ResAdapt: Adaptive Resolution for Efficient Video Reasoning

Anonymous authors
Paper under double-blind review

Abstract

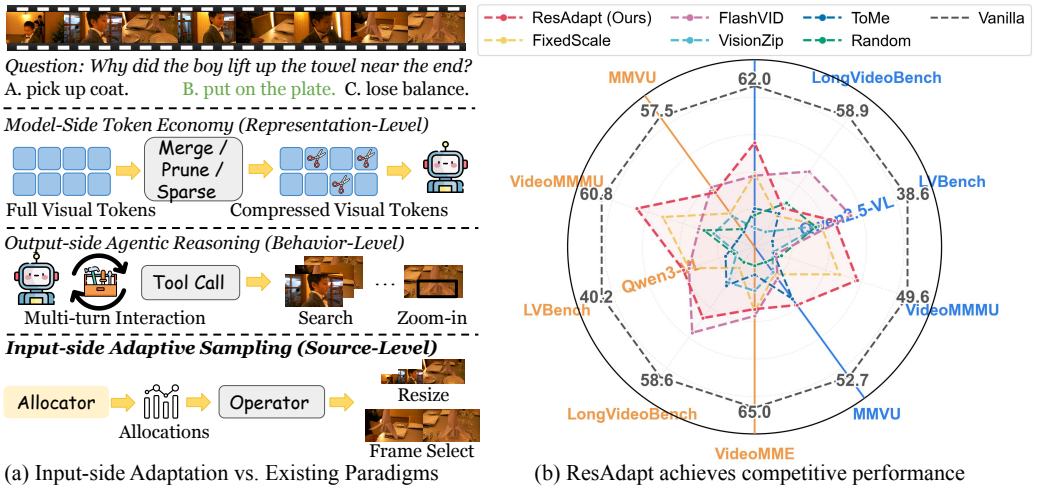
Multimodal Large Language Models (MLLMs) achieve stronger visual understanding by scaling input fidelity, but the resulting token explosion makes it prohibitive to jointly sustain high spatial resolution and long temporal context. Existing efficiency strategies only partially resolve this tension: *model-side* compression discards fine-grained evidence after encoding, while *output-side* agentic reasoning adds multi-pass latency. We argue that the true bottleneck lies in the initial volume of pixels processed, and introduce **ResAdapt**, an **input-side adaptation** framework that dynamically allocates visual budgets *before* encoding. ResAdapt couples a lightweight Allocator with a frozen MLLM backbone, formulated as a contextual bandit. To train the Allocator, we propose **Cost-Aware Policy Optimization (CAPO)** to translate sparse rollout feedback into a stable accuracy–cost signal, complemented by a temporal-similarity regularizer that prevents redundant allocations across visually similar frames. Extensive evaluations demonstrate that ResAdapt establishes a new Pareto frontier in low-budget video QA and reasoning-augmented temporal grounding. By reinvesting saved spatial compute into extended temporal coverage, ResAdapt supports up to 16× more frames at the same visual budget while delivering over 15% performance gains. Our results highlight that learning a sparse, content-dependent, single-pass allocation policy is a highly effective route to long-context multimodal reasoning under strict efficiency constraints. Code is available at `https://anonymous.4open.science/r/ResAdapt`.

1 Introduction

Multimodal Large Language Models (MLLMs) achieve stronger visual understanding by scaling input fidelity, yet the resulting visual-token growth makes jointly sustaining high spatial resolution and long temporal context prohibitive (Guo et al., 2025a; Bai et al., 2025a; Liu et al., 2025a; Shu et al., 2025; Shao et al., 2025b). In practice, this trade-off is central to video reasoning: reducing resolution risks losing the small visual cues that determine the answer, whereas shortening the clip removes the temporal context needed for long-horizon inference. Even architecturally efficient encoders (Zhang et al., 2026; Liu et al., 2025b) do not remove this tension; they merely shift where it becomes painful.

Existing efficiency methods typically fall into two paradigms (Figure 1a), both of which intervene late in the pipeline. *Model-side* approaches prune or merge tokens post-encoding (Khaki et al., 2025; Xu et al., 2025; Bolya et al., 2022; Tao et al., 2025); however, discarded fine-grained evidence cannot be recovered, and irregular token layouts disrupt optimized attention kernels (Dao, 2024; Kwon et al., 2023; Zheng et al., 2024). Conversely, *output-side* agentic methods rely on iterative retrieval or zooming (Zhang et al., 2025b; Yang et al., 2025d; Shen et al., 2025b; Zheng et al., 2025b). While improving coverage, they introduce multi-turn latency and risk missing decisive cues if the initial coarse view is over-compressed.

To overcome these limitations, we shift focus from post-encoding compression to pre-encoding pixel allocation. We propose **Input-side adaptation**, which reallocates the visual budget *before* encoding. Our method, **ResAdapt**, employs a lightweight allocator to predict per-frame visual budgets conditioned on coarse visual features and the text query. This budget is then materialized via operations like dynamic resolution resizing or frame selection. Consequently, the backbone processes a standard token sequence in a single pass,



(a) Input-side Adaptation vs. Existing Paradigms (b) ResAdapt achieves competitive performance

Figure 1: Input-side Adaptation improves the visual-token efficiency frontier. (a) Comparison of efficiency paradigms. While model-side methods compress tokens post-encoding and output-side agents rely on iterative retrieval, ResAdapt dynamically allocates per-frame visual budgets *before* encoding. **(b)** Performance of Qwen2.5-VL-7B using 32 frames at ∼10% visual token retention. ResAdapt establishes Pareto frontier, yielding the most significant gains on reasoning benchmarks.

seamlessly integrating with modern inference engines (Dao, 2024; Kwon et al., 2023). Unlike prior slow–fast pipelines (Yang et al., 2025a; Zhang et al., 2026) that rely on query-agnostic heuristics, ResAdapt learns a query-aware allocation policy directly from task rewards.

Optimizing this allocator presents significant challenges: the allocation space is continuous, the budget operator is non-differentiable, and naive accuracy–cost trade-offs often collapse into degenerate, low-budget policies. To address these issues, we introduce **Cost-Aware Policy Optimization (CAPO)**, which transforms sparse rollout feedback into a stable, asymmetric learning signal. Coupled with a temporal-similarity regularizer that penalizes redundant high-resolution allocations across adjacent frames, ResAdapt emerges as a trainable, content-aware policy rather than a handcrafted heuristic.

Empirically, ResAdapt establishes a new efficiency–accuracy Pareto frontier for video reasoning. As shown in Figure 1b, at an aggressive 90% token reduction, ResAdapt consistently outperforms existing token economy methods across comprehensive benchmarks. By eliminating spatial redundancy, the saved compute can be seamlessly reinvested to expand temporal coverage—processing 16× more frames under the same budget to unlock substantial performance gains. Furthermore, the learned policy exhibits *active perception*, autonomously allocating high resolutions to information-dense frames in a single forward pass without requiring explicit saliency supervision. Our main contributions are as follows:

1. We introduce **ResAdapt**, an *input-side adaptation* framework that formulates dynamic per-frame visual budgeting as a contextual bandit problem, fully preserving the native architecture and hardware optimizations of MLLMs.
2. We propose **CAPO** with a temporal similarity regularizer, providing a stable, asymmetric learning signal to jointly optimize accuracy and cost without hand-crafted heuristics.
3. Through extensive experiments and ablations, we show that ResAdapt achieves better efficiency–accuracy Pareto frontier across video QA and temporal grounding tasks.

2 Background and Problem Formulation

2.1 Preliminaries

Given a text query $\boldsymbol{q}$ and a video $\mathcal{V} = \{f_t\}_{t=1}^{T}$, a backbone policy π_ϕ encodes frames at fixed fidelity and autoregressively generates a rollout $\boldsymbol{y} = (y_1, \ldots, y_L)$:

$$\pi_\phi(\boldsymbol{y} \mid \boldsymbol{q}, \mathcal{V}) = \prod_{j=1}^{L} \pi_\phi(y_j \mid y_{<j}, \boldsymbol{q}, \mathcal{V}). \tag{1}$$

Visual cost scales with total pixel volume, whereas answer-critical evidence is temporally sparse, causing inefficiency.

To control pre-encoding cost, we introduce an Allocator policy π_θ that emits a per-frame allocation vector $\boldsymbol{s} = (s_1, \ldots, s_T)$ conditioned on the input $\boldsymbol{x} = (\boldsymbol{q}, \mathcal{V})$:

$$\boldsymbol{s} \sim \pi_\theta(\cdot \mid \boldsymbol{x}), \qquad s_t \in [s_{\min}, s_{\max}], \tag{2}$$

and applies a *visual budget operator* $\mathcal{O}$ to each frame: $\tilde{f}_t = \mathcal{O}(f_t, s_t)$. The backbone then generates from the transformed input $\tilde{\boldsymbol{x}} = (\boldsymbol{q}, \{\tilde{f}_t\}_{t=1}^T)$:

$$\pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}}) = \prod_{j=1}^{L} \pi_\phi(y_j \mid y_{<j}, \tilde{\boldsymbol{x}}). \tag{3}$$

The framework is operator-agnostic, supporting resizing, frame selection, or other pre-encoding budget controls.

2.2 Problem Formulation

Because the Allocator acts once before decoding, the outer problem is a *Contextual Bandit* with continuous allocation vector $\boldsymbol{s} \in [s_{\min}, s_{\max}]^T$. For joint training, the two-stage policy factorizes as $p_{\theta,\phi}(\boldsymbol{s}, \boldsymbol{y} \mid \boldsymbol{x}) = \pi_\theta(\boldsymbol{s} \mid \boldsymbol{x})\, \pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}})$.

With $C(\boldsymbol{s})$ denoting visual cost and $Q(\boldsymbol{x}, \boldsymbol{y})$ the response quality, Lagrangian relaxation of a budgeted objective yields the unconstrained utility:

$$\max_{\theta,\phi} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\, \boldsymbol{s}\sim\pi_\theta,\, \boldsymbol{y}\sim\pi_\phi} \left[Q(\boldsymbol{x}, \boldsymbol{y}) - \lambda\, C(\boldsymbol{s}) \right], \tag{4}$$

for trade-off coefficient $\lambda \geq 0$. Section 3 instantiates this objective with an Input-side adaptation policy, CAPO, and temporal regularization. Detailed derivations are in Appendix D.

3 Method

Figure 2 summarizes the Input-side adaptation framework. At inference, the Allocator predicts one allocation per frame and applies a pre-encoding operator before the video reaches the backbone in a single pass. In the experimental instantiation studied here, $\mathcal{O}$ is bilinear resizing, so the allocation becomes a resize factor s_t and $\tilde{f}_t = \mathcal{R}(f_t, s_t)$. At training, rollout feedback updates the Allocator and, optionally, the backbone.

3.1 Joint RL Optimization Framework

The objective in Eq. (4) motivates a one-step expected-reward formulation. For a fixed context $\boldsymbol{x}$, the ideal rollout reward is $R^{\text{ideal}}_{\boldsymbol{s},\boldsymbol{y}} = Q(\boldsymbol{x}, \boldsymbol{y}) - \lambda\, C(\boldsymbol{s})$, yielding the target:

$$\max_{\theta,\phi}\ \mathcal{J}(\theta, \phi) = \mathbb{E}_{\pi_\theta(\boldsymbol{s}|\boldsymbol{x})} \left[\mathbb{E}_{\pi_\phi(\boldsymbol{y}|\tilde{\boldsymbol{x}})} \left[R^{\text{ideal}}_{\boldsymbol{s},\boldsymbol{y}} \right] \right]. \tag{5}$$

Direct optimization of Eq. (5) via score-function estimators (detailed in Appendix D) is brittle due to three challenges: 1) **Policy parameterization**: π_θ must efficiently predict continuous allocations; 2) **Credit assignment**: the raw reward $R^{\text{ideal}}_{\boldsymbol{s},\boldsymbol{y}}$ is high-variance and often causes budget collapse; 3) **Temporal structure**: rollout-level rewards lack explicit signals to prevent redundant allocations across adjacent frames. We address these sequentially below.

3.2 Allocator Architecture

To resolve the parameterization bottleneck, the Allocator processes coarse visual features with negligible overhead (see Appendix C). Each frame f_t is encoded by a frozen lightweight visual encoder, while the query is encoded separately. A shallow decoder alternates temporal self-attention with gated cross-attention to the query, producing hidden states $\{h_t\}_{t=1}^T$.

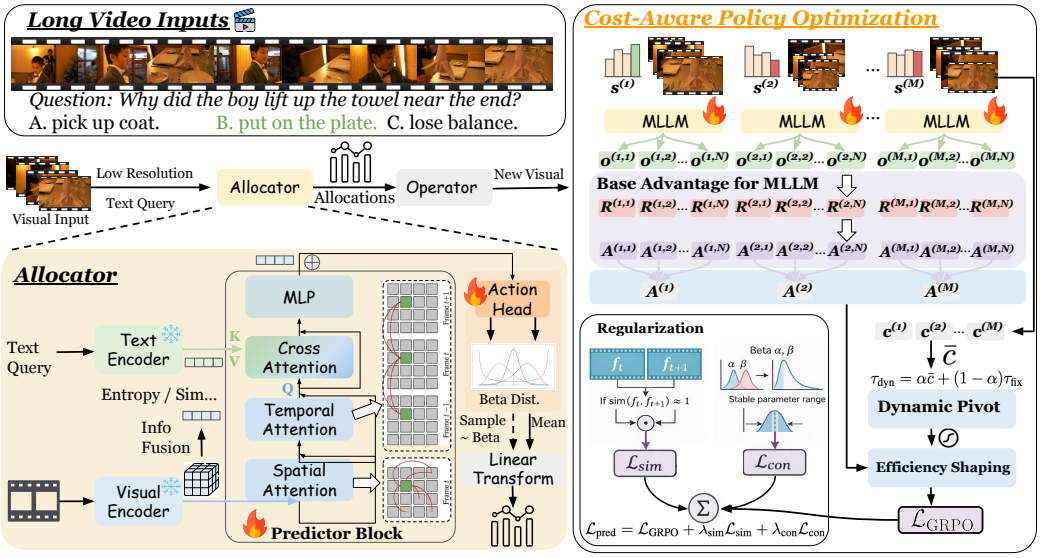


Figure 2: ResAdapt framework. (a) At inference, the Allocator π_θ maps coarse visual features and the query to latent actions $a_t \sim \mathrm{Beta}(\alpha_t, \beta_t)$, which parameterize per-frame input allocations. In the resize instantiation used in our experiments, these allocations are realized as scales $s_t \in [s_{\min}, s_{\max}]$, and the resized frames are processed by the MLLM in a single call. **(b)** During training, CAPO reshapes group-relative advantages with a dynamic cost pivot τ_{dyn}, while temporal-similarity regularization suppresses redundant high-budget allocation on adjacent similar frames.

To preserve exploration, we parameterize latent actions with Beta distributions, mapping their bounded support to $[s_{\min}, s_{\max}]$:

$$a_t \sim \mathrm{Beta}(\alpha_t, \beta_t), \qquad s_t = s_{\min} + a_t\,(s_{\max} - s_{\min}). \tag{6}$$

Conditioned on $\{h_t\}$, the latent policy factorizes as $\log q_\theta(\boldsymbol{a} \mid \boldsymbol{x}) = \sum_{t=1}^{T} \log \mathrm{Beta}(a_t; \alpha_t, \beta_t)$, inducing the continuous allocation policy $\pi_\theta(\boldsymbol{s} \mid \boldsymbol{x})$.

3.3 Cost-Aware Policy Optimization (CAPO)

A flat penalty on visual cost $C(\boldsymbol{s})$ often collapses the policy toward uniformly tiny budgets. To solve the credit assignment bottleneck, CAPO replaces the raw penalty with a shaped surrogate learning signal.

Compute metric. For the resize operator, frame f_t resized by s_t produces $n_t(s_t) \propto \lceil s_t H_t / P \rceil \lceil s_t W_t / P \rceil$ tokens. Physical compute is measured by the *token retention ratio* $\rho(\boldsymbol{s}) = \sum n_t(s_t) / \sum n_t(1)$, well-approximated by the average quadratic scale. To reduce variance during optimization, we use the smoother linear proxy $c_m = (\bar{s} - s_{\min})/(s_{\max} - s_{\min})$ for allocation m.

Notation bridge. During training, $R^{\mathrm{task}}_{m,n}$ denotes the task reward of rollout (m, n), $A^{\mathrm{base}}_{m,n}$ the GRPO group-normalized advantage, and $A_{m,n}$ the final CAPO-shaped advantage. Appendix D details their relation to the ideal reward. We also define $u_{m,n} \in \{0, 1\}$ as a binary correctness indicator (e.g., exact-match for QA, thresholded score for generation).

Dynamic cost pivot. A fixed threshold does not track the evolving policy, whereas a purely group-dependent statistic is too noisy. CAPO therefore interpolates between a fixed target τ_{fix} and the prompt-local mean $\bar{c}_{\mathrm{group}} = \frac{1}{M}\sum_{m=1}^{M} c_m$:

$$\tau_{\mathrm{dyn}} = \kappa_{\mathrm{mix}}\,\bar{c}_{\mathrm{group}} + (1 - \kappa_{\mathrm{mix}})\,\tau_{\mathrm{fix}}, \tag{7}$$

where $\kappa_{\mathrm{mix}} \in [0, 1]$ controls adaptivity.

Asymmetric shaping. With τ_{dyn} as pivot, we apply a correctness-dependent bonus or penalty with $\lambda_- > \lambda_+ > 0$:

$$S_{m,n} = \begin{cases} \lambda_+ \, \sigma\left(\dfrac{\tau_{\text{dyn}} - c_m}{\tau_{\text{s}}}\right) & \text{if } u_{m,n} = 1, \\ -\lambda_- \, \sigma\left(\dfrac{c_m - \tau_{\text{dyn}}}{\tau_{\text{s}}}\right) & \text{if } u_{m,n} = 0, \end{cases} \tag{8}$$

Efficient correct rollouts receive a moderate bonus, whereas costly incorrect rollouts receive a stronger penalty. The sigmoid temperature τ_{s} smooths the transition around the pivot.

Final CAPO advantage. Let $\tilde{A}_{m,n} = A^{\text{base}}_{m,n} + \lambda_{\text{capo}} \, S_{m,n} - \gamma \, c_m$. The final advantage is

$$A_{m,n} = \begin{cases} \max\left(\tilde{A}_{m,n}, \varepsilon_+\right) & \text{if } u_{m,n} = 1, \\ \tilde{A}_{m,n} & \text{if } u_{m,n} = 0, \end{cases} \tag{9}$$

where $\lambda_{\text{capo}} > 0$ scales CAPO shaping, $\gamma \geq 0$ applies a residual global cost penalty, and the floor $\varepsilon_+ > 0$ ensures that correct low-cost rollouts retain a positive learning signal. The dominant anti-collapse term is the pivoted asymmetric shaping in $S_{m,n}$.

3.4 Regularization and Training Objective

While CAPO balances global task reward and efficiency, it acts strictly at the rollout level. As observed empirically (Section 4.4), this can lead to the temporal structure bottleneck: the policy may allocate uniformly low-variance scales across highly redundant frames. To enforce localized selectivity, we augment the optimizer with structural regularization.

Temporal similarity loss ($\mathcal{L}_{\text{sim}}$). CAPO optimizes global quality–efficiency but does not penalize redundant allocations on near-duplicate frames. Reusing the coarse features $\boldsymbol{f}_t$ from Sec. 3.2, we penalize similar adjacent pairs:

$$\mathcal{L}_{\text{sim}} = \frac{1}{T-1} \sum_{t=1}^{T-1} w_t \cdot \max(0, \log s_t + \log s_{t+1} + \eta_{\text{sim}}), \tag{10}$$

where $w_t = \sigma((\cos(\boldsymbol{f}_t, \boldsymbol{f}_{t+1}) - \tau_{\text{sim}})/\gamma_{\text{sim}})$ activates when frames exceed a similarity threshold τ_{sim}. No penalty is incurred when $s_t s_{t+1} \leq e^{-\eta_{\text{sim}}}$.

Concentration loss ($\mathcal{L}_{\text{con}}$). To prevent Beta distributions from collapsing to near-deterministic spikes, we softly cap total concentration at $\kappa_{\max} > 0$: $\mathcal{L}_{\text{con}} = \frac{1}{T} \sum_{t=1}^{T} \max(0, \alpha_t + \beta_t - \kappa_{\max})$. Together, $\mathcal{L}_{\text{sim}}$ encourages differentiated allocation, while $\mathcal{L}_{\text{con}}$ preserves exploration.

Practical training objective. We optimize both policies in a single GRPO-style loop (Zheng et al., 2025a; Yu et al., 2025). For each prompt $\boldsymbol{x}$, the Allocator samples M allocation trajectories $\boldsymbol{s}_{1:M}$; each transformed input $\tilde{\boldsymbol{x}}^{(m)}$ produces N response rollouts. CAPO computes rollout advantages $A_{m,n}$, serving as the shared learning signal.

Allocator objective. We aggregate rollout advantages per allocation, $A^{\text{CAPO}}_m = \frac{1}{N} \sum_n A_{m,n}$, and optimize the per-frame clipped surrogate objective:

$$\mathcal{L}_\theta = -\frac{1}{MT} \sum_{m=1}^{M} \sum_{t=1}^{T} \min\Big(r^{(m)}_{\theta,t} \, A^{\text{CAPO}}_m, \text{clip}\big(r^{(m)}_{\theta,t}, 1-\varepsilon, 1+\varepsilon\big) \, A^{\text{CAPO}}_m\Big), \tag{11}$$

where $r^{(m)}_{\theta,t} = q_\theta(a^{(m)}_t \mid \boldsymbol{x}) / q_{\theta_{\text{old}}}(a^{(m)}_t \mid \boldsymbol{x})$. The full loss is $\mathcal{L}_{\text{alloc}} = \mathcal{L}_\theta + \lambda_{\text{sim}} \mathcal{L}_{\text{sim}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}$.

Backbone update. Conditioned on the sampled allocations, the backbone is optionally updated (for ResAdapt-RL) with the token-level clipped surrogate objective using advantages $A_{m,n}$ and ratios $r^{(m,n)}_{\phi,j}$ (see Appendix D). In practice, $\mathcal{L}_{\text{alloc}}$ and $\mathcal{L}_\phi$ are optimized alternately.

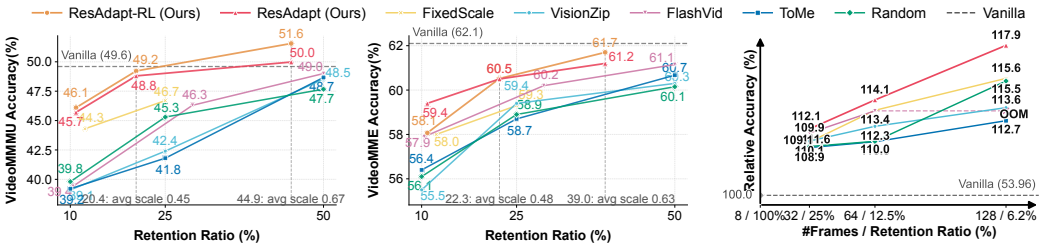


Figure 3: Efficiency–accuracy trade-offs and temporal reallocation. (a,b) VideoMMMU and VideoMME versus visual-token retention ratio R. ResAdapt is on or near the Pareto frontier, with the clearest advantage on reasoning-heavy settings at low retention. **(c)** Relative gain from trading spatial resolution for temporal coverage under a fixed 8-frame-equivalent budget.

4 Experiments

4.1 Setup

Implementation. The Allocator π_θ adopts the lightweight SmolVLM architecture (Marafioti et al., 2025) for high-throughput prediction. We instantiate input-side allocation via continuous per-frame resizing, providing the smooth action space required by our optimizer, with frame dropping naturally emerging at the zero-budget limit. The Allocator is trained on Qwen2.5-VL-7B-Instruct (Bai et al., 2025b) and additionally evaluated for zero-shot transfer on Qwen3-VL-8B-Instruct (Bai et al., 2025a). We report two paradigms: **ResAdapt-RL**, which jointly updates the Allocator and the backbone, and **ResAdapt**, which applies the trained Allocator to a frozen backbone to assess plug-and-play generalization.

Baselines. We compare against **heuristic methods** (Random Drop, FixedScale), **model-side compression** (ToMe (Bolya et al., 2022), FlashVid (Fan et al., 2026), VisionZip (Yang et al., 2025c)), and **inference augmentation** (VideoAuto-R1 (Liu et al., 2026)). Efficiency is measured by the visual-token retention ratio R. For inference-augmented models, R accounts strictly for visual encoder tokens. Since several baselines only support discrete operating points, we align budgets approximately and emphasize Pareto frontier trade-offs.

Benchmarks. Evaluations span *video QA* (VideoMME (Fu et al., 2025a), MMVU (Zhao et al., 2025b), LongVideoBench (Wu et al., 2024), MLVU (Zhou et al., 2025), VideoMMMU (Hu et al., 2025), LVBench (Wang et al., 2025b)), *temporal grounding* (Charades-STA (Gao et al., 2017), ActivityNet (Fabian et al., 2015), NExT-GQA (Xiao et al., 2024)), and *image understanding* (MathVista (Lu et al., 2023), MMMU (Yue et al., 2024), OCRBench (Liu et al., 2024), ChartQA (Masry et al., 2022), AI2D (Kembhavi et al., 2016), TextVQA (Singh et al., 2019)). Unless otherwise stated, analyses default to Qwen2.5-VL-7B with 32 frames. All evaluations utilize lmms-eval (Zhang et al., 2024a). Full hyperparameters, prompts, reward formulations, and detailed configurations are deferred to Appendix B.

4.2 Main Results

We evaluate ResAdapt across video QA, temporal grounding, and zero-shot image transfer (Table 1). Our analysis focuses on the extreme low-budget regime ($\sim$10% token retention), where standard compression typically collapses, demonstrating the robustness of input-side adaptation.

Video QA: Preserving Reasoning under Extreme Compression. Post-encoding token reduction methods (e.g., ToMe, VisionZip) indiscriminately discard semantic features, causing precipitous drops in reasoning accuracy under high compression. In contrast, ResAdapt successfully isolates sparse but critical evidence. On the reasoning-heavy VideoMMMU benchmark, it maintains a substantial margin (up to +6.5 points) over baselines at matched budgets. This resilience highlights the core advantage of *pre-encoding* allocation: by dynami-

Table 1: Evaluation Results on Video QA Benchmarks. Retention ratio R reflects visual token count; *Reasoning* (✓/✗) indicates chain-of-thought use; **bold** marks the best result. ResAdapt yields larger gains on the reasoning benchmark than on the perception benchmarks.

Backbone	Method	Retention Ratio R	Reasoning	Video Perception				Video Reasoning		Grounding (mIoU)		NExT-GQA	
				VideoMME	LongVideoBench	MMVU	MLVU	VideoMMMU	LVBench	Charades	ActivityNet	Acc	mIoU
Qwen2.5-VL-7B	*32 Frames*												
	Vanilla	100%	✗	62.0	58.9	52.7	63.1	49.6	38.6	47.3	22.6	78.9	28.0
	Random Drop	25.0%	✗	56.9	57.8	49.6	58.3	45.3	36.7	25.7	11.7	77.5	16.6
	ToMe (Bolya et al., 2022)	25.0%	✗	58.7	58.0	51.0	58.7	41.8	37.7	26.0	12.1	77.8	16.3
	VisionZip (Yang et al., 2025c)	25.0%	✗	59.4	57.1	49.8	57.9	42.4	36.5	-	-	-	-
	FlashVid (Fan et al., 2026)	29.3%	✗	60.2	**58.6**	51.1	59.2	46.3	36.9	26.6	12.0	**78.1**	16.5
	FixedScale	25.0%	✗	60.0	56.8	51.2	59.8	46.7	37.3	24.9	14.1	77.7	12.3
	ResAdapt (Ours)	23.8%	✗	**60.3**	58.2	**51.9**	**60.1**	**48.8**	**37.9**	**35.6**	**15.3**	76.6	**23.2**
	Random Drop	10.0%	✗	56.1	55.6	47.1	56.5	39.8	35.2	24.6	11.1	76.3	15.4
	ToMe (Bolya et al., 2022)	10.0%	✗	56.4	55.2	48.9	58.0	39.2	33.6	27.4	12.2	77.3	15.7
	VisionZip (Yang et al., 2025c)	10.0%	✗	55.5	54.5	47.6	57.3	39.1	35.3	-	-	-	-
	FlashVid (Fan et al., 2026)	10.4%	✗	57.9	**56.8**	47.9	57.7	39.4	**36.5**	25.1	11.8	**77.4**	16.1
	FixedScale	12.3%	✗	58.0	55.1	47.7	57.5	44.3	35.4	**32.0**	**13.3**	76.1	13.7
	ResAdapt (Ours)	11.4%	✗	**59.4**	55.4	**49.2**	**58.4**	**45.7**	35.9	27.2	12.5	74.3	**20.4**
	VideoAuto-R1 (Liu et al., 2026)	100%	✓	63.2	58.9	55.0	60.0	53.6	41.5	41.5	34.4	73.6	33.8
	+ **ResAdapt (Ours)**	23.8%	✓	60.4	57.1	53.2	61.1	51.2	38.7	49.1	44.7	79.3	35.3
	+ **ResAdapt (Ours)**	11.4%	✓	59.3	56.3	51.8	59.3	49.1	36.7	30.0	24.4	74.7	24.7
	128 Frames												
	Vanilla	100%	✗	65.3	60.3	53.1	66.5	47.9	42.0	52.8	34.4	79.8	29.9
	Random Drop	25.0%	✗	64.9	61.2	50.8	64.8	48.1	41.3	20.7	18.8	80.3	10.7
	ToMe (Bolya et al., 2022)	25.0%	✗	65.1	**61.6**	51.9	63.1	46.6	**42.1**	20.7	19.1	**80.3**	10.9
	VisionZip (Yang et al., 2025c)	25.0%	✗	64.8	61.3	51.1	64.5	47.3	41.5	-	-	-	-
	ResAdapt (Ours)	22.9%	✗	**65.6**	60.2	**52.8**	**65.9**	**51.1**	**42.1**	**42.0**	**24.3**	78.1	**27.2**
	Random Drop	10.0%	✗	63.0	59.0	45.8	63.4	46.7	38.0	24.7	17.0	79.4	12.8
	ToMe (Bolya et al., 2022)	10.0%	✗	60.6	56.3	44.2	63.5	41.8	39.5	17.9	16.4	79.1	11.1
	VisionZip (Yang et al., 2025c)	10.0%	✗	61.8	56.1	44.8	63.2	42.1	38.2	-	-	-	-
	FixedScale	12.3%	✗	**64.1**	**60.9**	**49.6**	**64.5**	46.9	**40.3**	22.7	**18.3**	77.9	11.3
	ResAdapt (Ours)	11.1%	✗	63.8	58.6	49.0	64.3	**49.2**	39.9	**28.9**	17.2	76.2	**23.9**
	VideoAuto-R1 (Liu et al., 2026)	100%	✓	64.7	59.1	56.7	65.1	52.2	41.2	28.9	33.5	68.0	31.0
	+ **ResAdapt (Ours)**	23.8%	✓	66.2	60.2	53.5	66.0	52.6	41.8	49.1	44.7	79.3	35.3
	+ **ResAdapt (Ours)**	11.4%	✓	64.7	57.8	52.4	64.6	51.3	39.5	34.2	35.7	76.6	29.4
Qwen3-VL-8B	*32 Frames*												
	Vanilla	100%	✗	65.0	58.6	57.5	64.0	60.8	40.2	46.4	31.8	78.7	34.2
	Random Drop	25.0%	✗	61.3	58.4	57.1	60.2	53.4	37.8	12.1	10.0	77.2	15.6
	ToMe (Bolya et al., 2022)	25.0%	✗	62.4	57.4	56.0	60.8	49.1	36.4	43.1	32.6	77.1	31.7
	VisionZip (Yang et al., 2025c)	25.0%	✗	61.8	57.2	54.4	60.6	51.5	37.3	-	-	-	-
	FlashVid (Fan et al., 2026)	30.0%	✗	**63.9**	**59.0**	54.8	**61.9**	55.1	**38.5**	**47.7**	**36.8**	**77.8**	**33.9**
	ResAdapt (Ours)	23.8%	✗	62.6	57.5	55.3	61.0	**58.4**	**38.5**	39.9	28.5	75.1	30.2
	Random Drop	10.0%	✗	58.8	54.7	53.2	56.6	47.1	35.5	4.4	5.0	74.3	11.3
	ToMe (Bolya et al., 2022)	10.0%	✗	59.2	55.5	53.1	58.5	42.7	35.8	41.8	34.1	**79.2**	**34.0**
	VisionZip (Yang et al., 2025c)	10.0%	✗	59.9	55.4	53.7	58.8	45.8	35.4	-	-	-	-
	FlashVid (Fan et al., 2026)	12.2%	✗	**61.0**	**57.1**	**54.8**	59.1	47.8	37.1	**44.6**	**35.2**	75.6	31.8
	FixedScale	12.3%	✗	60.8	54.9	53.8	58.4	52.6	37.1	37.9	28.4	74.2	29.9
	ResAdapt (Ours)	11.4%	✗	60.7	56.6	54.6	**59.6**	**56.1**	**37.3**	33.6	27.2	71.8	28.2
	128 Frames												
	Vanilla	100%	✗	69.4	64.3	58.5	72.7	63.0	45.7	45.6	33.9	81.1	36.6
	Random Drop	25.0%	✗	67.2	61.3	**56.8**	67.4	55.3	42.4	27.4	26.3	79.3	22.4
	ToMe (Bolya et al., 2022)	25.0%	✗	67.2	**62.0**	55.9	70.4	53.5	43.1	-	-	-	-
	VisionZip (Yang et al., 2025c)	25.0%	✗	67.1	61.3	55.7	69.2	56.8	41.2	-	-	-	-
	ResAdapt (Ours)	22.9%	✗	**67.4**	61.9	56.3	**70.8**	**59.6**	**43.3**	**39.8**	**30.0**	76.8	**33.3**
	Random Drop	10.0%	✗	64.1	58.3	**55.4**	62.4	55.5	38.8	21.9	24.8	76.9	19.9
	ToMe (Bolya et al., 2022)	10.0%	✗	64.7	58.6	55.1	67.3	46.3	40.5	**38.1**	**31.4**	**77.4**	31.5
	VisionZip (Yang et al., 2025c)	10.0%	✗	64.2	59.1	54.2	66.8	47.6	39.4	-	-	-	-
	FixedScale	12.3%	✗	66.7	59.5	54.4	67.7	56.3	41.7	38.1	29.5	75.4	32.6
	ResAdapt (Ours)	11.1%	✗	**66.8**	**60.2**	**55.4**	**69.4**	**58.2**	**42.6**	33.7	28.4	73.2	**43.9**

cally assigning pixels based on content, ResAdapt protects high-frequency visual details exactly where complex reasoning demands it.

Trading spatial redundancy for temporal context. The primary benefit of this efficiency is the ability to reinvest saved spatial budgets into broader temporal coverage. By extending the context window from 32 to 128 frames using the budget freed by ResAdapt, we surpass the dense 128-frame baseline's performance while remaining computationally cheaper (Figure 3). This confirms a critical insight: reasoning models benefit significantly more from a longer, adaptively compressed timeline than a shorter, fully dense one.

Temporal Grounding: Anchoring and Denoising. Grounding tasks are inherently compression-sensitive as they demand precise spatio-temporal anchors. While naive frame dropping destroys these anchors and severely degrades performance (Appendix E.6), ResAdapt mitigates this by heavily compressing static segments while explicitly preserving the full resolution of critical transition events.

Synergy with long-context reasoning. ResAdapt exhibits a powerful synergy with long-context models. Scaling to 128 frames often causes dense models to struggle with distraction, paradoxically degrading grounding accuracy. Integrating ResAdapt not only recovers this loss but yields a net improvement over the short-context baseline. This reveals an emergent *denoising* effect: by suppressing irrelevant visual noise, ResAdapt actively guides the model's attention toward valid temporal anchors, achieving higher accuracy with under 10% of the original token cost.

Exploratory Image Transfer. Zero-shot transfer to static images confirms our policy does not overfit to video dynamics, though text-dense images naturally require a minimum spatial resolution threshold (Appendix E.7, Table 7).

Table 2: Latency breakdown (ms, ↓) on Qwen2.5-VL-7B with single-GPU Allocator and 4-GPU vLLM engine. Averaged over 200 runs after 5 warm-up; E2E latency = Scale Time + Gen. Time.

Method	#Frames	Retention Ratio R	Scale						Inference			Total	
			TFLOPs	Text Enc.	Visual Enc.	Scale Pred.	Scale Apply	Scale Time	TFLOPs	TTFT	Gen. Time	TFLOPs	E2E Time
Vanilla	16	100%	–	–	–	–	–	–	111.4	378.9	527.9	111.4	527.9
ResAdapt	16	76.3%	1.5	19.8	94.1	85.6	6.3	205.8	77.2 (↓30.7%)	272.5 (↓28.1%)	370.7 (↓29.8%)	80.1 (↓28.1%)	576.5 (↑9.2%)
ResAdapt	16	52.8%	1.5	19.9	102.9	94.5	8.4	225.7	51.5 (↓53.8%)	261.5 (↓31.0%)	313.1 (↓40.7%)	54.4 (↓51.2%)	538.8 (↑2.1%)
ResAdapt	16	28.9%	1.5	20.4	103.4	92.2	9.0	225.0	31.0 (↓72.2%)	227.2 (↓40.0%)	237.9 (↓54.9%)	33.9 (↓69.6%)	462.9 (↓12.3%)
Vanilla	32	100%	–	–	–	–	–	–	222.5	723.3	881.9	222.5	881.9
ResAdapt	32	74.4%	2.9	19.9	204.1	97.4	14.4	335.9	153.9 (↓30.8%)	589.4 (↓18.5%)	627.6 (↓28.8%)	159.7 (↓28.2%)	963.5 (↑9.2%)
ResAdapt	32	51.5%	2.9	20.0	193.2	92.0	16.2	321.4	102.4 (↓54.0%)	505.0 (↓30.2%)	467.1 (↓47.0%)	108.2 (↓51.4%)	788.5 (↓10.6%)
ResAdapt	32	28.2%	2.9	20.3	190.4	90.3	17.3	318.3	61.4 (↓72.4%)	451.8 (↓37.5%)	332.6 (↓62.3%)	67.2 (↓69.8%)	650.9 (↓26.2%)
Vanilla	64	100%	–	–	–	–	–	–	444.6	1457.5	2059.6	444.6	2059.6
ResAdapt	64	73.2%	5.8	19.8	389.5	95.8	26.4	531.5	307.3 (↓30.9%)	1093.1 (↓25.0%)	1327.0 (↓35.6%)	318.9 (↓28.3%)	1858.5 (↓9.8%)
ResAdapt	64	50.7%	5.8	20.1	382.1	94.9	29.9	527.0	204.3 (↓54.0%)	991.8 (↓31.9%)	740.5 (↓64.0%)	215.9 (↓51.4%)	1267.5 (↓38.5%)
ResAdapt	64	27.8%	5.8	20.0	371.6	90.2	34.8	516.6	122.2 (↓72.5%)	899.2 (↓38.3%)	511.4 (↓75.2%)	133.8 (↓69.9%)	1028.0 (↓50.1%)
Vanilla	128	100%	–	–	–	–	–	–	888.9	2936.3	4877.0	888.9	4877.0
ResAdapt	128	74.2%	11.6	20.1	766.3	95.0	53.1	934.5	614.1 (↓30.9%)	2286.6 (↓22.1%)	2323.6 (↓52.4%)	637.3 (↓28.3%)	3258.1 (↓33.2%)
ResAdapt	128	51.4%	11.6	20.2	755.3	93.8	59.4	928.7	408.0 (↓54.1%)	2071.0 (↓29.5%)	1496.0 (↓69.3%)	431.2 (↓51.5%)	2424.7 (↓50.3%)
ResAdapt	128	28.2%	11.6	20.4	734.5	92.0	68.6	915.5	243.9 (↓72.6%)	1766.7 (↓39.8%)	1061.8 (↓78.2%)	267.1 (↓70.0%)	1977.3 (↓59.5%)

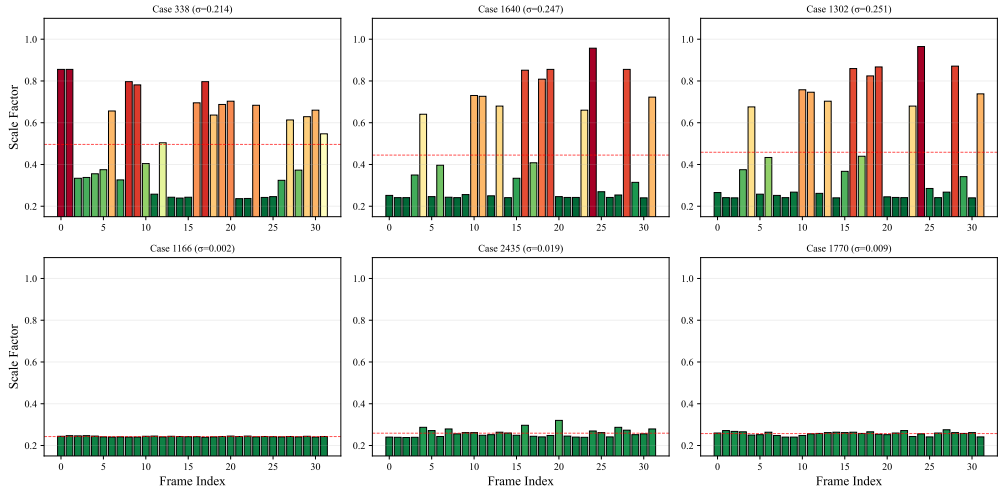


Figure 4: Emergent active perception. Per-frame scale s_t over frame index for VideoMME, grouped by intra-video scale diversity σ. High-diversity videos show localized scale spikes on scene changes or text overlays, while low-diversity videos remain uniformly compressed.

4.3 Runtime Overhead

We measure pipeline latency to determine when the Allocator's upstream cost is amortized by downstream token savings. Table 2 reports latency using a dedicated single-GPU Allocator coupled with a 4-GPU vLLM engine. By adjusting the maximum allowed scale, ResAdapt acts as a continuous accuracy–speed dial, spanning conservative (R≈74%) to aggressive (R≈28%) compression.

The break-even point depends heavily on context length. At conservative compression, end-to-end (E2E) wall-clock savings emerge at ≥64 frames (**−9.8%**). Under aggressive compression, E2E savings begin as early as 16 frames (**−12.3%**) and accelerate to **−59.5%** at 128 frames, alongside a **78%** reduction in generation time. This scaling trajectory reflects the quadratic complexity of attention: as sequence length grows, the downstream backbone savings compound much faster than the linear Allocator overhead, making ResAdapt exceptionally well-suited for long-context generation.

4.4 Analysis and Ablation

Emergent Active Perception. ResAdapt learns a highly sparse temporal allocation policy. As shown in Figure 4, the Allocator concentrates high resolution on brief, informative events

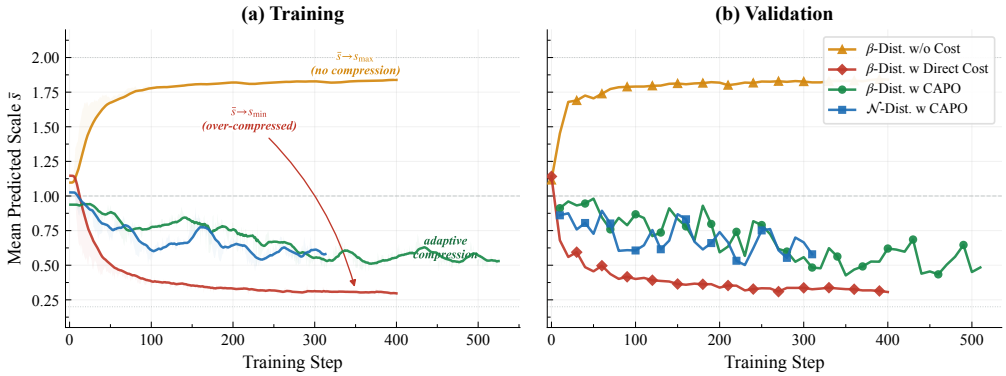


Figure 5: Reward-design ablation. Mean predicted scale $\bar{s}$ during training and validation. Direct cost penalties collapse to the minimum scale, whereas CAPO variants converge to stable intermediate operating points.

(e.g., text overlays, scene transitions) while heavily compressing redundant segments. This behavior goes beyond a trivial positional prior: high-scale allocations emerge as localized, content-dependent bursts. This confirms that the model selectively expends its pixel budget precisely where reasoning demands it (see Appendix E.1.1 for detailed global allocation statistics).

CAPO Ablation. We evaluate the necessity of CAPO's asymmetric cost shaping (Appendix E.2.2). While exact parametric forms (β- vs. $\mathcal{N}$-CAPO) yield marginal differences, CAPO consistently outperforms direct cost penalties. As illustrated dynamically in Figure 5, direct penalties cause the policy to collapse to the minimum scale, whereas CAPO stabilizes at an intermediate operating point, effectively rewarding selective allocation.

Temporal Regularization. While CAPO shapes the cost objective, it does not prevent uniform scaling across visually redundant frames. Removing the temporal similarity loss $\mathcal{L}_{\text{sim}}$ causes the policy to collapse into near-constant scaling (Appendix E.2.1). Reintroducing $\mathcal{L}_{\text{sim}}$ restores sharp frame-level differentiation, demonstrating that temporal regularization is essential for breaking symmetry and enforcing sparse allocation.

Further Analysis. Although trained exclusively for continuous resizing, ResAdapt's predicted scales generalize zero-shot to discrete frame selection, outperforming uniform sampling at lower budgets (Appendix E.3). Furthermore, while ResAdapt effectively redistributes spatial budgets, we provide a qualitative analysis of its boundary cases—such as missing extremely brief, simple cues—in Appendix E.5.

5 Conclusion

We introduce ResAdapt, an input-side adaptation framework that shifts visual efficiency optimization from post-encoding token compression to pre-encoding budget control. By employing a lightweight Allocator trained via Cost-Aware Policy Optimization (CAPO) and temporal-similarity regularization, ResAdapt learns a sparse, content-dependent policy to dynamically resize frames before visual encoding. This approach effectively preserves critical high-frequency details while heavily compressing redundant segments. Extensive evaluations demonstrate that ResAdapt establishes a new Pareto frontier in low-budget video QA and significantly enhances reasoning-augmented long-context grounding by reinvesting spatial savings into extended temporal coverage. While the current open-loop design precludes iterative recovery of missed evidence, our findings highlight pre-encoding allocation as a highly promising and generalizable paradigm for efficient long-context multimodal reasoning.

References

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9392–9401, 2025.

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, Sibo 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, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, et al. Scaling rl to long videos. *arXiv preprint arXiv:2507.07966*, 2025.

Zeyuan Chen, Kai Zhang, Zhuowen Tu, and Yuanjun Xiong. Soft tail-dropping for adaptive visual tokenization. *arXiv preprint arXiv:2601.14246*, 2026.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024.

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*, pp. 961–970, 2015.

Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. Flashvid: Efficient video large language models via training-free tree-based spatiotemporal token merging. *arXiv preprint arXiv:2602.08024*, 2026.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24108–24118, 2025a.

Shenghao Fu, Qize Yang, Yuan-Ming Li, Xihan Wei, Xiaohua Xie, and Wei-Shi Zheng. Love-r1: Advancing long video understanding with an adaptive zoom-in mechanism via multi-step reasoning. *arXiv preprint arXiv:2509.24786*, 2025b.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. Framefusion: Combining similarity and importance for video token reduction on large vision language models. *arXiv preprint arXiv:2501.01986*, 2024.

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*, pp. 5267–5275, 2017.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

Weiyu Guo, Ziyang Chen, Shaoguang Wang, Jianxiang He, Yijie Xu, Jinhui Ye, Ying Sun, and Hui Xiong. Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. *arXiv preprint arXiv:2503.13139*, 2025b.

Zefeng He, Xiaoye Qu, Yafu Li, Siyuan Huang, Daizong Liu, and Yu Cheng. Framethinker: Learning to think with long videos via multi-turn frame spotlighting. *arXiv preprint arXiv:2509.24304*, 2025.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*, 2025.

Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, 2025.

Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. Multi-granular spatio-temporal token merging for training-free acceleration of video llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23990–24000, 2025.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pp. 235–251. Springer, 2016.

Samir Khaki, Junxian Guo, Jiaming Tang, Shang Yang, Yukang Chen, Konstantinos N Plataniotis, Yao Lu, Song Han, and Zhijian Liu. Sparsevila: Decoupling visual sparsity for efficient vlm inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23784–23794, 2025.

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.

Jialuo Li, Bin Li, Jiahao Li, and Yan Lu. Divide, then ground: Adapting frame selection to query types for long-form video understanding. *arXiv preprint arXiv:2512.04000*, 2025a.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025b.

Hao Liang, Jiapeng Li, Tianyi Bai, Xijie Huang, Linzhuang Sun, Zhengren Wang, Conghui He, Bin Cui, Chong Chen, and Wentao Zhang. Keyvideollm: Towards large-scale video keyframe selection. *arXiv preprint arXiv:2407.03104*, 2024.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*, 2025a.

Shuming Liu, Mingchen Zhuge, Changsheng Zhao, Jun Chen, Lemeng Wu, Zechun Liu, Chenchen Zhu, Zhipeng Cai, Chong Zhou, Haozhe Liu, et al. Videoauto-r1: Video auto reasoning via thinking once, answering twice. *arXiv preprint arXiv:2601.05175*, 2026.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4122–4134, 2025b.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025c.

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.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, et al. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

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*, pp. 3505–3506, 2020.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22857–22867, 2025.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. *arXiv preprint arXiv:2505.21334*, 2025a.

Kele Shao, Keda Tao, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. When tokens talk too much: A survey of multimodal long-context token compression across images, videos, and audios. *arXiv preprint arXiv:2507.20198*, 2025b.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. Fastvid: Dynamic density pruning for fast video large language models. *arXiv preprint arXiv:2503.11187*, 2025a.

Xiaoqian Shen, Min-Hung Chen, Yu-Chiang Frank Wang, Mohamed Elhoseiny, and Ryo Hachiuma. Zoom-zero: Reinforced coarse-to-fine video understanding via temporal zoom-in. *arXiv preprint arXiv:2512.14273*, 2025b.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26160–26169, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Mingyang Song, Haoyu Sun, Jiawei Gu, Linjie Li, Luxin Xu, Ranjay Krishna, and Yu Cheng. Adareasoner: Dynamic tool orchestration for iterative visual reasoning. *arXiv preprint arXiv:2601.18631*, 2026.

Guangyu Sun, Archit Singhal, Burak Uzkent, Mubarak Shah, Chen Chen, and Garin Kessler. From frames to clips: Training-free adaptive key clip selection for long-form video understanding. *arXiv preprint arXiv:2510.02262*, 2025.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. *arXiv preprint arXiv:2502.21271*, 2025.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18992–19001, 2025.

Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025a.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22958–22967, 2025b.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*, 2024.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-r1: Post-training large vision language model for temporal video grounding. *arXiv preprint arXiv:2503.13377*, 2025c.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

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*, pp. 13204–13214, 2024.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. *arXiv preprint arXiv:2510.09608*, 2025.

Biao Yang, Bin Wen, Boyang Ding, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai keye-vl 1.5 technical report. *arXiv preprint arXiv:2509.01563*, 2025a.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, et al. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19803–19813, 2025b.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025c.

Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, and Lidong Bing. Longvt: Incentivizing "thinking with long videos" via native tool calling. *arXiv preprint arXiv:2511.20785*, 2025d.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

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*, pp. 9556–9567, 2024.

Boqiang Zhang, Lei Ke, Ruihan Yang, Qi Gao, Tianyuan Qu, Rossell Chen, Dong Yu, et al. Penguin-vl: Exploring the efficiency limits of vlm with llm-based vision encoders. *arXiv preprint arXiv:2603.06569*, 2026.

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Yaqi Xie, Katia Sycara, Haitao Mi, and Dong Yu. Vscan: Rethinking visual token reduction for efficient large vision-language models. *arXiv preprint arXiv:2505.22654*, 2025a.

Congzhi Zhang, Zhibin Wang, Yinchao Ma, Jiawei Peng, Yihan Wang, Qiang Zhou, Jun Song, and Bo Zheng. Rewatch-r1: Boosting complex video reasoning in large vision-language models through agentic data synthesis. *arXiv preprint arXiv:2509.23652*, 2025b.

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, 2024a. URL https://arxiv.org/abs/2407.12772.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20857–20867, 2025c.

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. *arXiv preprint arXiv:2506.22139*, 2025d.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, et al. Thyme: Think beyond images. *arXiv preprint arXiv:2508.11630*, 2025e.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024b.

Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Ming Li, Qilong Wu, Kaipeng Zhang, and Chen Wei. Pyvision: Agentic vision with dynamic tooling. *arXiv preprint arXiv:2507.07998*, 2025a.

Yilun Zhao, Haowei Zhang, Lujing Xie, Tongyan Hu, Guo Gan, Yitao Long, Zhiyuan Hu, Weiyuan Chen, Chuhan Li, Zhijian Xu, et al. Mmvu: Measuring expert-level multi-discipline video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8475–8489, 2025b.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, 2024.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025b.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.

Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Zhenheng Yang, and Yang You. Focus: Efficient keyframe selection for long video understanding. *arXiv preprint arXiv:2510.27280*, 2025.

Yuanhao Zou, Shengji Jin, Andong Deng, Youpeng Zhao, Jun Wang, and Chen Chen. Air: Enabling adaptive, iterative, and reasoning-based frame selection for video question answering. *arXiv preprint arXiv:2510.04428*, 2025.

Limitations and future work.

ResAdapt improves the efficiency–accuracy trade-off for long-video MLLMs, but the current evidence is still bounded by four concrete design choices.

(i) Front-end overhead is amortized only in the long-context regime. The Allocator adds a fixed pre-encoding cost—coarse visual encoding, cross-frame fusion, and distribution prediction—before any backbone savings are realized. When the sequence is short ($T \leq 32$), this constant overhead can offset a meaningful fraction of the downstream attention reduction, so the clearest wall-clock gains appear only when temporal context is long (Sec. 4.3). Reducing this fixed cost through cached video features, cheaper front-ends, or distilled allocation rules is therefore an important next step.

(ii) Allocation is limited by coarse visual evidence. The Allocator observes frozen coarse features $f_t \in \mathbb{R}^D$ rather than the full high-resolution frame. This is sufficient to detect broad redundancy and scene structure, but it is weaker on small text, subtle objects, and brief answer-critical cues embedded in otherwise simple frames (Figure 20). Multi-scale conditioning, motion-aware features, or lightweight local refinement would help close this gap without giving up the speed advantage of the current front-end.

(iii) The present study validates the framework through one video-centric instantiation. Our formulation is general input-side adaptation, but the experiments instantiate the operator with resizing and train the policy primarily on video tasks. As a result, transfer beyond this regime is uneven: the learned policy sometimes recognizes image inputs that need more fidelity, yet it does not deliver uniformly efficiency-preserving gains on static-image benchmarks (Table 7). Extending training to mixed image–video data and alternative operators such as hard frame selection remains open.

(iv) Allocation is open-loop rather than reasoning-aware. All budget decisions are committed before the backbone processes any visual token. The policy therefore cannot revise a mistaken low-resolution choice after partial reasoning or uncertainty signals emerge. A natural extension is closed-loop allocation, where early backbone states trigger re-encoding, budget revision, or a second visual pass only when needed.

Software and Data

The code for this paper is available at: https://github.com/Xnhyacinth/ResAdapt

A Related Work

Input-side adaptation. Reducing visual cost *before* encoding typically involves keyframe selection (Liang et al., 2024; Zhu et al., 2025; Sun et al., 2025; Tang et al., 2025), recently augmented by query-aware iterative search (Zou et al., 2025; Li et al., 2025a; Guo et al., 2025b; He et al., 2025). Alternatively, multi-resolution encoding routes frames to varying resolutions based on inter-frame similarity (Yang et al., 2025a; Zhang et al., 2026) or query conditioning (Zhang et al., 2025d; Chen et al., 2026). However, these rely on handcrafted heuristics, binary routing, or fixed resolution bins. In contrast, ResAdapt learns an optimal, continuous input-side allocation policy directly from task rewards via RL, generalizing across pre-encoding operators like resizing and selection.

Model-side token economy. Post-encoding methods compress visual tokens in the embedding space via merging (Bolya et al., 2022), saliency-guided pruning (Chen et al., 2024; Yang et al., 2025c; Shang et al., 2025; Zhang et al., 2025c), progressive dropping (Xing et al., 2024; Zhang et al., 2024b), and diversity-based allocation (Alvar et al., 2025; Yang et al., 2025b; Zhang et al., 2025a). Video extensions further exploit spatiotemporal redundancy through token separation (Huang et al., 2025; Shen et al., 2025a), hierarchical merging (Hyun et al., 2025), and segment-level fusion (Tao et al., 2025; Fu et al., 2024; Shao et al., 2025a). These techniques are complementary to ResAdapt; they operate *after* encoding, whereas we determine the initial pixel budget to prevent irreversible loss of high-frequency details.

Output-side agentic reasoning. To recover efficiency, iterative reasoning methods keep the input fixed but re-query the model after retrieving frames or zooming into regions. These employ either static cropping/clipping operators (Zheng et al., 2025b; Wang et al., 2025a; Song et al., 2026) or dynamic code-generation tools (Zhang et al., 2025e; Zhao et al., 2025a; Hong et al., 2025; Wang et al., 2024). While precise, they are inherently multi-pass and suffer from latency and control overhead. ResAdapt demonstrates that a *single-pass* pre-encoding allocation policy can achieve similar precision without iterative interaction.

RL for multimodal reasoning. RL post-training, successful in LLMs (Shao et al., 2024; Guo et al., 2025a), has recently been adapted for multimodal reasoning (Liu et al., 2025c; Yu et al., 2025; Zheng et al., 2025a) and video understanding via iterative evidence refinement (Feng et al., 2025; Li et al., 2025b; Liu et al., 2026; Yang et al., 2025d; Chen et al., 2025; Wang et al., 2025c; Fu et al., 2025b). Our approach is orthogonal: rather than training output-side reasoning policies, we utilize RL for *input-side perception control* to optimize frame-level visual allocations under a strict accuracy–cost trade-off. To this end, CAPO specifically prevents the degenerate low-budget collapse caused by naive cost penalties.

B Implementation Details

B.1 Training Data

Data Composition. We build the training set from the difficulty-filtered data of VideoAuto-R1 (Liu et al., 2026), keeping only image and video samples and discarding pure-text examples. To improve coverage of visually demanding subdomains, we additionally sample 16,500 video instances from Video-R1 (Feng et al., 2025), focusing on OCR, free-form QA, and regression-style tasks. The merged pool contains approximately 93.4K training samples. We manually remove all evaluation examples from our benchmark suite to avoid leakage.

B.2 Training Configuration

Unless otherwise noted, training runs for one epoch with global batch size 128 and AdamW. The learning rate is 2×10^{-5} for the Allocator and 1×10^{-6} for the backbone, with weight decay 0.01 and gradient clipping at 1.0. We set the maximum video token budget to 8,192, use T=128 frames during training, and allow scales in the range $[s_{\min}, s_{\max}] = [0.2, 1.8]$, which permits both downscaling and selective upscaling. CAPO samples M=16 allocation trajectories per prompt and N=1 rollout per trajectory. Training is conducted on 32 H100 GPUs with VeRL (Sheng et al., 2025), DeepSpeed (Rasley et al., 2020), and vLLM (Kwon et al., 2023). Evaluation uses lmms-eval (Zhang et al., 2024a); unless stated otherwise, we cap response length at 256 tokens and increase it to 4,096 for reasoning models.

B.3 Reward Design

We provide full details complementing Sec. 3.3. The base scalar reward $R^{\text{task}}_{m,n}$ is task-specific; efficiency enters later through CAPO advantage shaping rather than through a raw additive reward term.

Base Task Reward ($R^{\text{task}}_{m,n}$). We consider four task types:

- *Question Answering.* For math problems, we extract the numeric answer and compare it to the ground truth within a tolerance of 10^{-2}. For multiple-choice questions, we extract the option letter. For other QA tasks, we compare normalized strings (*e.g.*, case-folded, whitespace-stripped). This yields the binary reward
$$R_{\text{QA}}(\hat{o}, o) \in \{0, 1\}.$$
- *Free-form Generation.* For open-ended tasks, we compute the ROUGE-L score between the generated answer $\hat{o}$ and the reference o:
$$R_{\text{Gen}}(\hat{o}, o) = \text{ROUGE-L}(\hat{o}, o) \in [0, 1].$$
- *Temporal Grounding.* Let the ground-truth segments be $\mathcal{G} = \{[s_j, e_j]\}_j$ and the predicted segments be $\widehat{\mathcal{G}} = \{[\hat{s}_k, \hat{e}_k]\}_k$ (each set may contain one or multiple intervals). We compute the temporal IoU and select the best-matching pair:
$$R_{\text{TG}}(\widehat{\mathcal{G}}, \mathcal{G}) \;=\; \max_{[\hat{s},\hat{e}]\in\widehat{\mathcal{G}},\, [s,e]\in\mathcal{G}} \text{tIoU}([\hat{s}, \hat{e}], [s, e]) \in [0, 1].$$
If no valid segment can be parsed from the output, we assign $R_{\text{TG}}(\widehat{\mathcal{G}}, \mathcal{G}) = 0$.
- *Grounding QA.* We parse both the textual answer and the predicted temporal segments from the model output, compute $R_{\text{QA}}(\hat{o}, o)$ and $R_{\text{TG}}(\widehat{\mathcal{G}}, \mathcal{G})$, and sum them:
$$R_{\text{GQA}}(\hat{o}, \widehat{\mathcal{G}}; o, \mathcal{G}) \;=\; R_{\text{QA}}(\hat{o}, o) + R_{\text{TG}}(\widehat{\mathcal{G}}, \mathcal{G}) \;\in [0, 2].$$

These task-specific metrics define the scalar base reward $R^{\text{task}}_{m,n}$. CAPO additionally uses a binary success indicator $u_{m,n} \in \{0, 1\}$: for exact-match QA tasks we use the binary correctness outcome directly, whereas for continuous metrics (ROUGE-L, temporal IoU, and their grounding-QA combination) we threshold the scalar score at 0.35, matching the implementation. When format validation is enabled, a weighted format term is added before GRPO normalization, but $u_{m,n}$ is computed from the task metric alone.

Format Reward. We employ a binary format reward $R_{\text{fmt}}(\hat{o}) \in \{0, 1\}$ enforced via strict regex validation. The output must contain exactly one `<think>...</think>` block and one `<answer>...</answer>` block, with the final answer enclosed in `\\boxed{...}` within the `<answer>` tags:
$$R_{\text{fmt}}(\hat{o}) = \begin{cases} 1 & \text{if format matches regex,} \\ 0 & \text{otherwise.} \end{cases}$$

In the implementation, malformed outputs receive a penalty before weighting, and the format term enters the scalar reward with weight 0.2.

Table 3: Prompt template used for CAPO training. The template presents video frames and the task question, requires intermediate reasoning inside <think> tags, and places the final answer in \boxed{} within <answer> tags. This structure enables automatic reward extraction from MLLM outputs.

Prompt Template for Training with Thinking

System Prompt:
You are a helpful assistant.
You FIRST think about 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 answer MUST BE enclosed within <answer> </answer> tags.
The final answer MUST BE put in \boxed{} and the \boxed{} expression MUST BE contained entirely within the <answer> </answer> tags.
Do not include any reasoning or explanations outside these tags.

B.4 Prompt Template

We employ the standard prompt for GRPO training, shown in Table 3. The model generates a reasoning trace within <think> </think> tags (optional for ResAdapt since reasoning is handled by the MLLM π_ϕ, but maintained for compatibility with reasoning-based baselines), followed by the final answer enclosed in \\boxed{}.

C Complexity Analysis

We derive formal computational bounds for ResAdapt to clarify when Allocator overhead is negligible relative to the savings induced in the backbone. For readability, we assume a standard Transformer backbone with quadratic self-attention and a uniform native resolution $H \times W$ over T frames; the extension to heterogeneous resolutions is immediate by replacing HW with per-frame products $H_t W_t$.

Baseline cost. Let P denote the ViT patch size. A vanilla MLLM encoding T frames at full resolution incurs a total visual token count of:

$$N_0 \;=\; T \cdot \left\lceil \frac{H}{P} \right\rceil \left\lceil \frac{W}{P} \right\rceil \;\approx\; \frac{THW}{P^2}. \tag{12}$$

Adaptive cost and token retention ratio. For the resize instantiation analyzed in this paper, frame f_t is rescaled by factor $s_t \in [s_{\min}, s_{\max}]$, producing $n_t(s_t) = \lceil s_t H/P \rceil \lceil s_t W/P \rceil \approx s_t^2 \cdot HW/P^2$ tokens. Summing over the sequence and normalizing by N_0 yields the *token retention ratio*:

$$N^{\text{adapt}} = \sum_{t=1}^{T} n_t(s_t) \;\approx\; \frac{HW}{P^2} \sum_{t=1}^{T} s_t^2, \qquad \rho \;\triangleq\; \frac{N^{\text{adapt}}}{N_0} \;=\; \frac{1}{T} \sum_{t=1}^{T} s_t^2. \tag{13}$$

Because the learned Beta policy places most redundant frames near $s_{\min}$ (Figure 8), ρ is much smaller than 1 in practice; across our evaluation suite, $\rho \in [0.06, 0.16]$.

Quadratic FLOPs reduction. For an L_{mllm}-layer MLLM with hidden dimension D_{mllm}, self-attention cost scales quadratically in the visual sequence length: $\Phi(N) = O(L_{\text{mllm}} N^2 D_{\text{mllm}})$. Substituting $N^{\text{adapt}} = \rho \cdot N_0$ gives:

$$\Phi_{\text{mllm}}^{\text{adapt}} \;=\; O\Big(L_{\text{mllm}} \cdot \rho^2 N_0^2 \cdot D_{\text{mllm}} \Big), \tag{14}$$

a reduction by a factor of ρ^2 relative to full-resolution processing. At the representative operating point $\rho = 0.11$, we obtain $\rho^2 \approx 0.012$, corresponding to roughly 83× fewer backbone attention FLOPs.

Allocator overhead. The Allocator processes $N_c = T \cdot \lceil H/P_c \rceil \lceil W/P_c \rceil$ coarsely pooled tokens across L_{pred} layers with dimension D_{pred}, where $P_c \gg P$ is the coarse spatial stride. Its cost and relative overhead are:

$$\Phi_{\text{pred}} = O\Big(L_{\text{pred}} \cdot N_c^2 \cdot D_{\text{pred}}\Big), \qquad \frac{\Phi_{\text{pred}}}{\Phi_{\text{mllm}}^{\text{base}}} = O\left(\frac{L_{\text{pred}}\, D_{\text{pred}}}{L_{\text{mllm}}\, D_{\text{mllm}}} \cdot \left(\frac{P}{P_c}\right)^4\right) \ll 1. \tag{15}$$

Substituting our implementation parameters (P_c=14, L_{pred}=4, D_{pred}=1,024 versus L_{mllm}=28, D_{mllm}=3,584), the Allocator accounts for less than 3% of inference FLOPs. The decision stage is therefore small compared with the backbone computation it helps eliminate.

Net speedup. Combining the above under the first-order approximation $\Phi_{\text{mllm}}^{\text{base}} \gg \Phi_{\text{pred}}$:

$$\text{Speedup} \approx \frac{\Phi_{\text{mllm}}^{\text{base}}}{\Phi_{\text{mllm}}^{\text{adapt}} + \Phi_{\text{pred}}} \approx \frac{N_0^2}{(N^{\text{adapt}})^2} = \frac{1}{\rho^2}. \tag{16}$$

At $\rho = 0.11$, this again yields a theoretical reduction of roughly 83× in backbone attention computation.

Temporal context scaling. The same savings admit a second interpretation in terms of *temporal coverage*. Under a fixed token budget B, a vanilla MLLM can process only $T_0 = BP^2/(HW)$ full-resolution frames, whereas the resize instantiation of ResAdapt used in our experiments can process T_0/ρ adaptively resized frames. This yields an effective $1/\rho \approx 6$–$16\times$ increase in temporal horizon at comparable compute, which is exactly the trade-off exploited by the long-context experiments in Sec. 4.2.

Remark (acceleration transparency). A practical consequence of Input-side adaptation is that the backbone still receives an ordinary visual-token sequence, only shorter. As a result, ResAdapt remains compatible with optimized attention stacks such as FlashAttention, vLLM (Kwon et al., 2023), and SGLang (Zheng et al., 2024) without kernel-level modifications. By contrast, model-side pruning and merging often create irregular token layouts that are harder to route through the same optimized kernels and may require fallback implementations or architecture-specific engineering.

D Derivation of Joint RL Formulation

This appendix collects derivations omitted from Sec. 3 for space and clarifies how the one-step contextual MDP (Contextual Bandit) introduced in Sec. 2.2 motivates the practical surrogate objectives optimized by ResAdapt. All derivations are stated for a single context (video and query); the full objective is the expectation over the dataset $\mathcal{D}$.

Notation. The prompt context is $\boldsymbol{x} = (\boldsymbol{q}, \mathcal{V})$. The Allocator policy $\pi_\theta(\boldsymbol{s} \mid \boldsymbol{x})$ samples a continuous allocation vector $\boldsymbol{s} = (s_1, \ldots, s_T)$. A deterministic transformation constructs the operator-transformed input $\tilde{\boldsymbol{x}} = (\boldsymbol{q}, \{\mathcal{O}(f_t, s_t)\}_{t=1}^T)$; in the experimental instantiation, $\mathcal{O}$ is bilinear resizing. The MLLM backbone policy $\pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}})$ then samples a full response rollout $\boldsymbol{y} = (\boldsymbol{r}, \boldsymbol{o})$, where $\boldsymbol{r}$ is the reasoning trace and $\boldsymbol{o}$ is the final answer.

D.1 One-Step Contextual MDP and the Joint Objective

As defined in Sec. 2.2, the system is a one-step contextual MDP. In this setting, there are no sequential state transitions across time steps t; the episode terminates after the allocation $\boldsymbol{s}$ is sampled and the corresponding rollout $\boldsymbol{y}$ is produced. Consequently, the value functions collapse to the immediate reward, and the standard Policy Gradient Theorem simplifies drastically without requiring temporal discount factors or credit assignment across Markov states.

The joint distribution of the allocation and the rollout factorizes conditionally:

$$p_{\theta,\phi}(\boldsymbol{s}, \boldsymbol{y} \mid \boldsymbol{x}) = \pi_\theta(\boldsymbol{s} \mid \boldsymbol{x})\, \pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}}). \tag{17}$$

For a single context with ground-truth answer $\boldsymbol{o}^\star$, the marginal answer probability under the transformed input is

$$p_{\theta,\phi}(\boldsymbol{o}^\star \mid \boldsymbol{x}) = \mathbb{E}_{\pi_\theta(\boldsymbol{s}|\boldsymbol{x})}\Big[\mathbb{E}_{\pi_\phi(\boldsymbol{r}|\tilde{\boldsymbol{x}})}\big[\pi_\phi(\boldsymbol{o}^\star \mid \tilde{\boldsymbol{x}}, \boldsymbol{r})\big]\Big]. \tag{18}$$

Because $\log(\cdot)$ is monotone, maximizing $\log p_{\theta,\phi}(\boldsymbol{o}^\star \mid \boldsymbol{x})$ would be equivalent, but the RL derivation below does not require introducing the logarithm. It only requires a scalar utility evaluated after sampling $(\boldsymbol{s}, \boldsymbol{y})$. We therefore abstract the answer-quality term as a rollout utility $Q(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{y} = (\boldsymbol{r}, \boldsymbol{o})$, and treat it as parameter-independent once the rollout is sampled. This is a modeling abstraction rather than an exact reformulation: when Q is chosen as an answer-aligned task score, the resulting RL problem is a surrogate to likelihood maximization. This lets us define the ideal rollout reward

$$R^{\text{ideal}}_{\boldsymbol{s},\boldsymbol{y}} = Q(\boldsymbol{x}, \boldsymbol{y}) - \lambda\, C(\boldsymbol{s}), \tag{19}$$

and optimize the one-step expected return

$$\max_{\theta,\phi}\ \mathcal{J}(\theta, \phi) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{E}_{\pi_\theta(\boldsymbol{s}|\boldsymbol{x})}\Big[\mathbb{E}_{\pi_\phi(\boldsymbol{y}|\tilde{\boldsymbol{x}})}\Big[R^{\text{ideal}}_{\boldsymbol{s},\boldsymbol{y}}\Big]\Big]. \tag{20}$$

D.2 Policy Gradient and Alternating Optimization

Because the objective involves two distinct parameterized policies, its gradients follow the score-function estimator (the likelihood-ratio / REINFORCE identity). This is the underlying policy-gradient structure; GRPO does not change that structure, but replaces the raw reward with normalized advantages and clipped surrogates for practical optimization. Taking the gradient of $\mathcal{J}(\theta, \phi)$ with respect to the backbone parameters ϕ:

$$\begin{aligned}\nabla_\phi \mathcal{J}(\theta, \phi) &= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\pi_\theta(\boldsymbol{s}|\boldsymbol{x})}\left[\nabla_\phi \int \pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}}) R^{\text{ideal}}_{\boldsymbol{s},\boldsymbol{y}} d\boldsymbol{y}\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\pi_\theta(\boldsymbol{s}|\boldsymbol{x})}\mathbb{E}_{\pi_\phi(\boldsymbol{y}|\tilde{\boldsymbol{x}})}\left[R^{\text{ideal}}_{\boldsymbol{s},\boldsymbol{y}}\, \nabla_\phi \log \pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}})\right].\end{aligned} \tag{21}$$

Taking the gradient of $\mathcal{J}(\theta, \phi)$ with respect to the Allocator parameters θ:

$$\begin{aligned}\nabla_\theta \mathcal{J}(\theta, \phi) &= \mathbb{E}_{\boldsymbol{x}}\left[\nabla_\theta \int \pi_\theta(\boldsymbol{s} \mid \boldsymbol{x}) \left(\int \pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}}) R^{\text{ideal}}_{\boldsymbol{s},\boldsymbol{y}} d\boldsymbol{y}\right) d\boldsymbol{s}\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\pi_\theta(\boldsymbol{s}|\boldsymbol{x})}\left[\left(\int \pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}}) R^{\text{ideal}}_{\boldsymbol{s},\boldsymbol{y}} d\boldsymbol{y}\right) \nabla_\theta \log \pi_\theta(\boldsymbol{s} \mid \boldsymbol{x})\right].\end{aligned} \tag{22}$$

In practice, we substitute the ideal reward $R^{\text{ideal}}_{\boldsymbol{s},\boldsymbol{y}}$ with the shaped CAPO advantage $A_{m,n}$, leading to the clipped surrogate objectives $\mathcal{L}_\phi$ and $\mathcal{L}_\theta$ described in the main text.

For the backbone update, the standard token-level clipped surrogate objective is:

$$\mathcal{L}_\phi = -\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{1}{L_{m,n}}\sum_{j=1}^{L_{m,n}} \min\Big(r_{\phi,j}^{(m,n)}\, A_{m,n},\ \text{clip}\big(r_{\phi,j}^{(m,n)}, 1-\varepsilon, 1+\varepsilon\big)\, A_{m,n}\Big), \tag{23}$$

where $L_{m,n}$ is the rollout length and

$$r_{\phi,j}^{(m,n)} = \frac{\pi_\phi(y_j^{(m,n)} \mid y_{<j}^{(m,n)}, \tilde{\boldsymbol{x}}^{(m)})}{\pi_{\phi_{\text{old}}}(y_j^{(m,n)} \mid y_{<j}^{(m,n)}, \tilde{\boldsymbol{x}}^{(m)})}. \tag{24}$$

The exact change-of-variables mapping $\boldsymbol{a} \mapsto \boldsymbol{s}$ yields the log-probability:

$$\log q_\theta(\boldsymbol{a} \mid \boldsymbol{x}) = \sum_{t=1}^{T} \log \text{Beta}(a_t; \alpha_t, \beta_t). \tag{25}$$

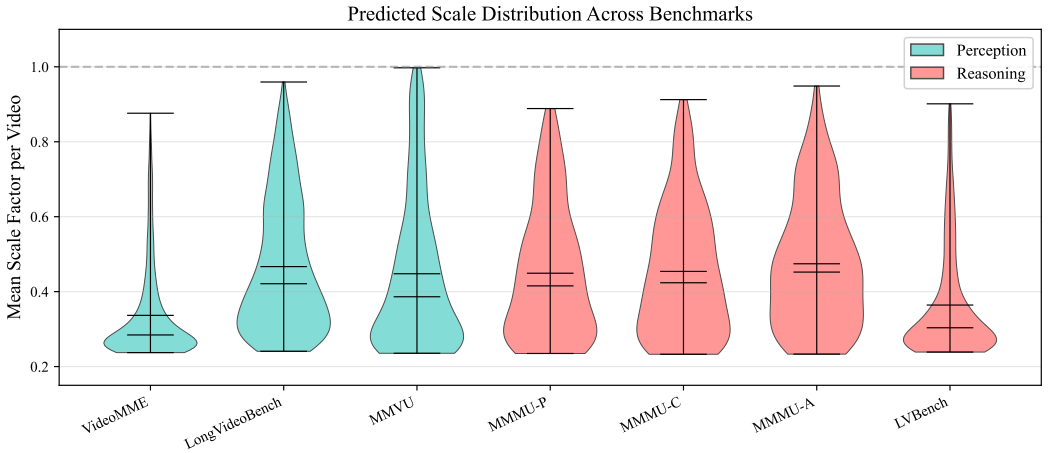


Figure 6: Per-video mean scale across benchmarks. Kernel density estimates of the per-video mean scale $\bar{s}$. Reasoning-heavy benchmarks shift toward larger $\bar{s}$ than perception-heavy ones, indicating that the learned policy spends more fidelity where fine-grained evidence is more likely to matter.

Similarly, the gradient with respect to the Allocator parameters θ relies on the marginalized reward $R_{\boldsymbol{s}}^{\text{ideal}} = \mathbb{E}_{\pi_\phi(\boldsymbol{y}|\tilde{\boldsymbol{x}})}[R_{\boldsymbol{s},\boldsymbol{y}}^{\text{ideal}}]$:

$$\nabla_\theta \mathcal{J}(\theta, \phi) = \mathbb{E}_{\boldsymbol{x}} \mathbb{E}_{\pi_\theta(\boldsymbol{s}|\boldsymbol{x})} \left[R_{\boldsymbol{s}}^{\text{ideal}} \nabla_\theta \log \pi_\theta(\boldsymbol{s} \mid \boldsymbol{x}) \right]. \tag{26}$$

To optimize this objective with GRPO, we introduce importance sampling from behavior policies $\pi_{\theta_{\text{old}}}$ and $\pi_{\phi_{\text{old}}}$. A naive joint importance weight $\frac{\pi_\theta \pi_\phi}{\pi_{\theta_{\text{old}}} \pi_{\phi_{\text{old}}}}$ suffers from compounded variance. We therefore use an **alternating block-coordinate ascent** approximation. When updating the MLLM (ϕ), we fix the Allocator to its behavior policy ($\pi_\theta = \pi_{\theta_{\text{old}}}$), making its importance ratio exactly 1. The off-policy surrogate gradient for ϕ becomes:

$$\nabla_\phi \mathcal{J}_{\text{surr}}(\phi) = \mathbb{E}_{\pi_{\theta_{\text{old}}}} \mathbb{E}_{\pi_{\phi_{\text{old}}}} \left[\frac{\pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}})}{\pi_{\phi_{\text{old}}}(\boldsymbol{y} \mid \tilde{\boldsymbol{x}})} R_{\boldsymbol{s},\boldsymbol{y}}^{\text{ideal}} \nabla_\phi \log \pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}}) \right]. \tag{27}$$

Using the log-derivative identity $\nabla_\phi r_\phi = r_\phi \nabla_\phi \log \pi_\phi$ where $r_\phi = \pi_\phi / \pi_{\phi_{\text{old}}}$, this motivates the surrogate objective:

$$\mathcal{L}_\phi^{\text{ideal}} = \mathbb{E}_{\pi_{\theta_{\text{old}}}} \mathbb{E}_{\pi_{\phi_{\text{old}}}} \left[r_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}}) R_{\boldsymbol{s},\boldsymbol{y}}^{\text{ideal}} \right]. \tag{28}$$

Conversely, when updating the Allocator (θ), we fix the backbone to its behavior policy ($\pi_\phi = \pi_{\phi_{\text{old}}}$). The corresponding ideal allocator surrogate is

$$\mathcal{L}_\theta^{\text{ideal}} = \mathbb{E}_{\pi_{\theta_{\text{old}}}} \left[r_\theta(\boldsymbol{s} \mid \boldsymbol{x})\, R_{\boldsymbol{s}}^{\text{ideal}} \right], \qquad r_\theta(\boldsymbol{s} \mid \boldsymbol{x}) = \frac{\pi_\theta(\boldsymbol{s} \mid \boldsymbol{x})}{\pi_{\theta_{\text{old}}}(\boldsymbol{s} \mid \boldsymbol{x})}, \tag{29}$$

where $R_{\boldsymbol{s}}^{\text{ideal}} = \mathbb{E}_{\pi_{\phi_{\text{old}}}(\boldsymbol{y}|\tilde{\boldsymbol{x}})}[R_{\boldsymbol{s},\boldsymbol{y}}^{\text{ideal}}]$. In practice, this expectation is approximated by Monte Carlo rollouts under the frozen backbone.

D.3 Advantage Shaping and Monte Carlo Surrogates

The ideal linear penalty $-\lambda C(\boldsymbol{s})$ inside R^{ideal} often causes catastrophic collapse to minimum budgets. CAPO therefore replaces the raw reward with a cost-shaped, group-normalized advantage $A_{\boldsymbol{s},\boldsymbol{y}}$ (denoted $A_{m,n}$ in the main text). This replacement is *not* an unbiased baseline transformation of $R_{\boldsymbol{s},\boldsymbol{y}}^{\text{ideal}}$: the CAPO signal depends on the sampled allocation, the rollout outcome, and the within-group cost statistics. Instead, it defines a deliberately biased

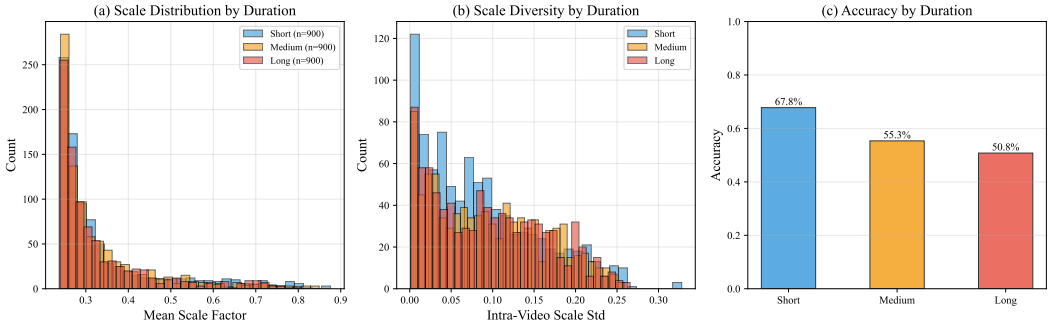


Figure 7: VideoMME broken down by video duration. As clip duration grows, the policy lowers the average scale, increases within-video scale diversity, and faces lower task accuracy. Longer clips are therefore processed more aggressively and more selectively.

surrogate objective that trades exact fidelity to the Lagrangian reward for lower variance and stronger budget control in practice.

Applying surrogate clipping to the exact joint ratios would couple all frame- and token-level factors, which is prohibitively noisy in practice. We therefore arrive at practical decoupled objectives. For a batch of M allocations and N rollouts per allocation, the MLLM sequence-level surrogate is:

$$\mathcal{L}_\phi^{\text{seq}} = -\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\min\Big(r_\phi^{(m,n)}\,A_{m,n},\ \text{clip}(r_\phi^{(m,n)},1-\varepsilon,1+\varepsilon)\,A_{m,n}\Big). \tag{30}$$

This sequence-level loss is already approximate because it uses the CAPO-shaped advantage in place of the ideal reward. To achieve finer credit assignment for the autoregressive MLLM, we further factorize $\pi_\phi(\boldsymbol{y} \mid \tilde{\boldsymbol{x}})$ into token-level probabilities, distribute the same rollout-level advantage $A_{m,n}$ to all tokens, and average over the sequence length $L_{m,n}$. Equation (23) should therefore be read as the standard token-level approximation to this sequence-level surrogate, not as an exact decomposition of the clipped joint ratio.

Conversely, when updating the Allocator (θ), we fix the MLLM ($\pi_\phi = \pi_{\phi_{\text{old}}}$) and use the aggregated advantage $A_m^{\text{CAPO}} = \frac{1}{N}\sum_n A_{m,n}$. Because the Allocator's output distribution factorizes conditionally across frames (Eq. 25), its score function decomposes additively:

$$\nabla_\theta \log \pi_\theta(\boldsymbol{s}^{(m)} \mid \boldsymbol{x}) = \sum_{t=1}^{T}\nabla_\theta \log \text{Beta}(a_t^{(m)};\,\alpha_t,\beta_t). \tag{31}$$

This additive log-probability structure supports low-variance frame-level credit assignment. Nevertheless, Eq. (11) remains a practical approximation to a trajectory-level clipped objective: conditional independence justifies decomposition of $\log \pi_\theta$, but not exact factorization of the nonlinear clipping term. We use the per-frame surrogate because it is substantially more stable in large-scale training.

E Supplementary Experiments and Analysis

This section first analyzes the learned allocation policy, then studies the two key ablation axes, and finally reports representative qualitative cases and a boundary-case transfer test beyond video. Unless otherwise noted, all plots use Qwen2.5-VL-7B with 32 uniformly sampled frames.

E.1 Behavioral Analysis of the Learned Policy

E.1.1 Global Allocation Statistics

The global distribution of predicted scales reveals the fundamental mechanism of ReSAdapt's efficiency. As shown in Figure 8, rather than distributing the resolution budget

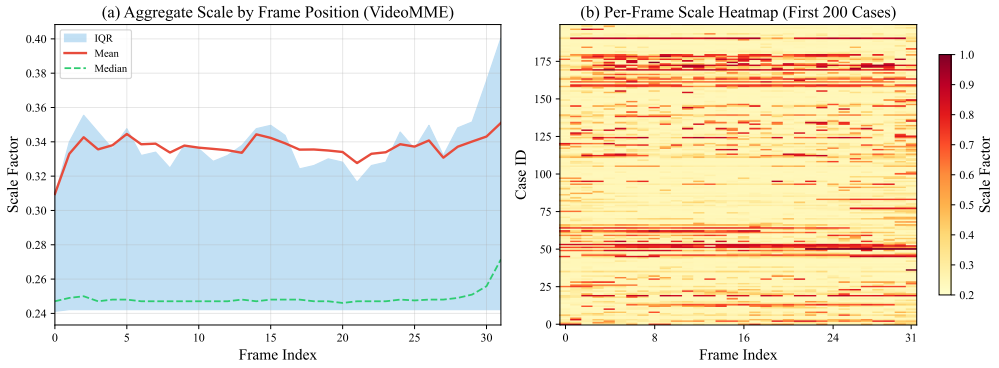


Figure 8: Global distribution of predicted scales. Across the entire dataset, the allocation policy exhibits a highly skewed distribution. The vast majority of frames are heavily compressed (near the minimum scale s_{min}), reserving the spatial token budget for a small subset of critical, information-dense frames. This global sparsity confirms that ResAdapt achieves efficiency by aggressively reducing redundancy rather than uniformly lowering resolution.

evenly, the policy learns a highly skewed allocation strategy. Most frames are downscaled close to the minimum allowed bound ($s_{\text{min}} = 0.2$), effectively discarding spatial redundancy in static or uninformative video segments. This massive compression on the majority of the video creates the budget headroom needed to process the few critical, evidence-bearing frames at much higher resolutions. This confirms that the learned behavior is genuinely selective, matching the localized scale spikes observed in the individual temporal profiles (Figure 4).

E.1.2 Benchmark-Level Budget Allocation

Figure 6 shows a clear benchmark-level ordering even though the policy never observes benchmark labels during training. Averaged across datasets, reasoning-oriented tasks use slightly higher mean scales than perception-oriented ones (0.435 vs. 0.417), with MMMU-Adaptation at the high end and VideoMME at the low end. The pattern is consistent with the main claim of the paper: the policy is not enforcing a fixed compression rule, but adapting its operating point to the expected visual difficulty of the task family.

E.1.3 Long-Context and Semantic Structure

Figure 7 is consistent with the long-context gains in the main paper. From short to long clips, the mean scale drops (0.342→0.336→0.332), but the within-video diversity rises (0.085→ ∼0.095). In other words, the policy does not merely compress longer videos more; it also becomes more selective inside them, which is exactly the regime where uniform resizing is least satisfactory.

Figure 9 refines the same story within a single benchmark. The policy spends the most budget on *Sports Competition* and the least on *Artistic Performance*, suggesting that even within VideoMME it distinguishes categories that are dense and spatially demanding from those that are visually simpler. This complements the main benchmark tables: the appendix focuses on *why* retained budgets differ, while the main text already reports the exact realized retention ratios.

E.1.4 Selectivity and Success

We next ask whether successful samples allocate budget *more selectively* within a clip. We quantify frame-level selectivity with the Gini coefficient of the predicted scales. High Gini means the policy concentrates budget on a small subset of frames; low Gini means the allocation is nearly uniform.

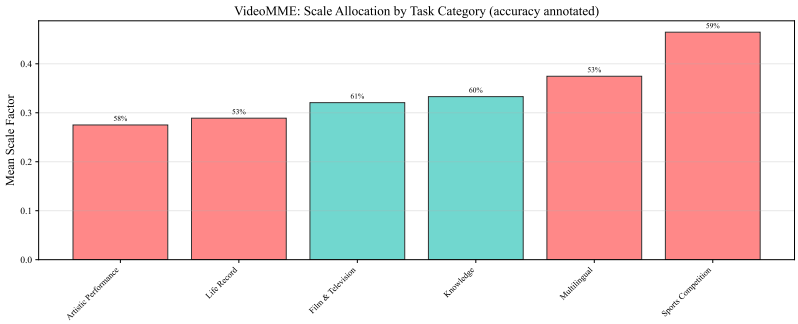


Figure 9: Scale allocation by VideoMME task category. Mean $\bar{s}$ varies substantially across categories, with larger budgets assigned to categories that contain crowded motion or finer local evidence. Accuracy annotations show that allocation is not a trivial proxy for which category is easiest.

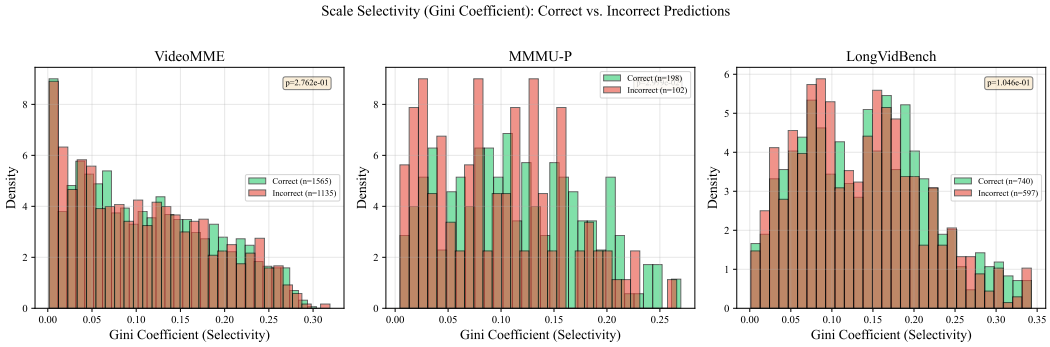


Figure 10: Selectivity versus prediction correctness on three representative benchmarks. Per-video Gini coefficients of the frame-level scales. Correct predictions tend to have higher Gini than incorrect ones, linking success to sharper concentration of resolution rather than merely larger average budgets.

Figure 10 shows that correct predictions consistently lie in the more selective regime, with the clearest separation on MMMU-P. This sharpens the mechanism claim of the appendix: success is associated not merely with keeping more pixels overall, but with concentrating them onto the frames that matter.

Robustness and failure modes. A final question is whether adaptive compression preserves existing correct answers or merely swaps one error pattern for another.

Figure 11 provides the right robustness interpretation for aggressive compression. Prediction stability remains high overall (about 89% of originally correct samples stay correct in the aggregate summary), so the policy is not helping only by randomly perturbing the answer distribution. However, error correction and error introduction are close enough that the effect should be read as *selective redistribution*: the policy repairs some failures, but it can also lose fine-grained evidence, especially when the decisive cue is brief or visually simple.

E.2 Ablation Studies

E.2.1 Temporal Similarity Ablation

We provide two complementary views of the temporal-similarity ablation: a cross-benchmark summary showing that the effect generalizes, and a single-benchmark diagnostic panel showing exactly how the allocation pattern changes.

Figure 12 makes the role of $\mathcal{L}_{\text{sim}}$ unusually clear. Without it, the policy collapses to near-uniform scales on every benchmark ($\sigma < 0.003$); with it, the same model family recovers substantial within-video variation, with 4×–693× larger diversity depending on the bench-

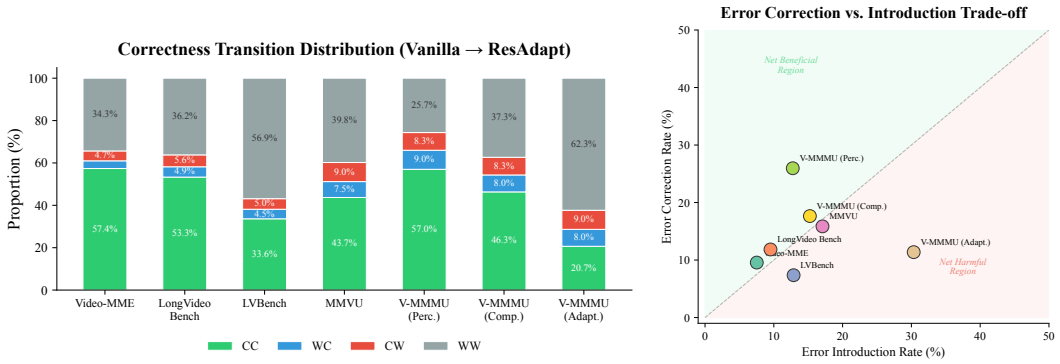


Figure 11: Sample-level robustness at 25% retention. Most originally correct predictions remain correct, but corrected and newly introduced errors are of comparable magnitude. Adaptive allocation is therefore selective rather than lossless.

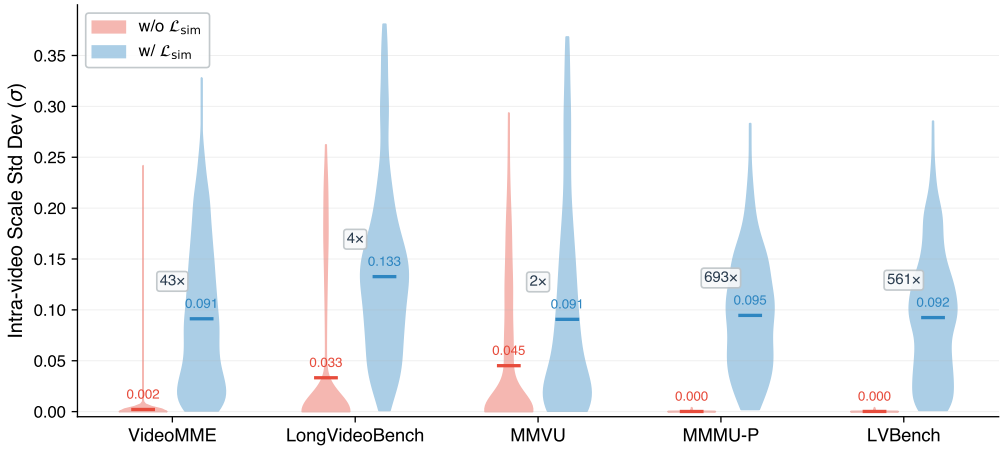


Figure 12: Cross-benchmark scale diversity with and without $\mathcal{L}_{\text{sim}}$. Per-video scale standard deviation σ across five benchmarks. Without the regularizer, diversity collapses toward zero; adding $\mathcal{L}_{\text{sim}}$ restores broad within-video variation on every benchmark.

mark. CAPO therefore controls *where* the global budget should sit, whereas $\mathcal{L}_{\text{sim}}$ prevents the trivial fixed-scale solution.

Quantitative confirmation. Figure 13 shows that this is not an artifact of any single statistic. The regularizer changes the global histogram, the per-video range, the frame-to-frame variation, and the Gini coefficient in the same direction, confirming that the benefit is structural rather than metric-specific.

E.2.2 Reward Design Ablation

We next examine whether different reward designs preserve a non-degenerate adaptive regime during training. All plots use EMA smoothing to suppress per-step noise; raw values remain visible as translucent traces.

Per-sample scale adaptivity. Figure 14 complements Figure 5 by measuring the per-sample scale range $s_{\max} - s_{\min}$ rather than the mean. CAPO preserves non-trivial adaptivity on validation, whereas direct cost collapses to the lower boundary and cost-free optimization drifts toward a nearly uniform high-scale policy.

Convergence and stability.

Figure 16 explains why CAPO works and the simpler baselines do not. The CAPO variants converge to stable interior solutions, whereas accuracy-only training saturates near $s_{\max}$ and

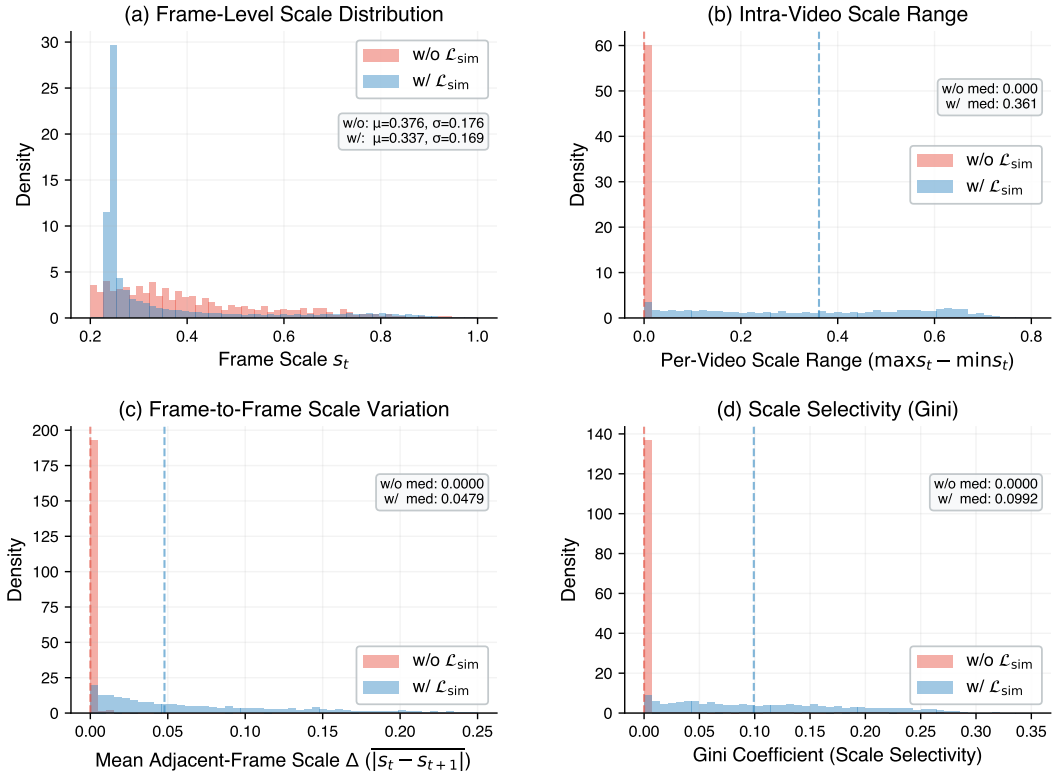


Figure 13: Four diagnostics of the $\mathcal{L}_{\text{sim}}$ ablation on VideoMME. With the regularizer, the frame-scale histogram becomes bimodal, the per-video range expands, adjacent-frame variation increases, and the Gini coefficient rises. The policy moves from near-uniform allocation to a genuinely selective regime.

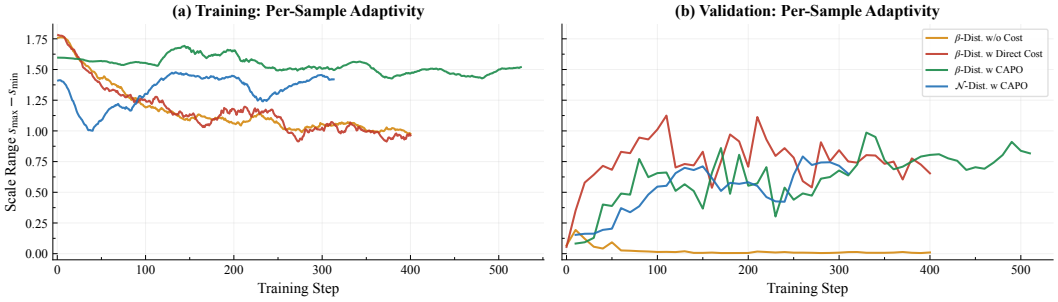


Figure 14: Per-sample scale adaptivity under different reward designs. Scale range $s_{\max} - s_{\min}$ over training on **(a)** training and **(b)** validation splits. CAPO keeps a non-trivial adaptive range, whereas direct cost collapses and cost-free training saturates.

direct cost collapses to $s_{\min}$. This is consistent with CAPO's intended role: balancing task reward and budget pressure without falling into either trivial boundary solution. The key result is therefore not merely convergence, but convergence to a non-degenerate operating point where content-adaptive allocation is still available.

E.3 Additional Ablation Studies

E.4 Qualitative Case Studies

We present four representative case studies that complement the aggregate analysis above: two task-contrast examples from Video-MMMU, one evidence-localization success from VideoMME, and one failure case. Each visualization (Figures 17–20) renders 32 uniformly

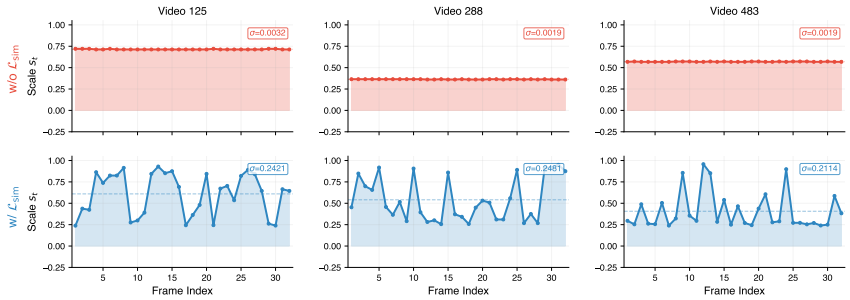


Figure 15: $\mathcal{L}_{\text{sim}}$ ablation: per-frame scale profiles. Without temporal-similarity regularization, the Allocator approaches near-uniform scaling; with it, the policy concentrates resolution on selected frames and suppresses redundant neighbors.

Table 4: Distribution family ablation for CAPO. The two variants follow the same training protocol.

Variant	$\bar{s}$	VideoMME	LongVideoBench	MMVU	VideoMMMU			LVBench
					Per.	Comp.	Adap.	
β-CAPO	0.54	60.3	**58.2**	51.2	65.0	**54.3**	28.7	**37.6**
$\mathcal{N}$-CAPO	0.60	**61.0**	57.4	**51.8**	**66.0**	50.0	**30.3**	37.2

sampled frames at their assigned scale inside a fixed grid; warmer borders indicate larger predicted scales.

Task-dependent operating regimes. Figures 17 and 18 contrast two Video-MMMU tasks from a visually similar educational domain that nevertheless demand very different allocations. In the comprehension example, the relevant evidence is concentrated in a small set of diagram-bearing slides, so the policy adopts a sparse operating regime and suppresses the explicitly irrelevant quiz frame. In the adaptation example, the downstream reasoning depends on reading a dense numeric table, so the same policy shifts to a much higher-budget regime and preserves high fidelity much more broadly. The contrast shows that the policy responds to what the task will require, not just to generic visual clutter.

Evidence localization and failure. The VideoMME success case in Figure 19 shows a more local version of the same phenomenon: the answer depends on short text overlays embedded in otherwise repetitive footage, and the policy magnifies only those evidence-bearing moments. Figure 20 shows the failure mode that remains. The decisive cue is temporally brief and visually simple, so the policy enlarges a nearby frame but compresses the frame that actually contains the fork. This diagnosis matches the quantitative robustness analysis: ResAdapt is strong at concentrating budget, but still vulnerable when the decisive evidence is both subtle and short-lived.

Summary. Together, these case studies support the same three conclusions as the quantitative appendix: the policy changes its operating regime with the task, concentrates fidelity on evidence-bearing frames, and fails in interpretable ways when subtle cues are missed. The qualitative examples therefore reinforce the claim that ResAdapt learns a meaningful input-allocation strategy rather than a fixed compression heuristic.

E.5 Failure Modes Analysis

Adaptive allocation does not act as a lossless compression layer. In practice, ResAdapt usually preserves many originally correct predictions, but it can still miss decisive evidence, especially when the relevant cue is visually simple and appears only briefly. Because the policy is open-loop, it cannot revise allocations after reasoning begins or recover evidence that was undersampled in the initial pass. We therefore interpret its gains as selective redistribution of visual budget rather than as guaranteed preservation of all useful information.

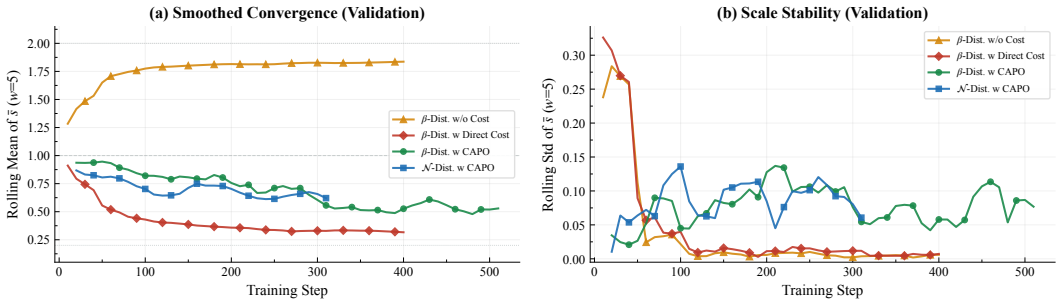


Figure 16: Validation-time convergence under different reward designs. CAPO variants converge to stable intermediate operating points, while cost-free training saturates at the upper boundary and direct cost collapses to the lower boundary. Stability alone is not sufficient; the key is where the policy stabilizes.

Table 5: Operator generalization. Zero-shot transfer of ResAdapt scores to frame selection. Combining top-K selection with adaptive resizing from 128 candidate frames outperforms uniform sampling baselines at lower token budgets.

Method	VideoMME	LongVideoBench	LVBench	MMVU
Budget: 8 frames				
Vanilla	54.0	53.9	33.3	48.9
Top-8 Select	52.2	51.1	32.0	49.2
Budget: 16 frames				
Vanilla	58.9	56.0	36.1	50.9
Threshold Select	58.0	57.4	36.4	51.0
Avg. Budget (Retention Ratio)	12.2f (9.5%)	23.2f (18.1%)	16.7f (13.0%)	17.2f (13.4%)
Top-32 Select + Resize	60.6	57.2	38.9	50.2
Avg. Budget (Retention Ratio)	11.7f (9.1%)	16.9f (13.2%)	13.7f (10.7%)	14.1f (11.0%)
Budget: 32 frames				
Vanilla	62.3	58.7	39.5	52.0
Top-32 Select	59.7	55.7	37.0	51.2
Top-64 Select + Resize	62.5	58.4	40.0	52.3
Avg. Budget (Retention Ratio)	23.8f (18.6%)	36.2f (28.3%)	24.1f (18.8%)	32.5f (25.4%)

E.6 Temporal Grounding Full Results

Table 6 provides the full temporal grounding evaluation results that were omitted from the main text for space.

E.7 Boundary-Case Transfer Beyond Video

The paper's main claims target video QA and temporal grounding, so we place image transfer at the end of the appendix as a boundary-case analysis rather than as supporting evidence for the main contribution. Table 7 is still informative: the learned video policy sometimes identifies image inputs that warrant additional fidelity, as in ChartQA, but it does not yet yield reliable efficiency-preserving transfer on text-dense image tasks. The result is therefore best read as scope clarification. It suggests that input-side adaptation is broader than the resize-on-video setting studied here, while also showing that a video-trained policy should not be assumed to transfer cleanly to static images.

Table 6: Evaluation Results on Temporal Grounding Benchmarks. Grounding is much more compression-sensitive.

Backbone	Method	Retention Ratio R	Reasoning	Temporal Grounding Benchmark									
				Charades-STA				ActivityNet				NExT-GQA	
				0.3	0.5	0.7	mIoU	0.3	0.5	0.7	mIoU	Acc	mIoU
Qwen2.5-VL-7B	*32 Frames*												
	Vanilla	100%	✗	71.0	51.4	26.0	47.3	30.4	18.0	8.9	22.6	78.9	28.0
	Random Drop	25.0%	✗	39.4	23.2	11.0	25.7	15.2	8.1	3.7	11.7	77.5	16.6
	ToMe (Bolya et al., 2022)	25.0%	✗	39.5	23.9	11.4	26.0	16.0	8.4	4.0	12.1	77.8	16.3
	FlashVid (Fan et al., 2026)	31.3%	✗	40.7	24.2	11.3	26.6	15.8	8.4	3.8	12.0	**78.1**	16.5
	FixedScale	25.0%	✗	36.7	24.7	12.3	24.9	18.6	9.4	4.3	14.1	77.7	12.3
	ResAdapt (Ours)	16.2%	✗	**53.8**	**34.8**	**17.0**	**35.6**	**19.8**	**10.8**	**5.2**	**15.3**	76.6	**23.2**
	Random Drop	10.0%	✗	36.9	23.2	11.6	24.6	14.3	7.5	3.6	11.1	76.3	15.4
	ToMe (Bolya et al., 2022)	10.0%	✗	41.3	26.9	14.1	27.4	16.0	8.4	**4.0**	12.2	77.3	15.7
	FlashVid (Fan et al., 2026)	12.6%	✗	38.2	22.9	11.1	25.1	15.4	8.1	3.7	11.8	**77.4**	16.1
	FixedScale	12.3%	✗	**48.0**	**31.5**	**15.4**	**32.0**	**17.5**	**8.9**	**4.0**	**13.3**	76.1	13.7
	FixedScale	6.3%	✗	39.9	26.8	13.3	26.7	15.2	8.1	3.9	11.9	74.1	15.4
	ResAdapt (Ours)	6.8%	✗	41.0	27.8	14.0	27.2	16.3	8.5	3.9	12.5	74.3	**20.4**
	VideoAuto-R1 (Liu et al., 2026)	100%	✓	60.0	48.3	27.2	41.5	50.8	34.1	17.4	34.4	73.6	33.8
	+ **ResAdapt (Ours)**	6.8%	✓	43.5	30.1	15.8	30.0	35.4	21.5	10.0	24.4	74.7	24.7
	128 Frames												
	Vanilla	100%	✗	77.5	60.3	34.1	52.8	47.9	30.9	17.5	34.4	79.8	29.9
	Random Drop	25.0%	✗	32.3	19.6	7.9	20.7	26.7	13.9	6.3	18.8	**80.3**	10.7
	ToMe (Bolya et al., 2022)	25.0%	✗	32.4	19.8	7.9	20.7	27.2	14.4	6.4	19.1	**80.3**	10.9
	ResAdapt (Ours)	16.1%	✗	**63.5**	**43.6**	**21.3**	**42.0**	**33.1**	**19.3**	**10.2**	**24.3**	78.1	**27.2**
	Random Drop	10.0%	✗	37.8	23.8	11.2	24.7	23.8	12.0	5.3	17.0	**79.4**	12.8
	ToMe (Bolya et al., 2022)	10.0%	✗	27.9	16.2	7.3	17.9	22.9	11.8	5.5	16.4	79.1	11.1
	FixedScale	12.3%	✗	34.7	22.3	10.5	22.7	**25.0**	**13.8**	5.9	**18.3**	77.9	11.3
	FixedScale	6.3%	✗	42.6	28.4	14.3	28.3	22.8	12.8	5.7	17.1	75.7	12.9
	ResAdapt (Ours)	6.8%	✗	**43.5**	**29.8**	**15.0**	**28.9**	23.5	12.9	**6.1**	17.2	76.2	**23.9**
	VideoAuto-R1 (Liu et al., 2026)	100%	✓	40.3	33.7	22.1	28.9	49.4	34.3	18.5	33.5	68.0	31.0
	+ **ResAdapt (Ours)**	16.1%	✓	72.8	53.0	27.5	49.1	65.8	44.9	23.8	44.7	79.3	35.3
	+ **ResAdapt (Ours)**	6.8%	✓	50.1	33.2	16.6	34.2	53.4	34.0	16.4	35.7	76.6	29.4
Qwen3-VL-8B	*32 Frames*												
	Vanilla	100%	✗	73.0	49.0	21.4	46.4	44.6	28.3	15.5	31.8	78.7	34.2
	Random Drop	25.0%	✗	16.2	8.6	3.8	12.1	12.4	6.7	3.2	10.0	77.2	15.6
	ToMe (Bolya et al., 2022)	25.0%	✗	68.7	42.1	17.6	43.1	45.9	28.8	15.6	32.6	77.1	31.7
	FlashVid (Fan et al., 2026)	31.3%	✗	**72.9**	**52.3**	**25.1**	**47.7**	**51.9**	**33.4**	**19.0**	**36.8**	**77.8**	**33.9**
	ResAdapt (Ours)	16.2%	✗	64.4	37.3	16.3	39.9	40.0	24.4	13.0	28.5	75.1	30.2
	Random Drop	10.0%	✗	4.1	1.8	0.7	4.4	4.7	2.4	1.0	5.0	74.3	11.3
	ToMe (Bolya et al., 2022)	10.0%	✗	67.6	39.3	16.6	41.8	46.3	31.0	**19.2**	34.1	**79.2**	**34.0**
	FlashVid (Fan et al., 2026)	12.6%	✗	**68.8**	**46.9**	**22.9**	**44.6**	**49.9**	**31.5**	17.4	**35.2**	75.6	31.8
	FixedScale	12.3%	✗	61.3	34.3	14.6	37.9	39.6	24.2	13.1	28.4	74.2	29.9
	FixedScale	6.3%	✗	52.7	28.2	11.3	33.2	37.0	22.3	12.0	27.0	71.5	28.0
	ResAdapt (Ours)	6.8%	✗	53.6	29.0	11.8	33.6	37.5	22.5	12.3	27.2	71.8	28.2
	128 Frames												
	Vanilla	100%	✗	72.8	46.0	20.1	45.6	45.8	31.1	19.2	33.9	81.1	36.6
	Random Drop	25.0%	✗	41.6	25.2	10.6	27.4	36.1	21.1	12.7	26.3	79.3	22.4
	ResAdapt (Ours)	16.1%	✗	**64.4**	**37.0**	**15.9**	**39.8**	**40.6**	**26.7**	**15.7**	**30.0**	76.8	**33.3**
	Random Drop	10.0%	✗	32.6	19.0	7.8	21.9	33.5	18.6	11.5	24.8	76.9	19.9
	ToMe (Bolya et al., 2022)	10.0%	✗	61.6	33.8	13.3	**38.1**	**42.4**	**27.6**	**16.6**	**31.4**	**77.4**	31.5
	FixedScale	12.3%	✗	**61.7**	**34.9**	**14.7**	**38.1**	39.9	26.2	15.3	29.5	75.4	32.6
	FixedScale	6.3%	✗	53.7	28.2	11.8	33.6	37.9	24.3	14.3	28.1	73.0	39.1
	ResAdapt (Ours)	6.8%	✗	54.3	28.0	11.7	33.7	38.3	24.5	14.4	28.4	73.2	**43.9**

Q: Evaluate five statements about Urban Geography City Models (concentric zone, Hoyt sector, multiple nuclei, galactic, Latin American); identify which are correct. *Please ignore the Quiz question in last frame of the video.*

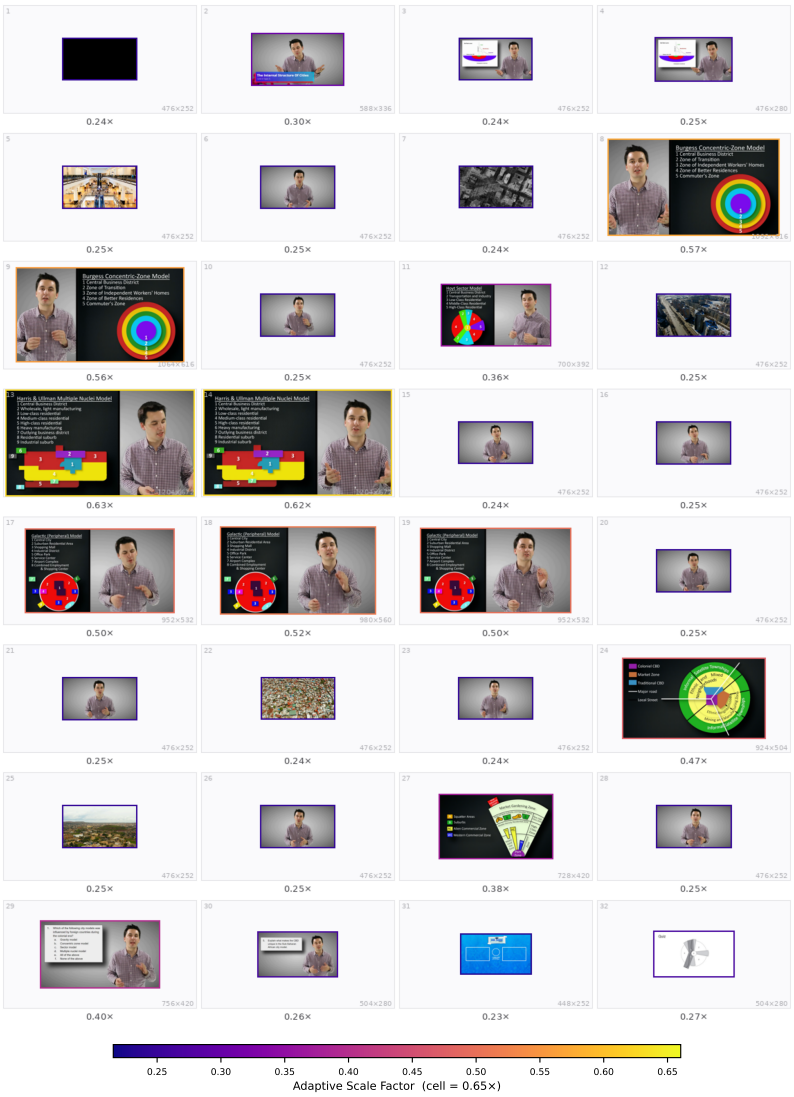


Figure 17: Case 1: Video-MMMU Comprehension (Hu et al., 2025) (Vanilla × → ResAdapt ✓). The policy concentrates resolution on diagram-bearing slide frames, compresses lecturer-only frames, and suppresses the final quiz frame that the prompt explicitly marks as irrelevant.

Q: Watch and learn the video content. Then apply what you learned to answer: Table 11.47 provides a survey of the youngest online entrepreneurs (ages 17–30) whose net worth $\geq$ \$1M. We want to know whether ages and net worth are independent. χ^2 test statistic = ------

Figure 18: Case 2: Video-MMMU Adaptation (Hu et al., 2025) (Vanilla $\times$ $\rightarrow$ ResAdapt $\checkmark$). When the answer depends on reading a numeric table and performing a χ^2 computation, the policy keeps a much higher global budget and strongly upscales the table-bearing frames.

Q: When is the zodiacal light visible from the video? (A) Mar. 19, (B) Mar. 24, (C) Mar. 25, (D) Mar. 29.

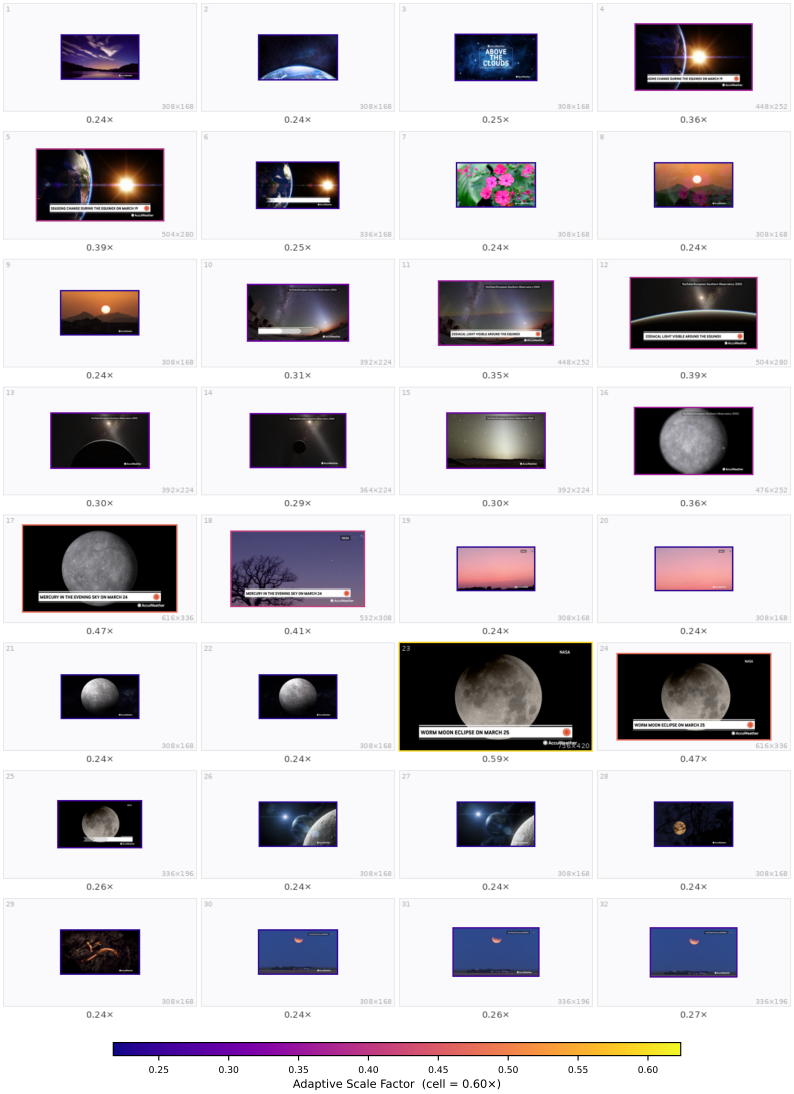


Figure 19: Case 3: VideoMME (Fu et al., 2025a) (Vanilla × → ResAdapt ✓). Frames containing the decisive date overlays are enlarged, while the largely homogeneous sky footage is compressed. The policy spends budget on answer-bearing evidence rather than on the surrounding context.

Q: Which item does the man throw into the trash at the beginning of the video? (A) A fork, (B) A pair of chopsticks, (C) A box of noodles, (D) A spoon.

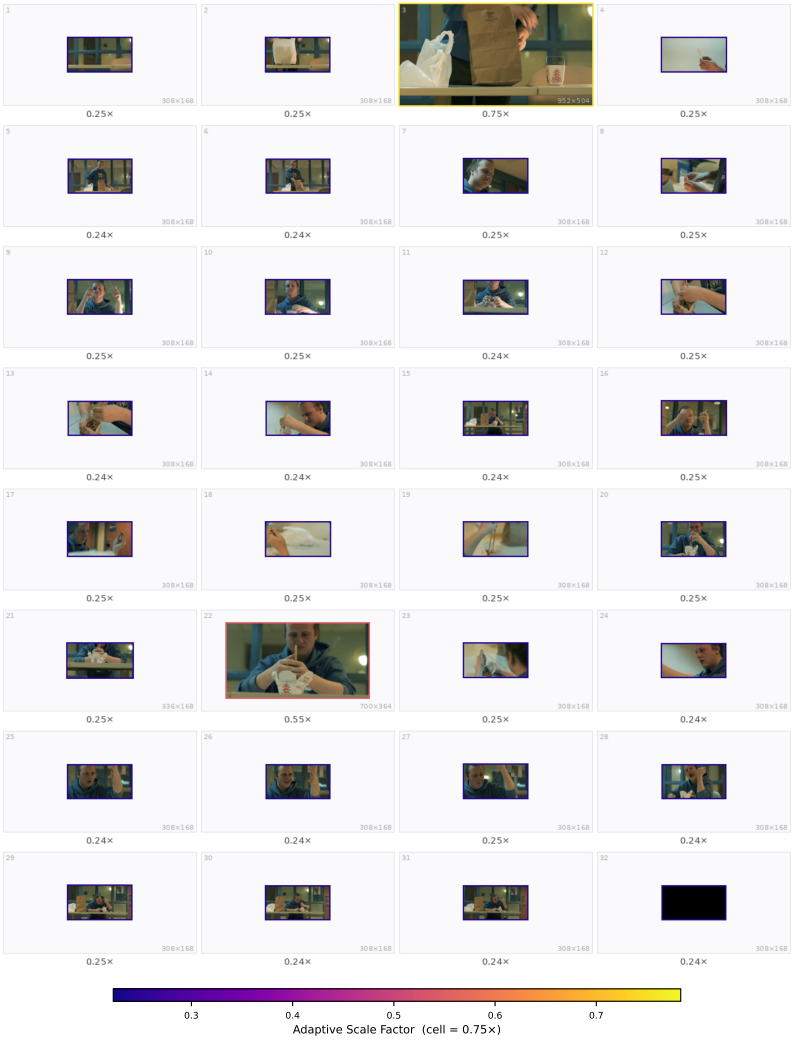


Figure 20: Case 4: VideoMME (Fu et al., 2025a) (Vanilla ✓ → ResAdapt ×; failure case). A nearby frame is enlarged, but the actual fork-bearing frame is compressed. The decisive fine detail is therefore lost at exactly the wrong moment.

Table 7: Exploratory zero-shot transfer to image benchmarks. Parenthetical values denote per-task retention ratio R, and ResAdapt-RL additionally fine-tunes the MLLM via RL.

Model	MathVista testmini	MMMU val	OCRBench	ChartQA	AI2D	TextVQA val
Qwen2.5-VL-7B	49.1(100%)	50.9(100%)	84.2(100%)	83.9(100%)	82.5(100%)	82.9(100%)
Random Drop	44.8(50%)	49.0(50%)	74.8(50%)	71.6(50%)	80.3(50%)	78.1(50%)
ToMe (Bolya et al., 2022)	46.2(50%)	49.6(50%)	79.3(50%)	78.1(50%)	81.9(50%)	81.2(50%)
VisionZip (Yang et al., 2025c)	47.2(50%)	48.6(50%)	79.6(50%)	77.9(50%)	81.9(50%)	81.3(50%)
ResAdapt(Qwen2.5-VL-7B)	45.5(42%)	51.0(29%)	80.0(64%)	85.9(105%)	81.4(41%)	69.6(30%)
ResAdapt-RL(Qwen2.5-VL-7B)	46.7(42%)	50.9(29%)	80.8(64%)	86.6(105%)	81.1(41%)	70.1(30%)
Qwen3-VL-8B	56.1(100%)	53.4(100%)	85.0(100%)	84.0(100%)	83.5(100%)	82.1(100%)
Random Drop	47.3(50%)	48.7(50%)	62.9(50%)	70.2(50%)	79.7(50%)	76.6(50%)
VisionZip (Yang et al., 2025c)	47.8(50%)	50.3(50%)	70.5(50%)	75.0(50%)	80.5(50%)	79.3(50%)
ToMe (Bolya et al., 2022)	49.6(50%)	50.6(50%)	70.3(50%)	75.2(50%)	80.5(50%)	79.4(50%)
ResAdapt(Qwen3-VL-8B)	52.5(42%)	50.9(29%)	82.7(64%)	83.2(105%)	81.2(41%)	67.8(30%)