APPO: Attention-guided Perception Policy Optimization for Video Reasoning

Supplementary Material

Contents

A. Details of APPO

Our APPO algorithm primarily consists of two core steps: Attention-guided Frame Selection and Intra-group Perception Tokens Re-weighting, as mentioned in Section 3.2 of main submission. Therefore, in this section, we mainly discuss rationality and correctness of these two steps in detail.

A.1. Attention-guided Frame Selection

To understand the limitations of MLLMs' visual perception, recent works [5] have studied the attention patterns of MLLMs when answering visual questions. Inspired by these works, we also explored the attention patterns in video scenarios. The Fig. A and Fig. C present the attention patterns of response tokens to the video frames, respectively. It can be found that: (1) For the same video, the model exhibits different patterns of attention weights on video frames as the question changes; (2) The model's incorrect answers are due to either missing crucial frames or assigning too low attention weights to these frames. In particular, as shown in Fig. A, the question is "What is the mouse writing on the blackboard?", and the answer mainly appears in the last part of the video. The model correctly focused on the last few frames, especially when it provided correct information such as "very nice". Similarly, in Fig. C, the question is "What is the kitten doing when the blue cat turns its head for the second time?", and the answer mainly appears in the middle part of the video ($16s \sim 21s$). Clearly,

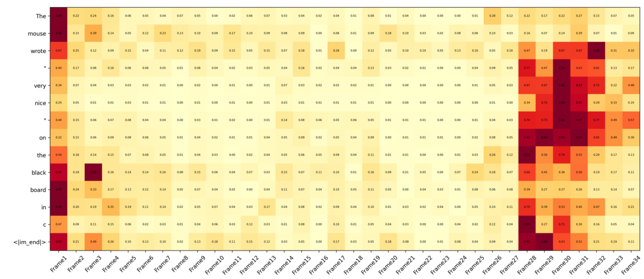


Figure A. The attention weights of response tokens to video frames. Question: "What did the mouse write on the blackboard?". The answer mainly appears in the last part of the video. The x-axis represents the video frame index (total 34 frames). The y-axis shows the response tokens. The darker the color, the higher the attention score.

the model did not pay sufficient attention to these video frames, resulting in an incorrect answer.

These empirical observations inspire us that the reasoning paths with higher rewards are more likely to focus on crucial frames. Therefore, as mentioned in Section 3.2 of main submission, we can divide all reasoning paths into two sets based on the reward scores, and then utilize the attention differences between these two sets to obtain frame-level guiding signals.

A.2. Intra-group Perception Tokens Re-weighting

The main goal of this step is to determine different learning intensities for intra-group perception tokens. Recent works [1, 4] demonstrate that the key reasoning tokens can be identified based on token-level distributional differences. Building upon this conclusion, Jisheng *et al.* [2] calculated the weights of tokens at each position for the entire sequence. Similarly, we argue that the intra-group perception tokens for each crucial frame could be treated as crucial fine-grained perception tokens, and they should be given different learning intensities based on information differences among them.

In particular, for each crucial frame $\mathcal{I}$ in ψ', each path contains certain perception tokens that focus primarily on this frame. The importance weights of this group of tokens at each position can be computed as follows:

$$D^{(k)} = \sum_{i=1}^{G} D_{\mathrm{KL}}\left(p(\Omega_{i,j}^{(k)}) \middle\| \mathbb{E}[\Omega_j^{(k)}]\right), \tag{1}$$

where $\mathbb{E}[\Omega_j^{(k)}]$ is computed by averaging the proability distribution of each response within $\Omega^{(k)}$, and the $D^{(k)}$ represents the importance weight of each token within $\Omega^{(k)}$.

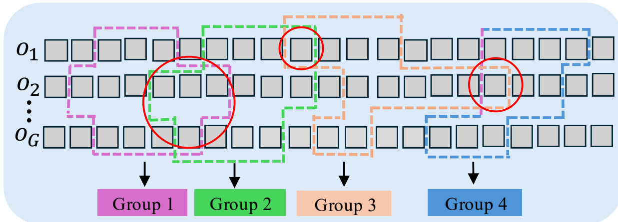


Figure B. The illustration of four intra-group perception tokens. The same token may attend to multiple crucial frames, as shown in the red solid circles in the figure.

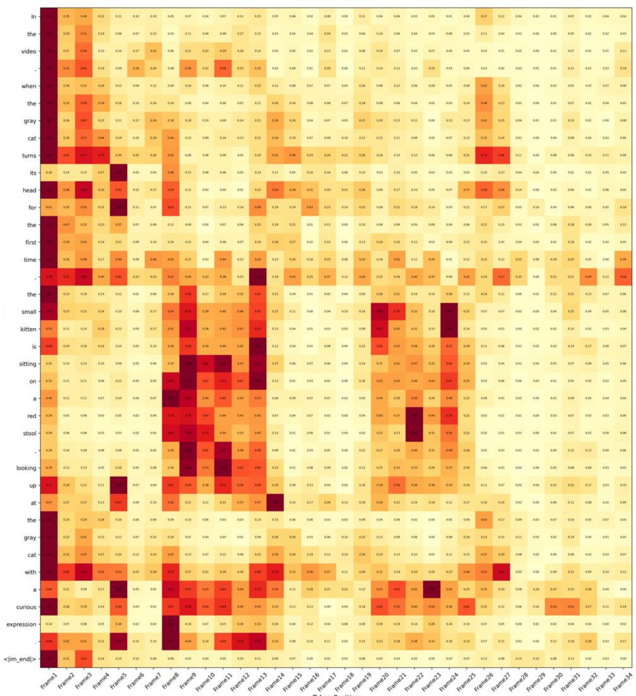


Figure C. The attention weights of response tokens to video frames. Question: "When the blue cat turned his head to look the second time, what was the small kitten doing?". The answer mainly appears in the middle part of the video ($15s \sim 20s$). The x-axis represents the video frame index (total 34 frames). The y-axis shows the response tokens. The darker the color, the higher the attention score.

Since the same token may attend to several crucial frames in ψ', these tokens may correspond to multiple optimization intensities, as shown in the red solid circles in Fig. B. Therefore, we additionally record the numbers of each token is optimized and the cumulative sum of the optimization intensities, then leverage the average intensity as the final token weights.

A.3. Relationship between answer correctness and focused frames

To analyze relationship between answer correctness and focused frames, we conducted statistics on NExTGQA test set, which features including keyframe labels to ensure reliability of this analysis. Specifically, we calculated average differences in focused frames between correct and incorrect answers from the same sample, shown in Tab. A. Approximately 20% of results *focus on the same video frames, regardless of correctness of final answer, making it difficult to select key frames by attention. In most cases* ($70\% \sim 80\%$)*, the key frames could be distinguished based on the correctness of final answer.* For those few special cases, it might be a meaningful direction for future exploration.

Table A. Differences in focused frames on NExTGQA test set.

Attention Difference	0s	$0 \sim 3s$	$3 \sim 6s$	$6 \sim 9s$	$9 \sim 12s$	$12 \sim 15s$	$> 15s$
Sample Nums	20.1%	23.5%	17.0%	12.6%	7.8%	6.2%	12.8%

B. Detailed Experimental Setup

For RL training, we limit the maximum number of response tokens to 512 for training efficiency. Each training step processes 16 input samples, with 8 rollouts per sample. Rollout sampling uses a temperature of 1.0 and top-p of 1.0. Specifically, for GRPO, we use a KL divergence coefficient of 0.01. For DAPO, the clip higher is set to 0.28. For our APPO, to obtain attention weights, we set the model's attention implementation to "eager" during sampling and average the weights across all attention heads. We use the official Hugging Face TRL library for training.

For evaluation, we use the vLLM [3] inference engine to accelerate, with a sampling temperature of 0.1, top-p of 0.001, top-k of 1, and a repetition penalty of 1.05. For accuracy calculation, we use regular expressions to match the predicted results with the ground truth. For mIoU calculation, we strictly compute the intersection and union between the predicted results and the labels.

Detailed hyperparameters duraing training and evaluation are shown in Tab. B.

C. Training Data Composition

To validate the effectiveness of our APPO compared to GRPO and DAPO, we collected subsets from existing dataset for RL training, including: $6K$ SEED-Bench-R1 training subset, $6K$ Perception Test multiple-choice subset, $3K$ NExT-GQA validation subset. To compare APPO with other video reasoning models, we selected $34K$ subset from Video-R1-$260K$ RL training data, as summaried in Tab. C.

D. Additional Experiment Results

D.1. Ablation results

To investigate the impact of the three selection strategies on the APPO algorithm, we conducted ablation experiments on SEED-Bench-R1 benchmark, as shown in Tab. D. It can be observed that the Soft selection strategy performs the best,

Table B. The hyperparameters used during training and evaluation.

Hyper-parameters	**Value**
Training	
Batch size	16
Gradient Accumulation Steps	1
Warmup	False
Rollout Numbers	8
Rollout Temperature	1.0
Rollout Top-P	1.0
Freeze Vision Encoder	True
KL divergence coefficient	1×10^{-2}
Learning rate	1×10^{-6}
GPUs	16
Optimizer	AdamW
Training Framework	TRL
Evaluation	
Inference Engine	vLLM
Temperature	0.1
Top-P	0.001
Top-K	1
Repetition Penalty	1.05

Table C. The detailed composition of $34K$ training subset selected from Video-R1-$260K$ RL training data.

Data Resource	**Sample Nums.**
LLaVA-Video	$\sim 5K$
STAR	$\sim 11K$
Perception Test	$\sim 6K$
NExT-QA	$\sim 5K$
CLEVRER	$\sim 6K$

Table D. Ablation results for different select strategies on SEED-Bench-R1 Benchmark.

Strategy	**L1 (In-Dist.)**	**L2 (OOD)**	**L3 (OOD)**
Hard	49.1	50.7	48.4
Soft	**50.5**	**51.3**	**49.3**
All	50.2	50.9	47.3

as it can fully promote fine-grained perception learning in both the high reward path set and the low reward path set.

D.2. Stability and robustness

Ablation studies in the main paper show that model performance is stable (within 0.5%) when hyperparameters are set within suitable ranges (*e.g.*, K_1: $15 \sim 20$, K_2: $2 \sim 4$, *etc.*). Additionally, five results with different seeds were used to calculate standard deviation (see Tab. E), showing stable improvements on 3/7B models.

For example, since APPO relies on the model's attention

Table E. Standard deviation results of Tab 1.

Model	SEED-Bench-R1 L1 (In-Dist.)	L2 (OOD)	L3 (OOD)	Perception Test	MVBench
Qwen2.5-VL 3B	0.238	0.319	0.279	0.301	0.257
Qwen2.5-VL 7B	0.103	0.183	0.201	0.163	0.095

Table F. The training efficiency comparison between DAPO and our APPO. Training efficiency refers to the training samples processed per second.

Model Scale	**Training Efficiency** DAPO	APPO (Ours)	**Percentage**
3B (30 frames)	0.325	0.264	81%
7B (16 frames)	0.462	0.429	93%

scores, this means that when sampling multiple reasoning paths, the model's attention implementation must be "eager." Additionally, existing inference acceleration frameworks, such as vLLM, have difficulty supporting the retrieval of attention scores, which limits the efficiency of the APPO algorithm.

E. Prompt Template

The prompt templates used for RL training on different tasks are as follows.

Prompt Template for Temporal Video Grounding Task

The video is {duration} seconds long, with {frame_nums} frames evenly sampled from it.
Based on the video content, think about the question deeply, select the correct option, and provide ONE or MORE time periods in the video where the clues corresponding to the correct option occur.
Question: {Question}
Output the thinking process in <think> </think> tags, final answer in <answer> </answer> tags and time clues in <time> </time> tags, i.e., <think> reasoning process here </think> <answer> answer here </answer> <time> 8.4 to 12.5, 25 to 30.2 </time>

Figure D. The prompt template for temporal video grounding task.

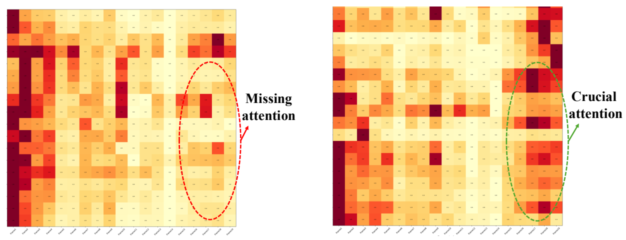
Prompt Template for RL Training

{Question} Output the thinking process in <think> </think> and final answer in <answer> </answer> tags, i.e., <think> reasoning process here </think><answer> answer here </answer>."

Figure E. The prompt template for RL training.

F. Visualization Results

As shown in Fig. G, we present visualization example from SEED-Bench-R1 Level-3 testset. It can be found that while GRPO and DAPO algorithms correctly answered question, it is noteworthy that our APPO algorithm successfully paid attention to the critical information of *handwashing*, which made the subsequent logical reasoning more coherent. Additionally, the attention visualization results in Fig. F demonstrate that APPO enables model to focus more on crucial frames during reasoning process, resulting in perception improvement.



(a) DAPO attention result. (b) APPO attention result.

Figure F. Attention comparison between DAPO and our APPO. **X-axis**: video frames. **Y-axis**: generated tokens during reasoning.

G. Limitations

While APPO achieves superior results than GRPO and DAPO in complex video reasoning scenarios, it also has some limitations. For example, since APPO relies on the model's attention scores, this means that when sampling multiple reasoning paths, the model's attention implementation must be "eager". Additionally, existing inference acceleration frameworks, such as vLLM, have difficulty supporting the output of attention scores. However, based on our experimental statistics, the training efficiency of APPO is $81\% \sim 93\%$ that of DAPO under the same training data and model scale, as shown in Tab. F. We will propose effective solutions to mitigate this limitation in the future works.

References

[1] Eric Bigelow, Ari Holtzman, Hidenori Tanaka, and Tomer Ullman. Forking paths in neural text generation. *arXiv preprint arXiv:2412.07961*, 2024. 1

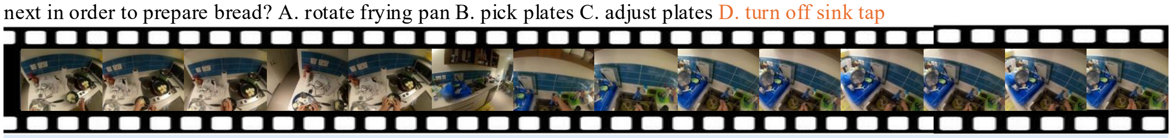
[2] Jisheng Dang, Jingze Wu, Teng Wang, Xuanhui Lin, Nannan Zhu, Hongbo Chen, Wei-Shi Zheng, Meng Wang, and Tat-Seng Chua. Reinforcing video reasoning with focused thinking. *arXiv preprint arXiv:2505.24718*, 2025. 1

[3] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023. 2

[4] Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Lin, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. Critical tokens matter: Token-level contrastive estimation enhances llm's reasoning capability. *arXiv preprint arXiv:2411.19943*, 2024. 1

[5] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. Mllms know where to look: Training-free perception of small visual details with multimodal llms. *arXiv preprint arXiv:2502.17422*, 2025. 1

Question: Considering the progress shown in the video and my current observation shown in the image, what action should I take next in order to prepare bread? A. rotate frying pan B. pick plates C. adjust plates D. turn off sink tap



GRPO: <think> The video shows that the person is preparing food on the stove. The next logical step would be to ensure the cooking area is clean and organized. Turning off the sink tap is a necessary action to prevent any water from splashing onto the stove or other cooking surfaces. This ensures safety and prevents any potential accidents. </think>\n<answer> D </answer>

DAPO: <think> The video shows that the person is preparing food on the stove and has just finished cooking something in a frying pan. The next logical step in preparing bread would be to clean up the kitchen area. The sink is visible with dishes and utensils, indicating it's time to wash them. Turning off the sink tap is a necessary step to ensure the water doesn't run while washing dishes. Therefore, the next action should be to turn off the sink tap. </think>\n<answer>D</answer>

APPO (Ours): <think>The task at hand involves preparing bread, which typically includes washing hands before handling food to maintain hygiene. The first image shows the person is cooking on the stove and has just finished making something in a frying pan. After washing hands, the next logical step would be to dry them, which is a common practice after washing to prevent cross-contamination or to proceed with food preparation tasks. The subsequent images show the drying of hands, which aligns with the typical sequence of washing and drying hands before handling food. The options provided do not include drying hands, so the closest relevant action from the given choices would be to turn off the sink tap, as it is a necessary step before drying hands. However, since the question asks for the next action in the context of preparing bread, and considering the sequence of typical food preparation steps, the most logical next action after washing hands would be to dry them, which is not directly listed but implied by the sequence of actions. Therefore, the closest option to the logical next step in the context of the given choices is to turn off the sink tap, as it is a prerequisite for drying hands.</think>\n\n<answer>D</answer>

Figure G. The visualization example from SEED-Bench-R1 Level-3 testset. The correct option is "D. turn off sink tap". Our APPO algorithm successfully paid attention to the critical information of handwashing, which made subsequent logical reasoning more coherent.