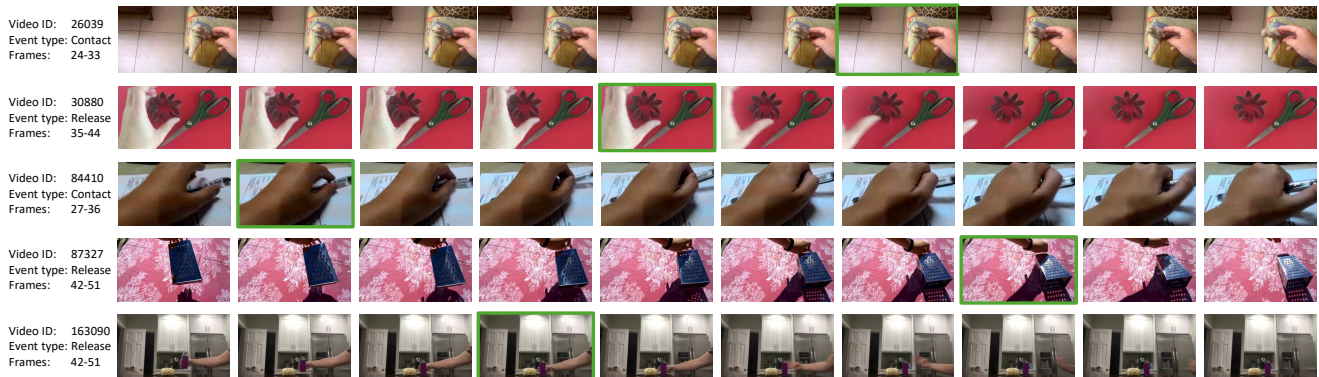
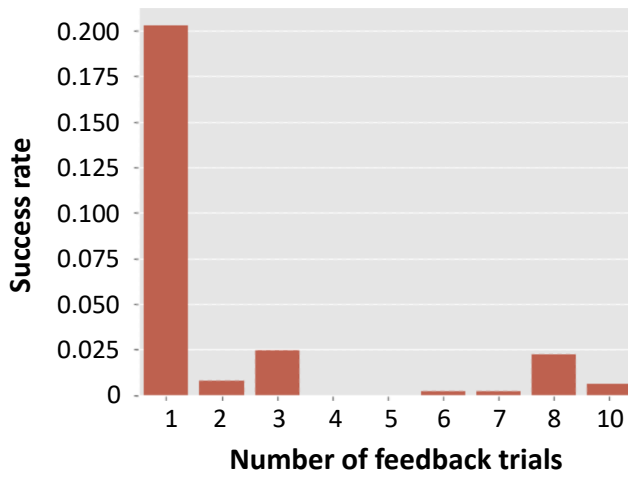
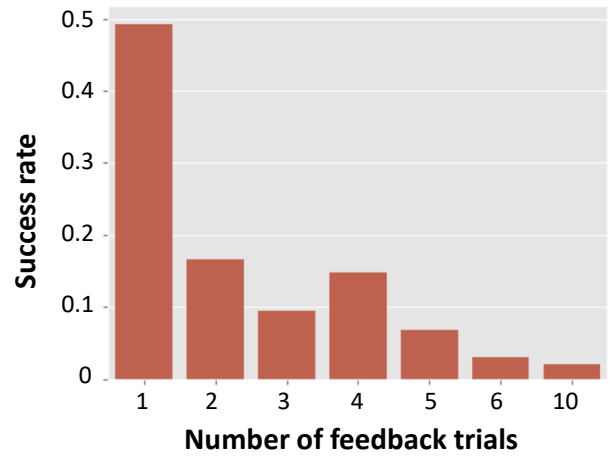


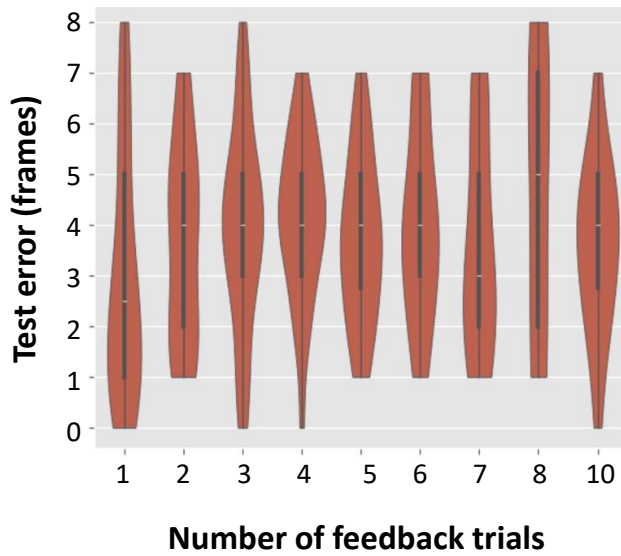
Figure S5. **Full interaction sequence examples from the evaluation dataset.** A green box marks the labeled event frame.



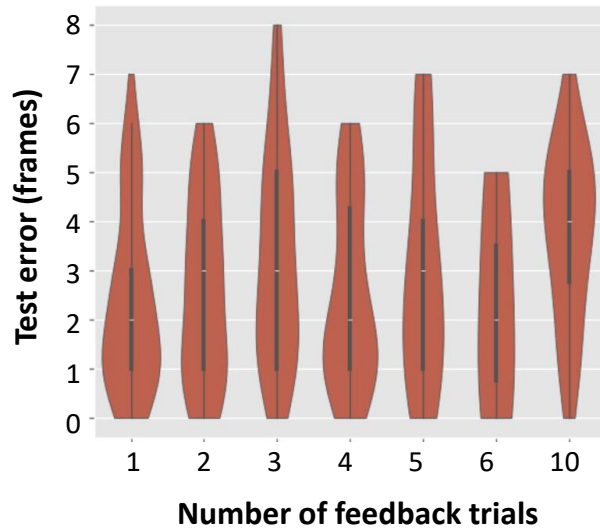
(a) Test prediction success rate - Without Reasoning



(b) Test prediction success rate - With Reasoning



(c) Distribution of test error (frames) - Without Reasoning



(d) Distribution of test error (frames) - With Reasoning

Figure S6. **Effect of feedback on test success rate and error distribution.** In the TS with feedback experiment, the model goes through an iterative session, when it is provided with a feedback score until it predicts correctly the event frame of the second example. (a-b) The model's weighted success rate (see Sec. 2) on the test video vs. the number of trials in the feedback session, without and with Reasoning, respectively. (c-d) The distribution of the test error (in frames) without and with Reasoning, respectively. The results show that when the model is not required to provide reasoning for its prediction, the feedback session does not improve the model's performance in the test task and the test error increases with the number of feedback trials (a, c). On the other hand, when the model is required to provide reasoning, the test success rate decreases monotonically with the number of feedback trials up till the 6th iteration.

Table S7. Experimental dataset information.

Video ID	Action template	Object placeholders	Frame size (width,height)	Event clip ¹ (10 frames)		
				Start frame	Event frame	Event type
1979	Putting [something], [something] and [something] on the table	scale, eraser, sd card	(320, 240)	12 40 60	14 39 55	release release release
2648	Attaching [something] to [something]	dummy peach, peach tree	(427, 240)	11 30 46	18 36 52	contact contact release
3996	Putting number of [something] onto [something]	books, shelf	(427, 240)	8 29 47	13 33 50	release release release
4042	Pushing [something] so it spins	green candy	(427, 240)	7 18 27	10 21 34	release contact release
4144	Poking [something] so that it falls over	pen	(320, 240)	12 19 32	20 22 34	contact release release
9257	Piling [something] up	kool-aid packs	(320, 240)	8 34 52	14 38 57	release release release
12492	Putting [something], [something] and [something] on the table	keys, lock, bulb	(293, 240)	1 10 28	7 17 34	release release release
14990	Putting [something], [something] and [something] on the table	perfume bottle, naphthalene ball, silver ring	(320, 240)	22 32 48	25 37 52	release release release
17127	Putting [something], [something] and [something] on the table	prescribers guide book, medicine bottle, vape pen	(432, 240)	13 36 56	17 39 62	release release release
26039	Pushing [something] so that it falls off the table	toy	(427, 240)	1 13 24	9 20 30	contact contact contact
30880	Putting [something], [something] and [something] on the table	scissors, cookie cutter, grater	(427, 240)	12 35 56	17 39 62	release release release
41434	Stacking [number of] [something]	3, coins	(293, 240)	0 17 37	4 19 43	contact contact contact
57029	Taking [something] out of [something]	tools, toolbox	(427, 240)	0 14 40	1 17 45	contact contact contact
66464	Moving [part] of [something]	tuner, electric guitar	(320, 240)	12 26 36	14 32 40	contact release contact
67618	Putting [something], [something] and [something] on the table	bottle, tube, purse	(427, 240)	0 13 40	1 21 47	release release release
73232	Taking [something] out of [something]	cd, book	(427, 240)	4 11 20	8 18 23	contact contact release
74722	Taking [something] out of [something]	phone, drawer	(427, 240)	0 24 52	8 29 56	contact contact release

Continued on the next page

Table S7. . (cont.)

Video ID	Action template	Object placeholders	Frame size	Event clip (10 frames)		
				Start frame	Event frame	Event type
84410	Attaching [something] to [something]	pen's cover, pen	(427, 240)	6 18 27	10 25 28	contact release contact
87327	Putting [something], [something] and [something] on the table	grater, whisk, corkscrew	(427, 240)	4 28 42	11 31 50	release release release
92626	Poking [something] so that it spins around	flashlight	(427, 240)	2 29 41	7 31 44	contact release contact
95238	Attaching [something] to [something]	toy train engine, its coach	(427, 240)	3 13 40	7 18 43	contact contact contact
96903	Rolling [something] on a flat surface	perfume	(320, 240)	2 15 25	7 18 29	contact contact contact
153413	Putting [something], [something] and [something] on the table	fork, spoon, dish	(427, 240)	10 27 54	17 33 61	release release release
158080	Putting [something], [something] and [something] on the table	toothpick container, showpiece, padlock	(427, 240)	2 21 44	6 27 47	release release release
158915	Putting [something], [something] and [something] on the table	mug, spoon, gum	(427, 240)	1 20 31	5 25 34	release release contact
163090	Putting [something], [something] and [something] on the table	popcorn, vicks vaporub bottle, purple water bottle	(427, 240)	5 26 42	10 29 45	release release release
164784	Pushing [something] so that it almost falls off but doesn't	roll	(320, 240)	3 30 43	6 35 48	contact release contact
166894	Poking a stack of [something] without the stack collapsing	lincoln logs	(427, 240)	15 38 46	19 41 51	contact release contact
175159	Stacking [number of] [something]	5, hot pads	(427, 240)	6 20 39	9 26 41	contact contact contact
175167	Piling [something] up	water color containers	(427, 240)	4 25 41	8 31 44	contact release contact
181367	Piling [something] up	shoes	(320, 240)	0 19 29	2 26 30	contact release contact
186500	Pulling [something] onto [something]	nail clipper, envelope	(427, 240)	11 21 29	17 26 31	contact release contact
217743	Putting [something], [something] and [something] on the table	glass vase, child's shoe, coffee mug	(427, 240)	21 41 58	24 46 65	release release release

¹All video clips in this set are at 12 fps.

References

- [1] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haebel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The” something something” video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. [1](#)
- [2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning, 2025. [1](#)
- [3] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2022. Curran Associates Inc. [1](#), [2](#)
- [4] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2023. Curran Associates Inc. [1](#)