

# VideoAuto-R1: Video Auto Reasoning via Thinking Once, Answering Twice

## Supplementary Material

Shuming Liu<sup>1,2</sup> Mingchen Zhuge<sup>1,2</sup> Changsheng Zhao<sup>1</sup> Jun Chen<sup>1</sup>  
Lemeng Wu<sup>1</sup> Zechun Liu<sup>1</sup> Chenchen Zhu<sup>1</sup> Zhipeng Cai<sup>1</sup> Chong Zhou<sup>1</sup>  
Haozhe Liu<sup>1,2</sup> Ernie Chang<sup>1</sup> Saksham Suri<sup>1</sup> Hongyu Xu<sup>1</sup> Qi Qian<sup>1</sup> Wei Wen<sup>1</sup>  
Balakrishnan Varadarajan<sup>1</sup> Zhuang Liu<sup>3</sup> Hu Xu<sup>1</sup> Florian Bordes<sup>1</sup>  
Raghuraman Krishnamoorthi<sup>1</sup> Bernard Ghanem<sup>2</sup> Vikas Chandra<sup>1</sup> Yunyang Xiong<sup>1</sup>

<sup>1</sup>Meta AI <sup>2</sup>King Abdullah University of Science and Technology (KAUST) <sup>3</sup>Princeton University

In this appendix, we provide more details of our method and present more experimental results. Specifically, we provide the implementation details of VideoAuto-R1 in Section A. Then, we present the training data and its ablations in Section B. Next, we introduce the details of our reward design and related analysis in Section C. After this, we present the prompt template used in our ablation experiments in Section D, as well as the training curve in Section E. We also provide the algorithm details of our inference strategy in Section F. Then, we show additional experiments and further analysis in Section G. Next, we discuss our limitations in Section H. Finally, we provide more examples for visualization and discussion in Section I.

### A. Implementation Details

**Training.** During training, the maximum number of total video tokens is set to 4,096, and the maximum number of frames is set to 256. We use AdamW [12] as the optimizer, with a learning rate of  $1 \times 10^{-6}$ , weight decay of 0.01, and a maximum gradient norm of 1.0. A constant learning rate schedule without warm-up is employed. The KL penalty coefficient  $\beta$  is set to 0.01. The global batch size is set to 256, and we train the model for one epoch. The visual encoder remains frozen; only the projector and the LLM are fine-tuned. We leverage DeepSpeed [14] and vLLM [8] to accelerate the training. For GRPO rollout generation, we set the rollout size  $G$  to 16 and use a temperature of 1.0 to encourage exploration. Our training is conducted on 32 H100 GPUs for approximately 35 hours.

**Testing.** During testing, all evaluations are performed using `lmms-eval` [25] with greedy decoding (temperature 0). The maximum response length is set to 4,096 tokens, ensuring that no truncation occurs during evaluation. The early-exit confidence threshold  $\tau$  is set to 0.97, which is tuned on a small held-out subset of the validation data and

Table 1. **Training Dataset.** We include text, image, and video data during training, with a total of 83K samples.

Type	Size	Details
Text	6.4K	DAPO-Math [21]
Image	27.5K	ViRL [15], ThinkLite-Hard [17]
Video	49.4K	Video-R1 [6], TVBench [4], STI-Bench [11], MMR-VBench [27], Charades-STA [7], ActivityNet [5], Time-R1 [18], NExT-GQA [19]

generalizes well to unseen evaluation benchmarks. For the Qwen2.5-VL model, we allow up to 16K total video tokens and vary the maximum number of frames among  $\{64, 128, 256\}$ . For the Qwen3-VL model, we allow up to 128K total video tokens and sweep over  $\{64, 256, 2048\}$  frames. Following [1] and [2], we report the highest performance across these settings.

### B. Training Data

**Data Composition.** As described in the main paper, our training data consists of text, image, and video modalities. For text-based reasoning, we incorporate DAPO-Math [21]; for image-based reasoning, we include ViRL [15] and ThinkLite-Hard [17]. For video QA, we draw from several sources including Video-R1 [6], TVBench [4], STI-Bench [11], and MMR-VBench [27]. To enhance temporal grounding and grounding-based QA capabilities, we additionally include Charades-STA [7], ActivityNet [5], Time-R1 [18], and NExT-GQA [19]. All test samples from our evaluation benchmarks are manually excluded to prevent data leakage. The resulting training pool comprises approximately 137K samples.

Table 2. **Performance Comparison across Different Training Data and Filtering Strategy.** Note that we report the results under the RL with CoT setting. Combining text, image, and video data yields the best overall performance. Filtering out overly easy and hard samples consistently improves results while reducing dataset size, validating the effectiveness of our data curation pipeline.

Training Data	Filtered	Size	VideoMME	MVBench	VideoMMMU	Charades-STA
Text	✗	17K	63.3	62.6	45.8	38.6
Image	✗	50K	<b>65.6</b>	66.8	52.8	40.1
Video	✗	70K	64.7	<b>71.0</b>	<b>55.1</b>	<b>59.0</b>
Text + Image	✗	67K	66.1	67.4	53.3	41.6
	✓	34K	<b>67.0</b>	<b>68.5</b>	<b>56.4</b>	<b>42.0</b>
Text + Image + Video	✗	138K	65.4	71.0	55.4	59.7
	✓	83K	<b>66.1</b>	<b>71.2</b>	<b>56.4</b>	<b>59.8</b>

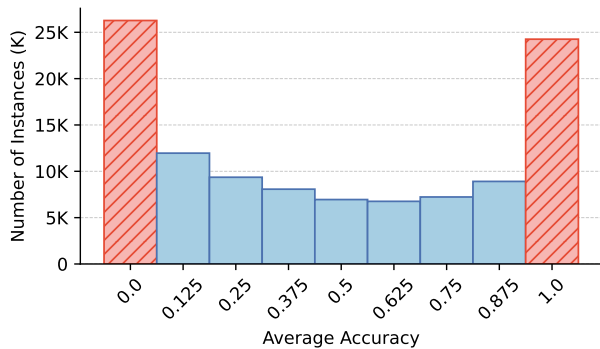


Figure 1. **Distribution of per-sample accuracy** in the initial training pool, estimated by evaluating 8 diverse responses per sample. Samples with all responses correct or all incorrect are considered too easy or too hard and are excluded from QA-based data.

**Filtering Pipeline.** We further curate a smaller, higher-quality subset from the initial data pool. First, we remove samples with invalid ground-truth (using `math-verify` for math problems and rule-based checks for QA problems). Next, for each remaining sample, we generate 8 responses using the base model (*i.e.*, Qwen2.5-VL-7B-Instruct [2]) with a high temperature. A smaller LLM (*i.e.*, Qwen3-30B-A3B-Instruct [20]) evaluates each response against the ground truth and assigns correct/incorrect labels. Samples for which all 8 responses are correct (too easy) or all are incorrect (too hard) are discarded, as they contribute little to GRPO-based reinforcement learning, as illustrated in Figure 1. This difficulty-based filtering is applied only to QA tasks; for temporal grounding, we retain all samples to mitigate the base model’s grounding weakness. After filtering, we finally obtain 83K samples. The detailed composition is listed in Table 1.

**Effectiveness of Data Filtering.** To evaluate the effectiveness of our data filtering pipeline, we analyze the results presented in Table 2. Two key observations emerge from this analysis. **First**, training solely on text data leads to a

noticeable drop in performance on video tasks compared to the Qwen baseline, suggesting a domain shift and poor generalization. Adding image data significantly improves video QA performance, particularly on VideoMMMU, highlighting the importance of image-based math and reasoning data. However, due to the absence of temporal grounding data, performance on the Charades-STA benchmark remains low. When combining text, image, and video data, the model achieves the best overall performance under both filtered and unfiltered settings. **Second**, in both the text+image and text+image+video configurations, removing overly easy or difficult samples leads to consistent performance gains. Additionally, this filtering reduces the number of training samples, thereby improving training efficiency. These findings validate the effectiveness of our data filtering strategy for GRPO-based reinforcement learning.

### C. Reward Designs

To complement the reward description in the main paper, we provide the details below. Our overall reward is defined as a weighted sum of the task reward and the format reward.

**Task Reward.** We consider three task types for computing task rewards: QA, temporal grounding, and grounding QA.

- *Question Answering.* For math problems, we use `math-verify` to compare the prediction with the ground truth; otherwise we compare normalized strings (*e.g.*, case-folded, whitespace stripped). This yields a binary reward

$$R_{QA}(o_i) \in \{0, 1\}.$$

- *Temporal Grounding.* Let the ground-truth segments be  $\mathcal{G} = \{[s_j, e_j]\}_j$  and the predicted segments be  $\hat{\mathcal{G}} = \{[\hat{s}_k, \hat{e}_k]\}_k$  (either set may contain one or multiple segments). We compute the temporal IoU and take the best matching pair with the largest tIoU. If no valid segment can be parsed, we assign  $R_{TG}(o_i) = 0$ .

$$R_{TG}(o_i) = \max_{[\hat{s}, \hat{e}] \in \hat{\mathcal{G}}, [s, e] \in \mathcal{G}} \text{tIoU}([\hat{s}, \hat{e}], [s, e]) \in [0, 1],$$

- *Grounding QA*. We parse the textual answer and the predicted segments from the model output, compute  $R_{QA}(o_i)$  and  $R_{TG}(o_i)$  as above, and sum them:

$$R_{GQA}(o_i) = R_{QA}(o_i) + R_{TG}(o_i) \in [0, 2].$$

**Format Reward.** In addition to task correctness, we use a binary format reward  $R_{fmt}(o_i) \in \{0, 1\}$  enforced via strict regex checks. For VideoAuto-R1, we require exactly two `\boxed{...}` answers, and in between one `<think>...</think>` block, with no additional text before, between, or after.

**Analysis of the Dual-Answer Reward Design.** In Section 4.2 of the main paper, we introduce the dual-answer reward design used during training. The key components of this design are the weight coefficients  $w_1$  and  $w_2$  assigned to the initial and reviewed answers, respectively, as well as the fallback bonus weight  $\alpha$ . Table 3 summarizes the effects of different choices for these coefficients.

First, when  $w_1 = w_2$ , the model assigns identical rewards to two distinct cases: (i) the first answer is correct but the second is wrong, and (ii) the first answer is wrong but the second is correct. However, our intention is to prioritize the correctness of the *reviewed answer*, since users who permit step-by-step reasoning with a sufficient compute budget expect the final answer to be reliable. Therefore, equal weighting fails to distinguish these two scenarios. By choosing  $w_1 < w_2$  (e.g., 0.9 : 1.1), the total reward becomes 0.9 for a “correct  $\rightarrow$  wrong” pattern, but 1.1 for “wrong  $\rightarrow$  correct”, thereby encouraging the model to produce accurate reviewed answers during RL.

Second, even with  $w_1 < w_2$ , the model still assigns the same reward when the first output is an incorrect guess or a fallback string “*Let’s analyze the problem step-by-step.*” The fallback string is not a wrong prediction; rather, it is an explicit and honest signal that the model identifies the task as difficult and intentionally defers reasoning to the next stage. Such behavior should be incentivized. By introducing the fallback bonus  $\alpha$ , as shown in the last column of Table 3, the model is able to clearly differentiate between an incorrect guess and a fallback indicator.

Finally, when both the initial and reviewed answers are correct, the model receives the highest possible reward, which aligns with our design goal.

## D. Prompt Template

In the main paper, we introduce the system prompt used in VideoAuto-R1, which adopts an *answer  $\rightarrow$  think  $\rightarrow$  answer* format. This prompt design avoids a cold-start stage and facilitates stable training with promising performance. Additionally, in Table 5 of the main paper, we explore alternative reinforcement learning settings.

Table 3. Effects of Dual-Answer Reward Coefficients.

First Answer	Second Answer	$w_1 = 1, w_2 = 1, \alpha = 0$	$w_1 = 0.9, w_2 = 1.1, \alpha = 0$	$w_1 = 0.9, w_2 = 1.1, \alpha = 0.3$
$\times$	$\times$	0	0	0
<i>Let’s analyze...</i>	$\times$	0	0	0
$\checkmark$	$\times$	1	0.9	0.9
$\times$	$\checkmark$	1	1.1	1.1
<i>Let’s analyze...</i>	$\checkmark$	1	1.1	1.4
$\checkmark$	$\checkmark$	2	2	2

**RL without Thinking.** As shown in Table 4, this variant directly applies GRPO without requiring any intermediate explanation. The model is prompted to provide only the final answer enclosed in a `\boxed{}` command.

Table 4. System Prompt for RL without Thinking.

You are a helpful assistant. Put your final answer in `\boxed{}`.

**RL with Thinking.** As shown in Table 5, this is the standard prompt for GRPO training. The model first generates a reasoning trace within `<think>` `</think>` tags, followed by the final answer enclosed in `\boxed{}`. This prompt format aligns with previous R1-style approaches such as Video-R1 [6] and VideoChat-R1 [9].

Table 5. System Prompt for RL with Thinking.

You are a helpful assistant.

FIRST, think through the reasoning process as an internal monologue, and THEN provide the final answer. The reasoning process MUST be enclosed within `<think>` `</think>` tags, and the final answer MUST be wrapped in `\boxed{}`.

## E. Training Curve

To better understand the behavior of VideoAuto-R1, we visualize the training curves of the task rewards for both the initial and reviewed answers during training, as shown in Figure 2. We highlight three key observations below.

**Reviewed Answer vs. Initial Answer.** For both Qwen2.5-VL-7B and Qwen3-VL-8B, the reviewed answer consistently achieves a higher task reward than the initial answer during training. This performance gap remains stable after convergence, indicating that the *answer–think–answer* paradigm effectively leverages intermediate reasoning to refine predictions. Moreover, this confirms that the dual-answer reward design (with  $w_2 > w_1$ ) can encourage the model to treat the second answer as a meaningful revision rather than a naive re-sampling of the first.

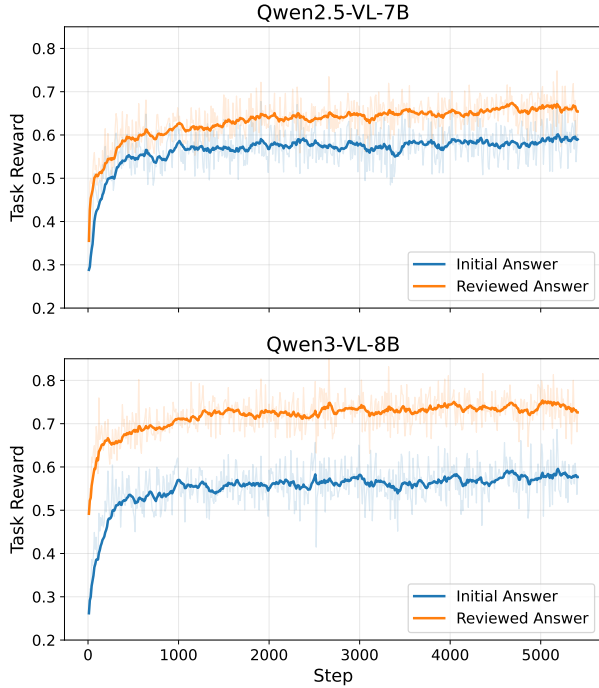


Figure 2. **Training Curves of VideoAuto-R1.** We show the average task reward for both initial and reviewed answers during GRPO training.

**Training Dynamics.** As training progresses, the task rewards for both answers increase. In the early stages, we observe a rapid improvement, followed by a slower but steady rise until convergence. This pattern suggests that GRPO quickly captures coarse task structure and gradually optimizes finer-grained reasoning capabilities over time.

**Impact of Backbone Capacity.** Throughout training, Qwen3-VL-8B consistently outperforms Qwen2.5-VL-7B in both answers. The stronger backbone benefits from better initialization and sustains a higher reward margin after convergence. These results demonstrate that VideoAuto-R1 scales effectively with model capacity: larger base models can more fully exploit dual-answer supervision and confidence-based reasoning, resulting in higher final results.

## F. Inference Strategy

At test time, VideoAuto-R1 employs a confidence-based early-exit mechanism to determine whether to stop after generating the initial direct answer or to proceed with a full chain-of-thought rationale followed by a reviewed answer. Algorithm 1 summarizes this procedure, which consists of three main steps: (1) generate the initial answer, (2) compute its confidence score, and (3) decide whether to exit early or continue reasoning.

For implementation simplicity, we terminate generation

---

### Algorithm 1 Inference Strategy of VideoAuto-R1

---

**Require:** Trained model  $p_\theta$ , video  $v$ , question  $q$ , confidence threshold  $\tau$ , fallback string  $f$

**Ensure:** Predicted answer  $\hat{a}$

- 1: Given input  $(v, q)$ , perform greedy decoding until the first `<think>` tag is generated.
  - 2: Let  $a_1 = (t_1, \dots, t_L)$  be the tokens inside the first box, and let  $y_{\leq \ell_0}$  denote the prefix up to (and including) the opening of  $a_1$
  - 3: **if**  $a_1 = f$  **then** ▷ designated fallback string
  - 4:      $s(a_1) \leftarrow -1e6$
  - 5: **else**
  - 6:     Compute length-normalized confidence  $s(a_1) \leftarrow \frac{1}{L} \sum_{\ell=1}^L \log p_\theta(t_\ell \mid y_{\leq \ell_0 + \ell - 1}, x)$
  - 7: **end if**
  - 8: **if**  $s(a_1) \geq \log \tau$  **then** ▷ early exit
  - 9:     Accept the initial answer
  - 10:    **return**  $\hat{a} \leftarrow a_1$
  - 11: **else** ▷ continue reasoning
  - 12:     Resume decoding from the current prefix
  - 13:     Generate rationale  $r$  enclosed in `<think>...</think>` and the second boxed answer  $a_2$
  - 14:     **return**  $\hat{a} \leftarrow a_2$
  - 15: **end if**
- 

early by detecting the appearance of the opening `<think>` tag during greedy decoding. We then extract the token sequence enclosed in the first `\boxed{}` block, which always precedes the `<think>` tag. Since the initial answer  $a_1$  typically consists of only a few tokens, this strategy enables low-overhead confidence computation while providing substantial savings in decoding latency and token budget whenever early exit is triggered.

## G. Additional Experiments

In this section, we present additional experiments and analyses to complement the findings reported in the main paper.

### G.1. Results on Image Benchmarks

Although VideoAuto-R1 is primarily designed for video understanding, we also evaluate its performance on several image reasoning benchmarks. As shown in Table 6, VideoAuto-R1 consistently outperforms the Qwen baseline on MathVista [13], MathVision [16], MathVerse [26], MMMU [23], MMMU-Pro [24], and MM-Vet [22]. For example, it improves from 69.4% to 73.7% on MathVista and from 51.3% to 53.8% on MMMU.

These improvements are largely attributable to the inclusion of image-centric math and reasoning data during training, which strengthens the model’s visual reasoning skills

Table 6. **Evaluation Results on Image Benchmarks.** The Qwen baseline and VideoAuto-R1 are evaluated under the same settings.

Model	MathVista testmini	MathVision testmini	MathVerse testmini	MMMU val	MMMU-Pro overall	MM-Vet test
Qwen2.5-VL-7B	69.4	26.3	44.8	51.3	36.1	60.0
<b>VideoAuto-R1</b> <sub>(Qwen2.5-VL-7B)</sub>	<b>73.7</b>	<b>29.6</b>	<b>46.9</b>	<b>53.8</b>	<b>39.8</b>	<b>61.9</b>

Table 7. **Evaluation Results on Video QA Benchmarks with Different Frames.** For the Qwen2.5-VL models, we allow up to 16K total video tokens. For the Qwen3-VL models, we allow up to 128K total video tokens.

Model	Frames	Video Perception Benchmark				Video Reasoning Benchmark	
		VideoMME	MVBench	LongVideoBench	MMVU	VideoMMMU	MVP
Qwen2.5-VL-7B	64	63.1	67.0	59.7	<b>66.2</b>	<b>54.6</b>	35.8
Qwen2.5-VL-7B	128	65.9	67.0	60.6	66.2	54.7	35.8
Qwen2.5-VL-7B	256	<b>66.0</b>	<b>67.1</b>	<b>60.9</b>	65.7	52.7	<b>36.5</b>
<b>VideoAuto-R1</b> <sub>(Qwen2.5-VL-7B)</sub>	64	64.6	71.0	60.0	<b>69.7</b>	<b>58.7</b>	39.2
<b>VideoAuto-R1</b> <sub>(Qwen2.5-VL-7B)</sub>	128	66.7	71.0	60.4	69.1	56.6	39.3
<b>VideoAuto-R1</b> <sub>(Qwen2.5-VL-7B)</sub>	256	<b>67.3</b>	<b>71.0</b>	<b>60.5</b>	68.6	56.7	<b>39.4</b>
Qwen3-VL-8B	64	67.3	69.4	63.4	69.9	<b>61.0</b>	40.4
Qwen3-VL-8B	256	70.9	69.4	66.0	69.6	59.9	40.5
Qwen3-VL-8B	2048	<b>72.5</b>	<b>69.4</b>	<b>67.6</b>	<b>69.9</b>	59.8	<b>40.5</b>
<b>VideoAuto-R1</b> <sub>(Qwen3-VL-8B)</sub>	64	67.9	71.8	63.9	71.0	<b>65.0</b>	42.7
<b>VideoAuto-R1</b> <sub>(Qwen3-VL-8B)</sub>	256	70.4	72.0	67.1	71.0	63.8	42.9
<b>VideoAuto-R1</b> <sub>(Qwen3-VL-8B)</sub>	2048	<b>71.7</b>	<b>72.0</b>	<b>67.4</b>	<b>71.1</b>	64.0	<b>43.0</b>

beyond the video domain. At the same time, the thinking once, answering twice design and dual-answer reward transfer naturally to static images, where the model can still benefit from an internal reasoning stage before giving a reviewed answer. Together, the results demonstrate that VideoAuto-R1 is not only effective for video understanding, but also exhibits strong generalization to challenging image benchmarks.

## G.2. Performance with Different Frames

In the main paper, we report the best-performing configurations of our model. Here, we present the complete results in Table 7 and analyze how the number of input frames affects performance on both perception and reasoning benchmarks.

Under a 16K video-token budget using Qwen2.5-VL, increasing the number of frames from 64 to 256 yields noticeable improvements on most perception benchmarks for both the Qwen baseline and VideoAuto-R1. For example, accuracy on VideoMME improves from 63.1% to 66.0%, and on LongVideoBench from 59.7% to 60.9%. However, the reasoning-oriented benchmark VideoMMMU shows weaker dependence on frame count, where performance slightly decreases with additional frames. This trend persists when switching to Qwen3-VL, which supports a larger 128K video-token budget and up to 2,048 frames.

Moreover, VideoAuto-R1 achieves consistent improvements compared to the Qwen baseline. For instance, even

under a 64-frame budget, VideoAuto-R1 improves upon the baseline performance from 63.1% to 64.6% on VideoMME, and from 66.2% to 69.7% on MMVU, demonstrating the effectiveness of our proposed approach across both low and high frame regimes.

## G.3. Analysis on Temporal Grounding Benchmarks

In the main paper, we emphasize that for grounding benchmarks, the initial answer is typically sufficient, so we exit early by default to save computation. In Table 8, we report the detailed grounding results when using the first boxed answer, the second boxed answer, and the confidence-based auto strategy.

**Initial vs. Reviewed Answer.** Unlike video QA benchmarks, temporal grounding shows almost no gap between the first and reviewed answers. For VideoAuto-R1, mIoU is the same for ActivityNet and NEXT-GQA when comparing the first and second boxed answers. On NEXT-GQA, the grounding QA accuracy also remains the same.

We hypothesize two reasons for this phenomenon. **First**, since the grounding procedure does not require multi-step logical deduction, the model can map the queried event to a time span directly from perception. Once the model has localized a segment in the first answer, additional textual reasoning has limited room to further improve the IoU. **Second**, since we lack the SFT stage to teach the model how to explicitly reason on the grounding task, the model cannot

Table 8. **Comparison of Different Inference Strategies on Temporal Grounding Benchmarks.** We compare the results using the first boxed answer, the second boxed answer, or the confidence-based early-exit answer. We observe that on grounding benchmark, the first boxed answer is typically sufficient, so we early-exit without further reasoning to save computation.

Model	Inference Strategy	0.3	ActivityNet			NEXT-GQA	
			0.5	0.7	mIoU	Acc	mIoU
VideoAuto-R1 (Qwen2.5-VL-7B)	First Answer	69.2	48.5	27.3	47.6	80.6	36.7
	Second Answer	69.2	48.5	27.3	47.6	80.6	36.7
	Auto	69.2	48.5	27.3	47.6	80.6	36.7

easily refine the predicted segments. Consequently, the reasoning stage rarely corrects localization errors, leading to nearly identical scores. In practice, this suggests that for grounding tasks, RL still shows significant improvements compared to baseline or SFT, but it is often unnecessary to rely on long and language-based thinking rationales.

**Reasoning Traces on QA vs. Grounding.** To better understand this behavior, we examine representative reasoning traces of VideoAuto-R1 between grounding and QA tasks, as shown in Figure 6, 7, and 9. On video QA benchmarks, the thinking rationale usually contains multi-step analysis: enumerating visual evidence, performing arithmetic, or checking answer options. In contrast, grounding traces are much shorter. The model typically identifies the relevant event or shot, notes when it appears and disappears in the video, and then outputs the corresponding timestamps or intervals.

These qualitative observations align with the quantitative results in Table 8: for temporal grounding benchmarks, explicit reasoning provides limited additional benefit over the direct localization. Therefore, we use the direct answering results on grounding benchmarks for VideoAuto-R1.

#### G.4. Analysis of the Early-Exit Threshold

Figure 3 further studies the impact of the early-exit threshold  $\tau$  on accuracy and think ratio under our confidence-based routing. As  $\tau$  increases, early exit becomes more conservative, leading to a monotonic rise in the think ratio. Therefore,  $\tau$  provides a direct and continuous control knob to trade efficiency for accuracy within a unified inference rule.

On reasoning-intensive benchmarks, higher  $\tau$  consistently improves accuracy alongside increased reasoning usage. For VideoMMMU, rising  $\tau$  from 0.86 to 0.98 improves accuracy from 57.5% to 58.7% while increasing the think ratio from 29% to 55%. Similarly on MVP, accuracy increases from 39.16% to 39.37% as the think ratio rises from 20% to 51%. These trends indicate that when the initial answer is less reliable, the reviewed-answer stage offers meaningful corrective benefits for these reasoning samples.

In contrast, on perception-oriented VideoMME, accuracy remains essentially unchanged across thresholds, whereas the think ratio still increases. This suggests diminishing returns from additional reasoning for easy per-

ceptual queries. Based on these observations, we set  $\tau = 0.97$  as a robust default that preserves satisfied accuracy on reasoning-heavy tasks while limiting unnecessary CoT invocation on perception-heavy data, without requiring dataset-specific tuning.

#### G.5. Analysis of the Impact of Cold-Start SFT

In our training framework, we deliberately omit chain-of-thought SFT and proceed directly to RL. Traditionally, SFT is used to (1) teach the CoT output format, (2) imitate the CoT reasoning process, and (3) acquire general knowledge from newly collected data. However, with modern base models that are already trained on massive corpora, the marginal benefit for (1) and (3) is limited. Moreover, collecting large-scale, high-quality CoT traces for (2) is expensive and often noisy.

Table 9. **Ablation on Cold-Start CoT SFT.**

Setting	VideoMME	MVBench	VideoMMMU
Qwen2.5-VL baseline	66.0	67.1	54.7
SFT with Video-R1-CoT data	60.1	64.0	53.8
RL with thinking	66.1	71.2	56.4
SFT $\rightarrow$ RL with thinking	61.7	64.3	53.5

In early experiments, SFT on Video-R1-CoT data [6], which has both the intermediate reasoning traces and final answer, not only failed to improve performance, but actually degraded the Qwen2.5-VL baseline, a phenomenon also observed in prior work [3, 10]. Table 9 summarizes this effect. Pure SFT substantially hurts performance across all three benchmarks. When we apply GRPO starting from the SFT checkpoint (“SFT  $\rightarrow$  RL with thinking”), the final model remains significantly worse than RL applied directly on the base model.

These results suggest that low-quality CoT supervision can distort the behavior of a strong base model and create a poor initialization for RL. We therefore focus on directly incentivizing the base model’s reasoning via GRPO-style reinforcement learning.

#### H. Limitations

In this section, we mainly discuss three limitations of our work and leave them as future work.

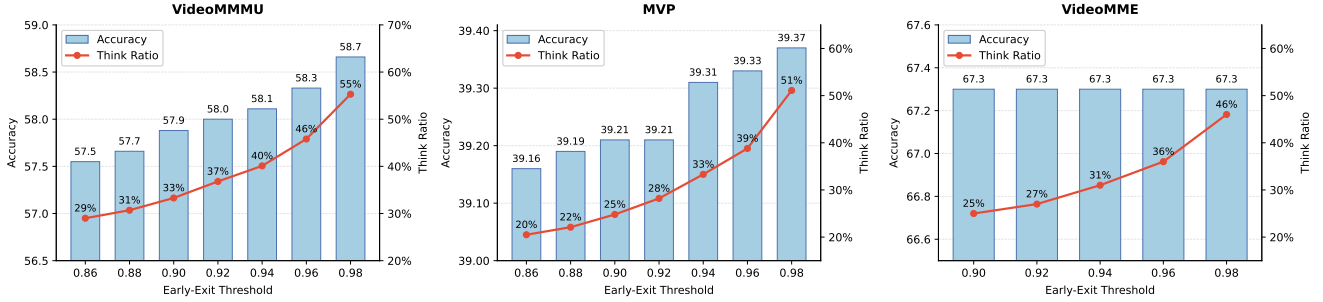


Figure 3. Effect of the Early-Exit Threshold on Accuracy and Think Ratio. In practice, we set  $\tau = 0.97$  for all datasets.

**First**, our distinction between direct answering and reasoning is currently made purely at test time via a confidence-based early-exit rule on the first boxed answer. While this mechanism is simple and effective, it does not explicitly shape the confidence distribution during training. A natural extension would be to incorporate the probability of the first boxed answer into the training objective itself: for simple questions, the model should be encouraged to assign high confidence to a correct direct answer, whereas for genuinely hard questions it should learn to keep the initial confidence low and defer to the reasoning stage. Jointly optimizing both accuracy and calibrated confidence could further improve the reliability of the early-exit policy.

**Second**, our current reasoning mechanism relies strictly on language-based chain-of-thought. While effective for symbolic and logical tasks, we observe that such textual reasoning yields limited improvements on perception-oriented QA and temporal grounding benchmarks compared to direct answering. This suggests that purely semantic rationales may be insufficient to correct fine-grained visual perception errors or refine precise temporal boundaries once the initial visual encoding is fixed. Future work could explore interleaved multimodal reasoning paradigms, such as “thinking with frames”, where the model explicitly revisits video segments or visual features during the reasoning process to enhance perceptual precision and grounding accuracy.

**Third**, the existing video reasoning benchmarks are still limited in scope and difficulty. Many datasets contain relatively short clips and perception-oriented questions. More advanced benchmarks that stress long-range temporal dependencies, compositional logic, and counterfactual reasoning, rather than just math or symbolic-heavy problems, are needed to more faithfully evaluate and compare the reasoning capabilities of MLLMs.

**Fourth**, truly “*must-think*” video data, where multi-step reasoning is indispensable rather than merely helpful, remains scarce. Constructing high-quality, large-scale video datasets that explicitly require deep reasoning (for example, multi-event causal chains, non-trivial temporal puzzles, or physically challenging scenarios) is therefore an urgent and valuable direction for future work. In the meantime, explor-

ing the advanced reasoning pattern for the grounding task is also an interesting direction.

## I. Qualitative Examples

In this section, we provide additional qualitative results to support our analysis.

In Figure 4, we first present a failure case of VideoChat-R1 [9], where the direct answer is correct but the CoT-reasoned result is incorrect. Although the model generates a seemingly reasonable step-by-step rationale, it suffers from hallucinations. For example, it mistakenly describes dancing details that are not present at the end of the video. These errors often stem from a single step of misperception or flawed reasoning, yet they ultimately lead to incorrect final answers. In contrast, the direct answer provides an accurate and concise response for such perception-oriented tasks.

In Figure 5, we also show a success case of VideoChat-R1 on VideoMMM. Unlike perception-oriented examples, this question involves a science problem based on an instructional video. In this context, the chain-of-thought reasoning process demonstrates a clear advantage: the model performs step-by-step deduction, correctly computes equations, and arrives at the final numerical result, which would be challenging via direct answering alone.

Next, we present qualitative results from VideoAuto-R1 across different benchmark types. In Figure 6, we illustrate the model’s outputs on temporal grounding tasks. For these examples, the reasoning trace is typically straightforward—often limited to identifying when the action begins and ends. In many cases, the initial and reviewed answers are identical. Based on this observation, we apply early-exit directly on temporal grounding tasks without invoking further reasoning, which leads to reduced computation without sacrificing accuracy.

In Figure 7, we show results on perception-oriented QA benchmarks. For these relatively simple visual questions, VideoAuto-R1 consistently provides accurate responses in the initial answer, often accompanied by a high confidence score (e.g., over 99%). These examples trigger early-exit behavior, allowing the model to maintain strong accuracy

while improving inference efficiency.

In Figures 8, we showcase examples from reasoning-intensive QA benchmarks. Compared to perception-oriented tasks, the reasoning traces here are significantly longer, with more detailed deduction steps. Notably, the model’s confidence in the initial answer is relatively low in such cases, allowing our confidence-based inference mechanism to trigger reasoning effectively.

## References

- [1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibao Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. 1
- [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 2
- [3] Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training rl-like reasoning large vision-language models. *arXiv preprint arXiv:2504.11468*, 2025. 6
- [4] Daniel Cores, Michael Dorkenwald, Manuel Mucientes, Cees GM Snoek, and Yuki M Asano. Lost in time: A new temporal benchmark for videollms. *arXiv preprint arXiv:2410.07752*, 2024. 1
- [5] Caba Heilbron Fabian, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–970, 2015. 1
- [6] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-rl: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025. 1, 3, 6
- [7] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017. 1
- [8] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. 1
- [9] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-rl: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025. 3, 7
- [10] Yunxin Li, Xinyu Chen, Zitao Li, Zhenyu Liu, Longyue Wang, Wenhan Luo, Baotian Hu, and Min Zhang. Veripo: Cultivating long reasoning in video-llms via verifier-guided iterative policy optimization. *arXiv preprint arXiv:2505.19000*, 2025. 6
- [11] Yun Li, Yiming Zhang, Tao Lin, XiangRui Liu, Wenxiao Cai, Zheng Liu, and Bo Zhao. Sti-bench: Are mllms ready for precise spatial-temporal world understanding? *arXiv preprint arXiv:2503.23765*, 2025. 1
- [12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1
- [13] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 4
- [14] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506, 2020. 1
- [15] Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025. 1
- [16] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. 4
- [17] Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. *arXiv preprint arXiv:2504.07934*, 2025. 1
- [18] Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-rl: Post-training large vision language model for temporal video grounding. *arXiv preprint arXiv:2503.13377*, 2025. 1
- [19] Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13204–13214, 2024. 1
- [20] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 2

- [21] Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gao-hong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. 1
- [22] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 4
- [23] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024. 4
- [24] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186, 2025. 4
- [25] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024. 1
- [26] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2024. 4
- [27] Kejian Zhu, Zhuoran Jin, Hongbang Yuan, Jiachun Li, Shangqing Tu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Mmr-v: What’s left unsaid? a benchmark for multimodal deep reasoning in videos. *arXiv preprint arXiv:2506.04141*, 2025. 1



What are the moves in the last scene of this dance?

- A. Kneel down on one knee and lean back.
- B. Passe and then chasse.
- C. Releve and then pirouette.
- D. Passe and then Grand jete.

GT: A

VideoChat-R1-Thinking (Direct-Answering) : A



VideoChat-R1-Thinking (CoT Reasoning) :

<think>

The video shows a ballet performance where the dancers are executing a series of movements. The last scene involves the dancers performing a sequence that includes a combination of steps and poses. The movements appear to be fluid and coordinated, typical of a ballet piece.

- At 0:10 to 1:50, the dancers perform a series of steps that involve passing their arms and legs in a synchronized manner.
- At 1:50 to 2:00, the dancers execute a grand jete, which is a leap where the dancer jumps into the air and extends their body horizontally before landing gracefully.

Given these observations, the correct answer should reflect the final movement performed by the dancers in the last scene.

</think>

<answer>D</answer>



There are two images here. One shows a girl in green clothing with braided hair, holding a clay container in front of a solid color background wall. The other shows a girl in black and white floral clothing with loose hair. According to the video, which character appears first?

- A. Boy with short hair and green stripes
- B. Boy with golden hair
- C. Girl in green clothing with loose hair
- D. Girl in green clothing with braided hair
- E. Girl in black and white floral clothing with loose hair

GT: D

VideoChat-R1-Thinking (Direct-Answering) : D



VideoChat-R1-Thinking (CoT Reasoning) :

<think>

The video starts with a title card for "metkids" and then transitions to a scene where hands are shown kneading clay on a surface. After this, there is a cut to a girl in green clothing with braided hair holding a clay container. This sequence suggests that the girl in green clothing with braided hair appears after the initial clay preparation scenes.

The question asks which character appears first. Since the video begins with the clay preparation and then cuts to the girl in green clothing with braided hair, it's clear she is not the first character to appear. Therefore, the correct answer should be related to the girl in black and white floral clothing with loose hair, as she is the only one mentioned before the girl in green clothing with braided hair.

</think>

<answer>E</answer>



Figure 4. **Failure case of VideoChat-R1.** The model hallucinates visual content in its reasoning trace, leading to an incorrect answer. Direct answering performs better in this perception-oriented task.

Absolute pressure & Gage pressure

**Definition:** Absolute pressure ( $P_{abs}$ ) is the pressure measured relative to a vacuum (0 Pa).  
 • More often used in thermodynamic calculations

**Definition:** Gage pressure ( $P_{gage}$ ) is the pressure measured relative to the ambient pressure ( $P_a$ ), which is usually the local atmospheric pressure.  
 • Easier to measure  $P_{gage}$  compared to  $P_{abs}$ .  
 • More often used in fluid mechanics calculations.

$P_{abs} = P_{gage} + P_a$

$P_{abs} = P_{gage} + P_a$

$P_{abs} = P_{gage} + P_a = 14.7 \text{ psi}$

$P_{abs} = P_{gage} + P_a = 32.9 \text{ psi} + 14.7 \text{ psi}$

$P_{abs} = P_{gage} + P_a = 46.7 \text{ psi}$  (absolute)

"psi" is often used to denote  $P_{abs}$ .  
 "psig" is sometimes used to denote  $P_{gage}$ .

This video was created with support from the California State University Course Redesign with Technology program.

Written by  
 Dr. Paul Nisenson, Dr. Angela Shi, Dr. Henry Kuo  
 Department of Mechanical Engineering  
 California State Polytechnic University, Pomona

Produced and narrated by  
 Dr. Paul Nisenson

Quiz

A wind tunnel draws in sea-level standard air from the room and accelerates it into a 1-m by 1-m test section. A pressure transducer in the test section wall measures  $\Delta p = 45 \text{ mm water}$  between inside and outside. Estimate the absolute pressure at the nose of the model.

Options:

- A. 112000 Pa      B. 101350 Pa      C. 98000 Pa      D. 99000 Pa      E. 95280 Pa  
 F. 110000 Pa      G. 107000 Pa      H. 94000 Pa      I. 105000 Pa      **GT: B**

**VideoChat-R1-Thinking (Direct-Answering) : E** ✘

**VideoChat-R1-Thinking (CoT Reasoning) :**

<think>

To estimate the absolute pressure at the nose of the model, we need to understand the relationship between gauge pressure, absolute pressure, and atmospheric pressure.

From the video, we know that:

- Absolute pressure  $P_{abs}$  is the pressure measured relative to a vacuum (0 Pa).
- Gauge pressure  $P_{gage}$  is the pressure measured relative to the ambient pressure  $P_a$ , which is usually the local atmospheric pressure.
- The formula for gauge pressure is:  $P_{gage} = P_{abs} - P_a$ .

Given:

We can use the conversion factor between mm water and psi to find the gauge pressure in psi:

1 mm water = 0.002316 psi

45 mm water =  $45 \times 0.002316 \approx 0.10422 \text{ psi}$

Now, using the formula  $P_{gage} = P_{abs} - P_a$ :

$P_{abs} = 0.10422 \text{ psi} + 14.7 \text{ psi} \approx 14.80422 \text{ psi}$

$0.10422 \text{ psi} = P_{abs} - 14.7 \text{ psi}$

Since 1 psi is approximately equal to 6894.757293 Pa, we convert the absolute pressure to pascals:

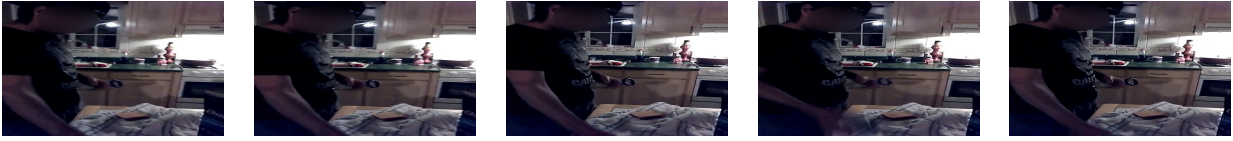
$P_{abs} \approx 14.80422 \times 6894.757293 \approx 102,000 \text{ Pa}$

The closest option to this value is 101350 Pa.

</think>

**<answer>B</answer>** ✔

Figure 5. Success Case of VideoChat-R1 on VideoMMMU. The question involves scientific analysis with equations. Chain-of-thought reasoning leads to the correct answer, whereas direct answering would likely fail.



Locate the start and end timestamps of the video segment corresponding to the description: *the person was pouring a bottle of beer*. Please provide the start and end timestamps (in seconds, precise to one decimal place) in the format `\boxed{[start, end]}`.

**VideoAuto-R1 : `\boxed{[3.2, 10.8]}`**

`<think>`At 3.2 seconds, the person is seen holding a bottle and tilting it towards their mouth, which suggests they are pouring something into their mouth. At 10.8 seconds, they finish pouring and put the bottle down. This action aligns with the described behavior of pouring a bottle of beer.`</think>`

`\boxed{[3.2, 10.8]}`

GT: [1.1, 9.4]



Locate the start and end timestamps of the video segment corresponding to the description: *Another man is shown being pulled back onto the raft*. Please provide the start and end timestamps (in seconds, precise to one decimal place) in the format `\boxed{[start, end]}`.

**VideoAuto-R1 : `\boxed{[42.5, 58.3]}`**

`<think>`At 42.5 seconds, we see another individual who has fallen into the river. The rafters work together to pull him back onto the raft, ensuring everyone's safety. This action demonstrates teamwork and coordination among the group.`</think>`

`\boxed{[42.5, 58.3]}`

GT: [70, 73]



Question: *what did the person do after the turtle finished eating the leaf?*

Options: A. drink B. look to his left C. smiled D. look backwards E. walk away

Please answer the question and then provide the video segment that best supports your answer. The start and end timestamps must be in seconds with exactly one decimal place.

Output format: `\boxed{answer <> [start, end]}`. For example: `\boxed{A <> [20.3, 30.8]}`

**VideoAuto-R1 : `\boxed{E <> [31.5, 46.9]}`**

`<think>`At 31.5 seconds, the person is seen walking away from the turtle, which indicates they have finished interacting with it. This action aligns with the option E, "walk away." The other options do not fit the observed behavior in the video.`</think>`

`\boxed{E <> [31.5, 46.9]}`

GT: E <> [34.3, 45.8]

Figure 6. **VideoAuto-R1 on Temporal Grounding Tasks.** The reasoning trace is simple and redundant with the initial answer, enabling effective early-exit without full CoT reasoning.



Question: What is the action performed by the person in the video?

Options:

(A) boiling (B) putting (C) stirring (D) cooking

Put your final answer in `\boxed{}`.

GT: C

**VideoAuto-R1** : `\boxed{C}` (confidence 0.99, early-exit)

<think>

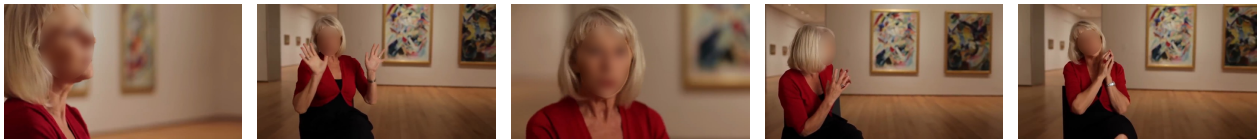
The person in the video is actively engaged in cooking, specifically stirring food in a pan with a spatula. This action aligns with the option C, stirring.

The other options do not fit the observed activity:

- Boiling (A) involves heating water to a boil, which is not what is happening here.
- Putting (B) implies placing something into the pan, but the person is not doing that; they are stirring.
- Cooking (D) is too broad and includes all cooking activities, including stirring, so it is not the specific action being performed here.

</think>

`\boxed{C}`



A woman is sitting inside a gallery. She is wearing a red coat and black clothes. Her hair is blond, and she has a watch on her wrist. There are two paintings hanging on the wall behind her. When she mentions 'his work is incredibly worked out and um,' what change occurs to the woman onscreen?

Options:

- A. The woman walks to admire four artworks.      B. The woman starts introducing the artworks.  
 C. The woman changed her clothes.                      D. The woman sits down to admire four artworks.  
 E. The woman's hair was tied up.

Put your final answer in `\boxed{}`.

GT: A

**VideoAuto-R1** : `\boxed{A}` (confidence 0.99, early-exit)

<think>

The woman is initially seen sitting, but as she mentions \"his work is incredibly worked out and um,\" she stands up and walks towards the wall where four artworks are displayed. This action aligns with the option A, which states that she walks to admire four artworks.

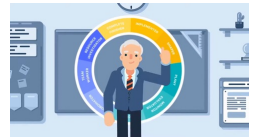
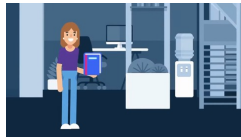
The other options do not fit the sequence of events shown in the video. The woman does not change her clothes, nor does she sit down to admire the artworks after standing up. The mention of her hair being tied up is not relevant to the actions taken in the video.

</think>

`\boxed{A}`



Figure 7. **VideoAuto-R1 on Perception-Oriented QA Tasks.** High-confidence initial answers trigger early exit, improving inference efficiency.



A team member consistently generates innovative ideas but struggles to follow through with implementation and often overlooks practical details. According to Belbin's model, which role most likely describes this individual, and what is a potential weakness they might exhibit?

Options:

- A. Implementer; Inflexible and close-minded
- B. Plant; Might ignore incidentals and might be too pre-occupied to communicate properly
- C. Shaper; Easily provoked and may offend others
- D. Completer-Finisher; Excessively worried and reluctant to delegate work
- E. Coordinator; Seen as manipulative and work-shy
- F. Team Worker; Indecisive and tends to avoid confrontation
- G. Resource Investigator; Overly optimistic and can lose interest quickly
- H. Monitor Evaluator; Lacks drive and ability to inspire, overly critical
- I. Specialist; Contributes only on a narrow front and dwells on technicalities

Put your final answer in `\boxed{}`.

GT: B

**VideoAuto-R1 : `\boxed{B}` (confidence 0.94, continue CoT reasoning)**

`<think>`

The individual who consistently generates innovative ideas but struggles to follow through with implementation and often overlooks practical details aligns with the characteristics of a "Plant" role. The Plant role is described as innovative, creative, problem-solving, and free-thinking, which matches the description of generating ideas. However, the potential weakness mentioned for the Plant role is that they might ignore incidentals and might be too pre-occupied to communicate properly, which fits the scenario of overlooking practical details and possibly not communicating effectively.

Therefore, the correct answer is B. Plant; Might ignore incidentals and might be too pre-occupied to communicate properly.

`</think>`

`\boxed{B}`



Figure 8. **VideoAuto-R1 on Reasoning-Oriented QA Tasks.** The reasoning trace is longer and more detailed, with clear step-by-step deductions.



The data that follow are the square footage (in 1,000 feet squared) of 28 homes. The sample mean = 2.50 and the sample standard deviation = 0.8302. The distribution can be written as  $X \sim U(1.5, 4.5)$ . What is  $P(x < 3.5 | x < 4)$ ?

Options:

- A. 0.875      B. 0.55      C. 0.8      D. 0.75      E. 0.7  
 F. 0.625      G. 0.95      H. 0.9      I. 0.5      J. 0.85

Put your final answer in  $\boxed{\}$ .

GT: C

**VideoAuto-R1 :  $\boxed{D}$**  (confidence 0.92, continue CoT reasoning)

<think>

To find  $P(x < 3.5 | x < 4)$ , we need to use the concept of conditional probability. The formula for conditional probability is:  $P(A | B) = \frac{P(A \cap B)}{P(B)}$ .

In this context, let  $A$  be the event  $x < 3.5$  and  $B$  be the event  $x < 4$ . Since  $A$  is a subset of  $B$ ,  $P(A \cap B) = P(A)$ .

First, we need to find  $P(x < 3.5)$  and  $P(x < 4)$ . The distribution is uniform from 1.5 to 4.5, so the probability density function  $f(x)$  is:

$$f(x) = \frac{1}{4.5 - 1.5} = \frac{1}{3}$$

The cumulative distribution function  $F(x)$  is:

$$F(x) = \int_{1.5}^x \frac{1}{3} dt = \frac{x - 1.5}{3}$$

Now, we calculate  $P(x < 3.5)$  :

$$P(x < 3.5) = F(3.5) = \frac{3.5 - 1.5}{3} = \frac{2}{3},$$

Next, we calculate  $P(x < 4)$  :

$$P(x < 4) = F(4) = \frac{4 - 1.5}{3} = \frac{2.5}{3}.$$

Now, we can find the conditional probability:

$$P(x < 3.5 | x < 4) = \frac{P(x < 3.5)}{P(x < 4)} = \frac{\frac{2}{3}}{\frac{2.5}{3}} = \frac{2}{2.5} = \frac{4}{5} = 0.8.$$

Therefore, the correct answer is: C. 0.8.

</think>

$\boxed{C}$



Figure 9. **VideoAuto-R1 Performing Complex Math Reasoning.** The model applies probability and integration, revising an incorrect initial answer to the correct one through structured reasoning.