AdaThinkV: Adaptive Thinking for Token-Efficient Video Reasoning

Jingqi Tian[1]*, Haoji Zhang[1]*, Lin Chen[2]*, Hongbo Jin[3], Haonan Xu[2], Tianrui Zhu[1], Xingming Shui[1], Shilin Ma[1], Wenjing Yang[2], Yansong Tang[1]†

[1]Tsinghua Shenzhen International Graduate School, Tsinghua University [2]Alibaba Group [3]Peking University
{tjq25@mails, tang.yansong@sz}.tsinghua.edu.cn

Abstract

Chain-of-thought (CoT) reasoning can improve performance on difficult video questions but often wastes decoding tokens on simple ones. We study whether a video multimodal large language model can adapt its reasoning effort to each question. We propose AdaThinkV, an adaptive framework for video reasoning that learns whether to reason explicitly without offline difficulty labels, manually tuned confidence thresholds, or an external router. During reinforcement learning, AdaThinkV samples matched rollouts in explicit reasoning and direct answering modes for each prompt. ThinkGain estimates the prompt-level utility of explicit reasoning by balancing its accuracy gain against additional response length, providing supervision for both conditional response generation and autonomous mode selection. For difficult prompts, limited rollout exploration can yield groups in which every response is unsuccessful and accuracy rewards show little variation, providing insufficient signal for learning. We therefore introduce Variance Recovery Policy Optimization (VRPO), which retains and progressively expands these groups to recover informative signals from prompts that are difficult yet solvable. At inference, AdaThinkV selects a response mode and generates the response in a single autoregressive sequence. Across a unified suite of video reasoning evaluations, AdaThinkV achieves a mean accuracy of 40.79 with an average of 257.20 output tokens, outperforming the strongest evaluated adaptive baseline by 2.98 points while using 22.7% fewer tokens.

Introduction

Explicit chain-of-thought (CoT) can improve complex temporal reasoning in video multimodal large language models (MLLMs) (Wei et al. 2022; Feng et al. 2025), yet not every question warrants a long visible trace. Simple perception can often be handled directly, whereas event ordering, state tracking, and causal analysis may require multi-step reasoning. Always reasoning wastes tokens and can overcomplicate simple questions, whereas always answering directly can underthink compositional ones. As Fig. 1 illustrates, a capable video MLLM should learn not only *how* to reason, but also *whether* explicit reasoning is worth its cost.

Adaptive reasoning methods switch between concise and explicit reasoning (Tu et al. 2025; Zhang et al. 2025b; Fang, Ma, and Wang 2025). For video, VideoAuto-R1 selects a response mode based on the confidence of a preliminary answer (Liu et al. 2026). Confidence, however, is only an indirect proxy: uncertainty does not imply that further reasoning will help, while high confidence can coexist with missed temporal evidence. The relevant question is not whether the model is uncertain, but whether additional reasoning improves the answer enough to justify its token cost. We formulate this quantity as the *conditional net utility of reasoning*: the empirical accuracy gain from explicit reasoning minus its additional generation cost for the same input.

We introduce **AdaThinkV**, which learns to choose between explicit reasoning and direct answering without offline difficulty labels, confidence thresholds, or an external router. We first propose ThinkGain to estimate the prompt-specific utility of explicit reasoning from controlled `THINK`/`ANSWER` rollouts, accounting for both accuracy gain and additional response length. However, a fixed rollout group may not sufficiently explore difficult prompts, yielding only unsuccessful responses with little variation in accuracy rewards. Dynamic sampling addresses this issue by oversampling prompts and filtering out zero-variance groups to retain a fixed number of effective prompts, potentially biasing observed reward dispersion upward (Yu et al. 2025). To mitigate this bias, we introduce Variance Recovery Policy Optimization (VRPO), which progressively expands the same group while retaining existing samples for more reliable variance estimation. In inference, AdaThinkV generates the mode marker and response in one autoregressive sequence.

Across unified evaluation settings covering temporal reasoning, video mathematics, and scientific video understanding, AdaThinkV attains 40.79 mean accuracy with 257.20 output tokens. Relative to VideoAuto-R1-Qwen3, it improves accuracy by 2.98 points with 22.7% fewer tokens; relative to Qwen3-VL-8B-Thinking, it gains 3.66 points with 95.8% fewer tokens.

Our main contributions are fourfold:

- We formulate adaptive video reasoning as prompt-level utility estimation and introduce AdaThinkV, which selects whether to reason explicitly without difficulty labels, confidence thresholds, or an external router.
- We introduce ThinkGain, a paired-rollout estimator of prompt-level reasoning utility that balances accuracy improvement against output length.
- We introduce VRPO, a complementary rollout allocation strategy that retains and expands groups with unsuccess-

*These authors contributed equally.
†Corresponding author.

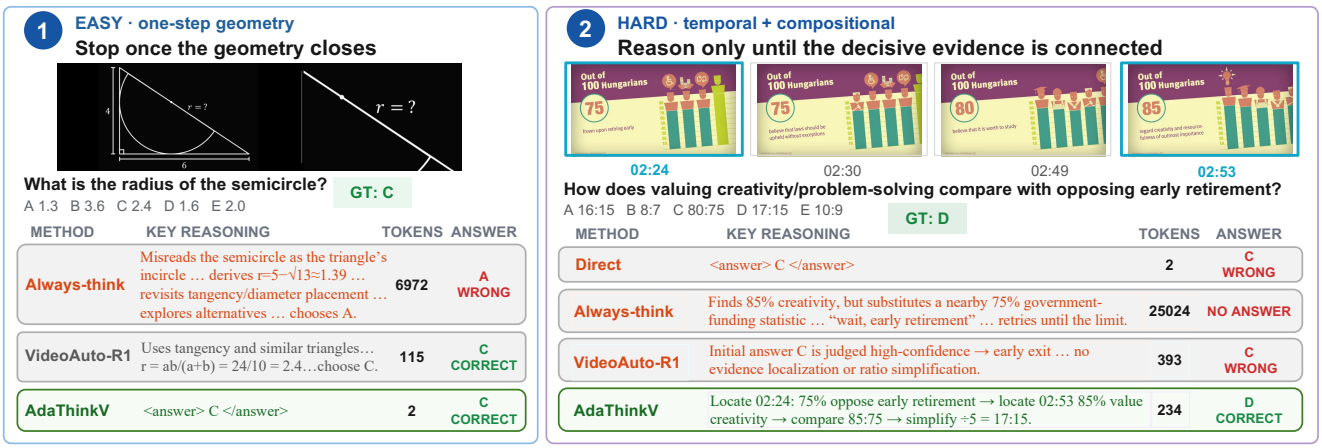

Figure 1: Why does response mode selection matter? Excessive reasoning can overcomplicate simple questions, while confidence-routed VideoAuto-R1 may stop before collecting evidence needed for compositional reasoning. AdaThinkV selects direct or explicit reasoning and terminates autoregressively.

ful, low-dispersion rewards to recover informative RL signals from difficult prompts.

- Experiments across diverse video benchmarks show that AdaThinkV improves accuracy and efficiency over adaptive and reasoning baselines. Controlled ablations reveal that ThinkGain primarily improves the accuracy–token trade-off, whereas VRPO raises absolute accuracy.

Related Work

Video reasoning and efficiency. Video reasoning has advanced through reinforcement learning, temporal grounding, structured distillation, and explicit reasoning traces (Feng et al. 2025; Lin et al. 2026; Zhu et al. 2025; Wang et al. 2025b; Tian et al. 2025; Fei et al. 2025; Maaz et al. 2025). Efficiency-oriented methods shorten reasoning traces, compress visual inputs, or scale perception and test-time computation (Zhong et al. 2025; Yan et al. 2025; Wang et al. 2025c; Buch et al. 2025; Zhang et al. 2025a). These methods improve reasoning or reduce generation and perception costs, but most do not directly optimize whether visible reasoning is worthwhile for each question. AdaThinkV studies this cost-aware, per-question decision in video reasoning.

Adaptive reasoning. AutoThink (Tu et al. 2025) and AdaptThink (Zhang et al. 2025b) choose between direct and explicit reasoning, while Thinkless (Fang, Ma, and Wang 2025) separates mode control from response optimization. SwitchReasoner uses fixed direct and thinking branches to construct sample-level supervision from their relative success and adds a global controller to balance mode usage (Fang et al. 2026). Other methods control latent reasoning or the disclosure of visible traces (Li et al. 2026; Wei et al. 2026). For video reasoning, VideoAuto-R1 selects a mode using the confidence of a preliminary answer (Liu et al. 2026). AdaThinkV instead estimates a length-aware utility from prompt-matched branches, applies a dead zone to uncertain comparisons, and separates marker supervision from conditional continuation learning. It generates the marker and response in one autoregressive sequence, requiring neither a preliminary answer nor an external router.

Group relative RL and rollout allocation. GRPO optimizes a policy with group-relative advantages and no value model, while DAPO extends it with dynamic sampling that filters groups with no accuracy variation (Shao et al. 2024; Yu et al. 2025). Later methods recover signals from such groups through modified credit assignment, advantage shaping, or reference-guided repair (Jin et al. 2026; Le et al. 2025, 2026). Other work reduces rollout waste through prompt filtering (Zheng et al. 2025; Li et al. 2025a) or allocates rollouts using predicted statistics, pilot samples, and adaptive budgets (Nguyen et al. 2026; Kim et al. 2026). VRPO instead retains the current unsuccessful group and adds balanced mode rollouts until task success or sufficient accuracy reward dispersion emerges, recovering a learning signal without reference-guided repair or a learned prompt allocator.

Method

Overview

Given a video v and a question q, let $x = (v, q)$. AdaThinkV generates a response with an autoregressive policy. Let $c \in \{\texttt{THINK}, \texttt{ANSWER}\}$ denote the response mode and let $m(c)$ denote its mode marker. The two response formats are

$$\begin{aligned} \texttt{THINK}: &\ \texttt{<think>}\, h\, \texttt{</think>} \\ &\ \texttt{<answer>}\, a\, \texttt{</answer>}, \\ \texttt{ANSWER}: &\ \texttt{<answer>}\, a\, \texttt{</answer>}, \end{aligned}$$

where h is an explicit reasoning trace and a is the final answer. Only the opening <think> or <answer> acts as the mode marker; the closing tags are format delimiters. All tokens after the marker form a continuation $u = (u_1, \ldots, u_T)$. A serialized mode marker may map to one or more tokenizer tokens. Writing $m(c) = (m_1, \ldots, m_{K_c})$, its probability is

$$\pi_\theta(m(c) \mid x) = \prod_{k=1}^{K_c} \pi_\theta(m_k \mid x, m_{<k}),$$

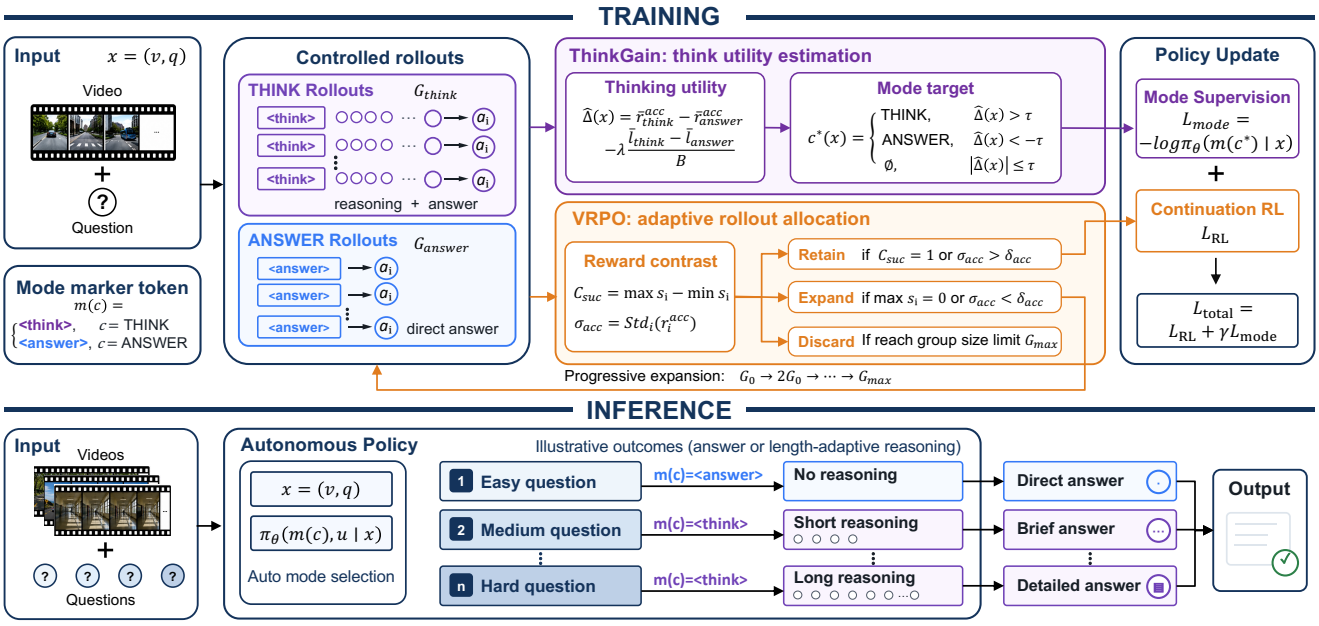

Figure 2: AdaThinkV RL training and inference pipeline. ThinkGain constructs mode supervision from prompt-level THINK/ANSWER branches. VRPO expands groups with weak reward contrast signals. At inference, the model autoregressively generates a mode marker followed by a variable-length continuation in a single sequence.

and the complete response factorizes as

$$\pi_\theta(m(c), u \mid x) = \pi_\theta(m(c) \mid x) \prod_{t=1}^{T} \pi_\theta(u_t \mid x, m(c), u_{<t}).$$

We first initialize the policy through an SFT cold start on balanced examples in both formats and then optimize it with reinforcement learning. Both modes must be explored to estimate when reasoning is useful, while group-relative optimization requires variation in accuracy rewards. As shown in Fig. 2, **ThinkGain** estimates prompt-level mode utility through controlled branches, whereas **Variance Recovery Policy Optimization (VRPO)** complements it by allocating additional rollouts to weak-signal groups. The shared training loop optimizes autonomous mode selection and conditional continuation generation.

ThinkGain: Prompt-Matched Estimation of Reasoning Gain

Controlled rollouts. For a rollout group of size G, we force a fraction ρ of responses to begin with the THINK mode marker and the remainder with the ANSWER mode marker:

$$G_{\text{think}} = \lfloor \rho G \rfloor, \qquad G_{\text{answer}} = G - G_{\text{think}}.$$

Mode identity is assigned by rollout position rather than inferred from generated text. Because both branches use the same input and decoding configuration, their within-prompt comparison removes between-prompt difficulty variation while retaining finite-sample generation noise. It estimates conditional differences in accuracy and continuation length. When VRPO expands a group, ThinkGain is recomputed over all rollouts while maintaining balanced allocation.

Mode utility with a length cost. For response y_i, let r_i^{acc} be its normalized task accuracy reward, where $0 \le r_i^{\text{acc}} \le 1$, and let $\ell_i = |u_i|$ be its continuation length measured by the tokenizer after the opening mode marker. For mode c, define

$$\bar{r}_c^{\text{acc}} = \frac{1}{G_c} \sum_{i:c_i=c} r_i^{\text{acc}}, \qquad \bar{\ell}_c = \frac{1}{G_c} \sum_{i:c_i=c} \ell_i.$$

ThinkGain estimates the net utility of explicit reasoning on prompt x as

$$\hat{\Delta}(x) = \bar{r}^{\text{acc}}_{\text{think}} - \bar{r}^{\text{acc}}_{\text{answer}} - \lambda \frac{\bar{\ell}_{\text{think}} - \bar{\ell}_{\text{answer}}}{B}. \quad (1)$$

In Eq. (1), B normalizes length and λ controls the extra decoding cost of THINK relative to ANSWER. A positive value indicates that the empirical accuracy gain from explicit reasoning outweighs its additional generation cost.

We use a margin $\tau \ge 0$ to define a dead zone and convert this estimate into a mode target:

$$c^*(x) = \begin{cases} \texttt{THINK}, & \hat{\Delta}(x) > \tau, \\ \texttt{ANSWER}, & \hat{\Delta}(x) < -\tau, \\ \emptyset, & |\hat{\Delta}(x)| \le \tau. \end{cases} \quad (2)$$

Because $\hat{\Delta}(x)$ is a Monte Carlo estimate based on finitely many samples, $c^*(x)$ is a utility preference rather than a ground-truth difficulty label. The dead zone suppresses small estimated differences but does not guarantee statistical certainty. We therefore treat the supervision as noisy. Additional ablations and diagnostic analyses are provided in the supplement.

Mode selection learning. Forced mode markers define a stratified intervention rather than actions sampled from the

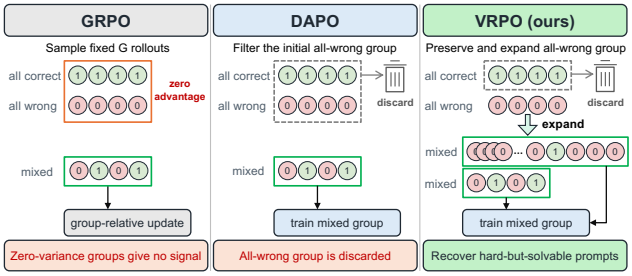

Figure 3: Rollout allocation comparison. GRPO uses fixed groups, DAPO replaces zero-contrast prompts, and VRPO retains and expands unsuccessful groups with low accuracy-reward dispersion.

old mode policy. We mask these tokens from the RL loss, so only the continuation is optimized conditional on the assigned mode. Autonomous mode selection is trained toward the target in Eq. (2) with a separate supervised likelihood evaluated before the mode marker:

$$\mathcal{L}_{\text{mode}} = -\frac{1}{N_{\text{mode}}} \sum_{\substack{x \in \mathcal{B}_{\text{RL}} \\ c^*(x) \neq \emptyset}} \log \pi_\theta(m(c^*(x)) \mid x), \qquad (3)$$

where N_{mode} is the number of prompts in the sum. We set this term to zero when $N_{\text{mode}} = 0$. Thus, marker positions receive no direct policy-gradient loss; their explicit token-level supervision is provided by Eq. (3). Continuation updates can still affect marker probabilities indirectly through the shared policy parameters.

Variance Recovery Policy Optimization

VRPO follows DAPO's policy optimization design but replaces its dynamic sampling rule (Yu et al. 2025). Rather than replacing a prompt with a weak learning signal and discarding its trajectories, VRPO preserves the current group and progressively adds rollouts. Fig. 3 contrasts this allocation rule with GRPO and DAPO.

Composite trajectory reward. We optimize retained trajectories with the total reward r_i as

$$\begin{aligned} r_i &= r_i^{\text{acc}} + r_i^{\text{fmt}} + r_i^{\text{ol}} + g_i, \\ g_i &= \alpha\, \mathbf{1}[c_i = c^*(x)]. \end{aligned} \qquad (4)$$

In Eq. (4), r_i^{fmt} scores compliance with the assigned serialization, and r_i^{ol} follows DAPO's soft penalty near the length limit. DAPO's hard mask removes completions truncated at this limit from the policy gradient loss. Since $c_i \in \{\texttt{THINK}, \texttt{ANSWER}\}$, no mode-consistency bonus is assigned when $c^*(x) = \emptyset$. Task correctness and VRPO allocation depend only on r_i^{acc}, so g_i supplements rather than replaces the accuracy reward. Because forced marker tokens are masked, g_i changes the advantages of continuation tokens but not the probability of the target marker. In contrast, $\mathcal{L}_{\text{mode}}$ acts directly on the autonomous marker probability. The two terms use the same target but optimize different conditional factors of the response.

Task success and reward dispersion. VRPO distinguishes a useful accuracy signal from task success. These concepts coincide for binary rewards based on exact answers but differ for continuous rewards such as temporal IoU. We define the binary accuracy s_i as

$$s_i = \begin{cases} r_i^{\text{acc}}, & \text{for binary rewards,} \\ \mathbf{1}[r_i^{\text{acc}} \geq \eta_{\text{task}}], & \text{for continuous rewards.} \end{cases}$$

For continuous tasks, η_{task} is fixed by the task success criterion rather than tuned on downstream test sets.

For the current group $\mathcal{Y}_x = \{y_i\}_{i=1}^{G_x}$, let

$$\bar{r}_x^{\text{acc}} = \frac{1}{G_x} \sum_{i=1}^{G_x} r_i^{\text{acc}}, \qquad C_{\text{suc}}(x) = \max_i s_i - \min_i s_i,$$

and define its empirical accuracy reward dispersion as

$$\sigma_{\text{acc}}(x) = \sqrt{\frac{1}{G_x} \sum_{i=1}^{G_x} (r_i^{\text{acc}} - \bar{r}_x^{\text{acc}})^2}.$$

A group provides a useful learning signal if it contains either contrast between successful and unsuccessful responses, $C_{\text{suc}}(x) = 1$, or sufficient accuracy reward dispersion, $\sigma_{\text{acc}}(x) > \delta_{\text{acc}}$. VRPO expands only groups in which all current responses are unsuccessful and their accuracy rewards remain insufficiently differentiated:

$$\text{action}(x) = \begin{cases} \text{retain}, & \begin{array}{l} C_{\text{suc}}(x)=1 \ \vee \\ \sigma_{\text{acc}}(x)>\delta_{\text{acc}}, \end{array} \\ \text{expand}, & \begin{array}{l} \max_i s_i=0,\ \sigma_{\text{acc}}(x)\leq\delta_{\text{acc}}, \\ G_x<G_{\max} \end{array} \\ \text{discard}, & \text{otherwise} \end{cases}. \qquad (5)$$

For continuous rewards, Eq. (5) retains a group in which all responses are successful when $\sigma_{\text{acc}}(x) > \delta_{\text{acc}}$ because its graded rewards still support relative learning. Such a group is discarded when its dispersion is insufficient. Each expansion preserves existing trajectories and the controlled mode ratio, and only retained groups contribute to $\mathcal{B}_{\text{RL}}$.

Expansion stops according to rewards, and the retained group is reused to estimate the ThinkGain target and compute the policy update. This reuse may introduce selection bias from optional stopping. We quantify the effect with diagnostics for each expansion stage and a comparison that uses an independently resampled update group in the supplement.

Optimization relative to each group. For each retained group, the trajectory advantage $\widehat{A}_i$ is obtained by normalizing the composite rewards across all G_x rollouts:

$$\widehat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^{G_x})}{\text{std}(\{r_j\}_{j=1}^{G_x}) + \epsilon}.$$

Here, $\epsilon > 0$ is a numerical stabilizer. The same trajectory advantage is applied to every continuation token in response i. For continuation token $u_{i,t}$, we define the importance ratio $s_{i,t}(\theta)$ as

$$s_{i,t}(\theta) = \frac{\pi_\theta(u_{i,t} \mid x, m(c_i), u_{i,<t})}{\pi_{\theta_{\text{old}}}(u_{i,t} \mid x, m(c_i), u_{i,<t})},$$

$$\widetilde{s}_{i,t}(\theta) = \text{clip}(s_{i,t}(\theta), 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}}).$$

We use $\epsilon_{\text{high}} > \epsilon_{\text{low}}$. The tighter lower clipping bound limits destructive probability decreases, whereas the looser upper

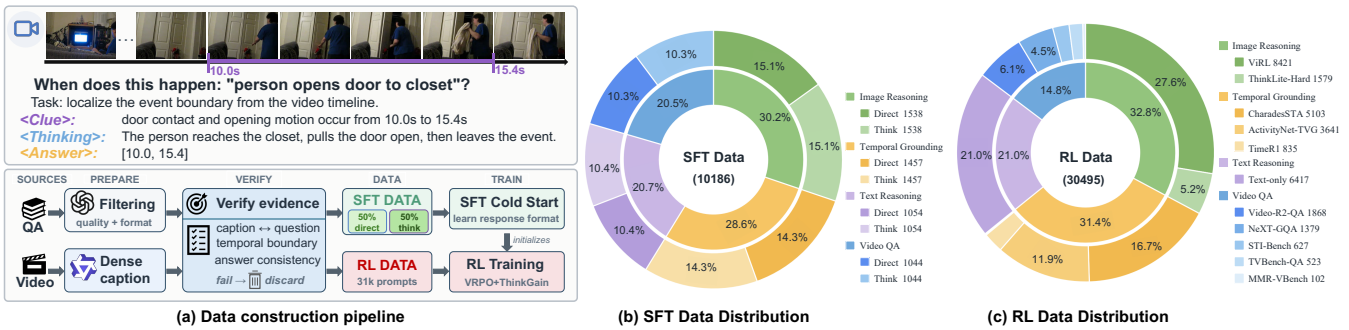

Figure 4: Training data construction pipeline and data distribution.

Model	Mode	VRB	VMath	VMath-M	MMRV	MMRV-CoT	SciV	Avg.	Tok.
Qwen2.5-VL-7B	Direct	3.54	29.76	55.95	42.00	34.77	27.90	32.32	400.82
Qwen3-VL-8B-Instruct	Direct	5.56	30.71	59.17	45.58	41.21	33.10	35.89	524.62
Qwen3-VL-8B-Thinking	Think only	6.11	38.81	62.38	42.72	42.24	30.50	37.13	6184.98
Temporal-RLT	Think only	2.64	29.52	55.77	41.29	35.96	26.80	32.00	343.20
Time-R1-7B	Think only	2.99	29.52	56.31	42.16	36.12	27.60	32.45	307.68
VideoRFT	Think only	2.22	25.71	54.94	40.89	40.49	26.00	31.71	427.35
Video-R1-7B	Think only	1.67	22.62	53.87	39.94	34.77	24.70	29.59	125.03
Video-RTS	Think only	3.13	28.33	56.31	42.24	37.07	28.40	32.58	483.78
LOVE-R1	Think only	2.50	18.57	49.94	41.21	40.02	14.10	27.72	848.18
Video-R2	Think only	1.81	29.29	56.55	39.62	35.72	27.70	31.78	346.05
VideoChat-R1.5	Think only	2.78	29.05	55.95	43.68	40.65	28.00	33.35	251.22
VideoAuto-R1-Qwen2.5	Auto	3.33	30.95	56.07	49.96	42.08	31.10	35.58	268.37
VideoAuto-R1-Qwen3	Auto	4.93	31.43	58.93	50.84	47.81	32.90	37.81	332.93
AdaThinkV w/o VRPO	Auto	6.94	36.67	59.76	46.62	45.51	34.40	38.32	235.73
AdaThinkV	Auto	**7.57**	**39.05**	**62.56**	**50.99**	**48.05**	**36.50**	**40.79**	257.20

Table 1: Comparison on video reasoning benchmarks. VRB, VMath, MMRV, and SciV denote VideoReasonBench, VideoMathQA, MMR-VBench, and SciVideoBench. Avg. is the average accuracy. Tok. is the average output token length.

bound allows advantageous but initially rare tokens to gain probability more quickly. Let μ_i be DAPO's binary hard overlong mask. It equals zero only when response i is truncated at the generated token limit. Following the clipped policy objective of PPO and the token aggregation of DAPO (Schulman et al. 2017; Yu et al. 2025), the optimization objective is

$$\mathcal{L}_{\mathrm{RL}} = -\frac{1}{|\mathcal{B}_{\mathrm{RL}}|} \sum_{x \in \mathcal{B}_{\mathrm{RL}}} \frac{1}{\sum_{i=1}^{G_x} \mu_i |u_i|} \sum_{i=1}^{G_x} \mu_i \sum_{t=1}^{|u_i|} \min\Big(s_{i,t}(\theta)\widehat{A}_i, \widetilde{s}_{i,t}(\theta)\widehat{A}_i \Big). \tag{6}$$

In Eq. (6), any prompt with zero unmasked continuation tokens is removed from $\mathcal{B}_{\mathrm{RL}}$, and $\mathcal{L}_{\mathrm{RL}} = 0$ if the buffer is empty. This normalization gives every unmasked continuation token equal weight within its prompt. It avoids the implicit $1/|u_i|$ weighting that arises when each response is averaged separately, and it prevents larger recovered groups from dominating the minibatch. The complete objective is

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{RL}} + \gamma \mathcal{L}_{\mathrm{mode}}. \tag{7}$$

In Eq. (7), γ controls mode supervision. As in DAPO, we use no KL penalty or separate reference model. We retain $\pi_{\theta_{\mathrm{old}}}$ only to compute policy optimization ratios. Continuation tokens are sampled from $\pi_{\theta_{\mathrm{old}}}$ conditional on the forced marker, so their ratios remain on policy for that conditional distribution. At test time, mode forcing is removed and the opening marker is generated autoregressively.

Experiments

Experimental Setup

Training data. Fig. 4 summarizes our SFT cold-start set of 10186 traces and RL set of 30495 prompt-only inputs, spanning image reasoning, temporal grounding, text reasoning, and video QA. Within each task family, the SFT set is balanced between direct-answer and explicit-reasoning traces. The RL set does not contain prefabricated mode-specific responses; instead, `THINK` and `ANSWER` exploration is balanced online through controlled rollout allocation. Data construction and filtering details are provided in the supplement.

Evaluation protocol. We evaluate every model in Tab. 1 with a unified pipeline. Unless stated otherwise, we sample between 2 and 256 frames at 2 FPS, impose a visual budget of 4096 tokens, and allow up to 32768 generated tokens subject to the context window. StreamingBench and OVO-Bench follow a query-time-causal protocol: for a query issued at timestamp t_q, frames are sampled only from the visible prefix $[0, t_q]$, and no post-query frames are provided to the

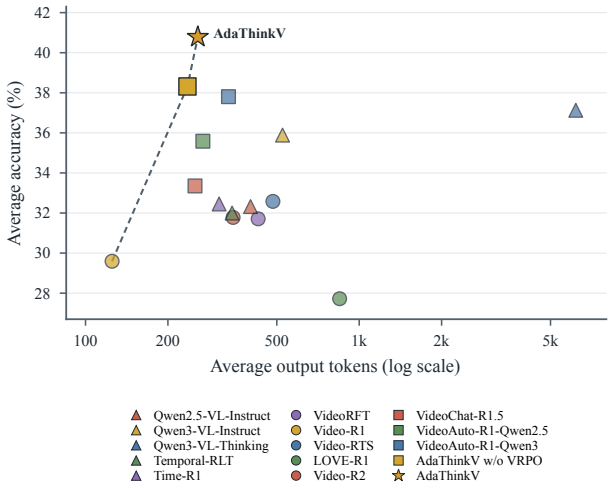

Figure 5: Accuracy–token trade-off. AdaThinkV advances the empirical Pareto frontier, achieving higher average accuracy with fewer generated tokens.

Model	V-MME	MVB	LongVB	MMVU	V-MMMU
Qwen2.5-VL-7B	66.0	67.1	60.9	66.2	54.7
Qwen3-VL-8B	**72.5**	69.4	67.6	69.9	61.0
Temporal-RLT	57.6	68.1	–	65.0	–
VideoRFT	59.8	62.1	–	68.5	51.1
Video-R1-7B	61.8	65.5	–	65.0	51.4
Video-RTS	63.0	–	56.6	66.4	52.7
VITAL	64.1	–	–	68.7	54.2
LongVILA-R1-7B	65.1	67.6	58.0	–	–
LOVE-R1	66.2	66.6	60.1	–	–
VideoChat-R1.5	65.2	70.6	61.4	–	49.6
VideoAuto-R1-Q2.5	67.3	71.0	60.5	69.7	58.6
VideoAuto-R1-Q3	71.7	72.0	67.4	71.1	65.0
AdaThinkV	72.1	**72.5**	**68.1**	**71.4**	**65.3**

Table 2: General video understanding benchmarks. V-MME, MVB, LongVB, and V-MMMU abbreviate Video-MME, MVBench, LongVideoBench, and Video-MMMU; Q2.5 and Q3 indicate the Qwen backbones.

model. All models share inputs, parsers, and scoring rules while retaining their native templates. Component ablations share initialization, training data, and optimization. VideoReasonBench uses its semantic judge, whereas the remaining benchmarks use exact accuracy. Benchmark-specific settings and a generation-cap sensitivity analysis are provided in the supplement.

Implementation details. Using Qwen3-VL-8B (Bai et al. 2025a) as the policy backbone, we train the SFT and RL stages for one epoch each, with a frozen visual encoder and a global batch size of 32. AdamW uses a learning rate of 1×10^{-6}, weight decay 0.01, and gradient clipping at 1.0. Rollouts use temperature 1.0, nucleus sampling with $p = 0.95$, top $k = 20$, and at most 4096 generated tokens. The clipping bounds are $\epsilon_{\text{low}} = 0.20$ and $\epsilon_{\text{high}} = 0.28$. ThinkGain uses $\rho = 0.5$, $\lambda = 0.1$, $B = 512$, $\tau = 0.05$, $\alpha = 0.3$, and $\gamma = 0.1$. VRPO starts from $G_0 = 8$, adds four rollouts per mode at each stage, and stops at $G_{\max} = 32$. For continuous rewards, it uses $\eta_{\text{task}} = 0.5$ and $\delta_{\text{acc}} = 0.05$, neither of which is tuned on downstream test sets. Under the Qwen3-VL tokenizer, the opening `<think>` marker is one token and `<answer>` is three tokens. We use the full-sequence marker likelihood in Eq. (3), without per-token length normalization, and mask every forced marker token from the continuation loss. Training is distributed across two eight-GPU nodes, for 16 NVIDIA H100 GPUs in total, with bfloat16 optimization, DeepSpeed ZeRO-1, colocated vLLM generation, and seed 42. Training cost is reported in Tab. 8, and detailed infrastructure is provided in the supplement.

	Charades-STA		ActivityNet		NExT-GQA	
Model	R@0.5	mIoU	R@0.5	mIoU	Acc.	mIoU
Qwen2.5-VL-7B	59.6	52.9	22.6	26.9	53.3	20.2
TimeChat	27.5	31.2	27.8	30.4	28.8	17.4
TimeSuite	67.1	–	–	–	–	–
TimeMarker	51.9	48.4	50.7	49.5	–	–
Temporal-RLT	67.9	57.0	38.4	39.0	78.7	37.3
Time-R1	72.2	58.8	**55.6**	52.1	–	–
VITAL	72.0	59.9	50.8	49.8	78.7	43.0
VideoChat-R1.5	71.6	60.6	32.3	35.3	–	–
VideoAuto-R1-Q2.5	70.8	60.0	48.5	47.6	80.6	36.7
VideoAuto-R1-Q3	74.9	63.7	54.3	51.9	81.1	44.2
AdaThinkV	**75.2**	**64.0**	54.9	**52.2**	**81.4**	**44.5**

Table 3: Temporal grounding benchmarks. Localization is measured by R@0.5 and mIoU; NExT-GQA also reports answer accuracy.

Model	StreamingBench	OVO-Bench
Qwen2.5-VL-7B	73.31	52.28
TimeChat-Online-7B	75.28	51.80
StreamForest-7B	77.26	56.60
HERMES-7B	79.44	59.20
Qwen3-VL-8B	78.65	63.51
AdaThinkV	**79.79**	**64.02**

Table 4: Streaming video understanding. StreamingBench reports RTVU accuracy; OVO-Bench averages its Real-Time and Backward tracks.

Main Results

Main video reasoning results. Tab. 1 reports results on VideoReasonBench, VideoMathQA, MMR-VBench, and SciVideoBench (Liu et al. 2025; Rasheed et al. 2026; Zhu et al. 2026; Deng et al. 2025). We compare general-purpose, specialized, and adaptive reasoning models (Bai et al. 2025b,a; Li et al. 2025a; Wang et al. 2025b,a; Feng et al. 2025; Wang et al. 2025c; Fu et al. 2025; Maaz et al. 2025; Yan et al. 2025; Liu et al. 2026). Under the unified protocol, AdaThinkV reaches an average accuracy of 40.79, exceeding VideoAuto-R1-Qwen3 by 2.98 points with 22.7% fewer generated tokens; the paired-bootstrap 95% confidence interval

Training recipe	Acc.	Fmt.	Tok.	Think
RL only	35.59	92.4	660.20	87.5
SFT only	36.80	99.1	260.48	33.9
SFT $\rightarrow$ RL	40.79	99.3	257.20	49.1

Table 5: SFT and RL stage ablation. Acc. and Tok. use the six-setting evaluation in Tab. 1; Fmt. and Think are percentages over the same examples.

Method	ThinkGain	VRB	VMath	MMRV	SciV	Avg.	Tok.
GRPO	No	5.42	30.24	42.35	31.80	27.45	612.40
SAPO	No	5.76	31.19	43.10	32.20	28.06	568.60
DAPO	No	6.04	31.67	43.85	32.60	28.54	524.25
VRPO	No	7.43	38.57	50.55	36.35	33.23	481.85
DAPO	Yes	6.94	36.67	46.62	34.40	31.16	235.73
VRPO	Yes	7.57	39.05	50.99	36.50	33.53	257.20

Table 6: ThinkGain and VRPO ablation. Results average VRB, VMath, MMRV, and SciV.

for this gain is $[2.21, 3.75]$. As shown in Fig. 5, both AdaThinkV variants lie on the empirical accuracy–token Pareto frontier. The average across the four canonical benchmarks is 33.53, while the average across benchmark families is 36.10.

General video understanding. Tab. 2 extends the comparison to Video-MME (Fu et al. 2024), MVBench (Li et al. 2023), LongVideoBench (Wu et al. 2024), MMVU (Zhao et al. 2025), and Video-MMMU (Hu et al. 2025). Against the broader set of baselines, including VITAL and LongVILA-R1 (Zhang et al. 2025a; Chen et al. 2025), AdaThinkV ranks first on four of the five benchmarks and remains within 0.4 points of the best Video-MME result.

Temporal grounding and streaming video. Tab. 3 evaluates Charades-STA (Gao et al. 2017), ActivityNet Captions (Krishna et al. 2017), and NExT-GQA (Xiao et al. 2024), while Tab. 4 covers StreamingBench and OVO-Bench under the query-time-causal visible-prefix protocol (Lin et al. 2024; Li et al. 2025b). Across the established temporal and streaming baselines (Ren et al. 2023; Zeng et al. 2024; Chen et al. 2024; Huang et al. 2025; Zeng et al. 2025; Zhang et al. 2026), AdaThinkV ranks first on all but one temporal-grounding metric and on both streaming benchmarks.

Ablation Study

SFT cold start and RL stages. Tab. 5 shows that adding RL after the balanced SFT cold start improves mean accuracy by 3.99 points while maintaining over 99% format compliance.

ThinkGain and VRPO. Tab. 6 compares GRPO, SAPO, and DAPO (Shao et al. 2024; Gao et al. 2026; Yu et al. 2025). ThinkGain improves DAPO by 2.62 points while more than halving output length; on VRPO, it adds 0.30 points while reducing length by 46.6%. Thus, ThinkGain primarily improves the accuracy–decoded-length trade-off, whereas VRPO adds 2.37 points with ThinkGain fixed. The gain is consistent across three seeds; further decompositions are reported in the supplement.

Inference modes. Tab. 7 shows that automatic inference is

Policy	Mode	Think	Tok.	Avg.
VideoAuto-R1	Direct	0.0	15.60	28.82
VideoAuto-R1	Think	100.0	2380.40	31.98
VideoAuto-R1	Auto	16.8	370.43	30.03
AdaThinkV	Direct	0.0	20.50	31.42
AdaThinkV	Think	100.0	693.95	34.24
AdaThinkV	Auto	47.2	257.20	33.53

Table 7: Inference mode comparison. Avg. and Tok. average VRB, VMath, MMRV, and SciV; Think is the percentage of `THINK` openings.

Setting	Avg. G	Seq. (M)	Tok. (M)	GPU-h	Acc.
Fixed $G = 8$ (DAPO)	8.0	0.244	62.75	494	26.00
Fixed $G = 16$	16.0	0.488	125.49	988	26.52
Fixed $G = 32$	32.0	0.976	250.99	1975	26.87
Fixed $G \in \{8, 16\}$ mix	11.6	0.354	91.00	718	26.25
Expand all zero-contrast	12.4	0.378	97.26	765	27.01
VRPO	11.6	0.354	90.98	716	27.71

Table 8: Training efficiency comparison between VRPO and other rollout settings. Accuracy is averaged over VRB, VMath, and SciV. The fixed mix is compute-matched to VRPO, whereas "Expand all zero-contrast" expands both all-wrong and all-correct groups.

0.71 points below forced `THINK` (95% CI $[-1.57, 0.15]$) with 62.9% fewer tokens, and 2.11 points above forced `ANSWER`. Across difficulty tertiles, the thinking rate rises from 25.6% to 69.1% and the mean `THINK` length from 151 to 612 tokens; prompt-level reasoning length is moderately correlated with difficulty ($\rho = 0.39$).

Rollout allocation. Tab. 8 shows that increasing fixed G from 8 to 32 nearly quadruples training cost for only a 0.87-point gain. At comparable compute, VRPO reaches 27.71 accuracy with 716 GPU-hours, compared with 26.25 at 718 GPU-hours for the fixed-allocation control.

Conclusion

We presented AdaThinkV, an adaptive framework for video reasoning that learns when explicit reasoning is worth its generation cost. ThinkGain estimates prompt-level reasoning utility by balancing accuracy improvement against additional output length, thereby supervising both mode selection and conditional response generation. VRPO retains and progressively expands all-unsuccessful, low-dispersion rollout groups, recovering informative learning signals from difficult yet solvable prompts. At inference, AdaThinkV generates the selected mode marker and response in a single autoregressive sequence, without an external router or a preliminary answer. Under the unified evaluation protocol, AdaThinkV outperforms the strongest evaluated adaptive baseline by 2.98 accuracy points while generating 22.7% fewer output tokens. Future work will extend this framework to image reasoning and other multimodal tasks and generalize its utility estimation and rollout allocation beyond scalar accuracy rewards to feedback from generative and rubric-based reward models.

References

Bai, S.; Cai, Y.; Chen, R.; Chen, K.; Chen, X.; et al. 2025a. Qwen3-VL Technical Report. arXiv:2511.21631.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; Zhong, H.; Zhu, Y.; Yang, M.; Li, Z.; Wan, J.; Wang, P.; Ding, W.; Fu, Z.; Xu, Y.; Ye, J.; Zhang, X.; Xie, T.; Cheng, Z.; Zhang, H.; Yang, Z.; Xu, H.; and Lin, J. 2025b. Qwen2.5-VL Technical Report. arXiv:2502.13923.

Buch, S.; Nagrani, A.; Arnab, A.; and Schmid, C. 2025. Flexible Frame Selection for Efficient Video Reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 29071–29082.

Chen, S.; Lan, X.; Yuan, Y.; Jie, Z.; and Ma, L. 2024. TimeMarker: A Versatile Video-LLM for Long and Short Video Understanding with Superior Temporal Localization Ability. arXiv:2411.18211.

Chen, Y.; Huang, W.; Shi, B.; Hu, Q.; Ye, H.; Zhu, L.; Liu, Z.; Molchanov, P.; Kautz, J.; Qi, X.; Liu, S.; Yin, H.; Lu, Y.; and Han, S. 2025. Scaling RL to Long Videos. arXiv:2507.07966.

Deng, A.; Yang, T.; Yu, S.; Spencer, L.; Bansal, M.; Chen, C.; Yeung-Levy, S.; and Wang, X. 2025. SciVideoBench: Benchmarking Scientific Video Reasoning in Large Multimodal Models. arXiv:2510.08559.

Fang, G.; Ma, X.; and Wang, X. 2025. Thinkless: LLM Learns When to Think. arXiv:2505.13379.

Fang, Y.; Fu, P.; Li, J.; Liang, J.; Huang, W.; Luo, R.; Zhang, S.; Luan, J.; Fung, Y. R.; and Ye, M. 2026. Switch-Reasoner: Learn When to Think in Multitask Mixtures via Reinforcement Learning. arXiv:2607.08572.

Fei, H.; Wu, S.; Ji, W.; Zhang, H.; Zhang, M.; Lee, M.-L.; and Hsu, W. 2025. Video-of-Thought: Step-by-Step Video Reasoning from Perception to Cognition. arXiv:2501.03230.

Feng, K.; Gong, K.; Li, B.; Guo, Z.; Wang, Y.; Peng, T.; Wang, B.; and Yue, X. 2025. Video-R1: Reinforcing Video Reasoning in MLLMs. arXiv:2503.21776.

Fu, C.; Dai, Y.; Luo, Y.; Li, L.; Ren, S.; Zhang, R.; Wang, Z.; Zhou, C.; Shen, Y.; Zhang, M.; Chen, P.; Li, Y.; Lin, S.; Zhao, S.; Li, K.; Xu, T.; Zheng, X.; Chen, E.; Ji, R.; and Sun, X. 2024. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis. arXiv:2405.21075.

Fu, S.; Yang, Q.; Li, Y.-M.; Wei, X.; Xie, X.; and Zheng, W.-S. 2025. LOVE-R1: Advancing Long Video Understanding with an Adaptive Zoom-in Mechanism via Multi-Step Reasoning. arXiv:2509.24786.

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. TALL: Temporal Activity Localization via Language Query. In *Proceedings of the IEEE International Conference on Computer Vision*, 5267–5275.

Gao, L.; Li, Z.; Jia, M.; Yuan, J.; Sun, H.; Sun, H.; and Li, X. 2026. Segment-Aligned Policy Optimization for Multi-Modal Reasoning. arXiv:2605.01327.

Hu, K.; Wu, P.; Pu, F.; Xiao, W.; Zhang, Y.; Yue, X.; Li, B.; and Liu, Z. 2025. Video-MMMU: Evaluating Knowledge Acquisition from Multi-Discipline Professional Videos. arXiv:2501.13826.

Huang, Z.; Li, X.; Li, J.; Wang, J.; Zeng, X.; Liang, C.; Wu, T.; Chen, X.; Li, L.; and Wang, L. 2025. Online Video Understanding: OVBench and VideoChat-Online. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jin, H.; Zhu, R.; Du, Z.; Jiang, X.; Tian, J.; Zhang, Q.; and Ding, J. 2026. DGPO: Distribution Guided Policy Optimization for Fine Grained Credit Assignment. arXiv:2605.03327.

Kim, W.; Yang, Z.; Yan, J. N.; and Liu, J. 2026. Spend Your Rollouts Where It Counts: Rollout Allocation for Group-Based RL Post-Training. arXiv:2605.26606.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Niebles, J. C. 2017. Dense-Captioning Events in Videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 706–715.

Le, D. A.; Nguyen, T.-P.; Nguyen, T. H.; Van, L. N.; and Le, T. 2026. Selective Off-Policy Reference Tuning with Plan Guidance. arXiv:2605.11505.

Le, T.-L. V.; Jeon, M.; Vu, K.; Lai, V.; and Yang, E. 2025. No Prompt Left Behind: Exploiting Zero-Variance Prompts in LLM Reinforcement Learning via Entropy-Guided Advantage Shaping. arXiv:2509.21880.

Li, H.; Han, S.; Liao, Y.; Luo, J.; Gao, J.; Yan, S.; and Liu, S. 2025a. Reinforcement Learning Tuning for VideoLLMs: Reward Design and Data Efficiency. arXiv:2506.01908.

Li, K.; Wang, Y.; He, Y.; Li, Y.; Wang, Y.; Liu, Y.; Wang, Z.; Xu, J.; Chen, G.; Luo, P.; Wang, L.; and Qiao, Y. 2023. MVBench: A Comprehensive Multi-modal Video Understanding Benchmark. arXiv:2311.17005.

Li, P.; Hou, B.; Zhu, Y.; Feng, Y.; Ye, K.; Lei, T.; Chen, Z.; Chen, T.; and Du, X. 2026. Adaptive Thinking: Large Language Models Know When to Think in Latent Space. In *International Conference on Learning Representations*.

Li, Y.; Niu, J.; Miao, Z.; Ge, C.; Zhou, Y.; He, Q.; Dong, X.; Duan, H.; Ding, S.; Qian, R.; Zhang, P.; Zang, Y.; Cao, Y.; He, C.; and Wang, J. 2025b. OVO-Bench: How Far Is Your Video-LLMs from Real-World Online Video Understanding? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Lin, H.; Lv, K.; Jiang, X.; Tian, J.; Du, Z.; Ding, J.; Zhang, Q.; and Jin, H. 2026. VISD: Enhancing Video Reasoning via Structured Self-Distillation. arXiv:2605.06094.

Lin, J.; Fang, Z.; Chen, C.; Wan, Z.; Luo, F.; Li, P.; Liu, Y.; and Sun, M. 2024. StreamingBench: Assessing the Gap for MLLMs to Achieve Streaming Video Understanding. arXiv:2411.03628.

Liu, S.; Zhuge, M.; Zhao, C.; Chen, J.; Wu, L.; Liu, Z.; Zhu, C.; Cai, Z.; Zhou, C.; Liu, H.; Chang, E.; Suri, S.; Xu, H.; Qian, Q.; Wen, W.; Varadarajan, B.; Liu, Z.; Xu, H.; Bordes, F.; Krishnamoorthi, R.; Ghanem, B.; Chandra, V.; and Xiong, Y. 2026. VideoAuto-R1: Video Auto Reasoning via Thinking Once, Answering Twice. arXiv:2601.05175.

Liu, Y.; Ouyang, K.; Wu, H.; Liu, Y.; Sui, L.; Li, X.; Zhong, Y.; Charles, Y.; Zhou, X.; and Sun, X. 2025. VideoReasonBench: Can MLLMs Perform Vision-Centric Complex Video Reasoning? arXiv:2505.23359.

Maaz, M.; Rasheed, H.; Khan, F. S.; and Khan, S. 2025. Video-R2: Reinforcing Consistent and Grounded Reasoning in Multimodal Language Models. arXiv:2511.23478.

Nguyen, H. T.; Nguyen, B.; Ma, W.; Zhao, Y.; She, R.; and Nguyen, V. A. 2026. Adaptive Rollout Allocation for Online Reinforcement Learning with Verifiable Rewards. arXiv:2602.01601.

Rasheed, H.; Shaker, A.; Tang, A.; Maaz, M.; Yang, M.-H.; Khan, S.; and Khan, F. S. 2026. VideoMathQA: Benchmarking Mathematical Reasoning via Multimodal Understanding in Videos. In *International Conference on Learning Representations*.

Ren, S.; Yao, L.; Li, S.; Sun, X.; and Hou, L. 2023. TimeChat: A Time-Sensitive Multimodal Large Language Model for Long Video Understanding. arXiv:2312.02051.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300.

Tian, J.; Du, Y.; Zhang, H.; Wang, Y.; Lee, I. N.; Bai, X.; Zhu, T.; Niu, J.; and Tang, Y. 2025. DDAVS: Disentangled Audio Semantics and Delayed Bidirectional Alignment for Audio-Visual Segmentation. arXiv:2512.20117.

Tu, S.; Lin, J.; Zhang, Q.; Tian, X.; Li, L.; Lan, X.; and Zhao, D. 2025. Learning When to Think: Shaping Adaptive Reasoning in R1-Style Models via Multi-Stage Reinforcement Learning. arXiv:2505.10832.

Wang, Q.; Yu, Y.; Yuan, Y.; Mao, R.; and Zhou, T. 2025a. VideoRFT: Incentivizing Video Reasoning Capability in MLLMs via Reinforced Fine-Tuning. arXiv:2505.12434.

Wang, Y.; Wang, Z.; Xu, B.; Du, Y.; Lin, K.; Xiao, Z.; Yue, Z.; Ju, J.; Zhang, L.; Yang, D.; Fang, X.; He, Z.; Luo, Z.; Wang, W.; Lin, J.; Luan, J.; and Jin, Q. 2025b. Time-R1: Post-Training Large Vision Language Model for Temporal Video Grounding. In *Advances in Neural Information Processing Systems*.

Wang, Z.; Yoon, J.; Yu, S.; Islam, M. M.; Bertasius, G.; and Bansal, M. 2025c. Video-RTS: Rethinking Reinforcement Learning and Test-Time Scaling for Efficient and Enhanced Video Reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Wei, J.; Guo, X.; Yu, P.; Zhang, X.; Ouyang, W.; Sun, S.; Wang, Q.; and You, C. 2026. When to Think, When to Speak: Learning Disclosure Policies for LLM Reasoning. arXiv:2605.03314.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Wu, H.; Li, D.; Chen, B.; and Li, J. 2024. LongVideoBench: A Benchmark for Long-Context Interleaved Video-Language Understanding. arXiv:2407.15754.

Xiao, J.; Yao, A.; Li, Y.; and Chua, T.-S. 2024. Can I Trust Your Answer? Visually Grounded Video Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13204–13214.

Yan, Z.; Li, X.; He, Y.; Yue, Z.; Zeng, X.; Wang, Y.; Qiao, Y.; Wang, L.; and Wang, Y. 2025. VideoChat-R1.5: Visual Test-Time Scaling to Reinforce Multimodal Reasoning by Iterative Perception. arXiv:2509.21100.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Fan, T.; Liu, G.; Liu, L.; Liu, X.; Lin, H.; Qi, Z.; Ma, B.; Sheng, G.; Tong, Y.; Zhang, C.; Zhang, M.; Zhang, W.; Zhu, H.; Zhu, J.; Chen, J.; Chen, J.; Wang, C.; Yu, H.; Dai, W.; Song, Y.; Wei, X.; Zhou, H.; Liu, J.; Ma, W.-Y.; Zhang, Y.-Q.; Yan, L.; Qiao, M.; and Wu, Y. 2025. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. arXiv:2503.14476.

Zeng, X.; Li, K.; Wang, C.; Li, X.; Jiang, T.; Yan, Z.; Li, S.; Shi, Y.; Yue, Z.; Wang, Y.; Wang, Y.; Qiao, Y.; and Wang, L. 2024. TimeSuite: Improving MLLMs for Long Video Understanding via Grounded Tuning. arXiv:2410.19702.

Zeng, X.; Qiu, K.; Zhang, Q.; Li, X.; Wang, J.; Li, J.; Yan, Z.; Tian, K.; Tian, M.; Zhao, X.; Wang, Y.; and Wang, L. 2025. StreamForest: Efficient Online Video Understanding with Persistent Event Memory. arXiv:2509.24871.

Zhang, H.; Gu, X.; Li, J.; Ma, C.; Bai, S.; Zhang, C.; Zhang, B.; Zhou, Z.; He, D.; and Tang, Y. 2025a. Thinking With Videos: Multimodal Tool-Augmented Reinforcement Learning for Long Video Reasoning. arXiv:2508.04416.

Zhang, H.; Yang, S.; Fu, J.; Ng, S.-K.; and Qiu, X. 2026. HERMES: KV Cache as Hierarchical Memory for Efficient Streaming Video Understanding. arXiv:2601.14724.

Zhang, J.; Lin, N.; Hou, L.; Feng, L.; and Li, J. 2025b. AdaptThink: Reasoning Models Can Learn When to Think. arXiv:2505.13417.

Zhao, Y.; Xie, L.; Zhang, H.; Gan, G.; Long, Y.; Hu, Z.; Hu, T.; Chen, W.; Li, C.; Song, J.; Xu, Z.; Wang, C.; Pan, W.; Shangguan, Z.; Tang, X.; Liang, Z.; Liu, Y.; Zhao, C.; and Cohan, A. 2025. MMVU: Measuring Expert-Level Multi-Discipline Video Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zheng, H.; Zhou, Y.; Bartoldson, B. R.; Kailkhura, B.; Lai, F.; Zhao, J.; and Chen, B. 2025. Act Only When It Pays: Efficient Reinforcement Learning for LLM Reasoning via Selective Rollouts. arXiv:2506.02177.

Zhong, Y.; Hu, Z.-Y.; Li, Y.; and Wang, L. 2025. Rethinking Chain-of-Thought Reasoning for Videos. arXiv:2512.09616.

Zhu, K.; Jin, Z.; Yuan, H.; Li, J.; Tu, S.; Cao, P.; Chen, Y.; Liu, K.; and Zhao, J. 2026. MMR-V: What's Left Unsaid? A Benchmark for Multimodal Deep Reasoning in Videos. In *International Conference on Learning Representations*.

Zhu, T.; Zhang, S.; Sun, Z.; Tian, J.; and Tang, Y. 2025. Memorize-and-Generate: Towards Long-Term Consistency in Real-Time Video Generation. arXiv:2512.18741.