WHEN REWARDS DISAGREE: REWARD WEIGHTING FOR TOKEN-EFFICIENT VIDEO REASONING

Anonymous authors
Paper under double-blind review

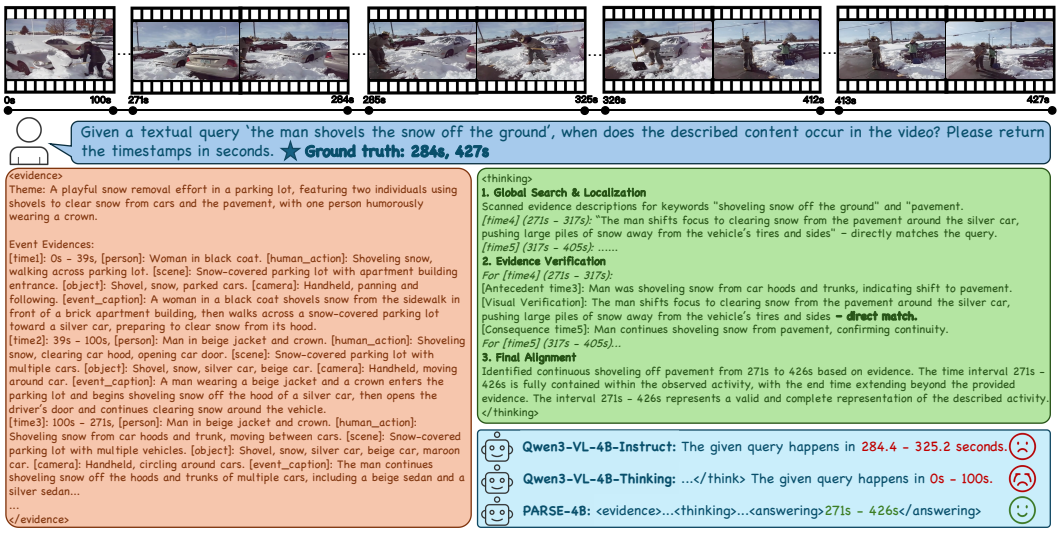

Figure 1: Overview of the proposed structured video reasoning framework. The model first organizes the video into temporally ordered, event-level evidence describing salient entities, actions, scenes, objects, and other relevant cues. This structured representation then constrains the subsequent reasoning process, allowing the model to focus on the relevant temporal dependencies and produce more grounded and interpretable predictions.

ABSTRACT

Human perception interprets video dynamics by constructing structured mental representations of entities, actions, and temporal boundaries before engaging in high-level reasoning. In contrast, current Video-LLMs typically rely on unstructured Chain-of-Thought (CoT), where crucial visual cues are obscured by verbose narration. While recent efforts ground reasoning in intermediate visual evidence, they rely on task-specific schemas that fail to generalize across diverse video understanding tasks. To address this, we introduce **Structured Event Evidence**, a unified, temporally ordered schema designed for general video reasoning. However, post-training a model to extract structured evidence while reasoning introduces a severe multi-objective optimization bottleneck, as candidate rollouts must simultaneously satisfy schema compliance, task accuracy, and reasoning budget. Standard reinforcement learning (RL) algorithms attempt to balance these objectives via static scalarization, enforcing a rigid accuracy–length trade-off across training and obscuring fine-grained reward signals. We resolve this by proposing **Group-Relative Advantage Balancing (GRAB)**, which dynamically computes Pareto-optimal objective weights within each rollout group via a minimum-norm criterion over centered rewards. Operating strictly in reward space, GRAB incurs virtually zero computational overhead ($< 0.002\%$ relative runtime to GRPO) and eliminates brittle weight tuning. Supported by our new **PARSE-60K** dataset (60K annotations across 35K videos) and a three-stage progressive curriculum, we train **PARSE-4B**. Across temporal grounding and video question-answering benchmarks, GRAB generates **13% to 39% fewer tokens** while maintaining competitive accuracy with exhaustively hand-tuned static weights—without requiring hyperparameter weight sweeps.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Large Vision-Language Models (LVLMs) have achieved remarkable success in static image understanding (Liu et al., 2024a; Bai et al., 2023) and are rapidly extending to video domains, where models must model temporal dynamics and reason over unfolding events (Zhang et al., 2023a; Li et al., 2024). Concurrently, Chain-of-Thought (CoT) prompting (Wei et al., 2022) has emerged as a cornerstone paradigm, encouraging models to generate intermediate reasoning trajectories prior to producing a final answer (Bai et al., 2025a) (Bai et al., 2025a).

However, applying unstructured, free-form CoT directly to video is inherently suboptimal. Unlike static images, video content is characterized by high spatio-temporal redundancy and events distributed across broad temporal intervals. Unstructured CoT frequently yields verbose and diffuse narrations that bury salient temporal boundaries and critical visual cues, causing reasoning-oriented models to underperform even basic instruction-tuned baselines (Bai et al., 2025a). In contrast, human perception builds structured mental representations, i.e., first segmenting continuous video into salient entities, actions, and temporal event boundaries, and subsequently performing higher-level cognitive reasoning over these grounded elements (Zacks & Swallow, 2007).

Motivated by this perceptual order, recent efforts attempt to ground video reasoning in explicit visual evidence (Meng et al., 2025; Huang et al., 2026; Zhang et al., 2026). However, existing methods typically formulate evidence representations for narrow, task-specific applications. To overcome this, we introduce **Structured Event Evidence**, a unified, task-agnostic intermediate representation. As illustrated in Figures 1 and 2, our approach organizes salient entities, actions, objects, scenes, and event boundaries into a compact, temporally ordered schema. This structured evidence anchors the subsequent reasoning process, keeping it concise, interpretable, and verifiable against observed temporal visual cues.

Despite its advantages, post-training a model to simultaneously extract structured evidence and execute complex reasoning introduces a severe multi-objective optimization challenge. Unlike standard CoT fine-tuning that primarily optimizes for answer accuracy, learning structured event evidence requires balancing three competing post-training objectives: schema compliance, task accuracy, and reasoning efficiency. Standard Group Relative Policy Optimization (GRPO) (Shao et al., 2024) collapses these multi-dimensional reward vectors into a scalar via fixed linear weighting, triggering two critical failure modes: *Conflicting Objectives*: The length reward and task accuracy metrics push optimization in opposite directions. A weak length penalty causes the model to generate redundant evidence (e.g., one event per second). Conversely, an aggressive penalty forces the model to be concise before it learns accuracy, pruning vital temporal evidence and causing performance to collapse (Wang et al., 2026; Li et al., 2026). *Indistinguishable Objectives*: Collapsing multi-dimensional rewards into a fixed weighted sum obscures individual component contributions. Two candidates earning different reward vectors (e.g., satisfying format and accuracy vs. satisfying format and budget) can yield identical scalar averages, preventing the group from determining which specific objective each response met.

To resolve these optimization bottlenecks without manual hyperparameter tuning or expensive per-objective gradient steps, we propose **Group-Relative Advantage Balancing (GRAB)**. GRAB formulates post-training as group-relative multi-objective policy optimization, dynamically deriving scale-invariant advantage weights by solving a lightweight minimum-norm problem over centered reward vectors within each rollout group. Furthermore, to prevent optimization instability when learning structure and reasoning simultaneously, GRAB operates on top of a three-stage progressive training curriculum supported by our newly constructed **PARSE-60K** dataset.

Our primary contributions are summarized as follows:

- **Multi-Objective Failure Analysis:** We identify and formalize two core optimization bottlenecks in post-training structured video reasoning, i.e., *objective conflicts* and *scalar advantage degeneracy*, demonstrating why static scalarization in standard GRPO fails.
- **The GRAB Algorithm:** We introduce Group-Relative Advantage Balancing (GRAB), a scale-invariant multi-objective RL framework that dynamically balances group-relative advantages via minimum-norm quadratic optimization in reward space. GRAB incurs virtually zero computational overhead and is independent of model parameter scale.
- **Unified Framework & Benchmark:** We present *Structured Event Evidence*, a unified schema for grounded video reasoning, along with **PARSE-60K** (60K annotations over

35K videos) and a three-stage progressive curriculum. Our resulting model, **PARSE-4B**, achieves state-of-the-art performance across both temporal grounding and complex video question-answering benchmarks.

2 Related Work

Video understanding LVLMs. Following the success of image-to-text LVLMs (Liu et al., 2024a), early Video-LLMs largely treated videos as sequences of static frames and relied on token compression to fit spatio-temporal information within a limited context budget (Maaz et al., 2023; Cheng et al., 2024). While effective for coarse video understanding, this frame-centric formulation can overlook temporal grounding and event-level dependencies (Liu et al., 2024b; Chen et al., 2024a). More recent work has introduced time-aware architectures to better capture temporal dynamics (Ren et al., 2024; Huang et al., 2024; Guo et al., 2024), as well as memory- and tool-augmented mechanisms for reasoning over long, untrimmed videos (Song et al., 2024; Wang et al., 2024). Nevertheless, many Video-LLMs still depend heavily on sparse frame sampling and selection heuristics rather than explicitly modeling how events evolve and relate over time (Fu et al., 2024; Xiao et al., 2021). Our approach instead learns an explicit structured representation of salient events and their temporal dependencies before free-form reasoning, providing a compact event-level prior for downstream reasoning and reducing reliance on brittle frame selection.

Reasoning and CoT in LVLMs. CoT prompting improves language-model reasoning by eliciting intermediate steps (Wei et al., 2022), while Multimodal-CoT extends this paradigm by conditioning such reasoning on visual inputs (Zhang et al., 2023b). Video presents a particular challenge for unconstrained CoT: visual inputs are highly redundant, while fine-grained temporal cues and event relationships may be sparse and easily overlooked. As a result, free-form visual reasoning can become verbose, drift away from the relevant evidence, or introduce unsupported details when intermediate steps are insufficiently grounded (Chen et al., 2023). Prior work addresses this in several ways. Tool-augmented approaches, including ReAct-style and MM-ReAct-style methods, introduce external verification or perception steps during reasoning (Yao et al., 2023b; Yang et al., 2023), while methods such as Tree-of-Thoughts and self-consistency explore multiple candidate reasoning trajectories (Yao et al., 2023a; Wang et al., 2023). For video reasoning specifically, recent approaches broadly follow two directions. The first lets the model interact with the visual input during inference, for example by invoking tools or revisiting and re-cropping relevant frames (Zeng et al., 2026; Ding et al., 2025; Lee et al., 2026), at the cost of additional inference-time computation. The second preserves end-to-end reasoning but explicitly grounds it in intermediate evidence, such as timestamped clues, evidence chains, or evidence flows (Meng et al., 2025; Huang et al., 2026; Zhang et al., 2026). We follow this second direction. However, prior evidence representations are tailored to a specific task, so a single reward suffices. Our *Structured Event Evidence* compactly encodes salient events, entities, actions, scenes, and temporal relations as a shared intermediate representation for both task families, which constrains subsequent CoT reasoning to stay concise, relevant, and grounded in the video.

Reinforcement learning in LVLMs. Reinforcement Learning from Human Feedback (RLHF) has become a standard approach for aligning LLMs with human preferences, typically using scalar rewards optimized with PPO (Ouyang et al., 2022; Touvron et al., 2023; Schulman et al., 2017). More recently, GRPO (Shao et al., 2024) has emerged as an efficient alternative that uses group-wise relative comparisons and avoids explicit value-function estimation, while remaining compatible with rule-based rewards. In multimodal and video settings, GRPO has been applied to tasks including reasoning, temporal grounding, and counting (Feng et al., 2025; Wang et al., 2025). Subsequent work improves different aspects of this optimization process. DAPO (Yu et al., 2025) introduces dynamic sampling and decoupled clipping to preserve useful gradients under sparse rewards, while GDPO (Liu et al., 2026) normalizes individual reward components so that weaker signals are not overwhelmed by larger ones. Both, however, still combine the components using fixed weights. So when objectives conflict, their disagreement is averaged rather than resolved. This limitation matters in structured video reasoning, where the model must satisfy structural constraints, task performance, and a reasoning budget at the same time. Multi-objective RL provides a natural framework for such settings, since it seeks solutions that balance competing objectives rather than optimizing a single scalar target (Kong et al., 2025; Li et al., 2025b). Related work on efficient text reasoning treats the reasoning budget as a genuine competing objective rather than a weak regularizer. Wang et al. (2026) show that a larger length coefficient shortens outputs faster but degrades accuracy disproportionately, giving an unfavourable accuracy–length trade-off. Li et al. (2026) address this by adapting the length

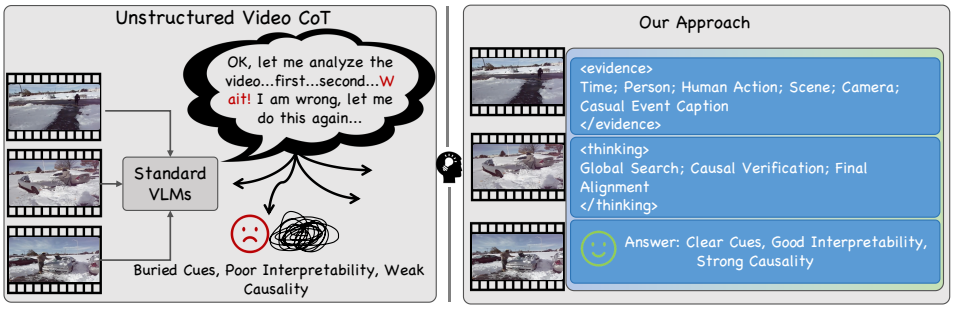

Figure 2: Comparison of unstructured video CoT and our structured reasoning approach. Our method first extracts structured event evidence and then performs evidence-grounded reasoning over temporal dependencies, yielding clearer cues and more interpretable predictions. Video from ActivityNet-Captions (Krishna et al., 2017).

coefficient online through a Lagrangian dual variable, and work on multi-objective alignment reaches a related conclusion that fixed scalarization is limiting (Lu et al., 2026). These approaches, however, adapt one global trade-off over training. We instead solve the weights inside each prompt group with a minimum-norm mix of the centered rewards rather than of policy gradients, which lets the optimization follow the local pattern of agreement and conflict among objectives while keeping task accuracy under a constrained reasoning budget.

3 Methodology

3.1 Problem Formulation and Reward Space

Unlike standard Video-LLMs that rely on unstructured Chain-of-Thought (CoT) and are prone to producing verbose narration, our framework grounds video reasoning in explicit *Structured Event Evidence* to anchor high-level reasoning to concrete temporal boundaries and entity interactions. Formally, given an input video V and a query q, the model generates a response sequence $o = (e, z, a)$ containing structured event evidence e, a multi-step reasoning path z, and a final answer a. During post-training, every response o is evaluated across M distinct objectives comprising schema compliance r_{schema}, task accuracy r_{acc}, and reasoning efficiency/budget r_{len} (Section G.1). We express the per-sample objective rewards as a vector $\mathbf{r}_i = [r_{i,1}, \ldots, r_{i,M}]^\top \in \mathbb{R}^M$.

3.2 Group-Relative Advantage Balancing (GRAB)

Motivations and Limitations of Static Scalarization. In standard Group Relative Policy Optimization (GRPO) (Shao et al., 2024), the vector reward $\mathbf{r}_i \in \mathbb{R}^M$ is scalarized as $\bar{r}_i = \mathbf{w}^\top \mathbf{r}_i$ using a fixed preference vector $\mathbf{w} \in \Delta^{M-1}$, where Δ^{M-1} denotes the probability simplex. Applying the same weights to every rollout group enforces a global, static trade-off among the M objectives. For joint evidence extraction and reasoning, this static scalarization exhibits two primary optimization failure modes:

1. *Conflicting Objectives*: The length reward r_{len} and task accuracy actively compete: insufficient penalties induce redundant evidence over-generation, whereas aggressive penalties prematurely prune critical temporal cues, degrading accuracy (Wang et al., 2026).
2. *Objective Indistinguishability:* Linear scalarization maps non-identical reward vectors $\mathbf{r}_i \neq \mathbf{r}_j$ to identical scalar advantages $\bar{r}_i = \bar{r}_j$ (Figure 3), collapsing multi-dimensional feedback and obscuring which specific objective a candidate satisfied.

While per-objective reward normalization (Liu et al., 2026) mitigates scale imbalances, it leaves directional objective conflicts unresolved. To address these limitations, we introduce Group-Relative Advantage Balancing (GRAB) to dynamically calculate scale-invariant, Pareto-optimal advantage weights for each rollout group without computationally expensive per-objective gradient operations. GRAB directly addresses this by dynamically deriving scale-invariant, group-relative advantage weights.

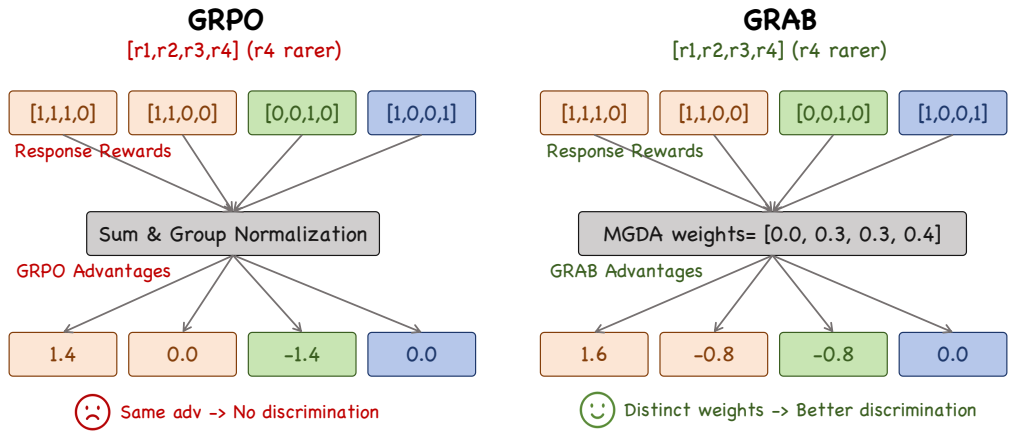

Figure 3: GRPO vs. GRAB on a rollout group. GRPO averages the four rewards, assigning identical scores and advantages to $[1, 1, 0, 0]$ and $[1, 0, 0, 1]$. GRAB applies dynamic minimum-norm weights (Algorithm 1) to centered rewards before z-score normalization, correctly distinguishing response preferences.

Group-Centered Reward Vectors. For a prompt q, we sample a group of G candidate responses $\{o_1, \ldots, o_G\} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$. Following the group-relative paradigm, we isolate the relative quality of response o_i on objective $m \in \{1, \ldots, M\}$ by subtracting the group mean:

$$\delta_{i,m} = r_{i,m} - \frac{1}{G} \sum_{j=1}^{G} r_{j,m}. \tag{1}$$

where $r_{i,m}$ denotes the reward for response o_i on objective m. We organize these relative scores into a centered reward matrix $\mathbf{D}_q \in \mathbb{R}^{G \times M}$ where $[\mathbf{D}_q]_{i,m} = \delta_{i,m}$. Each column represents a *directional preference vector* indicating whether responses perform above or below the group average for objective m.

Standardized Minimum-Norm Advantage Weighting. Inspired by the Multiple Gradient Descent Algorithm (MGDA) (Désidéri, 2012), GRAB solves for optimal objective weights in a standardized space. To ensure scale invariance, each column of $\mathbf{D}_q$ is divided by its sample standard deviation $\sigma_{q,m}$, yielding the standardized matrix $\hat{\mathbf{D}}_q$. Objectives with variance below a threshold $\sigma_{q,m} < \tau_\sigma$ are pruned, as they provide uninformative contrast across the group; we denote the set of remaining active objectives as $\mathcal{V}$. We apply the Frank-Wolfe algorithm (Jaggi, 2013) to identify the minimum-norm direction within the simplex $\Delta_{|\mathcal{V}|}$:

$$\boldsymbol{\alpha}_q^* = \arg\min_{\boldsymbol{\alpha} \in \Delta_{|\mathcal{V}|}} \left\| \hat{\mathbf{D}}_q \boldsymbol{\alpha} \right\|^2,$$
$$\text{where} \quad \Delta_{|\mathcal{V}|} = \{\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{V}|} \mid \textstyle\sum_{m \in \mathcal{V}} \alpha_m = 1,\ \alpha_m \geq 0\}. \tag{2}$$

Because every column in $\hat{\mathbf{D}}_q$ has norm $\|\hat{\mathbf{d}}_{q,m}\|^2 = G$, the Gram matrix simplifies directly to $\hat{\mathbf{D}}_q^\top \hat{\mathbf{D}}_q = G\,\mathbf{C}_q$, where $\mathbf{C}_q \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is the reward correlation matrix across the group. Thus, Eq. 2 is exactly equivalent to:

$$\boldsymbol{\alpha}_q^* = \arg\min_{\boldsymbol{\alpha} \in \Delta_{|\mathcal{V}|}} \boldsymbol{\alpha}^\top \mathbf{C}_q\, \boldsymbol{\alpha}, \tag{3}$$

If the resulting minimum norm is smaller than a threshold η, we fall back to uniform weights over $\mathcal{V}$. By optimizing over $\mathbf{C}_q$, $\boldsymbol{\alpha}_q^*$ dynamically suppresses highly correlated or redundant objectives and assigns higher relative weight to conflicting objectives that exhibit high variance across the rollout group.

Advantage Scalarization and Policy Update. After obtaining $\boldsymbol{\alpha}_q^*$, we project these weights back onto the raw centered rewards $\mathbf{D}_q$ to preserve scale fidelity, followed by group-wise z-score normalization:

$$A_i^{\text{raw}} = \sum_{m \in \mathcal{V}} \alpha_{q,m}^* \, \delta_{i,m}, \qquad A_i = \frac{A_i^{\text{raw}} - \mu_A}{\sigma_A + \epsilon}. \tag{4}$$

This two-step formulation—optimizing weights in standardized space for scale-invariance while applying them in centered space for fidelity—allocates relative advantages across objectives while maintaining GRPO's update scale. Finally, A_i is integrated into the clipped surrogate policy objective:

$$\mathcal{L}(\theta) = -\tfrac{1}{|\mathcal{B}|} \textstyle\sum_i \left[\min\left(\rho_i A_i,\ \text{clip}(\rho_i, 1{-}\varepsilon, 1{+}\varepsilon) A_i\right) - \beta \mathbb{D}_{\text{KL}}\right] \tag{5}$$

where $\rho_i(\theta) = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}$, $|\mathcal{B}|$ is the batch size, ε is the clipping hyperparameter, and β controls the KL penalty relative to reference policy π_{ref}.

3.3 Progressive Training Curriculum

While GRAB resolves multi-objective conflicts during post-training, attempting to optimize structural formatting, temporal grounding, and multi-step reasoning simultaneously from a raw base model leads to severe convergence instability. We therefore decompose training into a three-stage progressive curriculum:

1. **Evidence Training (Stage 1):** The model is trained exclusively to generate high-quality structured evidence, learning to ground salient entities, actions, scenes, objects, and temporal event boundaries using supervised cross-entropy loss.
2. **Thinking Warm-Start (Stage 2):** Conditioned on the extracted structured evidence, the model learns to execute multi-step reasoning through task-oriented supervision across diverse temporal grounding and visual reasoning tasks (detailed prompts in Appendix Section E.3).
3. **RL Post-Training (Stage 3):** Stage 3 applies GRAB during RL post-training to dynamically resolve the competing objectives of joint evidence extraction, reasoning quality, and concise generation.

3.4 PARSE-60K Dataset Engine

To provide the tailored supervision required across all three stages, we construct **PARSE-60K**, comprising 60K annotations over 35K clips sampled from Panda-70M (Chen et al., 2024b). We exclude Charades and ActivityNet from training, and remove any video whose ID occurs in a reported evaluation set. The dataset supports all three stages on a shared set of videos: Stage 1 provides event-boundary and entity annotations; Stage 2 provides step-by-step reasoning traces; and Stage 3 provides task-oriented prompts. Figure 4 summarises the construction pipeline; prompts, scoring rubrics, and additional implementation details are provided in Section D.

Because Panda-70M contains a single caption per video but no event timestamps, we first apply shot detection and merge or split the resulting segments to obtain clips of 10–30 seconds. Humans audit a randomly sampled half of the resulting boundaries. We generate evidence annotations using Qwen3-VL-235B-Instruct (Bai et al., 2025a) and Gemini-2.5-Pro (Comanici et al., 2025), which alternately serve as generator and judge. Human review covers 30K of the 60K annotations (three annotators; mean scores 8.5, 7.9, and 8.2). If any sample receives a score below 6, its entire batch of 1,000 is regenerated; this criterion triggered regeneration for 28.3% of batches. The RL data comprise 53% temporal grounding, 21% spatial VQA, 20% reasoning VQA, and 6% temporal VQA.

4 Experiments

We use *Qwen3-VL-4B-Instruct* as our base model with a three-stage training pipeline. Stages 1–2 use LoRA fine-tuning; Stage 3 (RL) performs full-parameter updates with four reward objectives (*Format, Linear IoU, Multi-choice Accuracy, Length*). We set $\text{fps} = 1$ and evaluate all Qwen3 baselines under

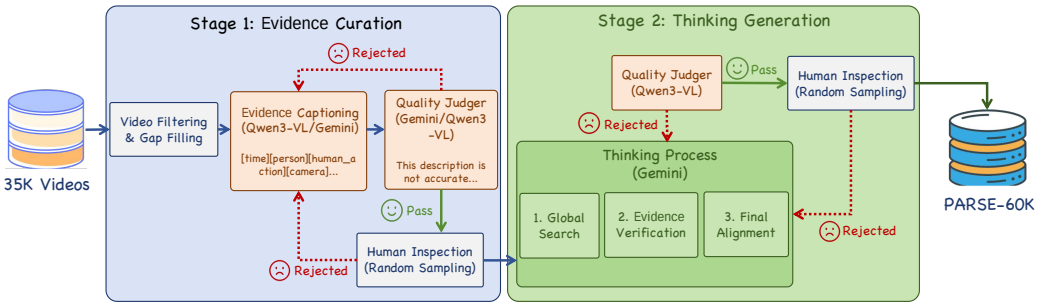

Figure 4: How we build PARSE-60K. We first drop any video whose ID appears in an eval set. Stage 1 then cuts events, fills gaps, writes evidence, and judges quality. Stage 2 writes thinking traces and judges them again.

the same setting; other baselines use official results (typically $\mathrm{fps} = 2$). Every number we produce ourselves is a three-seed mean, and Appendix H.7 reports the spread. ***Additional analyses including TempCompass/EventBench evaluation, 8B scalability, GRAB overhead, data quality, the scope of our claims, and reasoning quality are provided in Appendices.***

4.1 Ablation Studies

Ablation setup. We evaluate component effectiveness on Charades-TimeLens (Zhang et al., 2025a) and ActivityNet-Captions (Krishna et al., 2017) for temporal grounding, and VideoMME (Fu et al., 2024) and MLVU (Zhou et al., 2025) for general understanding. Table 1 reports component ablations and policy optimization baselines. The component ablations remove the evidence block (*w/o Evidence*), structured thinking (*w/o Thinking*), or RL (*w/o RL*). The optimization baselines include *GRPO*, *DAPO* (Yu et al., 2025), *GDPO* (Liu et al., 2026), *Fixed Weight (Accuracy)*, and *Lagrangian (Length)*. We evaluate the policy optimization methods with group sizes $G = 8$ and $G = 16$. All policy optimization variants start from the same Stage-2 checkpoint and use the same RL corpus, so they differ only in the optimization algorithm. For *Fixed Weight (Accuracy)*, the reward weights are $(w_{\text{fmt}}, w_{\text{task}}, w_{\text{len}}) = (1, 2, 1)$. For *Lagrangian (Length)*, the format and task weights remain fixed while one global coefficient on r_{len} is adjusted online. We tune its budget so that its final output length matches GRAB's. The update rule is given in Appendix H.5. Dataset details appear in Appendix C.

Component ablations. As shown in Table 1, the three stages serve different roles. Removing structured thinking (*w/o Thinking*) lowers both grounding benchmarks below the untrained backbone. This shows that the evidence becomes useful only when the model reasons over it. Without RL, the model retains the full evidence and thinking supervision and remains above the backbone on grounding, but its outputs are much longer than those of RL-trained models. Removing the evidence block (*w/o Evidence*) keeps the thinking supervision and RL, but loses to *w/o RL* on every accuracy column. RL alone therefore does not recover the benefit of the schema. Stages 1 and 2 teach the format, while Stage 3 trades accuracy against the reasoning budget.

Per-group weights versus one global coefficient. Lagrangian (Length) keeps the format-to-task ratio of Fixed Weight (Accuracy) and adapts only the length coefficient, using one global value for the entire run (Appendix H.5). We tune its budget so that it ends at the same output length as GRAB, which leaves accuracy as the only axis of comparison. Under this matched budget, GRAB is more accurate on every accuracy column in Table 1. Reaching a short policy is therefore not the difficulty; retaining accuracy while getting there is. An online dual variable still commits to a single exchange rate at any point in training, but the appropriate rate differs across prompt groups, because a group whose rollouts already agree can be shortened while a group whose rollouts still disagree cannot. GDPO falls short for the same reason, since it normalizes each reward but continues to mix them with fixed weights.

Fixed weights at a smaller budget. Fixed Weight (Accuracy) is the strongest static configuration we found, and it is the comparison that matters most for a multi-objective claim. GRAB matches or exceeds it on grounding at every IoU threshold while producing substantially shorter responses, and the two methods trade places on the remaining general-understanding columns of Table 1. The insight is not that one operating point dominates the other, but that GRAB reaches a competitive point without anyone selecting it. A static weighting commits to one exchange rate between task

Model	**Charades-TimeLens** (R1@IoU)				**ActivityNet-Captions** (R1@IoU)				**VideoMME**		**MLVU**		
	@0.3	@0.5	@0.7	Len	@0.3	@0.5	@0.7	Len	Acc	Len	TR	Ego	Len
Baselines 1fps													
Qwen3-VL-4B-Inst.* Bai et al. (2025a)	56.5	37.8	18.4	-	50.2	35.8	21.6	-	63.4	-	80.1	60.3	-
Qwen3-VL-4B-Think.* Bai et al. (2025a)	56.3	37.4	17.8	1439	47.8	31.7	18.8	1895	63.1	4016	79.5	59.0	5192
Ablations 1fps													
w/o Evidence = 16	54.0	37.0	16.5	553	50.7	34.3	23.2	1158	60.8	2961	76.2	56.8	3128
w/o Thinking = 16	50.1	34.2	14.1	673	45.5	31.2	17.8	2884	57.5	8432	75.6	55.4	10118
w/o RL	57.0	38.2	18.8	1174	53.4	36.6	23.7	3386	62.5	11258	81.2	61.3	13187
GRPO = 8	57.2	38.3	19.4	842	55.8	37.9	23.6	1955	60.2	4179	81.5	61.9	5635
DAPO = 8	57.4	38.7	19.1	816	56.0	37.3	24.0	1829	61.7	4108	80.1	61.4	5540
GDPO = 8	57.8	39.6	20.4	734	56.2	38.0	24.6	1703	62.6	3987	80.8	61.8	5376
GRAB = 8	59.1	41.1	21.6	**503**	56.7	40.2	25.1	**1116**	64.7	**3558**	82.6	63.3	4798
GRPO = 16	59.8	40.6	21.8	846	60.2	41.0	24.7	2038	65.6	4388	83.1	63.8	5327
DAPO = 16	61.2	41.3	21.2	714	61.2	41.3	26.1	1921	65.1	4313	83.6	64.1	5238
GDPO = 16	61.2	41.6	22.2	770	62.2	41.0	26.7	1722	65.6	4186	83.9	64.8	5083
Fixed Weight (Accuracy) = 16	63.2	42.9	24.5	931	63.9	44.0	28.0	2013	**68.0**	4310	86.1	**67.2**	5310
Lagrangian (Length) = 16	62.1	40.2	23.5	556	62.1	42.2	26.6	1212	65.7	3806	86.1	64.4	**4489**
GRAB = 16	**63.7**	**43.4**	**24.6**	579	**64.7**	**44.2**	**28.1**	1231	67.7	3736	**86.9**	66.4	4536

Table 1: Ablation study at the 4B scale on temporal grounding and general video understanding. We remove evidence, structured thinking, or RL, and compare GRAB against GRPO, DAPO, GDPO, fixed reward weighting, and a Lagrangian length baseline. @τ is Recall@1 at IoU threshold τ, VideoMME accuracy excludes subtitles, and TR and Ego are the Topic Reasoning and Ego Reasoning sub-tasks of MLVU. Len is the average number of generated tokens per response, so lower is better at comparable accuracy. A row ending in $= 8$ or $= 16$ gives the group size of its RL stage, and *w/o RL* has none. Entries marked "-" are not reported, since Qwen3-VL-4B-Instruct returns a bare answer whose length is not comparable with the reasoning rows.

Model	**Charades-TimeLens**			**ActivityNet-TimeLens**			**ActivityNet-Captions**		
	R1@0.3	R1@0.5	R1@0.7	R1@0.3	R1@0.5	R1@0.7	R1@0.3	R1@0.5	R1@0.7
Close-Source Models 2fps									
GPT-4o (Hurst et al., 2024)	60.6	44.5	23.5	55.2	41.4	25.8	-	-	-
GPT-5 (OpenAI, 2025)	59.3	42.0	22.0	57.4	44.9	30.4	-	-	-
Gemini-2.0-Flash (Comanici et al., 2025)	66.4	53.5	27.1	62.9	54.0	37.7	50.4	33.2	19.9
Gemini-2.5-Flash (Comanici et al., 2025)	68.7	56.1	30.6	66.8	57.5	41.3	51.2	34.7	21.0
Gemini-2.5-Pro (Comanici et al., 2025)	74.1	61.1	34.0	72.3	64.2	47.1	-	-	-
Open-Source Models 2fps									
VideoChat-R1-7B (Li et al., 2025a)	51.9	30.8	11.7	35.0	23.9	11.3	-	33.3	16.7
TRACE-7B (Guo et al., 2024)	-	-	-	-	-	-	54.0	37.7	24.0
Time-R1-7B (Wang et al., 2025)	57.9	32.0	16.9	44.8	31.0	19.0	58.1	39.0	21.4
Qwen2.5-VL-7B (Bai et al., 2025b)	59.7	37.8	16.6	44.1	31.0	16.1	34.5	20.8	11.2
Open-Source Models 1fps									
VideoChat-R1-7B (Li et al., 2025a)	48.7	27.1	9.7	-	-	-	-	-	-
Time-R1-7B (Wang et al., 2025)	55.9	30.4	13.5	-	-	-	56.9	37.2	20.1
TRACE-7B (Guo et al., 2024)	-	-	-	-	-	-	52.7	35.9	22.8
Qwen3-VL-4B-Instruct* (Bai et al., 2025a)	56.5	37.8	18.4	50.2	36.2	25.2	50.2	35.8	21.6
Qwen3-VL-4B-Thinking* (Bai et al., 2025a)	56.3	37.4	17.8	48.2	36.7	24.9	47.8	31.7	18.8
PARSE-4B	**63.7**	**43.4**	**24.6**	**54.7**	**41.4**	**26.7**	**64.7**	**44.2**	**28.1**

Table 2: Experiments on the public video temporal understanding benchmarks. * means this model is reproduced under the same video settings of our model, otherwise, the results are from official reports with more video frames.

performance and the reasoning budget before training begins (Wang et al., 2026; Li et al., 2026), and the sweep in Appendix H.6 shows that this choice moves both accuracy and length considerably. Every candidate rate costs a separate run, whereas GRAB resolves the weights inside each group and leaves no rate to tune.

Uniform-weight optimizer baselines. GRPO, DAPO, and GDPO use the uniform reward weights in Table 4. Against equal weighting, GRAB leads on every accuracy column at both group sizes and is also the most concise of the four methods, so the gain does not come from spending more tokens. Its margin over GRPO persists at both group sizes, which indicates that larger groups alone do not resolve the conflict between objectives. Averaging the reward components discards information that is present in the group, and recovering that information requires weighting the components rather than sampling more responses.

4.2 Comparisons with Existing Video-LLMs

We compare PARSE-4B against leading Video-LLMs on temporal grounding and general video understanding in Table 2 and Table 3, respectively. To ensure fair comparison, we evaluate key

	ETBench								VideoMME	MLVU		NExT-GQA
Metrics	RAR Acc	ECA Acc	RVQ Acc	TVG F1	EPM F1	TAL F1	TEM Rec	GVQ Acc	Acc	TR (Acc)	Ego (Acc)	Acc
Close-Source Models 2fps												
GPT-4o (Hurst et al., 2024)	27.8	27.3	57.7	40.4	4.5	20.0	13.6	-	71.9	-	-	-
GPT-4v (OpenAI, 2023)	33.3	40.9	46.2	27.0	1.8	18.0	23.9	-	59.9	-	-	-
Open-Source Models												
VideoLLama2-7B (Cheng et al., 2024)	28.8	27.4	28.0	0.1	0.0	0.0	0.0	-	-	-	-	-
VTimeLLM-7B (Huang et al., 2024)	28.4	31.0	28.0	7.6	1.9	18.2	6.9	1.9	-	-	-	17.4
TRACE-7B (Guo et al., 2024)	29.4	28.8	42.6	46.8	12.3	21.6	17.8	52.4	49.6	-	-	-
TinyLLaVA-Video-R1-3B (Zhang et al., 2025b)	-	-	-	-	-	-	-	-	46.6	-	-	-
Open-o3-Video (Meng et al., 2025)	-	-	-	-	-	-	-	-	63.6	-	-	-
Qwen2.5-VL-7B (Bai et al., 2025b)	41.0	42.0	51.4	20.1	2.9	15.3	22.3	32.7	62.9	74.9	50.0	59.5
Qwen2.5-VL-3B (Bai et al., 2025b)	38.4	47.2	52.2	19.9	2.6	14.8	18.1	55.1	61.5	-	-	-
Reproduced by us 1fps												
Qwen2.5-VL-3B + GRAB*	42.7	50.4	54.1	32.0	10.1	16.9	21.3	55.7	64.0	-	-	-
Qwen3-VL-4B-Instruct* (Bai et al., 2025a)	53.1	**54.1**	54.3	64.6	14.8	**20.1**	9.6	57.7	63.4	80.1	60.3	72.1
Qwen3-VL-4B-Thinking* (Bai et al., 2025a)	53.0	51.0	54.2	64.3	15.7	20.0	20.4	55.6	63.1	79.5	59.0	66.6
PARSE-4B	**56.1**	53.6	**57.2**	**67.3**	**16.9**	19.9	**27.9**	**59.6**	**67.7**	**86.9**	**66.4**	**73.6**

Table 3: Experiments on the public general video understanding benchmarks. TR and Ego mean Topic Reasoning and Ego Reasoning, respectively. Rows marked * are run by us at 1 fps, under the same video settings as PARSE-4B. The other rows are taken from official reports, which usually use more video frames.

open-source baselines under both the 2 fps and 1 fps settings. Beyond the quantitative gains, we highlight three qualitative insights that emerge from the results.

PARSE-4B improves over its own 4B backbone. Under identical evaluation settings, PARSE-4B improves over Qwen3-VL-4B-Instruct on every benchmark in Table 2 and on six of the eight ETBench sub-tasks in Table 3. Table 1 shows this gap is the full pipeline, not the schema alone: Stage 1–2 help a little on grounding, and Stage 3 RL does most of the work. This same-base comparison rules out that the gains stem from the backbone alone. We interpret the structured output as an *information bottleneck* complementary to the base model: the backbone supplies fine-grained perception, while the evidence schema imposes a discrete, time-ordered abstraction so that reasoning operates on compact event-level representations rather than thousands of redundant frame tokens.

Explicit event timestamps improve localization. A consistent pattern in Table 2 is that PARSE-4B improves at every IoU threshold over the same backbone. The explicit event timestamps encoded in the evidence provide hard anchors for the start and end of activities, enabling the evidence verification stage to align predictions with ground-truth intervals. The absolute gain is larger when the reference annotation is looser. ActivityNet-Captions has broad boundaries and improves the most, while ActivityNet-TimeLens re-annotates the same videos with tighter ones and improves less. The model finds the right region on both, and it refines the boundary itself mainly when the reference granularity is close to the event granularity our schema assumes.

Unstructured thinking is counterproductive for video. Across nearly all benchmarks in both tables, *Qwen3-VL-4B-Thinking* underperforms its *Instruct* counterpart, a trend also observed by (Bai et al., 2025a). The token budget is spent in the wrong place: unstructured CoT spends tokens on verbose self-corrections, repetitive scene narrations, and speculative tangents, leaving insufficient capacity for the actual temporal reasoning that matters. Front-loading a compact evidence block leaves downstream thinking tokens for evidence-grounded inference rather than redundant perception. On NExT-GQA, PARSE-4B reaches 73.6 against 72.1 for Instruct and 66.6 for Thinking, the widest gap to the Thinking baseline in either table, on a benchmark built around Why and How questions.

5 Conclusion

We present *PARSE-4B*, trained with GRAB, which weights the rewards inside each prompt group instead of fixing one weight vector for the whole run. GRAB arrives at a working point nobody selected: it improves on seven of nine accuracy columns over hand-tuned static weights at 13 to 39% fewer tokens, and it is more accurate on all nine than an online global length coefficient tuned to the same output length. Which member of the static family to use is a question GRAB does not ask. We instantiate the setting on *Structured Event Evidence* as the output format that creates the conflict between the schema, the task metric and the reasoning budget, and curate *PARSE-60K* to teach that format before RL. GRAB reads only the reward matrix of a group, so it carries over to any setting where several rewards score the same rollouts.

REFERENCES

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, et al. Qwen3-vl technical report. In *arXiv preprint arXiv:2511.21631*, 2025a.

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.

Jr-Jen Chen, Yu-Chien Liao, Hsi-Che Lin, Yu-Chu Yu, Yen-Chun Chen, and Frank Wang. Rextime: A benchmark suite for reasoning-across-time in videos. *Advances in Neural Information Processing Systems*, 37:28662–28673, 2024a.

Keqin Chen, Zhao Zhao, Yizhe Yang, Jiannan Liu, Zichang Tian, and Weijie Chen. Shikra: Unleashing multimodal LLM's referential dialogue magic. In *arXiv preprint arXiv:2306.15195*, 2023.

Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

Yang Ding, Yizhen Zhang, Xin Lai, Ruihang Chu, and Yujiu Yang. Videozoomer: Reinforcement-learned temporal focusing for long video reasoning. *arXiv preprint arXiv:2512.22315*, 2025.

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, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis, 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.

Yongxin Guo, Jingyu Liu, Mingda Li, Qingbin Liu, Xi Chen, and Xiaoying Tang. Trace: Temporal grounding video llm via causal event modeling. *arXiv preprint arXiv:2410.05643*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14271–14280, 2024.

Yanxiang Huang, Guohua Gao, and Zhaoyang Wei. Video evidence to reasoning: Efficient video understanding via explicit evidence grounding. *arXiv preprint arXiv:2601.07761*, 2026.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pp. 427–435. PMLR, 2013.

Lingxiao Kong, Cong Yang, Susanne Neufang, Oya Deniz Beyan, and Zeyd Boukhers. Emorl: Ensemble multi-objective reinforcement learning for efficient and flexible llm fine-tuning, 2025.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *International Conference on Computer Vision (ICCV)*, 2017.

Daeun Lee et al. Visioncoach: Reinforcing grounded video reasoning via visual-perception prompting. *arXiv preprint arXiv:2603.14659*, 2026.

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*, 2025a.

Yanhao Li, Lu Ma, Jiaran Zhang, Lexiang Tang, Wentao Zhang, and Guibo Luo. Leash: Adaptive length penalty and reward shaping for efficient large reasoning model. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2846–2856, 2026.

Yanwei Li, Chengyao Wang, and Jieru Zhu. Llama-vid: An image is worth 2 tokens in large video-language models. In *ECCV*, 2024.

Zhuo Li, Guodong Du, Weiyang Guo, Yigeng Zhou, Xiucheng Li, Wenya Wang, Fangming Liu, Yequan Wang, Deheng Ye, Min Zhang, and Jing Li. Multi-objective large language model alignment with hierarchical experts, 2025b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2024a.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Minjia Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Chang Wen Chen, and Ying Shan. E.t. bench: Towards open-ended event-level video-language understanding. In *Neural Information Processing Systems (NeurIPS)*, 2024b.

Yining Lu, Zilong Wang, Shiyang Li, Xin Liu, Changlong Yu, Qingyu Yin, Zhan Shi, Zixuan Zhang, et al. Learning to optimize multi-objective alignment through dynamic reward weighting. *Transactions of the Association for Computational Linguistics*, 2026.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL*, 2023.

Jiahao Meng, Xiangtai Li, Yunhai Tong, et al. Open-o3 video: Grounded video reasoning with explicit spatio-temporal evidence. *arXiv preprint arXiv:2510.20579*, 2025.

OpenAI. Gpt-4v(ision) system card, September 2023. URL `https://openai.com/index/gpt-4v-system-card/`.

OpenAI. Introducing gpt-5, 2025. Available from OpenAI announcement, August 7, 2025.

Long Ouyang et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

John Schulman et al. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Hugo Touvron et al. Llama: Open and efficient foundation language models. In *Proceedings of the 2023 Conference on Machine Learning and Systems*, 2023.

Chen Wang, Hexuan Deng, Yining Zhang, Yuchen Zhang, Jionghao Bai, Zhaochun Li, Ge Lan, and Yue Wang. Implicit compression regularization: Concise reasoning via internal shorter distributions in rl post-training. *arXiv preprint arXiv:2605.07316*, 2026.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision (ECCV)*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

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*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021.

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.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

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

Jeffrey M Zacks and Khena M Swallow. Event perception: a mind-brain perspective. *Psychological bulletin*, 133(2), 2007.

Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, et al. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. *arXiv preprint arXiv:2601.23224*, 2026.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *EMNLP*, 2023a.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, Ying Shan, and Limin Wang. Timelens: Rethinking video temporal grounding with multimodal llms. *arXiv preprint arXiv:2512.14698*, 2025a.

Wenhao Zhang, Kuanwei Lin, Xuyi Yang, Wei Gao, and Ge Li. Eflow: Learning evidence flow for long-video reasoning with adaptive reflection. *arXiv preprint arXiv:2607.00867*, 2026.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Tinyllava-video-r1: Towards smaller lmms for video reasoning. *arXiv preprint arXiv:2504.09641*, 2025b.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023b.

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 (CVPR)*, pp. 13691–13701, 2025.

CONTENTS OF APPENDIX

A Implementation Details **13**

B Limitations **14**

C Detailed Information of Evaluation Benchmarks **14**
C.1 Video Temporal Grounding Benchmarks 14
C.2 General Video Understanding Benchmarks 15

D Data Construction Pipeline Details **15**

E More details of PARSE-60K **17**
E.1 Video Source 17
E.2 RL Training Data Details 17
E.3 Data Curation Prompts 18

F Instruction Tuning Data Format **18**

G More details of Reinforcement Learning **19**
G.1 Detailed Rewards Design 19
G.2 The GRAB Algorithm 20
G.3 Live objectives per group 21
G.4 Frank-Wolfe Algorithm for Solving the Minimum-Norm Problem 21

H Additional Experiments and Analyses **22**
H.1 TempCompass and EventBench Evaluation 22
H.2 Scalability to 8B Model 22
H.3 GRAB Computational Overhead 22
H.4 Evidence Quality Evaluation 23
H.5 Lagrangian (Length) Baseline 23
H.6 Static Weight Sweep 23
H.7 Seed Variance 24
H.8 The Weights the Solver Returns 24
H.9 Reasoning Quality Evaluation 25

A IMPLEMENTATION DETAILS

We use *Qwen3-VL-4B-Instruct* as our base model. Since the 4B model is more susceptible to general-capability forgetting, we adopt stage-specific hyperparameter settings across our training stages. For Stage 1, we fine-tune with LoRA (rank 64, $\alpha = 128$), a learning rate of 1×10^{-5}, and weight decay of 1×10^{-4}. In Stage 2, we keep the same LoRA (Hu et al., 2022) configuration but increase the learning rate to 1×10^{-4} and weight decay to 2×10^{-4}.

For RL Stage 3, we perform full-parameter updates with a learning rate of 5×10^{-6}, $\text{num_generation} = 16$ for the final PARSE-4B model, and video max tokens per frame set to 64. The ablation study in Table 1 additionally reports all policy optimization variants under $\text{num_generation} = 8$. We set $\text{video_fps} = 1.0$ and $\text{epoch} = 1.0$ across all stages, with $\text{video_max_frames} = 256$ for Stages 1–2 and $\text{video_max_frames} = 128$ for Stage 3 to save memory. During evaluation, we re-evaluate all Qwen3 series models under the same settings as ours ($\text{fps} = 1$, $\text{max_frames} = 256$). For other models, results are taken from official reports, typically with $\text{fps} = 2$ and $\text{max_frames} = 2048$. Every number we produce ourselves is the mean of three random seeds, and Section H.7 reports the spread across them on Charades-TimeLens.

Table 4 collects the reward and solver hyperparameters of Stage 3, including the ones the length reward and the static baselines depend on.

Symbol	Meaning	Value
L_{max}	hard maximum response length (tokens)	4096
L_{buffer}	soft buffer below L_{max} (tokens)	2500
L_{target}	$L_{max} - L_{buffer}$, start of the length penalty	1596
w^{acc}	Fixed Weight (Accuracy), over (fmt, IoU, MC, len)	$(1, 2, 2, 1)$
w^{def}	weight vector of the GRPO, DAPO and GDPO runs	$(1, 1, 1, 1)$
$L^{\text{lag}}_{\text{target}}$	length target of the Lagrangian baseline	2500
τ_σ	variance threshold for the live set	10^{-6}
η	min-norm floor before the uniform fallback	10^{-3}
τ_{fw}	Frank–Wolfe weight tolerance	10^{-6}
$T_{\max}$	Frank–Wolfe iteration cap	50

Table 4: Reward and solver hyperparameters of the Stage-3 run. Weights are given over the four implemented rewards; since a prompt activates only one of the two task metrics, a group sees $(w_{\text{fmt}}, w_{\text{task}}, w_{\text{len}})$, which is $(1, 2, 1)$ for Fixed Weight (Accuracy) and $(1, 1, 1)$ for the default runs.

B LIMITATIONS

Our training corpus remains limited in scale, so expanding annotation volume and diversity is an important next step. Our main experiments also use a 4B-parameter model because of computational constraints, and scaling to larger architectures is future work that depends on additional compute. Two further limits come from the schema itself. First, we cut events into 10–30 second segments, which puts a floor on how precisely the model can place a boundary, and an adaptive or multi-scale decomposition of events would be the way to lift it. Second, the six fields are built around people and their actions, and a rule ties the action field to the person field, so on video without people, such as wildlife footage or screen recordings, several fields would fall back to "None" and the schema would carry less information.

C DETAILED INFORMATION OF EVALUATION BENCHMARKS

To comprehensively evaluate the capabilities of our model, particularly in temporal reasoning and open-ended generation, we utilize a diverse set of benchmarks categorized into two primary domains: Video Temporal Grounding and General Video Understanding.

C.1 VIDEO TEMPORAL GROUNDING BENCHMARKS

These benchmarks evaluate the model's ability to precisely localize specific moments within untrimmed videos based on textual queries.

- **Charades-TimeLens** (Zhang et al., 2025a): A high-quality dataset re-annotated based on the Charades-STA dataset, where 66.3% of low-quality annotations were rewritten. The final dataset contains 1,313 videos and 3,363 annotations, with an average video duration of 29.3 seconds. This benchmark focuses on daily indoor activities (e.g., cooking, reading) and is characterized by dense actions and complex temporal overlaps. Unlike the noisy annotations in the original dataset, Charades-TimeLens provides rigorously verified timestamps to evaluate fine-grained temporal localization. We report Recall@1 at IoU thresholds of 0.3, 0.5, and 0.7.
- **ActivityNet-TimeLens** (Zhang et al., 2025a): A re-annotated version of the ActivityNet-Captions (Krishna et al., 2017) dataset introduced in the TimeLens benchmark, where 69.7% of low-quality annotations were rewritten. The final dataset contains 1455 videos and 4500 annotations, with an average video duration of 134.9s. It addresses the issue of loose boundaries in the original annotations by providing stricter and more precise start/end times for temporal queries. Evaluating on this split offers a more rigorous assessment of a model's boundary alignment capabilities.

- **ActivityNet-Captions** (Krishna et al., 2017): A large-scale Video Temporal Grounding benchmark containing 4885 open-domain YouTube videos for evaluation, with 17031 annotations and an average video duration of 118s. It covers a wide range of complex human activities. The dataset is widely used for dense video captioning and temporal grounding. We evaluate performance using Recall@1 at IoU thresholds of 0.3, 0.5, and 0.7.

C.2 General Video Understanding Benchmarks

These benchmarks assess the model's holistic understanding, including temporal logic, structured reasoning, and ego-centric perception.

- **VideoMME** (Fu et al., 2024): A comprehensive Multi-modal LLM Evaluation benchmark designed for long-form video understanding. It covers diverse domains including movies, sports, and documentaries, with video lengths ranging from minutes to hours. It assesses the model's ability to process long-context information without subtitles.
- **MLVU** (Zhou et al., 2025): A holistic benchmark specifically tailored for long videos. It includes diverse tasks such as *Topic Reasoning*, *Anomaly Recognition*, *Ego Reasoning*, etc. In our experiments, we specifically focus on the Topic Reasoning (TR) and Ego Reasoning (Ego) sub-tasks to evaluate the model's sequence understanding and first-person perspective reasoning.
- **ETBench** (Liu et al., 2024b): A fine-grained benchmark designed to evaluate time-sensitive video understanding. We test the model on 8 different tasks: *Referred Action Recognition (RAR)*, *Event-Caption Alignment (ECA)*, *Referred Video Question-Answering (RVQ)*, *Temporal Video Grounding (TVG)*, *Episodic Memory (EPM)*, *Temporal Action Localization (TAL)*, *Temporal Event Matching (TEM)*, and *Grounded Video Question-Answering (GVQ)*. This benchmark is crucial for verifying whether the model's thinking process translates to accurate event-level deductions.
- **NExT-GQA** (Xiao et al., 2024): An extension of the NExT-QA dataset that introduces grounded video question answering. It focuses on causal (Why) and temporal (How) questions involving rich object interactions. Unlike standard QA, NExT-GQA requires the model to not only answer the question but also ground the answer in specific video segments, making it an ideal testbed for our reasoning-based approach.

D Data Construction Pipeline Details

We build PARSE-60K from 35K videos sampled from Panda-70M (Chen et al., 2024b), as summarized in Figure 4. We do not reuse Charades or ActivityNet. Because Panda-70M is also crawled from the web, we further drop every video whose ID appears in any eval set we report on. The pipeline consists of the following stages.

Event segmentation. Panda-70M has one caption per video and no event timestamps, so we make the timestamps ourselves. Shot detection proposes cut points. We merge or split those cuts so each segment is 10–30 seconds long. A human then audits these boundaries on a sampled half of the data, using the batch sampling of Section D, and fixes the cases where a camera cut is not an event change, for example one action split by a cut, or two actions in one shot. These timestamps are what the evidence generator sees.

Pre-processing and filtering. After the ID filter, we keep videos with many events. We treat one event as 10–30 seconds: for a video of length T we keep it only if $N \geq T/30$, so a 150-second video needs at least 5 events. If two checked segments leave a hole (for example $[25, 38]$ between $[15, 25]$ and $[38, 50]$), we also ask the model to describe that hole. Videos with $N < T/30$ are left out of evidence training, but we still use them in RL.

Evidence generation. We feed the verified dense timestamps into an evidence generator to produce the event evidence descriptions. Since evidence entries are fine-grained structured signals that require strong instruction following to achieve high data quality, we employ two complementary models: Qwen3-VL-235B-A22B-Instruct (Bai et al., 2025a) and Gemini-2.5-Pro (Comanici et al., 2025). To

prevent specific model preferences from compromising data quality, the two models alternate roles: when one serves as the generator, the other acts as a quality judge. The two strong models mutually critique and filter outputs, enabling evidence to capture detailed visual cues. The data produced in this stage is served for Stage 1.

Thinking trace generation. After obtaining the evidence, we generate challenging questions and prompt the LLM to produce explicit thinking traces conditioned on the question and the associated evidence information. Since thinking data demands a higher quality than evidence itself, we use Gemini-2.5-Pro exclusively as the generator in this stage, while Qwen3-VL serves as the quality judge. The data produced in this stage is served for Stage 2.

Structured thinking stages. Building upon the resulting Structured Event Evidence, we decompose reasoning into stages that progressively narrow, verify, and consolidate evidence. Specifically, we adopt a structured evidence-grounded reasoning process with three tightly coupled stages: 1. Global Search & Localization, 2. Evidence Verification, and 3. Final Alignment. As shown in Figure 1, the first phase *Global Search & Localization* requires the model to retrieve query-relevant clues from the structured evidence list to quickly narrow attention to a small set of highly relevant time windows. In the *Evidence Verification* stage, the model examines temporal coherence by analyzing the preceding and succeeding events around the hypothesized interval, and validates the localization through temporal event consistency rather than shallow pattern matching. To keep decisions in visual evidence, the model performs mandatory visual verification on the inferred interval using vision tokens paired with textual prompts. Before *Final Alignment*, the model performs a global consistency check to ensure the inferred temporal interval is fully contained within the observed activity sequences.

Human evaluation. To check data quality, we split the generated data into batches of $1{,}000$ annotations and draw 500 of each batch at random for manual inspection. This covers 30K of the 60K annotations, and the boundary audit above draws on the same sampled half, so all human review in the pipeline follows one protocol. The two thresholds then do different jobs: the automatic judge in Figure 7 gates a single sample, while the human score below gates a whole batch. Three annotators score each sampled instance from 1 to 10, following the same criteria as the automatic judge but they are more accurate and strict (Figure 7). Their average scores are 8.5, 7.9 and 8.2, with 87.1% of samples within one point.

Format completeness. For evidence, the `<evidence>` tag must be closed, the global theme line must be present, and every time segment must fill in all six fields (time, person, human action, scene, object, camera), with one line per input segment and no missing or invented timestamps. For thinking traces, `<evidence>`, `<thinking>` and `<answering>` must all be present, and the thinking block must actually walk through Global Search, Evidence Verification and Final Alignment.

Fluency. Whether the text reads naturally and says something specific. Evidence has to be concrete enough to check against the video, not "a person is doing something". In a thinking trace, each step has to move the reasoning forward rather than restate the conclusion in new words.

Hallucination. Whether everything stated can be found in the video. For evidence we look for entities, actions or timestamps that are not there. For thinking traces the bar is higher: on top of not inventing visual content, a trace may not cite a segment that is absent from the evidence list, and may not jump to the answer when the evidence does not support it.

We anchor the scale as follows. 9–10: all three are clean and the sample is kept as is. 7–8: minor wording problems, such as a vague or slightly repetitive description, but no hallucination and complete format. 4–6: missing format fields or mild hallucination, for instance one attribute that does not match the video. 1–3: the format is badly broken or large parts are invented. Any sample scoring below 6 sends its whole batch back for regeneration, which happened to 28.3% of batches.

The two data types are weighted differently. Evidence is scored mainly on format completeness and timestamp accuracy, since it is the input to downstream reasoning and one wrong timestamp propagates along the chain. Thinking traces get their format largely for free from the generation template, so they are scored mainly on hallucination and on whether the reasoning is valid; this is also why we use only Gemini-2.5-Pro to generate them.

E MORE DETAILS OF PARSE-60K

E.1 VIDEO SOURCE

We sample 35K videos from Panda-70M (Chen et al., 2024b) and keep only the raw video, not the original captions. We do not train on Charades-STA (Gao et al., 2017) or ActivityNet (Krishna et al., 2017), because we test on those sets. Panda-70M is also from the web, so a different source is not enough: we match video IDs against every eval set and drop any hit. After this filter, PARSE-60K and the eval sets share no video.

The downside is that we have to make event timestamps ourselves. As in Appendix D, we use shot detection, merge or split to 10–30 seconds, and let a human audit half of the boundaries. The timestamps follow our event length, not the style of any benchmark.

E.2 RL TRAINING DATA DETAILS

In this section, we provide a detailed statistical analysis of the dataset used for the Reinforcement Learning (RL) stage. The RL stage reuses the full corpus of 35K videos, specifically curated to optimize temporal reasoning and grounding capabilities through diverse task types and semantic domains.

Task Type Composition. As illustrated in Figure 5, the dataset follows a specific distribution across different task types to balance reward stability and semantic understanding. The majority of the dataset is dedicated to *Temporal Grounding* (53%). This high proportion is designed to leverage the objective nature of Temporal Intersection over Union (TIoU) metrics, which provide dense and stable reward signals during the policy optimization process. To ensure the model maintains robust logical reasoning capabilities beyond pure localization, the dataset includes a substantial portion of Visual Question Answering tasks, specifically *Spatial VQA* (21%) and *Reasoning VQA* (20%). These tasks require the model to interpret visual relationships and perform structured deductions. The remaining portion consists of *Temporal VQA* (6%). Every RL prompt therefore has a task metric, either temporal IoU for grounding or multiple-choice accuracy for the VQA types, so no group is left with only the format and length objectives.

Since the RL stage reuses the full corpus, only the task type assigned to each video differs from the evidence and thinking stages.

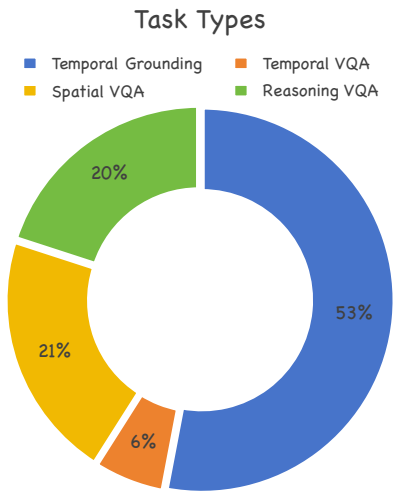

Figure 5: Distribution of task types in the RL training data. *Temporal Grounding* constitutes the largest share (53%), providing precise IoU-based supervision, while *Spatial VQA* (21%), *Reasoning VQA* (20%) and *Temporal VQA* (6%) broaden the training signal toward spatial, semantic and temporal understanding.

E.3 Data Curation Prompts

This section describes the prompts used in our data curation pipeline. We employ three specialized prompts for different stages: Evidence Generation, Quality Judge, and Thinking Generation. The complete prompts are illustrated in Figure 6, Figure 7, and Figure 8, respectively.

Evidence Generation Prompt. The Evidence Generation prompt (Figure 6) instructs the model to analyze video content and produce structured summaries for pre-defined time segments. Given a list of timestamps as input, the model generates a theme summary and event evidence for each segment. The prompt enforces a strict schema with six required components per segment: `[person]`, `[human_action]`, `[scene]`, `[object]`, `[camera]`, and `[event_caption]`. A critical dependency rule ensures logical consistency: if no person is present in the scene, the human action field must be marked as "None". The prompt also specifies that timestamps must be preserved in their original seconds format without conversion to MM:SS, and all output must be enclosed within `<evidence></evidence>` tags.

Quality Judge Prompt. The Quality Judge prompt (Figure 7) defines a strict evaluation protocol for validating generated evidence. It takes two inputs: the target timestamps that must be present and the model-generated structured text. The evaluation criteria are organized into four categories: (1) *Format Adherence* ensures proper tag structure, required headers, and all six mandatory fields in every line; (2) *Timestamp Alignment* verifies one-to-one correspondence between output lines and input segments, rejecting any missing or hallucinated timestamps; (3) *Logical Consistency* enforces the person-action dependency rule, ensuring that "None" person entries always correspond to "None" actions; and (4) *Content Quality* assesses completeness, fluency of captions, and specificity of descriptions. The judge outputs a structured JSON containing a numeric score (0-10), a regeneration flag, a list of critical errors, and qualitative feedback.

Thinking Generation Prompt. The Thinking Generation prompt (Figure 8) guides the model to produce evidence-based reasoning chains for video question answering. Given a user question and extracted evidence, the model must derive answers through the three-stage structured reasoning process of Section D. In *Stage 1: Global Search & Localization*, the model scans evidence descriptions for keywords related to the question, identifies candidate time segments based solely on textual evidence, and explicitly states which evidence IDs are selected as candidates. In *Stage 2: Evidence Verification*, the model validates candidates using antecedent-action-consequence logic: `[Antecedent]` establishes the pre-condition contrast, `[Verifying Key Visual Actions]` provides textual proof of the event, and `[Consequence]` confirms the post-condition result. In *Stage 3: Final Alignment*, it checks that the selected interval is contained within the observed activity sequence before committing to an answer. The output follows a `<thinking>...</thinking><answer>...</answer>` format, ensuring that reasoning is grounded in explicit structured evidence rather than implicit assumptions.

F Instruction Tuning Data Format

To effectively train the model to reason over structured evidence, we design a progressive prompt engineering strategy that guides the generation of training data across two distinct stages. This multi-stage approach ensures the model first learns to extract high-density visual evidence, and then integrates this skill into a coherent reasoning chain.

In the initial stage, the objective is to align the model's visual perception with our structured event schema, as displayed in Figure 9. We explicitly instruct the model to localize activity events and generate a detailed report containing a global theme and detailed event evidence. We enforce a rigid output format for each temporal segment, requiring specific fields such as time, person, human action, scene, object, camera, and an event caption. This design forces the model to decompose complex video dynamics into discrete, verifiable atomic elements rather than generating generic captions, effectively transforming raw video data into the structured representation used throughout the pipeline.

The final stage focuses on generating high-quality CoT data, as depicted in Figure 10. We combine the structured evidence from the first stage with challenging QA pairs. The prompt requires the model

to first recall the structured evidence context. Crucially, the subsequent thinking block is no longer trivial; it contains a detailed derivation process where the model performs global search, evidence verification, and alignment based on the provided evidence. This stage produces the reference thinking data used to warm-start the policy model, ensuring that the reasoning process is explicitly grounded in the extracted evidence.

G More details of Reinforcement Learning

G.1 Detailed Rewards Design

To guide the model toward generating structured, evidence-grounded, and concise reasoning chains, we design a composite reward function consisting of four distinct components: *Format Reward, Linear IoU Reward, Multi-choice Accuracy Reward, and Length Reward*.

Structured Thinking Format Reward (r_{fmt}). To enforce the structured reasoning schema proposed in our method, we design a rule-based format reward. This reward evaluates whether the model strictly follows the `<evidence>`, `<thinking>`, and `<answering>` XML structure and whether the `<thinking>` block explicitly traverses the required structured reasoning steps.

Let $\mathcal{K}$ be the set of mandatory structural keywords defined as:

$$\mathcal{K} = \{\textit{“Global Search”, “Evidence Verification”, “Final Alignment”, “Antecedent”, “Visual Verification”, “Consequence”}\} \tag{6}$$

Let $\mathcal{P}_{\text{basic}}$ denote the satisfaction of the basic regex pattern, $\mathcal{T}_{\text{valid}}$ denote the constraint that all XML tags appear exactly once, and $\mathcal{C}_{\text{think}}$ represent the content extracted from the `<thinking>` block. The format reward r_{fmt} is strictly defined as:

$$r_{\text{fmt}} = \begin{cases} 1.0 & \text{if } \mathcal{P}_{\text{basic}} \wedge \mathcal{T}_{\text{valid}} \wedge (\mathcal{K} \subseteq \mathcal{C}_{\text{think}}) \\ 0.5 & \text{if } \mathcal{P}_{\text{basic}} \wedge [\neg \mathcal{T}_{\text{valid}} \vee (\mathcal{K} \not\subseteq \mathcal{C}_{\text{think}})] \\ 0.0 & \text{otherwise (pattern mismatch)} \end{cases} \tag{7}$$

This reward scheme assigns a baseline score of 0.5 for adhering to the basic format structure, grants a full score of 1.0 only when the reasoning chain is structurally complete and structurally rigorous, and assigns 0.0 for format violations.

Task-Specific Performance Rewards (r_{task}). Depending on the downstream task type, we apply different performance metrics as rewards:

Linear IoU Reward: For temporal grounding tasks, we compute the Temporal Intersection over Union (TIoU) between the predicted time segments S_{pred} and ground truth segments S_{gt}. To handle ambiguity where a single continuous prediction covers multiple disjoint ground truth segments, we implement a hybrid scoring mechanism. Specifically, if $|S_{pred}| = 1$ and $|S_{gt}| > 1$, the reward is defined as $\max(\text{Coverage Ratio}, \text{Span IoU})$, ensuring the model is rewarded for correctly identifying the overall event span. Otherwise, we compute the average IoU across all segment pairs.

Multi-choice Accuracy Reward: For QA tasks, we strictly parse the predicted option (A/B/C/D) from the `<answering>` block. The reward is a binary indicator function: $r_{\text{acc}} = \mathbb{I}(\text{Pred} = \text{GT})$.

Every rollout is therefore scored on three objectives at once, namely schema compliance, the task metric, and the reasoning budget. A prompt is either a grounding prompt or a QA prompt, so only one of the two task metrics is active on any given prompt, and GRAB balances format, task metric and budget inside each group.

Length Reward (r_{len}). To prevent the “verbosity tax” often associated with CoT reasoning, we introduce a soft length constraint. Unlike hard truncation, we employ a piecewise linear decay function to gently discourage excessive token generation beyond a target budget. We penalize only over-long outputs, for two reasons. Hard truncation gives zero reward to every over-long rollout, so early in training many rollouts collapse to the same score and the group-relative advantage carries

little information. A two-sided penalty would pay the model for being short, which is exactly the shortcut that leads to cutting too much (Wang et al., 2026; Li et al., 2026). Our reward is flat below L_{target}, so being shorter than the budget earns nothing extra. This shape only limits how the conflict can appear; how much r_{len} counts against r_{acc} is still the real decision, and that is what GRAB sets automatically.

The budget binds during training. Stage-3 rollouts are drawn from a corpus that mixes grounding and QA prompts and average roughly 2,500 to 3,100 tokens, inside the $(L_{target}, L_{max}]$ band where r_{len} decays, so the length column varies within a group and survives the variance filter.

Let L be the generated response length, L_{max} be the hard maximum length, and L_{buffer} be a soft buffer zone. The target length is defined as $L_{target} = L_{max} - L_{buffer}$. We set $L_{max} = 4096$ and $L_{buffer} = 2500$, so the penalty starts at $L_{target} = 1596$ tokens. The length reward is calculated as:

$$r_{\text{len}}(L) = \begin{cases} 1.0 & \text{if } L \leq L_{target} \\ 1.0 - \frac{L - L_{target}}{L_{buffer}} & \text{if } L_{target} < L \leq L_{max} \\ 0.0 & \text{if } L > L_{max} \end{cases} \tag{8}$$

This design provides a full reward for concise answers within the target budget and applies a linear penalty only when the model enters the buffer zone, preventing the optimization instability caused by abrupt reward cliffs.

G.2 The GRAB Algorithm

Algorithm 1 states the full procedure that Section 3.2 describes in text.

Algorithm 1 GRAB

Input:
  $\mathbf{R} \in \mathbb{R}^{N_{\text{total}} \times M}$: rewards for N_{total} responses and M objectives
  $\mathbf{g} \in \mathbb{Z}^{N_{\text{total}}}$: group (prompt) ID for each response
  $\tau_\sigma > 0$: standard deviation threshold (e.g., 10^{-6})
  $\eta > 0$: min-norm floor (e.g., 10^{-3})
Output:
  $\mathbf{A} \in \mathbb{R}^{N_{\text{total}}}$: scalarized advantage for each response
Initialize $\mathbf{A} \leftarrow \mathbf{0}_{N_{\text{total}}}$
for each unique group $q \in \text{unique}(\mathbf{g})$ **do**
  // Step 1: Rewards Centralization (Eq. 1)
  $\mathcal{I}_q \leftarrow \{i : g_i = q\}$ ▷ Indices in group q
  $\mathbf{D}_q \leftarrow \mathbf{R}_{\mathcal{I}_q,:} - \text{mean}(\mathbf{R}_{\mathcal{I}_q,:}, \text{axis} = 0)$
  // Step 2: Standardized Conflict Resolution (Eq. 2)
  $\boldsymbol{\sigma}_q \leftarrow \text{std}(\mathbf{D}_q, \text{axis} = 0)$
  $\mathcal{V} \leftarrow \{m : \sigma_{q,m} > \tau_\sigma\}$ ▷ Live objectives
  if $\mathcal{V} = \emptyset$ **then**
    $\boldsymbol{\alpha}^* \leftarrow \frac{1}{M}\mathbf{1}_M$ ▷ Fallback to uniform
  else
    $\hat{\mathbf{D}}_q \leftarrow \mathbf{D}_q[:, \mathcal{V}]/\boldsymbol{\sigma}_q[\mathcal{V}]$ ▷ Standardization
    Solve: $\boldsymbol{\alpha}^*_{\mathcal{V}} = \arg\min_{\boldsymbol{\alpha}} \|\hat{\mathbf{D}}_q\boldsymbol{\alpha}\|^2$ s.t. $\boldsymbol{\alpha} \in \Delta_{|\mathcal{V}|}$ via Frank-Wolfe
    if $\|\hat{\mathbf{D}}_q\boldsymbol{\alpha}^*_{\mathcal{V}}\| < \eta$ **then**
      $\boldsymbol{\alpha}^*_{\mathcal{V}} \leftarrow \frac{1}{|\mathcal{V}|}\mathbf{1}$ ▷ near-zero min-norm → uniform
    end if
    $\boldsymbol{\alpha}^* \leftarrow \mathbf{0}_M; \quad \boldsymbol{\alpha}^*[\mathcal{V}] \leftarrow \boldsymbol{\alpha}^*_{\mathcal{V}}$
  end if
  // Step 3: Advantage Scalarization (Eq. 4)
  $\mathbf{A}^{\text{raw}}_q \leftarrow \mathbf{D}_q\boldsymbol{\alpha}^*$ ▷ Apply weights to centered rewards
  $\mathbf{A}[\mathcal{I}_q] \leftarrow (\mathbf{A}^{\text{raw}}_q - \text{mean}(\mathbf{A}^{\text{raw}}_q))/(\text{std}(\mathbf{A}^{\text{raw}}_q) + \epsilon)$
end for
return $\mathbf{A}$

G.3 LIVE OBJECTIVES PER GROUP

A prompt is either a grounding prompt or a QA prompt, so one of the two task metrics is constant within any group and is removed by the variance filter, leaving at most three live objectives. When exactly two are live, the two standardized columns have equal norm, so the shortest point on the segment between them is its midpoint and $\boldsymbol{\alpha}_q^* = (\frac{1}{2}, \frac{1}{2})$; the factor cancels in the z-score of Eq. 4. The adaptive weights therefore matter most for groups in which all three objectives vary.

G.4 FRANK-WOLFE ALGORITHM FOR SOLVING THE MINIMUM-NORM PROBLEM

To solve the minimum-norm optimization problem in Eq. (2), we employ the Frank-Wolfe algorithm, also known as the conditional gradient method (Jaggi, 2013), which is particularly well-suited for constrained convex optimization over polytopes such as the probability simplex.

Problem Formulation. Recall that our objective is to find the optimal weight vector $\boldsymbol{\alpha}^*$ that minimizes the squared norm of the aggregated direction in the standardized space:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha} \in \Delta_M} f(\boldsymbol{\alpha}), \quad \text{where} \quad f(\boldsymbol{\alpha}) = \left\| \hat{\mathbf{D}}_q \boldsymbol{\alpha} \right\|^2. \tag{9}$$

This is a convex quadratic program over the simplex $\Delta_M = \{\boldsymbol{\alpha} \in \mathbb{R}^M \mid \sum_{m=1}^M \alpha_m = 1, \alpha_m \geq 0\}$.

Algorithm Description. Starting from a uniform initialization $\boldsymbol{\alpha}^{(0)} = \frac{1}{M}\mathbf{1}_M$, each iteration t proceeds as follows:

Step 1: Gradient Computation. We compute the gradient of the objective at the current iterate:

$$\nabla f(\boldsymbol{\alpha}^{(t)}) = 2\hat{\mathbf{D}}_q^\top (\hat{\mathbf{D}}_q \boldsymbol{\alpha}^{(t)}). \tag{10}$$

Step 2: Linear Minimization Oracle (LMO). We solve a linear minimization problem over the simplex:

$$\mathbf{s}^{(t)} = \arg\min_{\mathbf{s} \in \Delta_M} \langle \mathbf{s}, \nabla f(\boldsymbol{\alpha}^{(t)}) \rangle. \tag{11}$$

Since Δ_M is a simplex, the solution is simply a one-hot vector placing all mass on the coordinate with the smallest gradient component:

$$\mathbf{s}^{(t)} = \mathbf{e}_{k^*}, \quad \text{where} \quad k^* = \arg\min_m [\nabla f(\boldsymbol{\alpha}^{(t)})]_m. \tag{12}$$

Step 3: Exact Line Search. We compute the update direction $\mathbf{d}^{(t)} = \mathbf{s}^{(t)} - \boldsymbol{\alpha}^{(t)}$ and determine the optimal step size via exact line search. For the quadratic objective, this admits a closed-form solution:

$$\gamma^{(t)} = \max\left(0, \min\left(1, \frac{-\langle \hat{\mathbf{D}}_q \boldsymbol{\alpha}^{(t)}, \hat{\mathbf{D}}_q \mathbf{d}^{(t)} \rangle}{\|\hat{\mathbf{D}}_q \mathbf{d}^{(t)}\|^2}\right)\right). \tag{13}$$

If $\|\hat{\mathbf{D}}_q \mathbf{d}^{(t)}\|^2 < \epsilon_{\text{fw}}$ (we use $\epsilon_{\text{fw}} = 10^{-12}$), we terminate early to avoid numerical instability.

Step 4: Weight Update. We update the weights via:

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} + \gamma^{(t)} \mathbf{d}^{(t)}. \tag{14}$$

Termination Criteria. The algorithm terminates when the change in weights falls below a tolerance threshold $\|\boldsymbol{\alpha}^{(t+1)} - \boldsymbol{\alpha}^{(t)}\| < \tau_{\text{fw}}$ (we use $\tau_{\text{fw}} = 10^{-6}$), or when the maximum number of iterations is reached (we use $T_{\max} = 50$). This tolerance is separate from the variance threshold τ_σ in Algorithm 1.

Near-zero min-norm fallback. Frank–Wolfe can return a mix whose direction $\|\hat{\mathbf{D}}_q \boldsymbol{\alpha}^*\|$ is close to zero. Every live column has the same norm, so this arises only when the columns cancel, never when they agree, and Algorithm 1 then falls back to uniform weights on $\mathcal{V}$ ($\eta = 10^{-3}$).

Convergence of the Frank–Wolfe Solver. The exact line search ensures monotonic decrease of the objective value at every iteration, guaranteeing convergence to the global optimum of this convex problem, which by Eq. (3) is the weight vector minimizing $\boldsymbol{\alpha}^\top \mathbf{C}_q \boldsymbol{\alpha}$ over the live objectives of that group.

Model	TempCompass			EventBench		
	Overall	Event Order	Attr. Change	Overall	Temporal Reason.	Causal Reason.
Qwen3-VL-4B-Instruct	68.71	74.76	70.40	62.71	64.51	66.87
Qwen3-VL-4B-Thinking	68.60	73.20	69.00	60.44	62.57	65.22
PARSE-4B	**70.64**	**76.96**	**73.85**	**63.48**	**65.42**	**69.47**

Table 5: Results on temporal and event reasoning benchmarks. TempCompass evaluates temporal perception; EventBench assesses event-level understanding including temporal and causal reasoning sub-tasks.

H Additional Experiments and Analyses

H.1 TempCompass and EventBench Evaluation

To further validate the temporal and event-level reasoning capabilities of our model, we evaluate on two additional benchmarks: TempCompass, which assesses temporal perception including event ordering and attribute change detection, and EventBench, which evaluates event-level understanding including temporal and causal reasoning sub-tasks. As shown in Table 5, PARSE-4B achieves 70.64% on TempCompass and 63.48% on EventBench, outperforming both Qwen3-VL-4B-Instruct and Qwen3-VL-4B-Thinking across all sub-tasks. Notably, the largest improvements appear in attribute change detection (+3.45%) and causal reasoning (+2.60%), confirming that structured event evidence effectively enhances the model's ability to capture fine-grained temporal dynamics and event dependencies.

H.2 Scalability to 8B Model

To validate that GRAB generalizes beyond the 4B scale, we train Qwen3-VL-8B with LoRA using both standard GRPO and our GRAB algorithm. As shown in Table 6, GRAB outperforms GRPO on all four benchmarks, with the largest gains on VideoMME and CTL-mIoU, the mean IoU on Charades-TimeLens. Our compute budget at this scale was limited, so both 8B runs use half of the RL data we use at 4B and a lower LoRA rank. Both runs carry the same setting, so the comparison between GRPO and GRAB is made under matched conditions.

Model (8B)	**VideoMME**	**CTL-mIoU**	**TempCompass**	**EventBench**
8B-GRPO	64.81	43.73	65.79	64.87
8B-GRAB	**66.98**	**46.81**	**66.23**	**66.10**

Table 6: GRAB vs. GRPO on the 8B model scale (LoRA fine-tuning). GRAB outperforms GRPO on all four benchmarks, so the margin is not confined to 4B. Both runs use half the RL data and a lower LoRA rank than the 4B experiments, so the absolute numbers sit below the 4B model and carry no scaling claim. CTL-mIoU is the mean IoU on Charades-TimeLens.

H.3 GRAB Computational Overhead

A key concern for multi-objective optimization algorithms is their computational cost. We measure the wall-clock time of the GRAB weight computation step and compare it against the total training step time. As shown in Table 7, GRAB adds negligible overhead (0.0013% of total step time) across all group sizes, as the core computation involves only a small quadratic program over the number of objectives ($M = 4$), independent of the model size.

Group Size	GRAB (ms)	One Step (ms)
4	0.68	52,789
8	0.69	108,401
16	1.00	211,389

Table 7: GRAB computational overhead. The weight computation is negligible ($< 0.002\%$) relative to the total training step time.

H.4 Evidence Quality Evaluation

These metrics measure agreement with the annotations used for supervision. We compute standard text generation metrics (F1, BLEU-4, ROUGE-L, CIDEr) against the held-out evidence annotations of PARSE-60K, where F1 is the token-level F_1 between the generated and reference evidence blocks after field-wise alignment. As shown in Table 8, PARSE-4B substantially outperforms the base Instruct model across all metrics, confirming that our multi-stage training pipeline effectively teaches the model to produce structured evidence in the target format.

Model	F1	BLEU-4	ROUGE-L	CIDEr
Qwen3-VL-4B-Instruct	44.2	10.6	27.1	4.1
PARSE-4B	**54.7**	**16.6**	**35.1**	**17.3**

Table 8: Evidence quality evaluation. PARSE-4B generates substantially higher-quality structured evidence compared to the base model.

H.5 Lagrangian (Length) Baseline

This baseline replaces the static length coefficient with the online dual variable of Li et al. (2026), so it adapts the accuracy–length trade-off during training with one global value. Everything else matches the GRAB run: same Stage-2 checkpoint, same RL corpus, same four rewards, $\mathrm{num_generation} = 16$, and the same learning rate and step count. The advantage is plain GRPO, that is, weighted sum followed by the group z-score.

The scalar reward is $R = w_{\text{fmt}} r_{\text{fmt}} + w_{\text{task}} r_{\text{task}} + \lambda\, r_{\text{len}}$, where w_{fmt} and w_{task} are frozen at the Fixed Weight (Accuracy) values and only λ moves. After each step we gather the mean completion length $\bar{L}$ over the whole batch and update

$$\lambda \leftarrow \mathrm{clip}\left(\lambda + \eta_\lambda \cdot \frac{\bar{L} - L_{\text{target}}}{L_{\text{target}}},\ 0,\ \lambda_{\max}\right), \tag{15}$$

with $\lambda_0 = 1$, dual step size $\eta_\lambda = 0.01$ and $\lambda_{\max} = 5$. We drive the update from $\bar{L}$ rather than from $\bar{r}_{\text{len}}$ because r_{len} is flat at 1.0 below L_{target} and would carry no signal once the policy is inside the budget. We tune L_{target} so that the run ends within a few percent of GRAB's output length, landing at $2{,}500$ training tokens, which leaves accuracy as the only axis on which the two differ.

H.6 Static Weight Sweep

Fixed Weight (Accuracy) in Table 1 is one setting of the static weights, and one point cannot tell us whether GRAB beats static weighting in general or only beats that point. The static family is smaller than it looks, because the weighted sum is z-scored inside each group. Multiplying the whole weight vector by a constant therefore leaves the advantage unchanged, so with three objectives able to be live the family has two free ratios rather than three weights. We sweep the one the length reward controls, holding w_{fmt} and w_{task} at their tuned values of 1 and 2 and varying w_{len} alone, which moves $w_{\text{len}}/w_{\text{task}}$ from 0 to 2.

Every run in the sweep starts from the same Stage-2 checkpoint and shares the RL corpus, the learning rate, the step count and $\mathrm{num_generation} = 16$, so the runs differ in w_{len} and nothing else.

We evaluate on Charades-TimeLens alone, whose videos average 29.3 seconds, since sweeping on the longer benchmarks is outside our compute budget. Two rows come for free, since Fixed Weight (Accuracy) and GRPO are both members of the static family and both already appear in Table 1 alongside GRAB at this group size. The two new runs take the Fixed Weight (Accuracy) vector and change w_{len} alone: setting it to 0 shows what accuracy a free budget buys, and $4\times$ its tuned value gives the short end, so the pair brackets GRAB's output length.

Table 9 lists the static runs ordered by output length, and what it shows first is how much the choice of weights matters: accuracy spans 37.8 to 42.9 R1@0.5 and length spans 468 to 1,180 tokens over four settings of a single ratio. The most accurate of them spends 931 tokens against GRAB's 579, and reaching it took four runs.

Run ($G = 16$)	$\mathbf{w}_{\text{len}}/\mathbf{w}_{\text{task}}$	**R1@0.3**	**R1@0.5**	**R1@0.7**	**Len**
Fixed Weight (Accuracy), w=$(1, 2, 0)$	0	62.4	42.1	23.8	1180
Fixed Weight (Accuracy), w=$(1, 2, 1)$	0.50	63.2	42.9	24.5	931
GRPO, uniform weights w=$(1, 1, 1)$	1.00	59.8	40.6	21.8	846
Fixed Weight (Accuracy), w=$(1, 2, 4)$	2.00	57.5	37.8	19.6	468
GRAB	per group	**63.7**	**43.4**	**24.6**	579

Table 9: Static reward weights on Charades-TimeLens at $G = 16$, written as $(w_{\text{fmt}}, w_{\text{task}}, w_{\text{len}})$ over the three objectives that can be live in a single group. Fixed Weight (Accuracy), GRPO and GRAB are the rows of the same name in Table 1. The other two runs take the Fixed Weight (Accuracy) vector and set w_{len} to zero or to four times its tuned value, leaving w_{fmt} and w_{task} alone, so the four static runs trace a single ratio. Output length falls monotonically in $w_{\text{len}}/w_{\text{task}}$ across them, so the column orders the family.

H.7 Seed Variance

Several margins in Table 1 are a point or two, so we report the spread over the three seeds on Charades-TimeLens for GRPO, Fixed Weight (Accuracy) and GRAB at $G = 16$. Every number we report elsewhere is the mean of these same three seeds.

Run ($G = 16$)	**R1@0.3**	**R1@0.5**	**R1@0.7**	**Len**
GRPO	59.8 ± 0.30	40.6 ± 0.30	21.8 ± 0.40	846 ± 12
Fixed Weight (Accuracy)	63.2 ± 0.30	42.9 ± 0.30	24.5 ± 0.40	931 ± 14
GRAB	$\mathbf{63.7 \pm 0.25}$	$\mathbf{43.4 \pm 0.25}$	$\mathbf{24.6 \pm 0.35}$	$\mathbf{579 \pm 9}$

Table 10: Spread across three random seeds on Charades-TimeLens. The means are the values reported in Table 1, and the table adds the standard deviation over the seeds.

H.8 The Weights the Solver Returns

Table 11 reports the weights the solver returns, averaged over the run and conditioned on the size of the live set.

In the three-objective groups the weights average 0.15 ± 0.11 on format, 0.39 ± 0.06 on the task metric and 0.46 ± 0.07 on length, with the spread taken across groups, where uniform weighting would put $1/3$ on each. Length carrying the largest average weight is what the criterion of Eq. 3 does when length is the column that moves against the others, and the operating point it produces is the one in Table 9: 43.4 R1@0.5 at 579 tokens. In the two-objective groups the weight is 0.50 with no spread, the value the algebra of Section G.3 gives. The last two rows compare GRAB with GRPO checkpoint by checkpoint at the same training step, over the checkpoints through step 100.

Quantity	Value
α^*_{fmt}, three live objectives	0.15 ± 0.11
α^*_{task}, three live objectives	0.39 ± 0.06
α^*_{len}, three live objectives	0.46 ± 0.07
α^*, two live objectives (self-check)	0.50 ± 0.00
R1@0.5 over GRPO, matched steps up to 100	$+0.7$ to $+2.1$
Mean length against GRPO, same steps	-8% to -19%

Table 11: Weights the solver returns, measured over the PARSE-4B run. The first three rows give the mean and the standard deviation of $\boldsymbol{\alpha}^*_q$ across the groups where all three objectives are live; the means sum to one, and uniform weighting would put $1/3$ on each. The fourth row is the value Section G.3 gives for two live objectives, measured on real groups rather than assumed. The last two rows compare the two runs checkpoint by checkpoint at the same training step, on Charades-TimeLens, with the range covering the checkpoints through step 100.

H.9 Reasoning Quality Evaluation

To evaluate the quality of the intermediate reasoning process (beyond final-answer accuracy), we employ Gemini-2.5-Pro as an automatic judge to rate model outputs on four dimensions: Overall quality, Hallucination avoidance, Logicality of the reasoning chain, and Interpretability of the structured output. Each dimension is scored on a 1–5 scale. As shown in Table 12, PARSE-4B matches or outperforms the Thinking baseline on all dimensions, with particularly strong gains in Interpretability ($+1.0$) and Logicality ($+0.25$), validating that structured event evidence enhances the reasoning process quality.

Model	Overall	Halluc.	Logic.	Interp.
Qwen3-VL-4B-Thinking	4.00	4.0	4.25	3.75
PARSE-4B	**4.41**	**4.0**	**4.50**	**4.75**

Table 12: Reasoning quality evaluation by Gemini-2.5-Pro. PARSE-4B produces more logical and interpretable reasoning chains.

Prompt example: Evidence Generation

Instruction: You are tasked with analyzing the provided video and generating a structured summary for specific pre-defined time segments.

INPUT DATA (Time Segments in Seconds):

{timestamps}

IMPORTANT: Your response MUST follow this EXACT format:

Theme: [Summary of the video's main subject]

Event Evidences:

[time1]Start_Sec - End_Sec, [person][human_action][scene][object][camera][casual_event_caption] \n

[time2]Start_Sec - End_Sec, [person][human_action][scene][object][camera][casual_event_caption]

... (Generate one line for EACH time segment provided in the input list)

Guidelines for Event Overview - FOCUS ONLY ON VISUAL EVENTS:

1. Perspective: You are watching the specific video segments defined by the input timestamps.

2. Segments Constraint: DO NOT segment the video yourself. You must STRICTLY iterate through the provided list of time segments. Analyze the video content strictly within these start and end times.

3. Timestamp Format: Use the exact seconds format provided in the input list (e.g., 0.83 - 19.86). Do not convert to MM:SS.

4. Description Structure: For each segment, include ALL these structured components:

[person]: Who is in the scene. Use a comma-separated list. If no human is present, write 'None'. (e.g., 'woman in white apron', 'None')

[human_action]: The specific physical movements or interactions performed by the human.

* DEPENDENCY RULE: If [person] is 'None', then [human_action] MUST be 'None'.

* Focus on INTERACTION with objects. Use a comma-separated list.

* (e.g., 'chopping vegetables', 'holding a phone', 'None')

[scene]: Location and setting details. (e.g., 'modern kitchen, white cabinets')

[object]: Key objects involved in the action. (e.g., 'wooden cutting board, silver knife')

[camera]: Camera techniques. (e.g., 'static third-person, zoom in')

[event_caption]: A detailed, natural description combining all elements sequentially. Describe exactly what happens within this specific time window.

5. Include all the content in between <evidence></evidence>.

Example (Note the dependency logic):

<evidence>

Theme: A nature documentary and a cooking tutorial.

Event Evidences:

[time1]: 0.83 - 19.86, [person]: Woman in white apron. [human_action]: Arranging ingredients, placing bowls. [scene]: Bright kitchen. [object]: Yellow bananas, glass bowls. [camera]: Static medium shot. [event_caption]: A woman wearing a white apron stands in a bright kitchen. She is arranging various ingredients, including bananas and glass bowls, on the counter. \n

...

</evidence>

Figure 6: Evidence Generation prompt used in the data curation pipeline to produce structured event evidence from videos.

Prompt example: Quality Judge

You are a strict Quality Assurance AI for video annotation data. Your task is to evaluate the output of a Vision-Language Model against a specific schema and set of logical rules.

INPUT DATA:

1. Target Timestamps: A list of time segments that MUST be present.

2. Model Output: The structured text generated by the model.

EVALUATION CRITERIA (Pass/Fail Rules):

1. Format Adherence (Critical):

- The output must be enclosed in <evidence></evidence> tags.
- It must contain the headers "Theme:" and "Event Evidence:".
- Each event line must strictly follow this pattern: [Start_Sec - End_Sec], [person]... [human_action]... [scene]... [object]... [camera]... [event_caption]...
- All 6 required tags must be present in every line: [person], [human_action], [scene], [object], [camera], [event_caption].

2. Timestamp Alignment (Critical):

- The output must contain exactly ONE line for EACH timestamp provided in the "Target Timestamps" list.
- No missing timestamps and no extra/hallucinated timestamps.
- The time format must be exactly as provided (e.g., "0.83 - 19.86"). Do NOT accept converted formats like MM:SS.

3. Logical Consistency (Critical Dependency Rule):

- Check every line for the Person-Action dependency.
- Rule: If [person] is 'None', then [human_action] MUST be 'None'.
- Rule: If [person] is NOT 'None', then [human_action] should typically describe an action (unless the person is static).

4. Content Quality:

- Completeness: No fields should be empty.
- Fluency: The [event_caption] must be a coherent, grammatical English sentence that accurately combines the details from the previous tags.
- Specificity: The description should include specific details (e.g., "silver pen" instead of just "object").

OUTPUT FORMAT:

Return a single JSON object. Do not output markdown code blocks or explanations outside the JSON.

```
{
  "score": (Integer 0-10, where 10 is perfect and <6 is a fail),
  "needs_regeneration": (Boolean, true if score < 8 or any Critical Rule is violated),
  "critical_errors": ["List of format or logic violations found, if any"],
  "quality_feedback": "Brief comment on the fluency and descriptive quality of the captions."
}
```

Figure 7: Quality Judge prompt used in the data curation pipeline to validate generated evidence.

Prompt example: Thinking Generation

```
# Role and Task
You are an expert AI video understanding analyst. You are provided with a User Question and a long list of Evidence extracted from a video.
Your objective is to generate a training sample that demonstrates evidence-based reasoning.
CRITICAL RULE: You must derive the answer solely by reading the Evidence. Do not assume or guess information that is not explicitly stated in the evidence descriptions.
# Reasoning Structure
Organize your thinking into two strictly separated sections.
Section 1: Global Search & Localization
- Describe how you scan the evidence text for keywords related to the User Question (e.g., "I looked for words like 'screwing', 'removing'...").
- Identify candidate segments based only on the event descriptions.
- Explicitly state which evidence IDs (e.g., [timeX]) you selected as candidates and why.
Section 2: Evidence Verification
- Verify your selected candidates using strict causal logic. Use these tags:
  - [Antecedent timeX]: (Pre-condition contrast — what happened before that sets up this event)
  - [Visual Verification]: (The visual proof in the text that confirms the event)
  - [Consequence timeX]: (Post-condition result — what happens after that confirms the event occurred)
Section3: Final Alignment
  - Conclude with your final answer based on the verified evidence.
# Example
## Example Input Data
User Question: "When does the player score?"
Evidence: ...[time11](40-45s Pass)... [time12](45-55s Goal)... [time13](55-65s Celebration)...
## Example Output
<thinking>
1. Global Search & Localization
Scanning the evidence descriptions from start to finish, I focused purely on the intent of "scoring a goal." I ignored general play in [time1]-[time10]. My search specifically looked for keywords like "Goal", "net shakes", or "score". I found that [time12] is the only segment containing the explicit description "Striker kicks ball into top right corner, Goal!". All other segments describe preparation or celebration. Thus, I identify [time12] as the sole candidate based on the text.
2. Evidence Verification
I will now verify this text-based candidate. [Antecedent time11]: Contrasting the 'defenders approaching' in the previous segment, the action here marks the release of that pressure. [Verifying Key Visual Actions]: The text in [time12] explicitly states "Goal!" and "Net shakes," which provides undeniable visual proof of the event. [Consequence time13]: The immediately following state of "Celebration" and "teammates hugging" confirms the goal was valid and not disallowed. Based on this causal chain, I conclude the scoring moment occurs in the [time12] window (45s-55s).
</thinking>
3. Final Alignment
[Output your justification]
<answer>
45s - 55s
</answer>
---
# Real Input Data
## User Question:
{user_question}
## Evidence:
{evidence}
# Output Format
<thinking>
[Your reasoning following the structure above...]
</thinking>
<answering>
[Your derived answer based on the factual evidence]
</answering>
```

Figure 8: Thinking Generation prompt used in the data curation pipeline to produce evidence-grounded reasoning traces.

Prompt example: Stage 1 - Evidence Training

{
"from": "human",
"value": "<video>\nLocalize a series of activity events in the video, generate a structured report containing a Theme and detailed Event Evidence. The output must strictly follow the format: [time][person][human_action][scene][object][camera][casual_event_caption] for each segment..."
},
{
"from": "gpt",
"value": "
<evidence>
Theme: Two men working on ski maintenance in a cluttered kitchen.

Event Evidence:
[time1]: 0s - 11.39s, [person]: Man in grey sweater, man in light green shirt. [human_action]: Walking around, sitting on chair, adjusting skis on stand, handling tools. [scene]: Cluttered kitchen with white cabinets, patterned tile floor, washing machine. [object]: Long black ski on red stand, laptop, various kitchen items. [camera]: High-angle static shot with fisheye lens. [event_caption]: In a cluttered kitchen, a man in a grey sweater sits on a chair while another man in a light green shirt moves around the room. The man in green adjusts a long black ski mounted on a red stand, handling tools and occasionally walking toward the washing machine area. The man in grey watches and occasionally shifts his position.

[time2]: 11.39s - 15.51s, [person]: Man in grey sweater, man in light green shirt. [human_action]: Sitting, standing, walking, bending over, picking up tools. [scene]: Same cluttered kitchen. [object]: Ski on stand, laptop, chair, kitchen counter items. [camera]: High-angle static shot with fisheye lens. [event_caption]: The man in the grey sweater remains seated while the man in the light green shirt walks around the kitchen, bends over to pick up tools near the ski stand, and moves back toward the counter ...."
}

Figure 9: Prompts for Stage 1 evidence training.

Prompt example: Stage 2 - Thinking Warm-Start

```
{
 "from": "system",
 "value": "{same_as_stage1.5}"
},
{
"from": "human",
"value": "
What is hanging on the rod above the man in the bathroom?
A. A purple towel folded neatly
B. A green shower cap with a floral pattern
C. A purple object shaped like a small ball
D. A blue hairdryer...
First, generate a structured report containing a Theme and detailed Event Evidence (spanning around 15-25
seconds) in <evidence></evidence> tags and strictly follow the format:
[time][person][human_action][scene][object][camera][event_caption] for each segment. Then, output thinking
process in <thinking></thinking> tags, and finally output the answer in <answering></answering> tags
"
},
{
"from": "gpt",
"value": "
        <evidence> {same as previous video evidence: [time1]...[person][human_action]} </
        evidence><thinking> 1. Global Search...2. Evidence Verification...3. Final Alignment <
        /thinking><answering> C. A purple object shaped like a small ball </answering>"
 }
```

Figure 10: Prompts for Stage 2 thinking warm-start. Orange parts mean these parts are different from the previous parts.