

Video-R4: Reinforcing Text-Rich Video Reasoning with Visual Rumination

Supplementary Material

6. Limitations

Despite these results, Video-R4 still has several limitations. First, the data curation pipeline relies on pre-extracted OCR results and object detections, so recognition errors or missing text can directly hurt both rumination trajectories and final answers. Second, the current tool interface supports only frame selection and spatial cropping with a bounded trajectory length, which may be insufficient for very long or fast-changing videos that require richer operations (e.g., tracking, retiming, or audio-aware cues). Third, our training data are primarily derived from M4-ViteVQA and a few related text-centric datasets, and experiments are conducted on a 7B backbone, leaving open questions about robustness under more diverse domains and larger model scales. Finally, the GRPO reward combines hand-designed proxies such as diversity, representativeness, and curiosity, which only approximate human notions of faithfulness and interpretability. Future work could relax these assumptions by broader operation types, more diverse optimization methods, and rewards.

7. Dataset Details

Dataset Statistics. Figure 5 presents the overall statistics of Video-R4-CoT-17k. The dataset is predominantly video-based, with images forming a smaller subset. The word cloud highlights frequent reasoning-related expressions such as “visual”, “information”, and various operation-oriented verbs. The question length distribution centers on medium-length prompts, while the plots of visual operation counts and conversation turns show that CoT trajectories typically require several visual operations and involve multi-round interactions. Figure 6 summarizes the statistics of Video-R4-RL-30k. The corresponding word cloud shows a more object-focused vocabulary (e.g., “object”, “person”, “left”, “color”), consistent with the concise, direct style characteristic of RL-refined queries.

Rule-Based Evidence Matching. The Rule-Based Evidence Matching algorithm is shown in Algorithm 1. For each training instance q , we denote by q^{text} the question text and by q^{ans} the answer expression, which may be a single string or a small set of candidates; the associated video is $v(q)$. Each instance carries two supervisory attributes: the temporal specification $\text{src}_1(q) \in \{\text{Single frame}, \text{Multi frame}\}$, indicating whether evidence is restricted to one frame or may span multiple frames, and the modality specification $\text{src}_2(q) \in \{\text{Text}, \text{Visual}\}$, indicating whether evidence is primarily textual (OCR) or visual (objects). We write

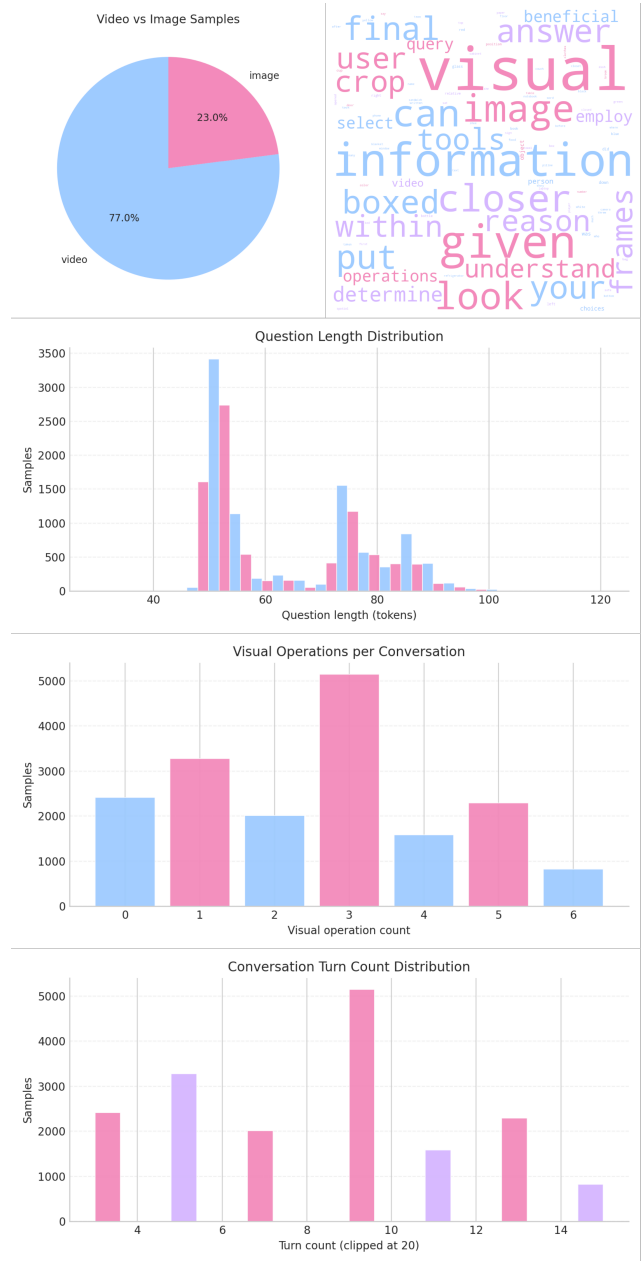


Figure 5. Overall statistics of the Video-R4-CoT-17k dataset, including the ratio of video versus image samples, word cloud of frequently appearing terms, question length distribution, distribution of visual operation counts per sample, and conversation turn count distribution.

$A_q = \text{tok}(q^{\text{ans}})$ and $W_q = \text{tok}(q^{\text{text}})$ for the normalized token sets of the answer and question, respectively. For a video v , let F_v be the set of candidate frames con-

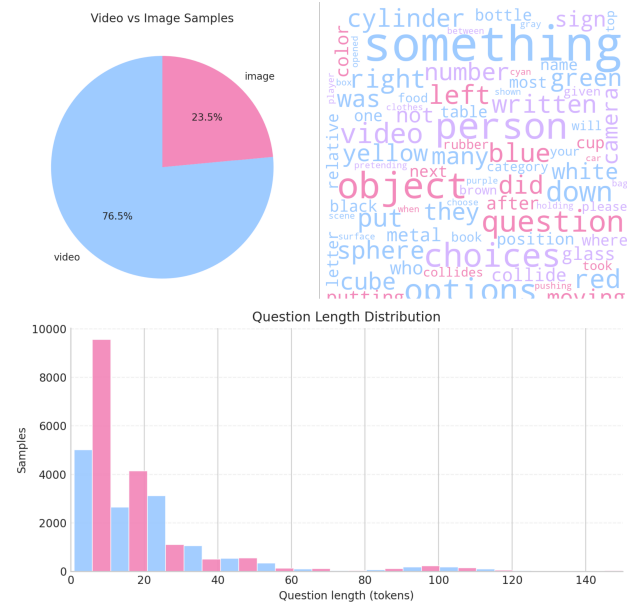


Figure 6. Overall statistics of the Video-R4-RL-30k dataset.

considered during evidence mining. For every frame $f \in F_v$, we assume paragraph-level OCR regions $\mathcal{P}_{v,f}$, fine-grained OCR text detections $\mathcal{T}_{v,f} = \{(s_{v,f,i}, b_{v,f,i}^{\text{text}})\}_i$ with strings and corresponding boxes, and object detections $\mathcal{O}_{v,f} = \{(\ell_{v,f,k}, b_{v,f,k}^{\text{obj}})\}_k$ with discrete labels and boxes. Matching and geometry are treated abstractly via the primitives $\text{score}_{\text{text}}(s, A) \in [0, 1]$ for text–answer relevance, $\text{score}_{\text{name}}(n, U) \in [0, 1]$ for name–token compatibility, $\text{iou}(b_1, b_2)$ for box overlap, $\text{extend}(b)$ for deterministic enlargement, and $\text{merge}(\mathcal{B})$ for minimal axis-aligned merging of a box set \mathcal{B} . The goal is, for each question q , to return a subset $\mathcal{R}_q \subseteq F_{v(q)}$ of relevant frames and a per-frame evidence region $B_{q,f}^{\text{ev}}$ obtained by combining textual and, when applicable, object cues. We denote by $b_{q,f}^{\text{text}}$ the best OCR-derived box selected in frame f for question q before paragraph refinement.

Template-Based Context Synthesis. We construct a set of multi-turn dialogues for each annotation through predefined templates. Each dialogue contains a system message, a question, the path to the input video, and a sequence of turns. The turns follow a chain-of-thought format. Each turn provides an analysis of the visual information obtained through the visual operation applied in the previous step. The first turn, instead, provides an overall description of the input video. The turn then continues with a brief reasoning segment that connects to the next action and ends with a statement describing the next visual operation, where the format of the visual operation follows [51], with “<tool_call>” labels to prompt the visual operations, and tool names and parameters are needed for a single function call. For clip-

ping, the parameters are the indices of the selected frames. For cropping, the parameters include a frame index and the bounding box coordinates. The final turn predicts the answer to the question with “{ }” format. The template produces a dialogue that contains several placeholders. These placeholders include the input video caption and a descriptive analysis of the visual observations obtained in each turn. They will be filled in during the following stages.

LMM-Based CoT Synthesis and Refinement. We use Qwen2.5-VL [1] to generate video captions, clip captions, and region captions. The input includes the original video frame sequence for the video captions and the text prompts. For clip and region captions, the original video frame sequence, the clips/regions in the current turn, and the text in the context serve as input. The think processes are then generated, focusing on whether the current visual cues obtained can answer the question sufficiently. Then we replace all the placeholders in the templates to get the multi-turn CoT trajectories. We use GPT-4o [25] to further refine the trajectories to make them more coherent, natural, and reasonable.

Quality Control Tool. We develop a quality control tool to quickly review all the QA queries and the corresponding synthesized trajectories. As shown in Figure 8, the tool supports quick browsing, sample saving, dropping functions, and a fixing mode, where human annotators can directly revise the content of the chain-of-thought trajectories, including both text and visual cues.

OCR Robustness Analysis. Because our data curation relies on pre-extracted OCR results and object detections, we analyze how recognition errors affect the quality of rumination trajectories. We conduct three analyses: (1) We sample 500 trajectories and manually verify OCR quality: 87% have accurate text recognition for key evidence regions, indicating that the filtering pipeline effectively removes low-quality cases. (2) We examine the distribution of trajectory types: 62% are text-only, 18% object-only, and 20% mixed, showing diversity beyond OCR-dependent samples. (3) We randomly drop 30% of text bounding boxes from the evidence pool and retrain; performance degrades by only 4.2% in accuracy, as Video-R4 can cross-verify evidence across multiple frames instead of relying on a single noisy observation. These results suggest that the benefit of rumination is robust at inference time even without high-quality OCR and stable bounding boxes. Moreover, the RL training set includes a substantial portion of unmatched samples with varying difficulty levels, encouraging the policy to explore beyond high-quality OCR patterns. The zero-shot generalization to MP-DocVQA and SlidesVQA (see main paper) further validates that Video-R4 does not overfit to specific

Algorithm 1 Rule-Based Evidence Matching

```

1: for each question  $q$  do
2:   initialize  $\mathcal{R}_q \leftarrow \emptyset$ 
3:   for each frame  $f \in F_{v(q)}$  do
4:     find best OCR match  $b_{q,f}^{\text{text}}$  using  $\text{score}_{\text{text}}(\cdot, A_q)$ 
5:     if a match exists then  $\mathcal{R}_q \leftarrow \mathcal{R}_q \cup \{f\}$ 
6:     end if
7:   end for
8:   for each  $f \in \mathcal{R}_q$  do
9:     refine  $b_{q,f}^{\text{text}}$  by selecting  $p^* \in \mathcal{P}_{v,f}$  with maximal iou, then set  $b_{q,f}^{\text{text}} \leftarrow \text{extend}(p^*)$ 
10:  end for
11:  if  $\text{src}_2(q) = \text{Text}$  then
12:    choose single or multiple frames according to  $\text{src}_1(q)$ 
13:    output refined text boxes  $\{b_{q,f}^{\text{text}}\}$ 
14:    continue
15:  end if
16:  for each  $f \in \mathcal{R}_q$  do
17:    collect object boxes whose names match  $A_q \cup W_q$  via  $\text{score}_{\text{name}}$ 
18:    merge all matched boxes with  $b_{q,f}^{\text{text}}$ :  $B_{q,f}^{\text{ev}} \leftarrow \text{merge}(\cdot)$ 
19:  end for
20:  select single or multiple frames according to  $\text{src}_1(q)$ 
21:  output  $\mathcal{R}_q$  and  $\{B_{q,f}^{\text{ev}}\}$ 
22: end for

```

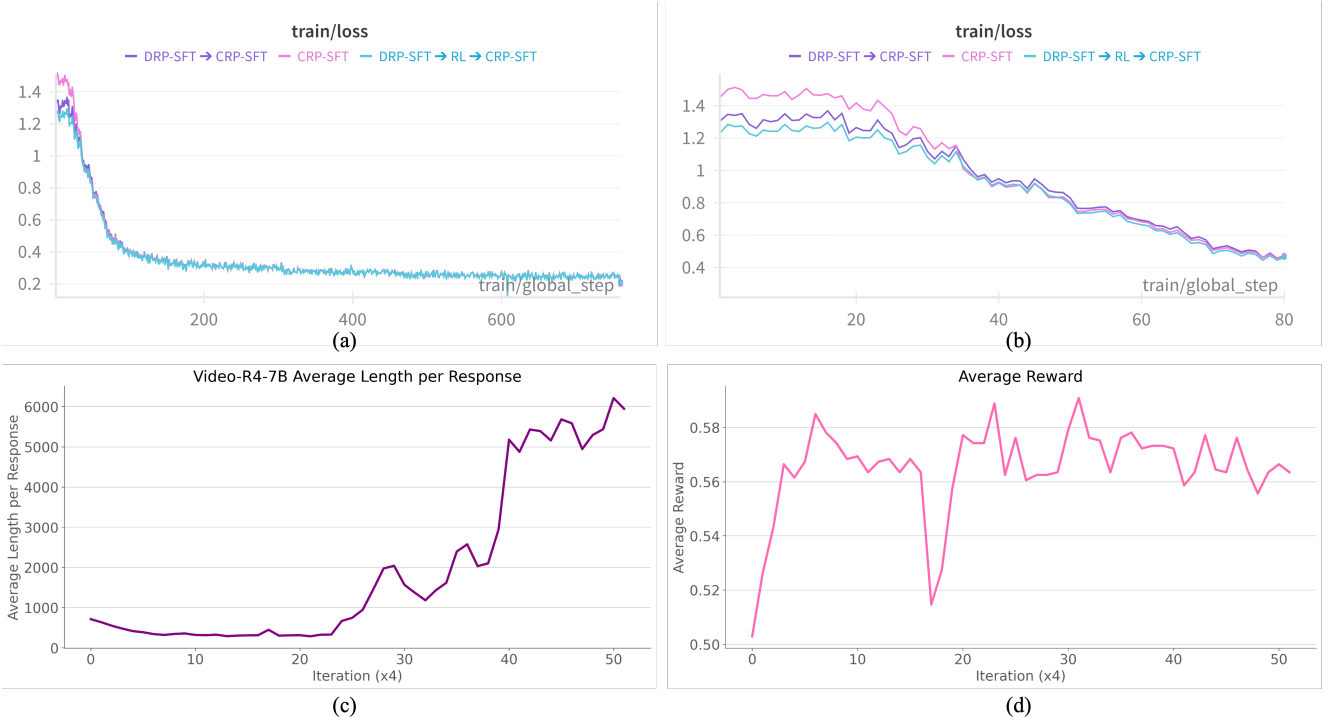


Figure 7. Comparison of training behaviors across fine-tuning strategies. Subfigures (a) and (b) show that models pre-finetuned on DRP-SFT data converge more quickly and achieve lower final loss when training on CRP-SFT, indicating that decomposing visual operations before interleaved training is beneficial. (c) Video-R4-7B progressively increases its response length during RL, suggesting emergent allocation of more thinking time. (d) Correspondingly, the average reward improves and remains stable across iterations.

OCR patterns but learns general visual reasoning capabilities.

Human Trajectory Validation. To verify whether our automatic pipeline produces human-like rumination patterns, we manually annotate a pilot set of 200 trajectories directly from raw videos without OCR guidance: annotators perform the task themselves and record their own frame selection and cropping operations. Preliminary analysis shows: (1) 83% agreement between human-annotated and our synthetic trajectories in terms of selected frames, and (2) human annotators use slightly fewer cropping operations, suggesting our trajectories are conservative but well-aligned with human strategies. These trajectories are further quality-checked via the annotation interface described above (Figure 8).

8. Evaluation Metrics

Average Normalized Levenshtein Similarity (ANLS). Exact-match metrics are brittle for text-centric VQA because minor OCR errors can flip a correct rationale into an incorrect string. Therefore, [7] proposed ANLS, which turns the normalized Levenshtein distance [30] between a prediction and reference into a similarity score with a cutoff. Let o_{q_i} be the model’s answer for question q_i , and $\{a_{ij}\}_{j=1}^M$ the set of M ground-truth strings. Denote by $NL(\cdot, \cdot) \in [0, 1]$ the normalized Levenshtein distance. With a threshold $\tau = 0.5$, the per-pair similarity is

$$s(a_{ij}, o_{q_i}) = \begin{cases} 1 - NL(a_{ij}, o_{q_i}) & \text{if } NL(a_{ij}, o_{q_i}) < \tau, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Then take the best match across references for each question and average over N questions:

$$\text{ANLS} = \frac{1}{N} \sum_{i=1}^N \left(\max_j s(a_{ij}, o_{q_i}) \right). \quad (9)$$

Predictions with edit distance $\geq \tau$ (over half the characters wrong) receive zero credit, while smaller deviations are rewarded proportionally. This softly penalizes OCR noise while still emphasizing exactness.

Exact Match (EM). This metric quantifies the proportion of predictions that exactly coincide with any of the ground truth answers, thereby providing a strict measure of answer correctness [47].

(Macro-averaged) F1 score. This metric assesses the token-level overlap between a prediction and the ground truth answer by treating both as bags of tokens and computing their F1 score. For each question, the highest F1 score

across all ground truth answers is selected, and the final metric is obtained by averaging these maxima over the full set of questions [47].

9. More Evidence to Support our Findings

Figure 7 (c) demonstrates Video-R4-7B naturally learns to solve reasoning tasks with more thinking time, which is evidence to support our Finding 1. Figure 7 (a) and (b) show the training curve of CRP-SFT under different settings, demonstrating that the models pre-finetuned on DRP-SFT data have a faster convergence. Even though losses converge during fine-tuning across different settings, the model fine-tuned on DRP-SFT data achieves better final performance on the benchmarks. This shows that it is helpful to learn different types of visual operations separately before interleaving them during training. As shown in Table 4, results on the validation set of M4-ViteVQA are also reported, demonstrating that Video-R4-7B establishes a new state-of-the-art in text-rich video understanding and reasoning.

10. Training Details

For DRP-SFT, we use 7k data from Video-R4-CoT-17k for fine-tuning. The learning rate of 1×10^{-6} is adopted. We fully fine-tune the model instead of using LoRA [16]. For the RL after DRP-SFT, we use accuracy and the curiosity reward. There are 15k samples from Video-R4-RL-30k used during the stage. Following [51], the curiosity reward’s hyperparameters are set as follows: $\alpha = 0.5$, $\beta = 0.05$, and $H = 0.3$. GRPO [15] is adopted as the policy optimization method. Eight responses are sampled for each sample. For the CRP-SFT, 10k samples from Video-R4-CoT-10k are used, and other hyperparameters are the same as those in DRP-SFT. For the RL after CRP-SFT, we adopt accuracy, diversity, representativeness, and curiosity reward, with the coefficients $\lambda_{\text{div}} = \lambda_{\text{rep}} = \lambda_{\text{cur}} = 1$ [84]. All the models are trained on one H100 80G GPU.

11. Inference Cost Analysis

We report the average inference time and the average number of rumination steps on 2,112 videos from the M4-ViteVQA Task 2 testset using 8 H100 80G GPUs. Table 5 shows that inference latency grows moderately with the rumination budget: from no rumination to budget=8 (averaging 5.2 rumination steps), inference time increases by 37% while accuracy improves by 15.5 points, demonstrating a favorable computation–accuracy trade-off. Furthermore, Figure 7(c) shows that the model progressively increases its response length during RL training, consistent with the test-time scaling phenomenon observed in the main paper.

Table 4. Performance comparison on the M4-ViteVQA validation set and testset.

Models	Task 1 - Split 1				Task 1 - Split 2				Task 2			
	Val		Test		Val		Test		Val		Test	
	Acc.(%)	ANLS(%)	Acc.(%)	ANLS(%)	Acc.(%)	ANLS(%)	Acc.(%)	ANLS(%)	Acc.(%)	ANLS(%)	Acc.(%)	ANLS(%)
JustAsk [67]	10.81	15.40	10.05	14.10	7.16	10.00	5.47	8.60	4.86	6.70	3.60	6.70
All-in-one-B [62]	11.47	15.30	10.87	14.80	6.85	9.20	5.66	7.80	4.20	5.00	3.28	4.60
Video-LLaVA-7B [37]	15.82	17.77	15.43	17.15	13.14	14.29	11.19	12.02	10.89	13.23	9.38	11.80
T5-ViteVQA [81]	23.17	30.10	22.17	29.10	17.59	23.10	16.68	23.80	12.30	16.10	9.29	13.60
VideoLLaMA2-7B [10]	20.04	21.73	20.76	23.55	18.30	19.63	18.33	20.45	19.68	23.62	16.54	21.80
Qwen2-VL-7B [63]	36.77	46.56	35.22	45.84	28.55	39.34	27.25	38.45	22.95	32.65	21.23	28.79
TEA-L [77]	37.49	46.38	34.78	43.71	28.27	36.32	28.43	38.13	22.83	30.21	18.83	28.90
NVILA-8B [44]	37.89	47.67	37.73	47.23	30.25	40.58	30.10	41.52	23.79	32.89	22.89	30.34
GAT-L [78]	38.01	47.53	38.30	48.23	31.35	41.33	30.90	41.81	24.54	33.30	22.13	30.75
Qwen2.5-VL [1]	22.22	48.67	26.53	44.91	17.84	46.72	24.34	39.60	22.31	42.21	32.81	50.82
Video-R1-7B [13]	38.10	50.80	37.10	48.25	38.40	49.62	33.67	44.94	47.77	58.52	43.16	53.37
Pixel-Reasoner [51]	54.44	63.57	52.91	61.44	54.69	62.58	48.88	58.23	63.78	69.93	58.97	65.32
Video-R4-7B (ours)	57.33	66.92	56.17	65.22	57.65	65.15	52.69	61.89	69.03	75.45	64.21	69.99
Human	-	-	85.27	89.30	-	-	78.41	82.80	-	-	82.26	85.10

Table 5. Inference cost analysis on M4-ViteVQA Task 2 testset (2,112 videos). Latency is measured per video on 8 H100 GPUs.

Method	Infer Time (ms)	Avg #Steps	Task2 Acc (%)
Qwen2.5-VL (baseline)	577.2	-	32.8
Video-R4 (no rumination)	658.8	0	48.7
Video-R4 (budget=1)	775.5	1.0	54.3
Video-R4 (budget=2)	815.2	1.7	58.7
Video-R4 (budget=4)	871.7	3.1	62.8
Video-R4 (budget=8)	905.4	5.2	64.2

12. Failure Case Analysis

Despite strong overall performance, we identify several common failure modes of Video-R4:

- Imprecise bounding boxes.** The model occasionally predicts bounding boxes that crop out part of the target text, causing downstream reasoning to miss key evidence. Because the cropped region is re-encoded and directly conditions subsequent steps, even small spatial errors can propagate through the rumination chain.
- Attention dispersion in dense text regions.** When a frame contains many visually similar text regions (e.g., spreadsheets, densely annotated UI screens), the model may attend to irrelevant but nearby text, diluting the evidence collected for the actual query.
- Unrecoverable low resolution.** If the target text is too small or blurred in every available frame, rumination cannot recover the missing information regardless of how many zoom-and-read cycles are performed. In such cases, the model tends to hallucinate plausible but incorrect text.
- Temporal text changes.** The model sometimes selects frames in which the target text has already changed or disappeared (e.g., a slide transition or a scrolling UI), leading to stale evidence that contradicts the correct answer.

Potential mitigation strategies include incorporating confidence-aware re-cropping to detect and correct spatial alignment errors, leveraging temporal consistency checks


across selected frames to filter out stale evidence, and training with adversarial or hard-negative samples that expose the model to these challenging scenarios during RL.

13. More Visualization Results

As shown in Figures 9 and 10, we present additional visualizations of the trajectory samples.


Sample 2 / 10079 | QID: 1 | VID: 02766 | Status: none | Jump to QID: | Go: [question_id]

user
What is written on the left white sign?




assistant
This video shows a busy urban intersection with various vehicles, including taxis and SUVs, navigating through traffic. Pedestrians are visible on the sidewalks, and the scene is set against a backdrop of tall buildings. Let me begin with a careful review of the frames. <tool_call={"name": "clipping", "arguments": {"target_frames": [2, 3, 4]}}></tool_call>

user
Here are the selected frames:




assistant
The frames show a busy urban intersection with vehicles and a white sign that reads "PHOENIX" and "park open 24 hours," which is somewhat helpful for answering the question. Let me see a few more frames. <tool_call={"name": "clipping", "arguments": {"target_frames": [7, 8, 9]}}></tool_call>

user
Here are the selected frames:




assistant
The frames show a busy urban intersection with vehicles and pedestrians, and the white sign reads "park" with arrows, which is somewhat helpful for answering the question. Let me see a few more frames. <tool_call={"name": "clipping", "arguments": {"target_frames": [0, 4, 5]}}></tool_call>

user
Here are the selected frames:




assistant
The visuals show a busy urban intersection with vehicles and a white sign that reads "park open 24 hours," which is somewhat helpful for answering the question. Still unclear—I'll inspect a region on a different frame. <tool_call={"name": "cropping", "arguments": {"target_image": 3, "bbox_2d": [0.177, 0.228, 0.298, 0.346]}}></tool_call>

user
Here is the cropped region:




assistant
The region shows a traffic light with a green arrow and a white sign below it that reads "WEST END AVE," which is somewhat helpful for answering the question. The crop seems off—I'll retarget in this frame. <tool_call={"name": "cropping", "arguments": {"target_image": 5, "bbox_2d": [0.03, 0.01, 0.466, 0.424]}}></tool_call>

user
Here is the cropped region:



assistant
The region shows a building with a white sign that reads "park open 24 hours" and "PHOENIX," which is somewhat helpful for answering the question. Let me tighten the crop on this area. <tool_call={"name": "cropping", "arguments": {"target_image": 5, "bbox_2d": [0.177, 0.228, 0.298, 0.346]}}></tool_call>

user
Here is the cropped region:



assistant
The white sign in the region reads "park open 24 hours," confirming the answer. The word "park" is prominently displayed on the left side of the sign. This matches the description provided in the earlier notes. Based on the inspected details, the answer is 'boxed(park)'.

Last Next Save Drop Too Long Needs Fix Cancel Edit Undo [status: none] Preview mode: review draft. Click Save again to finalize FIX, or Cancel Edit

Figure 8. Interface of the quality control tool used to review QA queries and synthesized chain-of-thought trajectories. The tool enables rapid browsing, frame inspection, saving or dropping samples, and in-place editing of both textual and visual reasoning steps to streamline annotation and correction workflows.



Figure 9. Trajectories visualization.

User What is written in the second line of the ppt text?



Video-R4 This video shows a laptop displaying a presentation on diabetes and insulin glargine. The presentation covers topics such as mechanism of action, pharmacokinetics, efficacy, and safety. Let me begin with a careful review of the frames. <tool_call>{"name": "clipping", "arguments": {"target_frames": [8, 9, 10]}}</tool_call>

System Here are the selected frames:



Video-R4 The visuals show a laptop with a presentation, but the text on the screen is not clearly visible, making it hardly helpful for identifying the second line of the PPT text. Let me check a span of frames. <tool_call>{"name": "clipping", "arguments": {"target_frames": [1, 2, 3]}}</tool_call>

System Here are the selected frames:



Video-R4 The visuals show a laptop with a presentation slide titled "Contents" listing topics, and the second line reads "Mechanism of action," which is highly helpful for answering the question. Signal is weak; refining the crop here. <tool_call>{"name": "cropping", "arguments": {"target_image": 2, "bbox_2d": [0.157, 0.445, 0.592, 0.884]}}</tool_call>

System Here is the cropped region:



Video-R4 The region shows a laptop with a presentation on the screen, but the text is not clearly visible, making it hardly helpful for determining the second line of the PPT text. Let me open up the region a bit here. <tool_call>{"name": "cropping", "arguments": {"target_image": 2, "bbox_2d": [0.137, 0.163, 0.791, 0.904]}}</tool_call>

System Here is the cropped region:




Video-R4 The region shows a laptop with a presentation slide listing topics, including "Mechanism of action" as the second line, which is highly helpful for answering the question. Let me tighten the crop on this area. <tool_call>{"name": "cropping", "arguments": {"target_image": 2, "bbox_2d": [0.611, 0.183, 0.771, 0.37]}}</tool_call>

System Here is the cropped region:



Video-R4 The region shows a presentation slide listing topics, with "Mechanism of action" as the second line. This directly supports that conclusion by clearly displaying the text in the specified position. Based on the inspected details, the answer is `\boxed{Mechanism of action}`.

User What is the point of Medvedev in the first game?



Video-R4 This video shows a tennis match with a player celebrating a point, followed by a commentator speaking. It then transitions to a practice session on a clay court. I'll first look more closely at the sequence. <tool_call>{"name": "clipping", "arguments": {"target_frames": [4, 5, 6, 7, 8]}}</tool_call>

System Here are the selected frames:



Video-R4 The visuals show a blurred person speaking into a microphone with text "NADAL 6 4" and "MEDVEDEV 2 6" visible, followed by a tennis practice session on a clay court; this is hardly helpful for determining the point of Medvedev in the first game. Let's review some nearby frames. <tool_call>{"name": "clipping", "arguments": {"target_frames": [5, 6, 7]}}</tool_call>

System Here are the selected frames:



Video-R4 The visuals show a tennis practice session on a clay court with two players, and there is no legible text present, making it hardly helpful for determining Medvedev's point in the first game. I'll explore a consecutive window of frames. <tool_call>{"name": "clipping", "arguments": {"target_frames": [3, 4, 5]}}</tool_call>

System Here are the selected frames:



Video-R4 The visuals show a tennis match with a scoreboard displaying "NADAL 2 6 6" and "MEDVEDEV 6 7 4," followed by a commentator and a practice session on a clay court; this is hardly helpful for determining Medvedev's point in the first game. No luck there; switching frame for another crop. <tool_call>{"name": "cropping", "arguments": {"target_image": 4, "bbox_2d": [0.046, 0.88, 0.284, 1.0]}}</tool_call>

System Here is the cropped region:



Video-R4 The region shows a hand holding a phone displaying a playlist titled "soft christmas songs," with text describing it as "not boppy christmas music; less jovial trumpets more crackling fireplaces," which is highly helpful for identifying the type of songs being played. Let me focus on a tighter area here. <tool_call>{"name": "cropping", "arguments": {"target_image": 9, "bbox_2d": [0.354, 0.225, 0.67, 0.393]}}</tool_call>

System Here is the cropped region:



Video-R4 The scoreboard shows "MEDVEDEV 6" for the first set, indicating that Medvedev won the first game with 6 points. Putting the evidence together, the answer is `\boxed{6}`.

Figure 10. More visualization with longer trajectories.