DYNFRAME: ADAPTIVE REASONING-DRIVEN MULTIMODAL FRAMEWORK WITH DYNAMIC FRAME AUGMENTATION FOR COMPLEX VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent video multimodal large language models (MLLMs) increasingly couple step-by-step reasoning with on-demand visual evidence retrieval, allowing models to revisit relevant video segments during inference. However, two structural gaps remain in existing thinking-with-video systems. **(i)** Sampling density is not a learnable decision: existing methods may let the model decide *where* to look, but the per-window frame rate is largely fixed. As a result, fine-grained evidence is often recovered through repeated retrieval calls or overly dense frame sampling, both of which increase inference context length and training difficulty. **(ii)** Retrieval and answer generation are usually optimized with a single trajectory-level advantage, so the "where to look" tokens and the "how to answer" tokens receive the same credit even when one is correct and the other is not. To address these gaps, we present **DynFrame**, a framework that emits the temporal window and the sampling density as native tokens within a single autoregressive pass; this learnable span–density retrieval acquires multi-granularity evidence with a single retrieval step. Based on the above tokenized retrieval interface, we further introduce **Segment-Decoupled GRPO (SD-GRPO)**, which splits each rollout at the retrieval boundary and assigns role-specific token-level advantages, separately crediting the sampling decision and the answer. Trained on the curated **DM-CoT-74k** and **DM-RL-45k**, DynFrame-4B is competitive with strong 7B–8B baselines across six benchmarks (NExT-GQA, Charades-STA, ActivityNet-MR, Video-MME, MLVU, LVBench), and DynFrame-8B sets a new state of the art on most metrics. Code, data, and model weights will be released.

1 INTRODUCTION

Multimodal large language models (MLLMs) (Bai et al., 2025b; Zhu et al., 2025; Li et al., 2025a; OpenAI, 2024; Comanici et al., 2025; Kimi Team et al., 2025; Bai et al., 2025a) have substantially advanced video question answering, temporal grounding, and long-form video understanding (Xiao et al., 2024; Gao et al., 2017; Krishna et al., 2017; Fu et al., 2025a; Zhou et al., 2025; Wang et al., 2025b). Building on chain-of-thought (CoT) reasoning (Wei et al., 2022; Shao et al., 2024a; Wang et al., 2025c), recent supervised and reinforcement-learning post-training methods further improve multi-step video inference (Feng et al., 2025a; Li et al., 2025c; Wang et al., 2025a). However, most video reasoners still operate after a fixed visual pass: a sparse set of frames is selected before generation, and all subsequent intermediate steps are purely textual (Maaz et al., 2024; Zhang et al., 2023; Jin et al., 2024; Zhang et al., 2024; Bai et al., 2025a). This design is fragile for long videos, where the answer may hinge on a short action, a brief object state change, or multiple clues scattered across a redundant temporal context. As reasoning chains grow, the model can drift away from the actual visual evidence and hallucinate missing fine-grained events.

A growing *thinking-with-video* line of work addresses this limitation by allowing models to actively revisit video evidence during inference. Existing systems instantiate visual revisiting through several retrieval interfaces: tool-based clip retrieval that calls an external module to crop or resample candidate video segments (Zhang et al., 2026; Yang et al., 2026), zoom-in or temporal-focusing mechanisms that inspect local regions at higher resolution (Fu et al., 2025b), multi-turn frame spotlighting or iterative perception that progressively refines clue-focused temporal regions (He et al., 2026; Yan et al., 2025),

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.

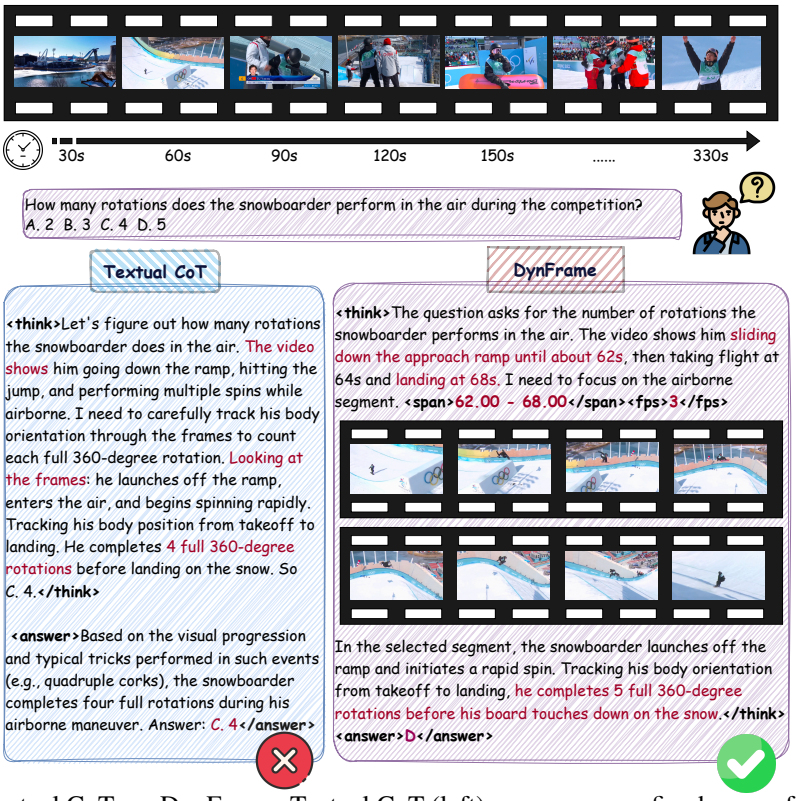


Figure 1: Textual CoT vs. DynFrame. Textual CoT (left) reasons over a fixed sparse frame set and misses the airborne segment, yielding a wrong rotation count. DynFrame (right) emits `<span>` and `<fps>` tokens *within* its reasoning to retrieve a denser, temporally focused frame set (a subset is shown) and continues reasoning over the augmented visual context to reach the correct answer.

and native interleaved tool invocation that couples evidence seeking with reasoning in a shared context (Zeng et al., 2026; Liu et al., 2026). These systems demonstrate the importance of revisiting visual evidence, but fine-grained evidence acquisition is often achieved by issuing repeated retrieval calls, refining temporal clues across turns, or appending additional high-resolution clips into an expanding context. Such multi-turn retrieval increases inference context length and also complicates training. This motivates a complementary question: *can a video MLLM itself learn a context-aware retrieval policy that adaptively selects the temporal span and sampling density so that multi-granularity evidence can be acquired more effectively?*

To address these challenges, we introduce **DynFrame**, a novel framework that represents the temporal window and sampling density as native tokens, making retrieval behavior an integral part of the model's reasoning process, as illustrated in Fig. 1. This turns frame-rate adaptation from a system hyperparameter into a learnable per-step decision. This learnable span–density interface enables task-adaptive frame acquisition: through training, the model learns to localize relevant evidence within the video and select an appropriate sampling density conditioned on the query and reasoning context. It can retrieve dense frames for brief, motion-sensitive events and sparse frames for long-range semantic understanding, acquiring multi-granularity evidence more effectively.

Furthermore, based on the explicit retrieval boundary created by the retrieval interface, we propose **Segment-Decoupled GRPO (SD-GRPO)**, which splits each rollout at the retrieval boundary and assigns role-specific token-level advantages, separately crediting the sampling decision and the answer reasoning. This provides targeted credit for retrieval quality and answer correctness, helping the model improve answer accuracy by grounding its reasoning in correctly retrieved evidence. To train DynFrame, we curate task-balanced **DM-CoT-74k** for supervised fine-tuning and **DM-RL-45k** for reinforcement learning, explicitly designed to cultivate robust and adaptive retrieval capabilities. Extensive experiments across six benchmarks spanning temporal grounding, grounded VideoQA, and long-form video understanding show that DynFrame-4B is competitive with strong 7B–8B baselines, while DynFrame-8B achieves state-of-the-art results on most metrics.

In summary, our contributions are three-fold:

- We introduce **DynFrame**, a multimodal reasoning framework that enables the model itself to learn a context-aware retrieval policy, jointly predicting the temporal span and sampling density as part of its reasoning process.
- We curate **DM-CoT-74k** and **DM-RL-45k** to train adaptive retrieval and retrieval-grounded reasoning, and propose **SD-GRPO**, which uses the explicit retrieval boundary to assign role-specific token-level advantages. This provides targeted credit for retrieval quality and answer correctness, supporting more accurate reasoning based on retrieved evidence.
- Across six benchmarks spanning temporal grounding, grounded VideoQA, and long-form video understanding, DynFrame-4B is competitive with strong 7B–8B baselines, while DynFrame-8B achieves state-of-the-art results on most metrics. Code, data, and model weights will be released.

2 Related Work

Multimodal chain-of-thought for video reasoning. Multimodal chain-of-thought extends textual CoT (Wei et al., 2022) by letting intermediate reasoning interact with visual evidence. Video-R1 (Feng et al., 2025a), VideoChat-R1 (Li et al., 2025c), VideoRFT (Wang et al., 2025a), and Temporal-RLT (Li et al., 2025b) improve multi-step inference through supervised fine-tuning or reinforcement learning, but mostly use visual tokens supplied before generation. Thinking-with-video methods instead revisit evidence during inference. VITAL (Zhang et al., 2026) and LongVT (Yang et al., 2026) use tool-based clip retrieval; LOVE-R1 (Fu et al., 2025b) uses adaptive zoom-in; VideoChat-R1.5 (Yan et al., 2025) performs iterative perception; Video-o3 (Zeng et al., 2026) and VideoTemp-o3 (Liu et al., 2026) explore native interleaved tool invocation, while Open-o3 Video (Meng et al., 2025) grounds reasoning in explicit timestamps and boxes without tool calls. Their retrieval granularity, however, remains largely governed by external tools, preset zoom/spotlight modes, or system-defined visual-token budgets. Matching evidence density to each question can therefore require repeated retrieval, expanding the context and complicating training.

Frame sampling for video MLLMs. Uniform sampling (Maaz et al., 2024; Zhang et al., 2023; Jin et al., 2024) can miss brief events. Query-conditioned selectors such as AKS (Tang et al., 2025), FOCUS (Zhu et al., 2026), and Frame-Voyager (Yu et al., 2025) usually determine the frame set before reasoning. Qwen2.5-VL and Qwen3-VL (Bai et al., 2025b;a) support different frame rates and timestamps, but the input pipeline selects sampling density; Video-RTS (Wang et al., 2025d) adds frames at test time based on output consistency. In retrieval-based systems, density is often controlled by tool protocols or preset slow/fast configurations (Zhang et al., 2026; Fu et al., 2025b). DynFrame jointly generates the temporal span and FPS as native tokens conditioned on the question and reasoning context, making within-window sampling density a learned action.

Reinforcement learning for visual reasoning and tool use. GRPO (Shao et al., 2024b) has been extended to visual reasoning (Huang et al., 2026; Zhang et al., 2025) and tool use (Zheng et al., 2026). Standard outcome-based GRPO shares one rollout-level advantage across generated tokens. VITAL (Zhang et al., 2026) reweights optimization by task and sample difficulty; LongVT (Yang et al., 2026) combines answer, format, and temporal-grounding rewards; and LOVE-R1 (Fu et al., 2025b) supplements multi-step QA optimization with separately supervised single-step zoom-in training. For general LLM agents, GiGPO (Feng et al., 2025b) combines episode- and step-level advantages through state-based action grouping. SD-GRPO adds a separately normalized span-and-FPS advantage to the pre-injection sampling segment while retaining answer-derived credit throughout the same trajectory.

3 Method

We propose **DynFrame**, an end-to-end trainable video reasoning framework that unifies adaptive frame retrieval with dynamically interleaved vision-language reasoning within a single generative process (Fig. 2). Built upon Qwen3-VL (Bai et al., 2025a), DynFrame introduces three key designs: (i) a *tokenized retrieval interface*, where the model specifies which temporal span and at what sampling

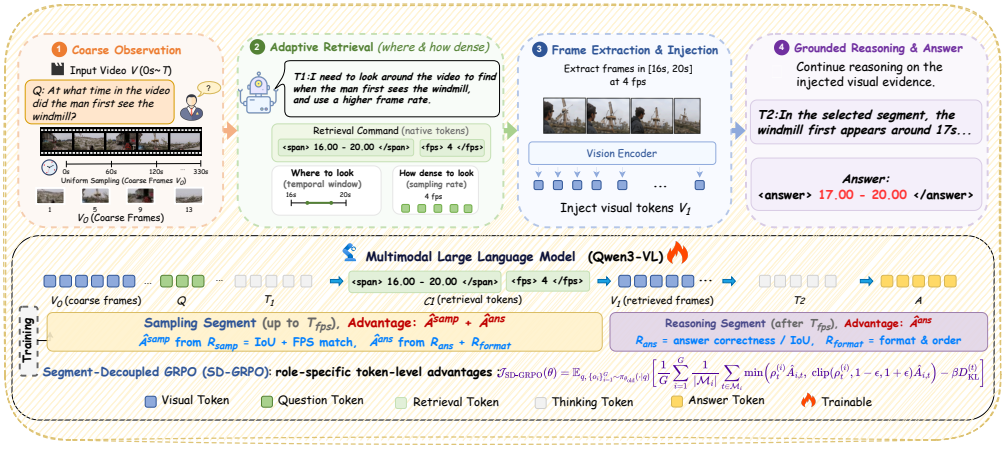


Figure 2: Overview of DynFrame. Tokenized temporal retrieval (, <fps>) triggers on-the-fly frame injection in a single autoregressive pass. SD-GRPO assigns segment-specific advantages at the retrieval boundary to credit sampling and answer reasoning separately.

density to retrieve by generating structured tokens; (ii) a *dynamic frame injection* mechanism that encodes the retrieved frames and inserts them into the decoding context on-the-fly; and (iii) *Segment-Decoupled GRPO*, which decouples rewards across response segments to separately optimize temporal selection and answer reasoning. We detail the dynamic multimodal CoT in §3.1, the two-stage training procedure in §3.2, and the dataset curation pipeline in §3.3.

3.1 Dynamic Multimodal Chain-of-Thought

3.1.1 Generation with Adaptive Frame Retrieval

Given a user question Q and an initial video observation V_0 (uniformly sampled frames), DynFrame generates a multimodal reasoning trajectory that interleaves textual reasoning with adaptive retrieval. In our main setting, this forms a three-stage process: *coarse reasoning* $\rightarrow$ *retrieval* $\rightarrow$ *grounded reasoning*. The model first produces an initial reasoning segment T_1 based on Q and a coarse understanding of V_0, then generates a retrieval command C_1 to request additional evidence from the original video. After the retrieved frames V_1 are injected, the model continues reasoning over the augmented visual context and produces the final answer A. The trajectory is:

$$\mathbf{s} = \{V_0,\, T_1,\, C_1,\, V_1,\, T_2,\, A\}. \tag{1}$$

In this work, we instantiate a single retrieval round. The same tokenized interface can be extended to multiple rounds by emitting additional retrieval commands.

Tokenized retrieval interface. We design a set of special tokens to express video evidence acquisition as part of the model output. The model emits t_s–t_e to specify a temporal window and <fps>f</fps> to specify the sampling frame rate, which together parameterize the retrieval command C_1. This turns frame selection from a fixed preprocessing choice into a learnable decision within the autoregressive generation trajectory, enabling adaptive, multi-granularity temporal sampling conditioned on the current reasoning context. Predicting both the temporal window and the sampling rate is essential, as different queries demand different granularities: a brief hand gesture requires dense frames within a narrow window, whereas long-form narrative understanding may only need sparse keyframes.

3.1.2 Dynamic Frame Injection

We perform on-the-fly frame injection during autoregressive generation. The closing token </fps> triggers retrieval: the system parses the preceding and <fps> fields to obtain (t_s, t_e, f) and requests

$$N = \lfloor (t_e - t_s) \times f \rfloor \tag{2}$$

frames uniformly distributed within $[t_s, t_e]$, capped by a maximum frame budget via uniform subsampling when necessary, and sampled separately from the initial observation. Retrieved frames

pass through the frozen vision encoder to produce decoder-compatible visual embeddings. For each temporal visual block, we construct a subsequence containing its timestamp on the original video timeline, vision boundary tokens, and $H'W'/m^2$ visual placeholder positions populated with the corresponding embeddings. Here, H', W' are the spatial patch-grid dimensions before spatial merging, and m is the spatial merge factor.

Let $\mathbf{x}_{1:L}$ denote the existing multimodal context through </fps>, and $\mathbf{v}_{1:K}$ the appended sequence of timestamp, boundary, and visual positions. We rebuild the KV cache by prefilling the complete extended context, then resume autoregressive decoding:

$$\mathbf{x}' = [\mathbf{x}_{1:L};\, \mathbf{v}_{1:K}], \qquad p_\theta(Y \mid \mathbf{x}') = \prod_{t=1}^{|Y|} p_\theta(y_t \mid \mathbf{x}', y_{<t}). \tag{3}$$

Here, Y denotes the subsequent reasoning and final answer, conditioned on both the preceding context and the retrieved visual evidence.

End-to-end optimization through discrete retrieval. We add , , <fps>, and </fps> to the model vocabulary as native retrieval tokens, generated alongside ordinary reasoning tokens. Together with the predicted span/FPS values, they express retrieval actions conditioned on the query and evolving multimodal context. These actions and the resulting visual observations form part of the same autoregressive trajectory, with retrieved evidence updating the context for subsequent reasoning and answer generation. This unified formulation enables end-to-end learning of retrieval and answer generation. The model can thus learn where to look and how densely to sample based on the query and the evidence needed to advance its reasoning.

3.2 Training Framework

Our training procedure consists of two stages: (1) cold-start supervised fine-tuning (SFT) to establish interleaved retrieval and reasoning behaviors, and (2) reinforcement learning via SD-GRPO to jointly improve temporal selection and grounded multimodal reasoning. We train on two curated multi-task datasets: **DM-CoT-74k** provides supervised trajectories with interleaved retrieval commands (span/FPS) and grounded reasoning, and **DM-RL-45k** provides question–video pairs with ground-truth temporal spans, FPS targets, and answers, enabling reward computation for both sampling quality and answer correctness. Full dataset details are in §3.3.

3.2.1 Cold-Start Supervised Fine-Tuning

We bootstrap DynFrame with SFT on **DM-CoT-74k** by maximizing the likelihood of interleaved reasoning and retrieval tokens:

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \log p_\theta(x_t \mid x_{<t}), \qquad \mathcal{M} = \{t \in \mathcal{A} : x_t \neq \texttt{<|video_pad|>}\}, \tag{4}$$

where $\mathcal{A}$ denotes non-padding positions in the assistant trajectory, including injected visual segments. We exclude <|video_pad|> tokens from the loss, as these positions contain visual embeddings rather than text targets. Timestamp tokens and the visual delimiters <|vision_start|> and <|vision_end|> remain supervised, preserving supervision over the timestamped visual structure.

3.2.2 Segment-Decoupled GRPO

GRPO with outcome supervision (Shao et al., 2024b) assigns the same group-relative advantage to all generated tokens in a rollout. After SFT, we observe that the model may retrieve relevant evidence yet answer incorrectly, or answer correctly using its initial video observations or prior knowledge despite an inaccurate retrieval. When the reward depends only on the final answer, it does not separately assess retrieval quality. Consequently, answer-based feedback can penalize useful retrieval decisions or reinforce poor ones during RL. To address this issue, we introduce **Segment-Decoupled GRPO (SD-GRPO)**, which adds retrieval-specific feedback before frame injection while retaining answer feedback throughout the rollout.

We initialize the policy from the SFT-trained model. At each RL iteration, given a question Q and video V, we sample G rollouts $\{o_i\}_{i=1}^{G}$ using the policy before the update, denoted by $\pi_{\theta_{\text{old}}}$.

The retrieval terminator </fps> at position T_{fps} divides each rollout into a *sampling segment* $o^{\text{samp}} = o_{1:T_{\text{fps}}}$, including preliminary reasoning and retrieval decisions, and a *reasoning segment* $o^{\text{reas}} = o_{T_{\text{fps}}+1:T}$. To evaluate retrieval quality and answer correctness separately, we define a sampling reward and an answer reward. We also use a format reward to encourage the model to adhere to our retrieval interface.

(1) Sampling reward. The sampling reward R_{samp} evaluates the quality of the frame acquisition decision, combining temporal span overlap with a smooth FPS matching score:

$$R_{\text{samp}} = \lambda_1 \cdot \text{IoU}\big([\hat{t}_s, \hat{t}_e],\ [t_s^*, t_e^*]\big) + \lambda_2 \cdot \left(1 - \frac{|\hat{f} - f^*|}{f_{\max}}\right), \tag{5}$$

where $[\hat{t}_s, \hat{t}_e]$ and $\hat{f}$ are the model's predictions, $[t_s^*, t_e^*]$ and f^* are the ground-truth annotations, and the FPS term decreases linearly with $|\hat{f} - f^*|$, normalized by $f_{\max}$=6 (all rates lie in $[1, 6]$ in our setting). The two terms play asymmetric but complementary roles: span IoU localizes *where* evidence lies, and the FPS term decides how densely to sample *within* that window, which matters mainly once the span is correct. Because Eq. (6) normalizes R_{samp} within each group, each term's influence scales with its coefficient times its within-group standard deviation. Despite their asymmetric roles, span IoU typically has a larger within-group variance than the FPS score, and the equal weighting λ_1=λ_2=0.5 performs best in our experiments.

(2) Answer reward. R_{ans} measures exact-match correctness for multiple-choice VideoQA and long-form video understanding, and temporal IoU between the final predicted and ground-truth spans for temporal grounding. For grounded VideoQA, it combines answer correctness with the temporal IoU of the predicted evidence window.

(3) Format reward. We additionally apply a rule-based format reward R_{format} to ensure the output follows the required structure (reasoning segment, span/FPS fields, and answer segment with correct ordering and pairing).

Token-level segment-decoupled advantage. We normalize the sampling reward and the answer-plus-format reward separately within each group:

$$\hat{A}_i^{\text{samp}} = \frac{R_{\text{samp},i} - \mu_{\text{samp}}}{\sigma_{\text{samp}} + \delta}, \qquad \hat{A}_i^{\text{ans}} = \frac{(R_{\text{ans},i} + R_{\text{format},i}) - \mu_{\text{ans}}}{\sigma_{\text{ans}} + \delta}, \tag{6}$$

where $(\mu_{\text{samp}}, \sigma_{\text{samp}})$ and $(\mu_{\text{ans}}, \sigma_{\text{ans}})$ are the within-group statistics of R_{samp} and $R_{\text{ans}} + R_{\text{format}}$, respectively, and δ is a small constant for numerical stability. Each optimized token receives:

$$\hat{A}_{i,t} = \begin{cases} \hat{A}_i^{\text{samp}} + \hat{A}_i^{\text{ans}} & \text{if } t \le T_{\text{fps}} \quad \text{(sampling segment)} \\ \hat{A}_i^{\text{ans}} & \text{if } t > T_{\text{fps}} \quad \text{(reasoning segment).} \end{cases} \tag{7}$$

Retrieval thus receives its own feedback without losing supervision from downstream answers; post-injection tokens receive only the answer-plus-format signal, since they cannot alter the completed retrieval decision.

Optimization. We use these advantages in the clipped GRPO objective:

$$\mathcal{J}_{\text{SD-GRPO}}(\theta) = \mathbb{E}_{q,\,\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|q)}$$
$$\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{M}_i|}\sum_{t\in\mathcal{M}_i}\min\big(\rho_t^{(i)}\hat{A}_{i,t},\ \text{clip}(\rho_t^{(i)}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\big) - \beta\, D_{\text{KL}}^{(t)}\right], \tag{8}$$

where $q = \{Q, V\}$ denotes the input question and video, and $\{o_i\}_{i=1}^{G}$ are G rollouts sampled from the behavior policy $\pi_{\theta_{\text{old}}}$. $\mathcal{M}_i = \{t : o_{i,t} \notin \mathcal{V}_{\text{pad}}\}$ excludes visual placeholder tokens (e.g., <|video_pad|>) from optimization. The importance ratio is $\rho_t^{(i)} = \pi_\theta(o_{i,t} \mid q, o_{i,<t}) / \pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})$, ϵ is the clipping range, and $D_{\text{KL}}^{(t)}$ is the per-token KL divergence against the reference policy π_{ref} with coefficient β.

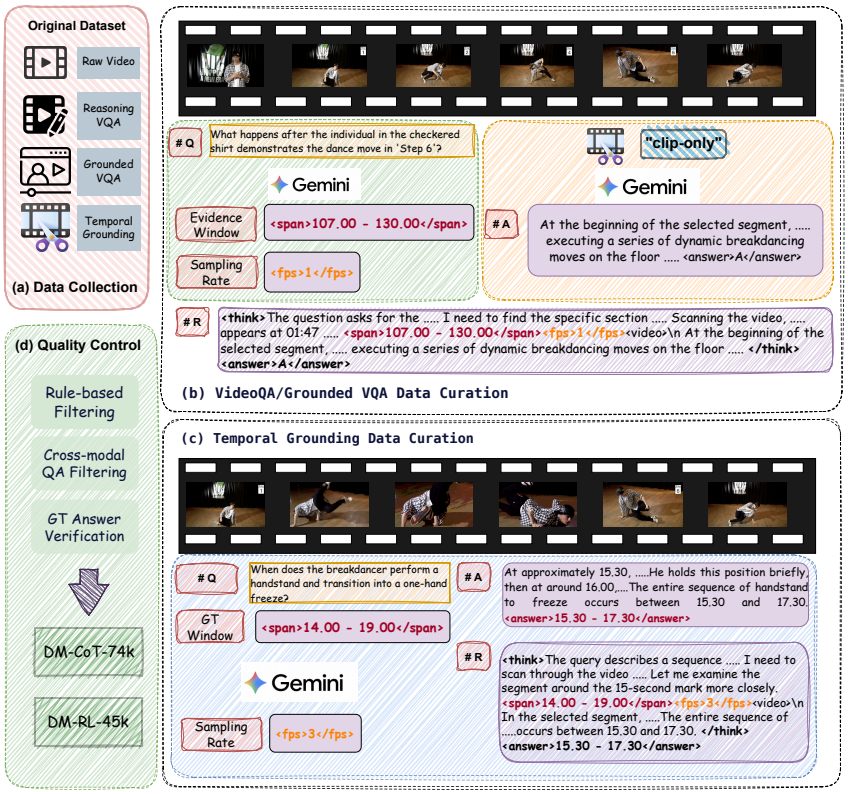


Figure 3: Data curation for DM-CoT-74k and DM-RL-45k. (a) Source benchmarks. (b) Gemini labels VideoQA spans, FPS, and answers under a prompt-level "clip-only" constraint. (c) Temporal grounding reuses ground-truth spans with Gemini FPS. (d) Rule-based and answer-consistency filters.

3.3 Training Data Curation

Training DynFrame requires trajectories that interleave textual reasoning with span+FPS retrieval commands and grounded answers. Since no public dataset provides this format, we curate two task-balanced mixtures over temporal grounding, VideoQA, and grounded VideoQA, sourced from Charades-STA (Gao et al., 2017), ActivityNet-MR (Krishna et al., 2017), Video-R1 (Feng et al., 2025a), ReXTime (Chen et al., 2024), and NExT-GQA (Xiao et al., 2024). As illustrated in Fig. 3, all samples are converted into the unified retrieval-augmented format of §3.1.1. All training samples are drawn from official training splits; evaluation samples are excluded.

For VideoQA and grounded VideoQA, whose training splits lack human-annotated evidence windows, we prompt Gemini-3-Pro (Google DeepMind, 2026) to identify the relevant evidence window, propose an FPS, and produce the answer under the constraint that it "only rewatched the proposed window", enforced at the prompt level during large-scale annotation. For temporal grounding, the human-annotated boundary is reused verbatim with a 0.5–2 s random margin to provide context, and Gemini is queried only for an activity-adaptive FPS conditioned on the span. Because no public video dataset provides per-segment FPS supervision, the FPS targets used during RL are inherited from this teacher; for temporal grounding, the IoU term in R_{samp} remains anchored to human-annotated boundaries.

We then apply two filtering stages: rule-based checks discard samples with missing or reversed retrieval fields, and an answer-consistency check drops teacher answers that disagree with the ground-truth label. Together they remove roughly 40% of raw teacher outputs. The final 74k SFT mixture comprises $\sim$30% temporal grounding, $\sim$45% VideoQA, and $\sim$25% grounded VideoQA; DM-RL-45k follows a similar ratio. Appendix B provides the full generation prompts, additional text-only verification, and manual spot checks.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate on six benchmarks spanning three task families: (i) *grounded VideoQA*—NExT-GQA (Xiao et al., 2024), with Acc and mIoU; (ii) *temporal sentence grounding*—Charades-STA (Gao et al., 2017) and ActivityNet-MR (Krishna et al., 2017), with R@$\{0.3, 0.5, 0.7\}$ and mIoU; (iii) *long-form video understanding*—Video-MME (w/o sub.) (Fu et al., 2025a), MLVU (M-Avg) (Zhou et al., 2025), and LVBench (Wang et al., 2025b), with multiple-choice accuracy.

Implementation. We build DynFrame on Qwen3-VL-Thinking at 4B and 8B with the visual encoder frozen. The initial pass uniformly samples at $f_1{=}2$ fps, and the adaptive retrieval round uses model-predicted $f_2 \in [1, 6]$ fps. SFT runs 2,000 steps at lr 1×10^{-5}, batch 256, on 64 H200 GPUs (AdamW); RL with SD-GRPO uses lr 1×10^{-6}, group $G{=}8$, temperature 1.0, for 1,000 further steps. Training configuration is detailed in Appendix A.1.

Table 1: Comparison of our method with existing methods across six benchmarks.

Model	Size	NExT-GQA		Charades-STA				ActivityNet-MR				V-MME	MLVU	LVB
		Acc	mIoU	R@.3	R@.5	R@.7	mIoU	R@.3	R@.5	R@.7	mIoU	w/o sub	M-Avg	Acc
General / Single-turn Video MLLMs														
Qwen2.5-VL (Bai et al., 2025b)	7B	59.5	15.4	64.7	43.1	22.8	43.6	41.6	23.2	9.5	28.9	65.1	70.2	45.3
InternVL3 (Zhu et al., 2025)	8B	–	–	68.0	43.0	26.0	45.4	37.0	22.0	12.0	25.8	66.3	71.4	–
Qwen3-VL (Thinking) (Bai et al., 2025a)	4B	76.3	32.9	79.2	68.5	46.4	59.0	49.8	30.4	14.1	34.5	68.9	75.7	53.5
Qwen3-VL (Thinking) (Bai et al., 2025a)	8B	71.3	39.7	81.7	70.6	47.8	59.9	51.8	32.4	15.6	36.2	71.8	75.1	55.8
Video-R1 (Feng et al., 2025a)	7B	77.3	–	71.0	56.0	31.6	50.4	56.4	38.4	17.7	38.1	61.4	–	–
Thinking-with-Video / Tool-Augmented Methods														
Temporal-RLT (Li et al., 2025b)	7B	78.7	37.3	80.2	68.3	44.5	57.9	56.9	38.4	20.2	39.1	57.6	–	–
VITAL (Zhang et al., 2026)	7B	78.7	43.0	83.1	72.0	46.7	59.9	70.9	50.8	31.6	49.8	66.1	–	46.2
LongVT (Yang et al., 2026)	7B	70.4	17.4	41.0	25.8	11.7	27.2	32.4	18.6	9.2	20.5	67.0	69.1	41.3
LOVE-R1 (Fu et al., 2025b)	7B	73.0	30.5	74.0	41.0	14.0	44.8	49.0	24.0	13.0	30.4	66.2	67.4	48.2
VideoChat-R1.5 (Yan et al., 2025)	7B	61.9	–	82.8	71.6	48.3	60.6	52.4	32.3	16.8	35.5	69.2	70.9	48.4
Video-o3 (Zeng et al., 2026)	7B	–	–	83.3	71.9	49.0	60.7	–	–	–	–	66.5	72.1	47.6
VideoTemp-o3 (Liu et al., 2026)	7B	76.4	33.4	–	–	33.0	57.8	–	–	26.7	45.3	64.5	–	43.0
DynFrame-4B (ours)	4B	77.6	38.7	82.4	69.8	45.1	59.4	70.4	49.2	28.6	47.5	69.2	76.3	54.8
DynFrame-8B (ours)	8B	**79.3**	**44.3**	**85.1**	**72.5**	**49.4**	**61.7**	**72.4**	**52.0**	**33.1**	**51.5**	**72.3**	**77.1**	**56.9**

4.2 COMPARISON WITH STATE-OF-THE-ART

Temporal sentence grounding. Charades-STA and ActivityNet-MR most directly probe the retrieval interface. As shown in Table 1, DynFrame-8B reaches 61.7 mIoU / 49.4 R@0.7 on Charades-STA, outperforming Video-o3 (60.7 / 49.0), VideoChat-R1.5 (60.6 / 48.3), and VITAL-7B (59.9 / 46.7). The 4B variant remains competitive with these strong 7B baselines, reaching 59.4 mIoU and 45.1 R@0.7. On ActivityNet-MR, DynFrame-8B obtains 72.4 / 52.0 / 33.1 / 51.5 on R@$\{0.3, 0.5, 0.7\}$ / mIoU, outperforming the competitive temporal grounding baseline VITAL-7B by $+1.5$ / $+1.2$ / $+1.5$ / $+1.7$. These gains indicate that predicting both the retrieval window and its sampling density is especially useful for precise temporal localization. **Grounded VideoQA.** On NExT-GQA, DynFrame-8B reaches 79.3 Acc and 44.3 mIoU, improving over VITAL-7B by $+0.6$ Acc and $+1.3$ mIoU and over Qwen3-VL-8B by $+8.0$ Acc and $+4.6$ mIoU. The 4B variant also achieves 77.6 Acc and 38.7 mIoU, outperforming Qwen3-VL-4B on both metrics. The simultaneous gain in answer accuracy and localization is consistent with SD-GRPO jointly optimizing the sampling decision and the final answer reasoning. **Long-form video understanding.** On Video-MME / MLVU / LVBench, DynFrame-8B reaches 72.3 / 77.1 / 56.9, improving over Qwen3-VL-8B by $+0.5$ / $+2.0$ / $+1.1$. It also surpasses recent tool-augmented baselines such as LongVT, LOVE-R1, VideoChat-R1.5, and Video-o3 on these long-form benchmarks. The 4B variant reaches 69.2 / 76.3 / 54.8, improving over Qwen3-VL-4B by $+0.3$ / $+0.6$ / $+1.3$ across all three benchmarks. These results show that learnable span–density retrieval benefits both precise grounding and long-form video understanding.

4.3 ABLATION STUDY

(a) Curated data versus executable retrieval. As illustrated in Table 2a, Qwen3-VL trained on DM-CoT-74k for identical SFT steps without retrieval or frame injection differs from backbone references by -2.2 / -12.3 / -0.6 / -2.6 on NExT-GQA / Charades-STA / MLVU / LVBench. Textual span and evidence supervision without matching frames may encourage ungrounded reasoning. DynFrame

Table 2: Ablations on the 8B model. N-GQA, V-MME, and LVB report answer accuracy; Cha reports mIoU; MLVU reports M-Avg.

(a) Curated SFT data vs. executable retrieval.

Variant	N-GQA	Cha	MLVU	LVB
Qwen3-VL backbone	71.3	59.9	75.1	55.8
SFT, no retrieval	69.1	47.6	74.5	53.2
DynFrame, SFT only	**77.9**	**60.8**	**75.9**	**56.3**

(b) Effectiveness of SD-GRPO.

Variant	N-GQA	Cha	MLVU	LVB
SFT only	77.9	60.8	75.9	56.3
+ vanilla GRPO	78.3	61.0	76.2	56.5
+ SD-GRPO	**79.3**	**61.7**	**77.1**	**56.9**

(c) Initial f_1; DynFrame uses dynamic f_2.

Variant	N-GQA	Cha	V-MME	LVB
Qwen3-VL, f_1=0.5	59.2	45.7	64.7	42.9
Qwen3-VL, f_1=2	71.3	59.9	71.8	55.8
DynFrame, f_1=0.5	79.1	61.5	72.3	56.9
DynFrame, f_1=2	**79.3**	**61.7**	**72.3**	**56.9**

(d) Retrieval FPS f_2 (DynFrame).

Variant	N-GQA	Cha	V-MME	LVB
f_2=2 (fixed)	76.9	59.8	70.2	56.2
f_2=3 (fixed)	77.0	60.2	70.4	56.1
f_2 dynamic	**79.3**	**61.7**	**72.3**	**56.9**

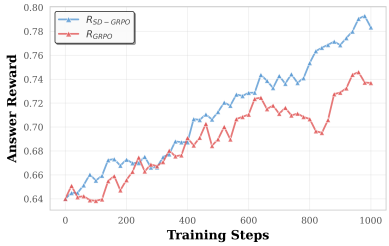
SFT supplies retrieved frames to subsequent reasoning and gains +8.8 / +13.2 / +1.4 / +3.1 over this control, supporting second-stage injection beyond supervision alone.

(b) SD-GRPO. To evaluate SD-GRPO's credit assignment across sampling and reasoning segments, we examine reward dynamics and downstream gains. Fig. 4 compares sampling and answer rewards computed from both methods' rollouts using the same definitions. This diagnostic scoring does not give GRPO segment-specific advantages: GRPO shares one trajectory-level advantage across generated tokens, whereas SD-GRPO adds sampling feedback before injection while retaining answer feedback throughout. SD-GRPO's higher late-stage rewards accompany downstream gains, as shown in Table 2b. GRPO improves Charades-STA / LVBench over SFT by +0.2 / +0.2, and SD-GRPO adds a further +0.7 / +0.4 over GRPO. SD-GRPO also improves MLVU by +1.2 over SFT.

(c) Robustness to f_1. To test whether DynFrame retrieves useful evidence from sparse initial observations, we reduce f_1 from 2 to 0.5 fps for both models. As shown in Table 2c, Qwen3-VL changes by −12.1 / −14.2 / −7.1 / −12.9 on NExT-GQA / Charades-STA / Video-MME / LVBench. DynFrame drops by only 0.2 on NExT-GQA and Charades-STA and is unchanged on the others, suggesting retrieval compensates for sparse initial sampling.



(a) Sampling reward R_{samp}.



(b) Answer reward R_{ans}.

Figure 4: Reward dynamics: SD-GRPO vs. vanilla GRPO.

(d) Dynamic FPS. To evaluate FPS prediction, we compare learned f_2 with fixed 2 and 3 fps. As shown in Table 2d, predicted FPS improves Charades-STA / Video-MME / LVBench by +1.9 / +2.1 / +0.7 over fixed f_2=2 and +1.5 / +1.9 / +0.8 over fixed f_2=3, supporting query-dependent density. Additional span/FPS controls are reported in Appendix A.3, and prediction analyses in Appendix C.

5 Conclusion

We presented DynFrame, a dynamic multimodal reasoning framework that turns visual evidence acquisition into a model-native decision. By predicting both the temporal window and the sampling density, DynFrame jointly decides where to retrieve and how densely to sample, acquiring task-adaptive, multi-granularity evidence with a single retrieval step. To train this behavior, we curated DM-CoT-74k and DM-RL-45k, and introduced Segment-Decoupled GRPO, which separately credits the sampling decision and the answer reasoning. Experiments across grounded VideoQA, temporal grounding, and long-form video understanding show that DynFrame-4B is competitive with strong 7B–8B baselines, while DynFrame-8B achieves new best results on most metrics.

AI USE STATEMENT

In this work, we used generative AI tools for *generating synthetic data sets, cleaning and reformatting datasets, providing feedback on experiments*. We have not used generative AI tools for *helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses, or supporting qualitative and thematic data analysis*, and *formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs* are not applicable to this work. Additionally, we used generative AI tools for *to aid or polish writing and improve readability*. We have reviewed all AI-assisted work, and two of the authors checked the code and all text in this paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not raise any ethical issues. All training and evaluation data come from public benchmarks; we did not collect sensitive personal data or release datasets that could potentially cause harm, and we will follow the licenses of the source datasets when releasing our annotations. All experiments were conducted in compliance with relevant institutional guidelines and legal regulations.

REPRODUCIBILITY STATEMENT

Section 3 specifies the tokenized retrieval interface, dynamic frame injection, and the SD-GRPO objective. Section 4.1 and Appendix A give the benchmarks, metrics, training settings, and hyperparameters. Section 3.3 and Appendix B describe the data sources, the complete data generation prompts, and the filtering and quality checks. We include the implementation code in the supplementary material, and to facilitate reproducibility, we will release the code, DM-CoT-74k, DM-RL-45k, and the model weights.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *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 Yu-Chiang F Wang. Rextime: A benchmark suite for reasoning-across-time in videos. *Advances in Neural Information Processing Systems*, 37:28662–28673, 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.

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. *Advances in Neural Information Processing Systems*, 38:99114–99137, 2025a.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2025b.

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

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

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5277–5285. IEEE, 2017.

Google DeepMind. Gemini 3 Pro Model Card, 2026. URL `https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf`.

Zefeng He, Xiaoye Qu, Yafu Li, Siyuan Huang, Daizong Liu, and Yu Cheng. Framethinker: Learning to think with long videos via multi-turn frame spotlighting. In *International Conference on Learning Representations*, volume 2026, pp. 74904–74933, 2026.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Fei zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13700–13710. IEEE, 2024.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 706–715, 2017.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025a. ISSN 2835-8856.

Hongyu Li, Songhao Han, Yue Liao, Junfeng Luo, Jialin Gao, Shuicheng Yan, and Si Liu. Reinforcement learning tuning for videollms: Reward design and data efficiency. *arXiv preprint arXiv:2506.01908*, 2025b.

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*, 2025c.

Wenqi Liu, Yunxiao Wang, Shijie Ma, Meng Liu, Qile Su, Tianke Zhang, Haonan Fan, Changyi Liu, Kaiyu Jiang, Jiankang Chen, et al. Videotemp-o3: Harmonizing temporal grounding and video understanding in agentic thinking-with-videos. *arXiv preprint arXiv:2602.07801*, 2026.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12585–12602, 2024.

Jiahao Meng, Xiangtai Li, Haochen Wang, Yue Tan, Tao Zhang, Lingdong Kong, Yunhai Tong, Anran Wang, Zhiyang Teng, Yujing Wang, et al. Open-o3-video: Grounded video reasoning with explicit spatio-temporal evidence. *arXiv preprint arXiv:2510.20579*, 2025.

OpenAI. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024a.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024b.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29118–29128. IEEE, 2025.

Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. Videorft: Incentivizing video reasoning capability in mllms via reinforced fine-tuning. *Advances in neural information processing systems*, 38:4350–4376, 2025a.

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

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*, 2025c.

Ziyang Wang, Jaehong Yoon, Shoubin Yu, Md Mohaiminul Islam, Gedas Bertasius, and Mohit Bansal. Video-RTS: Rethinking reinforcement learning and test-time scaling for efficient and enhanced video reasoning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 28126–28140, Suzhou, China, November 2025d. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1428.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800.

Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13204–13214. IEEE, 2024.

Ziang Yan, Yinan He, Xinhao Li, Zhengrong Yue, Xiangyu Zeng, Yali Wang, Yu Qiao, Limin Wang, and Yi Wang. Videochat-r1.5: Visual test-time scaling to reinforce multimodal reasoning by iterative perception. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 119152–119184. Curran Associates, Inc., 2025. doi: 10.52202/085713-3976.

Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, et al. Longvt: Incentivizing "thinking with long videos" via native tool calling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33816–33826, 2026.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, et al. Frame-voyager: Learning to query frames for video large language models. In *International Conference on Learning Representations*, volume 2025, pp. 84154–84179, 2025.

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 *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pp. 543–553, 2023.

Haoji Zhang, Xin Gu, Jiawen Li, Chixiang Ma, Sule Bai, Chubin Zhang, Bowen Zhang, Zhichao Zhou, Dongliang He, and Yansong Tang. Thinking with videos: Multimodal tool-augmented reinforcement learning for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32903–32914, 2026.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1859–1869, October 2025.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and XingYu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 126775–126798, 2026.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701. IEEE, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Zirui Zhu, Hailun Xu, Yang Luo, LIU YONG, Kanchan Sarkar, Zhenheng Yang, and Yang You. Focus: Efficient keyframe selection for long video understanding. In *International Conference on Learning Representations*, volume 2026, pp. 98463–98486, 2026.

SUPPLEMENTARY MATERIAL

A TRAINING CONFIGURATION AND ADDITIONAL ANALYSES

A.1 TRAINING CONFIGURATION

Table 3 summarizes the two training stages. SFT initializes DynFrame from Qwen3-VL-Thinking, and SD-GRPO continues from the corresponding SFT checkpoint. Both stages use the same span–FPS retrieval interface and a frozen visual encoder.

Table 3: Training hyperparameters and configuration of DynFrame.

Hyperparameter / configuration	**Stage 1: SFT**	**Stage 2: SD-GRPO**
Stage-specific training settings		
Initialization	Qwen3-VL-Thinking	SFT checkpoint
Training data	DM-CoT-74k	DM-RL-45k
Objective	Token likelihood	SD-GRPO
Learning rate	1×10^{-5}	1×10^{-6}
Training steps	2,000	1,000
Batch size	256	64
Optimizer	AdamW	AdamW
Training hardware	64× H200 GPUs	64× H200 GPUs
Rollouts per question (G)	–	8
Rollout temperature	–	1.0
Sampling reward weights (IoU / FPS)	–	0.5 / 0.5
FPS normalizer ($f_{\max}$)	–	6
KL coefficient / clipping range (β / ϵ)	–	0.02 / 0.1
Shared model and retrieval configuration		
Initial sampling rate (f_1)	2 fps	
Requested retrieval rate (f_2)	1–6 fps	
Retrieval rounds	1	
Loss masking		
Masked visual token	<|video_pad|>	<|video_pad|>

A.2 MASKING STRATEGY FOR INJECTED FRAMES

Masking convention. The <|video_pad|> token marks a visual-feature position in the token-ID sequence. Before decoder computation, the input embedding at that position is replaced by an encoded video feature; the placeholder ID remains available for identifying and masking the position. Thus, loss masking excludes the corresponding prediction targets, while the visual features remain available to attention. During SFT, labels at these positions are set to the ignore index `-100`, while timestamp tokens and <|vision_start|> / <|vision_end|> delimiters in assistant video segments remain supervised. During SD-GRPO, <|video_pad|> positions are excluded from the policy loss and KL term, while timestamps and vision boundaries are retained. Table 4 compares downstream performance under two training masks. Masking all video-related tokens, including timestamps and vision boundaries, degrades performance relative to masking only <|video_pad|> positions. These results support retaining supervision over temporal metadata and vision boundaries when aligning injected frames with the reasoning context.

Table 4: Masking strategy ablation on DynFrame-8B. NExT-GQA and Charades-STA report mIoU; Video-MME and LVBench report accuracy.

Masking strategy	NExT-GQA	Charades-STA	Video-MME	LVBench
All video tokens	18.1	26.4	36.3	39.4
Video pad only	**44.3**	**61.7**	**72.3**	**56.9**

A.3 Separating Span Selection and Density Prediction

To assess the contributions of span selection and adaptive sampling density, we cross the span source with the density policy on NExT-GQA. Comparing spans from the SFT-stage and final RL-stage checkpoints under the same density policy examines the effect of the span source. Comparing fixed $f_2 = 2$ with predicted FPS under the same span-source setting examines the benefit of density prediction. Table 5 reports answer accuracy for the four combinations. The density comparison is fixed versus predicted FPS, rather than a comparison between SFT-stage and RL-stage FPS predictors.

Table 5: Crossed span-source and density-policy control on NExT-GQA for DynFrame-8B. Each cell reports answer accuracy.

Span source	Fixed density ($f_2 = 2$)	Predicted density
SFT-stage span	75.8	77.9
RL-stage span	76.9	79.3

With SFT-stage spans, predicted FPS improves answer accuracy over fixed FPS by $+2.1$ points; with RL-stage spans, the gain is $+2.4$. Using RL-stage rather than SFT-stage spans adds $+1.1$ under fixed FPS and $+1.4$ under predicted FPS. Both component changes improve answer accuracy in the tested configurations.

B Data Generation Prompts and Quality Checks

We provide the complete prompts used to query Gemini-3-Pro (Google DeepMind, 2026) for constructing our training data. We design two complementary prompts tailored to the annotation characteristics of each task type. For VideoQA and grounded VideoQA, whose training splits lack human-annotated evidence windows (NExT-GQA annotates windows only in its validation and test splits, and ReXTime's training spans are machine-generated), we query Gemini to produce both temporal localization and answer reasoning from scratch. For temporal grounding, where ground-truth temporal boundaries already exist, we adopt a reformulative strategy by prompting Gemini to expand, clean, and canonicalize existing reasoning traces while preserving the original annotations. Both prompts share a unified adaptive FPS selection guideline to ensure consistent, content-aware sampling across the training mixture. Appendix B.3 reports quality and robustness checks.

B.1 VideoQA & Grounded VideoQA Prompt

The prompt in Figure 5 performs two-stage annotation in a single API call via a structured JSON with two fields. In `zoom_in_cot`, the model reasons about which temporal segment contains the required visual evidence and concludes with `<span>` and `<fps>` tags, without predicting or guessing the answer. In `answer_cot`, the model assumes it has watched **only** that segment at the specified FPS and reasons step by step to produce the final answer. A three-tier FPS guideline (1–2 fps for static scenes, 3–4 fps for moderate dynamics, 5–6 fps for rapid actions) is embedded to ensure activity-adaptive sampling density.

B.2 Temporal Grounding Prompt

The prompt in Figure 6 reformulates existing reasoning traces rather than annotating from scratch. Given a trace with ground-truth temporal boundaries, the model expands the span by a random margin of 0.5–2.0 s on each side for contextual retrieval, relocates the expanded `<span>` and an adaptive `<fps>` tag to the front of the chain, and removes redundant temporal descriptions. The original unexpanded boundaries serve as the supervision signal.

B.3 Data Quality and Robustness Checks

Verification and cleaning. We apply structural and answer-consistency filtering to the generated traces. A second, text-only teacher call checks whether the proposed span and FPS agree with the reasoning evidence and event dynamics. To verify that Gemini infers a reasonable FPS from the full

video and question in a single call, we crop 100 videos to Gemini's predicted spans offline and ask Gemini to re-predict the FPS from only the clip and the original question; the re-predicted FPS agrees with the original label on 99% of them. We also manually inspect 30 examples per source dataset for output format, retrieved span and FPS, reasoning, and task labels, checking that the spans and FPS match the visual cues. Training samples come from official training splits; evaluation samples are excluded from the training mixtures.

Robustness to the annotation model. We apply the same prompts and annotation pipeline to Qwen3-VL-235B. On samples retained by both teacher pipelines, the temporal windows agree at a mean IoU of 95.4%, and 98.6% of FPS labels differ by at most one frame per second. This measures agreement between the teachers' annotations, rather than downstream performance after retraining DynFrame with the alternative annotations.

Optimality of the teacher's FPS. To test whether the teacher's FPS choice is near-optimal independently of localization error, we fix the human-annotated evidence windows of NExT-GQA (available only in its validation and test splits) and vary only the FPS used to sample each crop. Answer accuracy is 83% at the teacher-selected density, compared with 81% at 1 fps above it and 77% at 1 fps below it. The teacher's choice performs best among these tested densities.

Prompt for VideoQA & Grounded VideoQA Data Generation

System Prompt: You are a video analysis assistant. Answer based solely on the visual content of the video. Rely strictly on visual cues; ignore audio. Respond in valid, raw JSON only.

User Prompt:
You will respond in valid, raw JSON with exactly two top-level keys.

JSON Structure:

```
{
  "zoom_in_cot": "Your step-by-step reasoning for selecting
   the specific video segment. This string MUST end strictly
   with the time span and FPS tags in the format
   <span>START - END</span> <fps>N</fps>.",

  "answer_cot": "Your step-by-step reasoning after 'watching'
   the selected segment. This string MUST end strictly with
   the final answer wrapped in
   <answer>OPTION_LETTER</answer>."
}
```

Workflow

1. Read the user's question and infer the precise visual evidence required.
2. Describe, step by step, how you would locate the segment that likely contains this evidence. Explain why this portion is critical.
3. Based on the nature and speed of the target activity, recommend an appropriate FPS for analysis.
4. Conclude your "zoom_in_cot" reasoning by appending the time span and FPS tags. **Do NOT** provide or guess the answer in this field.
5. After that, imagine you have now watched **ONLY** that segment at the specified FPS. Using the visual information in the segment, reason step-by-step to reach the answer.
6. This reasoning plus the final answer (wrapped in `<answer></answer>`) goes into "answer_cot".

FPS Selection Guideline

- **1–2 fps**: Static or quasi-static scenes with minimal temporal variation (e.g., reading text/signs, appearance attributes, object identification, colors, a person standing still).
- **3–4 fps**: Moderately dynamic scenarios with clear temporal progression (e.g., general activities, object interactions, walking, cooking step-by-step, gesture recognition, conversations).
- **5–6 fps**: Rapid, fine-grained actions requiring frame-by-frame analysis (e.g., fast actions, rapid motion, counting quick events, sports movements, catching/throwing a ball).

Strict Rules

- Output must be raw JSON only (no markdown fences like ```` ```json ````).
- Time span must use seconds with two decimal places (e.g., 5.00 - 9.00).
- The "zoom_in_cot" value must end with the closing tag `</fps>`. Do not add any punctuation, whitespace, or words after this tag.
- The final answer must appear only once, inside `<answer></answer>`, at the very end of "answer_cot".
- Inside `<answer></answer>`, output only the letter A, B, C, D… (do not include the option text).
- FPS must be an integer between 1 and 6.

Question: {question}
Options: {options}

Figure 5: Data generation prompt for VideoQA. We prompt Gemini-3-Pro (Google DeepMind, 2026) to jointly perform temporal evidence identification with adaptive FPS recommendation and clip-constrained answer generation in a single call via two structured JSON fields. The FPS selection guideline ensures the sampling rate matches the visual dynamics of the target activity.

Prompt for Temporal Grounding Data Generation

System Prompt: You are a video analysis assistant specializing in temporal grounding. Respond in valid, raw text only following the exact output format specified below.

User Prompt:
You are given a temporal grounding reasoning trace that contains a thinking process and an answer. Your task is to **reformulate** it into our canonical format, **expand the temporal window for contextual evidence**, and **recommend an appropriate FPS** for the identified temporal span. Perform the following operations:

Workflow

1. Starting from the original temporal span in the input, expand the start and end boundaries by a random margin of 0.5–2.0 seconds on each side to provide additional temporal context.
2. Move the expanded temporal span to the front of the thinking process, formatted as `<span>START - END</span>` (seconds with two decimal places, e.g., `<span>29.50 - 74.00</span>`).
3. Remove redundant or repeated temporal descriptions in the thinking process. Keep the reasoning concise and non-repetitive.
4. Based on the duration of the **expanded span** and the nature of the activity described, append an FPS tag `<fps>N</fps>` immediately after the `<span>` tag.
5. Extract only the **original** start and end timestamps as the final answer (e.g., `30.00 - 72.00`).

FPS Selection Guideline

- **1–2 fps**: Static or quasi-static scenes with minimal temporal variation (e.g., reading text/signs, appearance attributes, object identification, colors, a person standing still).
- **3–4 fps**: Moderately dynamic scenarios with clear temporal progression (e.g., general activities, object interactions, walking, cooking step-by-step, gesture recognition, conversations).
- **5–6 fps**: Rapid, fine-grained actions requiring frame-by-frame analysis (e.g., fast actions, rapid motion, counting quick events, sports movements, catching/throwing a ball).

Output Format

```
<think>
<span>EXPANDED_START - EXPANDED_END</span> <fps>N</fps>
[Rest of the cleaned reasoning...]
</think>
<answer>ORIGINAL_START - ORIGINAL_END</answer>
```

Strict Rules

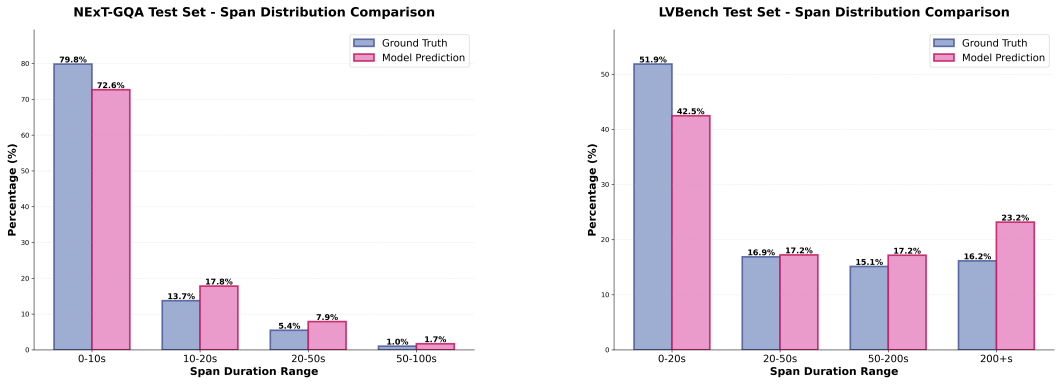
- Inside `<think>`, the first line must be `<span>EXPANDED_START - EXPANDED_END</span> <fps>N</fps>`.
- FPS must be an integer between 1 and 6.
- The `<span>` must contain the expanded temporal span.
- The `<answer>` must contain only the original, unexpanded start and end timestamps, with no additional text.
- Remove redundant temporal descriptions but preserve the core reasoning logic.

Input: {current_input} **Output:**

Figure 6: Data reformulation and FPS annotation prompt for temporal grounding. Given an existing reasoning trace with ground-truth temporal boundaries, we prompt Gemini-3-Pro (Google DeepMind, 2026) to canonicalize the format by expanding the temporal window for contextual evidence, relocating the contextual span to the front, cleaning redundant descriptions, recommending an activity-adaptive FPS, and extracting the original minimal answer.

C Analysis of Temporal Span and FPS Prediction

We analyze the temporal span and FPS predictions of DynFrame to illustrate its learned retrieval behavior.



(a) NExT-GQA test set. (b) LVBench test set.

Figure 7: Temporal span distribution comparison between ground-truth annotations and model predictions. (a) NExT-GQA; (b) LVBench. Across both benchmarks, the model systematically shifts probability mass from shorter to longer spans, predicting broader temporal windows to capture sufficient contextual evidence for reasoning.

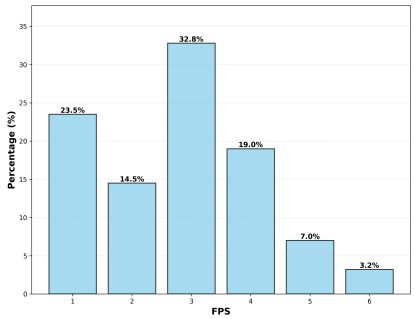
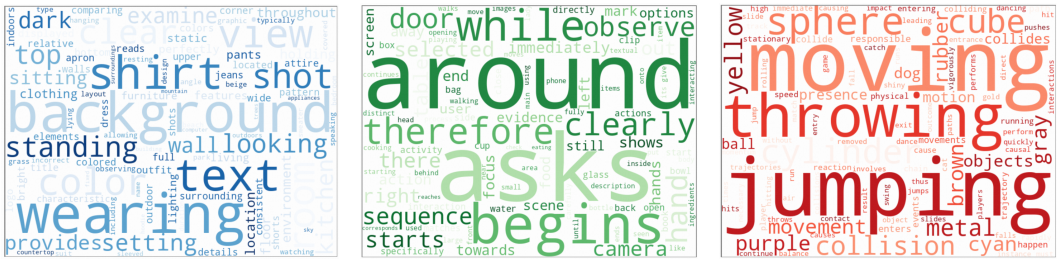


Figure 8: Predicted retrieval FPS distribution (f_2) of DynFrame on NExT-GQA.



(a) 1–2 FPS (b) 3–4 FPS (c) 5–6 FPS

Figure 9: Word clouds of content keywords by retrieval FPS band. Low FPS captures static descriptors; medium FPS captures sequential activities; high FPS captures rapid actions.

C.1 Temporal Span Prediction

Figure 7 compares the predicted and ground-truth span distributions on NExT-GQA (short-video) and LVBench (long-video). On NExT-GQA, ground-truth spans are heavily concentrated in the 0–10 s range, and the model's predictions closely follow this distribution while slightly redistributing mass toward longer spans, which suggests accurate localization on short-form videos. On LVBench, where ground-truth spans are distributed more evenly across all duration ranges and are notably longer overall, the model consistently predicts correspondingly broader temporal windows for extended segments—mirroring the same adaptive behavior observed on shorter videos. This consistency across vastly different video lengths suggests that the model has learned a robust, duration-aware grounding strategy rather than a dataset-specific bias, with predictions shifting moderately toward wider windows to capture sufficient contextual evidence for downstream reasoning.

C.2 Dynamic FPS Prediction

Figure 8 shows the distribution of predicted retrieval FPS (f_2) of DynFrame on NExT-GQA. Most predictions (89.8%) fall within 1–4 fps. This concentration is well-aligned with the nature of most video understanding tasks, where moderate frame rates already provide sufficient visual information for accurate reasoning. Nevertheless, for highly dynamic scenes that demand dense temporal sampling to capture rapid changes within very short intervals, DynFrame also predicts higher frame rates (5–6 fps; 10.2% of predictions), adapting sampling density to content dynamics.

To further investigate what drives different FPS predictions, we extract the most frequent content keywords within three FPS bands across all evaluation test sets and visualize them as deduplicated word clouds (Figure 9). A clear semantic gradient emerges across the bands. **Low FPS (1–2 fps)** is dominated by static descriptors: *standing*, *wearing*, *background*, *shirt*, *wall*, *text*—scene descriptions and appearance attributes with minimal temporal variation. **Medium FPS (3–4 fps)** features sequential-activity terms: *around*, *begins*, *sequence*, *asks*, *observe*—multi-step procedures and conversational interactions that require tracking temporal progression but not rapid motion. **High FPS (5–6 fps)** is characterized by rapid-action keywords: *moving*, *throwing*, *jumping*, *collision*, *sphere*, *cube*—fast-paced physical interactions where dense sampling is essential to capture critical state transitions. This semantic stratification suggests that DynFrame learns a meaningful mapping from content dynamics to sampling frequency. Together with the span prediction strategy analyzed above, the two mechanisms work synergistically: broader spans increase temporal coverage, while adaptive FPS controls sampling density within the retrieved window.

D ADDITIONAL QUALITATIVE EXAMPLES

We present additional qualitative examples to illustrate the behavior of DynFrame across diverse question types and video domains. Each example shows the model's full reasoning trajectory, including the predicted temporal span, adaptive FPS selection, and the grounded answer derivation.

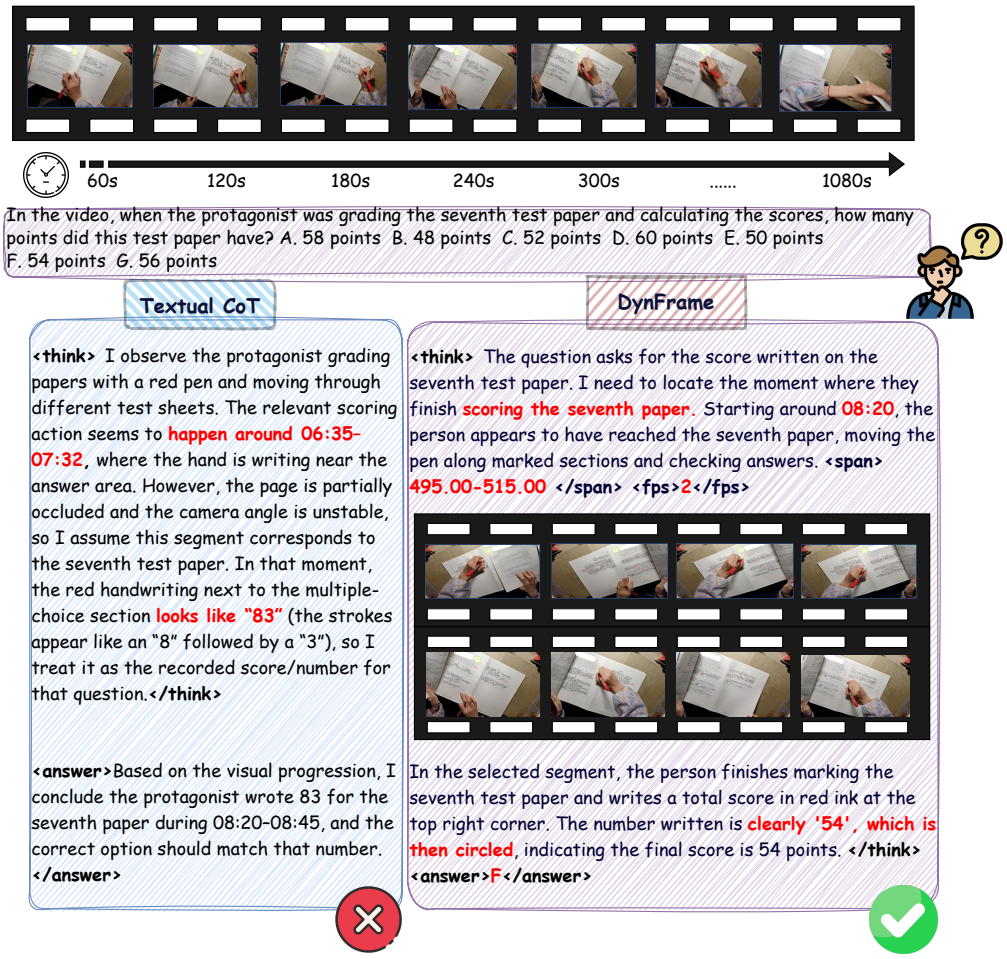


Figure 10: Case study 1: reading a handwritten score in a long video. DynFrame locates the 20 s segment in which the seventh test paper is graded and, at 2 FPS, reads the circled total of 54 points, whereas the textual CoT baseline misreads the score. A subset of the retrieved frames is shown.

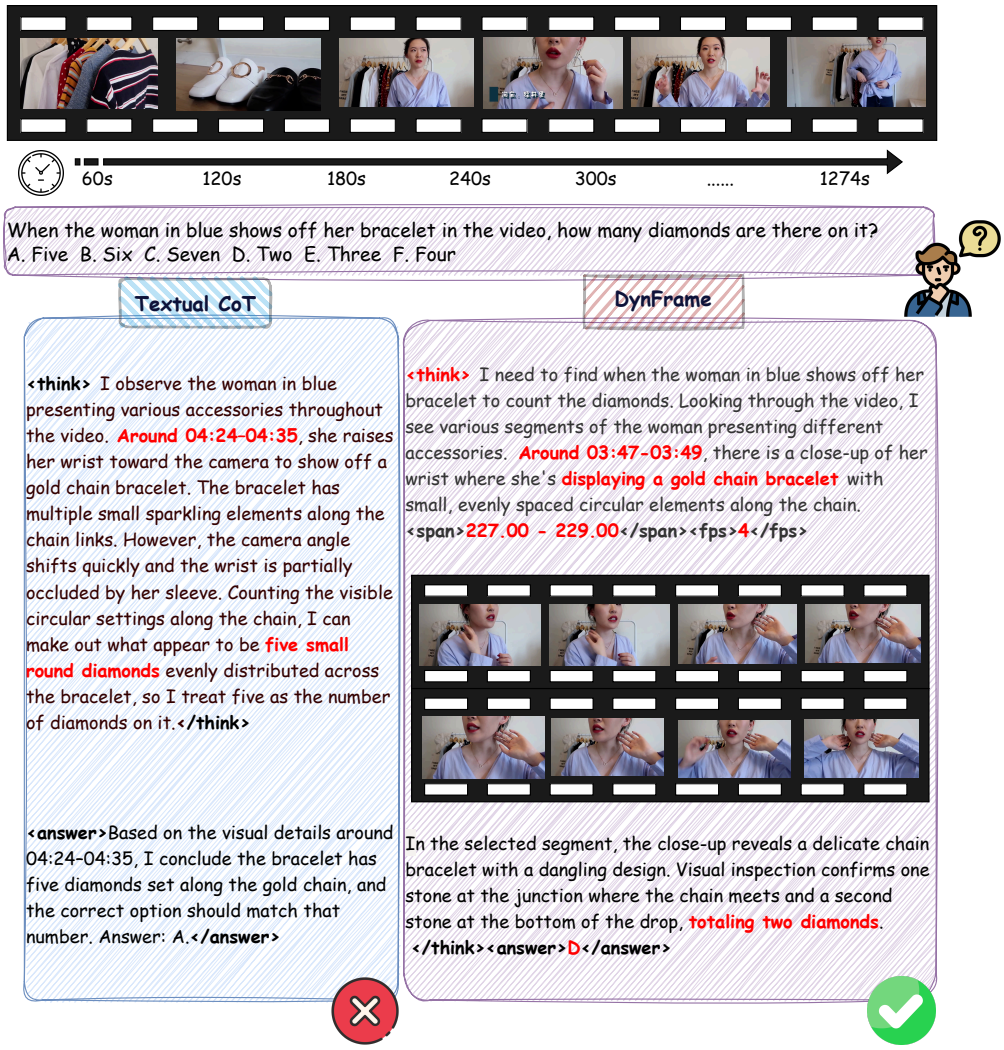


Figure 11: Case study 2: fine-grained object counting. DynFrame successfully locates the 2 s clip showcasing the bracelet in the long video and accurately counts the number of diamonds at 4 FPS, whereas the textual CoT baseline fails.