StreamReady: Learning *What* to Answer and *When* in Long Streaming Videos

Supplementary Material

In this supplementary material, we provide additional ablation studies in §A, followed by additional quantitative and qualitative analysis in §B. Next, we provide the details of ProReady-QA generation pipeline in §C, followed by additional implementation details in §D. Finally, we outline some directions for future research in §E.

A. Additional Ablation

We perform additional ablation studies on 3 tasks of ProReady-QA (REC, GSD, CTD), 2 tasks of StreamingBench (real-time perception and contextual reasoning), and VideoMME (long) to show the generalization of our framework on both proactive, and non-proactive streaming, and offline long-video benchmarks.

Contribution of Each Memory Bank. Table 6 (*top*) shows that short-term memory alone is insufficient for long streaming videos, since it captures only local context and quickly loses the long-range cues needed for temporal aggregation and causal reasoning. Adding long-term memory, thus, brings substantial gains across all datasets by capturing broader temporal context. The largest improvement comes from contextual memory, which allows the model to reuse prior QA information, maintain cross-turn consistency, and exploit long-range dependencies essential for multi-turn dialogue. This is especially evident on StreamingBench's contextual reasoning task and in the ARS gains on ProReady-QA, where stronger evidence modeling improves both accuracy and timing. While, VideoMME-Long benefits from long-term memory, contextual memory/reasoning is irrelevant for this benchmark due to its single-question format. Overall, combining hierarchical visual memory tree with lightweight contextual memory is key for StreamReady's reliable multi-turn reasoning in long-stream settings.

Design Choice of Visual Memory Tree. Existing long-term memory strategies typically rely on either similarity-based grouping [4, 37] or caption-based summarization [47, 54] to control memory growth and preserve information. Table 6 (*middle*) compares these approaches in both flat and hierarchical settings. Flat variants of both methods perform poorly, since collapsing all information into a single pool erases fine-grained cues needed for detailed evidence retrieval in long videos. Adding hierarchy alleviates this by organizing memory into increasingly abstract levels, reducing tokens while retaining the subtle patterns essential for long-form reasoning. Within this structure, similarity-based clustering consistently outperforms caption-based summarization, which is prone to semantic drift and incurs higher compute and latency costs. Our adaptive hierarchical clustering extends similarity-based designs across multiple abstraction levels, preserving temporal structure while keeping memory compact. This yields the strongest accuracy and ARS on ProReady-QA and the best performance on StreamingBench and VideoMME-Long, demonstrating the effectiveness of adaptive coarse-to-fine clustering for scalable long-term memory construction.

Design Choice of Query-Aware Reasoning. Table 6 (*bottom*) analyzes how different forms of query-aware reasoning affect performance. Without query-aware retrieval, the model cannot focus on the correct memory region or judge evidence sufficiency, leading to low accuracy and ARS. Short-term awareness offers modest gains but remains limited because it captures local changes but misses the long-range dependencies needed for long-term comprehension. Adding long-term awareness yields much larger improvements, with centroid-level reasoning consistently outperforming prototype-level reasoning across all datasets. Because centroids preserve finer visual details, it enables more precise evidence selection and better timing; whereas prototype-only reasoning loses cues needed for accurate temporal localization. The coarse-to-fine design of using prototypes for broad context and centroids for detailed refinement achieves the best results, delivering the strongest accuracy and ARS on all ProReady-QA tasks. A similar trend appears in StreamingBench and VideoMME-Long, where coarse-to-fine retrieval supports both global relevance filtering and precise evidence grounding. These results indicate that layered long-term reasoning not only improves overall performance but also enhances answer readiness by enabling more reliable assessment of evidence sufficiency.

B. Additional Analysis and Discussion

B.1. Model Behavior Analysis

Effect of Level Capacity and Depth of Visual Memory Tree. Figure 7 shows that each level of the visual memory tree has an optimal capacity beyond which accuracy plateaus while ARS declines. For level-1 raw frames, accuracy quickly stabilizes but ARS gradually drops as the FIFO buffer grows, since larger buffers slow retrieval and introduce timing penalties. The effect is sharper for level-2 centroids: moderate cluster counts retain the mid-level temporal detail needed to track subtle event changes across long videos, but larger centroid sets dilute discriminative structure and add latency, causing ARS to drop sharply. This matters across all benchmarks: ProReady-QA requires step-

Table 6. **Additional architectural ablation studies** on ProReady-QA, StreamingBench, and VideoMME long.

Method	ProReady-QA						StreamingBench		Vid-MME
	REC		GSD		CTD		Real	Context.	Long
	Acc.	ARS	Acc.	ARS	Acc.	ARS	Acc.	Acc.	Acc.
Contribution of Each Memory Bank									
Short-Term Memory ($\mathcal{M}_{V1}$)	12.7	0.51	24.1	0.56	31.7	0.49	41.3	12.4	36.7
+ Long-Term Memory ($\mathcal{M}_{V2}, \mathcal{M}_{V3}$)	37.6	0.66	55.3	**0.68**	40.7	0.58	69.4	32.1	**62.6**
+ Contextual Memory ($\mathcal{M}_{C}$)	**39.6**	**0.68**	**61.2**	**0.68**	**43.5**	**0.59**	**78.3**	**48.2**	N/A
Design choice of Visual Memory Tree									
Flat w/ similarity [4]	32.6	0.59	47.8	0.61	32.2	0.50	72.5	42.4	58.4
Flat w/ captioner [31]	34.5	0.50	51.7	0.52	34.7	0.43	71.2	43.8	60.1
Hierarchical w/ similarity [18]	37.7	0.57	59.2	0.63	40.6	0.54	74.5	45.2	62.3
Hierarchical w/ captioner [47]	35.8	0.54	60.3	0.58	39.4	0.51	72.6	44.3	62.1
Hierarchical w/ adaptive clustering	**39.6**	**0.68**	**61.2**	**0.68**	**43.5**	**0.59**	**78.3**	**48.2**	**62.6**
Design Choice of Query-Aware Reasoning									
No awareness	29.6	0.41	44.2	0.34	22.8	0.38	64.3	37.1	53.4
Short-term (ST) aware	30.1	0.41	49.6	0.37	31.8	0.39	68.2	37.5	56.2
ST + Centroid-level long-term aware (only $\mathcal{M}_{V2}$)	38.1	0.67	60.7	0.63	42.7	0.57	76.8	45.1	61.9
ST + Prototype-level long-term aware (only $\mathcal{M}_{V3}$)	32.3	0.44	52.2	0.54	39.1	0.46	64.7	41.4	59.4
ST + Coarse-to-fine long-term aware	**39.6**	**0.68**	**61.2**	**0.68**	**43.5**	**0.59**	**78.3**	**48.2**	**62.6**

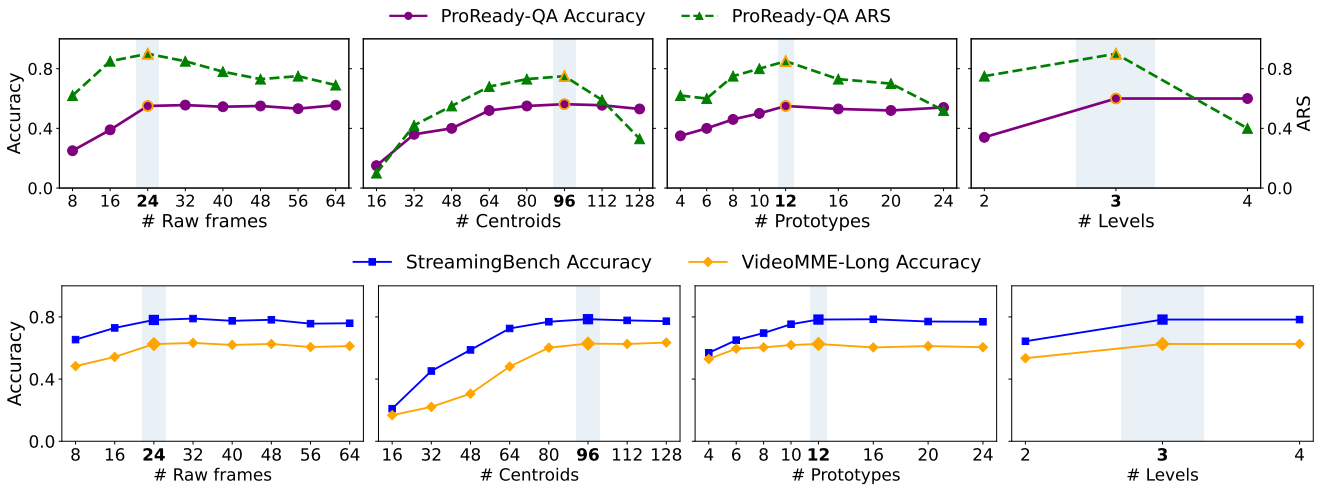

Figure 7. **Effect of level capacity and depth of visual memory tree on overall performance.** When varying the capacity of a specific level, the capacities of all other levels are held fixed. For the 2-level variant, the centroid and prototype capacities are merged into a single long-term level. For the 4-level variant, the prototype capacity is split across two abstraction layers to create two prototype levels. The selected capacity/depth is highlighted.

wise, causal, and clue-based evidence; StreamingBench relies on real-time cues and cross-turn context; VideoMME-Long demands precise long-range localization; so losing mid-level detail directly harms reasoning and timing. Level-3 prototypes show a similar pattern: small sets provide useful abstraction, while overly large sets add noise and latency without improving retrieval performance. Varying the depth of the tree further confirms that a 3-level structure balances abstraction and detail best; 2-level variants lose necessary mid-level cues, while 4-level designs over-fragment memory and introduce timing penalties. Overall, a configuration of 24 frames, 96 centroids, and 12 prototypes achieves strong accuracy and stable ARS by encoding long temporal structure efficiently while keeping retrieval fast.

Balancing Prototype-Centroid Retrieval for Effective Query-Aware Reasoning. Figure 8 examines how many prototypes and centroids should be retrieved (Eq. 4, 5) during query-aware long-term reasoning. Since retrieval

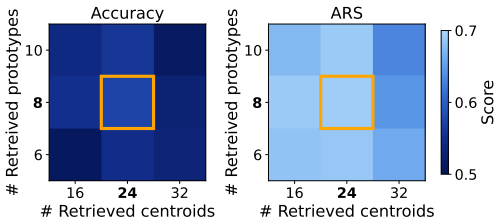

Figure 8. **Effect of number of query-aware retrieval slots** on ProReady-QA.

Table 7. **Token retrieval & reasoning strategies under identical budget.**

Method	Token	Acc.	Lat.
Full memory	108	56.4	8.7s
Single-stage	32	40.7	3.2s
Hierarchical	**32**	**56.6**	**3.1s**

proceeds in a coarse-to-fine manner, the choice of number of retrieval slots directly shapes both evidence quality and retrieval efficiency. Too few prototypes fail to cover the diverse high-level contexts present in long videos, reducing the chance that the correct event cluster is included in the search space explaining the weaker accuracy and ARS. Conversely, retrieving too many prototypes pulls in loosely related or noisy abstract clusters, weakening focus and increasing overhead. A similar trade-off appears for centroids: too few centroids miss relevant fine-grained states needed for tasks like step recognition or causal trigger detection, while too many centroids introduce excessive detail, slowing evidence localization and hurting ARS. The heatmaps show that the best results occur at the balanced setting of 8 prototypes and 24 centroids, which provides adequate semantic coverage through prototypes while maintaining enough centroid diversity to capture subtle evidence shifts without increasing retrieval latency. This additionally indicates that although the long-term memory is diverse, only a selective subset is needed per query, highlighting the effectiveness of our query-aware retrieval strategy.

While K-means is an effective way of memory construction, latency is governed by number of retrieved tokens and the reasoning strategy decides whether those tokens yield meaningful performance. In Table 7, all settings retrieve, reason over K-means' based memory, but StreamReady's hierarchical strategy preserves performance under small token budget and comparable latency.

Importance of Relevance-Gated Contextual Reasoning. While we perform contextual reasoning through a similarity-based gating and relevance-filtering mechanism over the contextual memory bank, we also evaluate a variant that attends to all prior QA pairs without any filtering. This naive design reduced accuracy by roughly $4-5\%$ on StreamingBench, showing that blindly incorporating the entire QA history introduces noise from unrelated turns. In long streaming videos, such unfiltered cross-attention forces the model to process irrelevant reasoning traces, weakening answer correctness and disrupting temporal alignment by pulling retrieval toward outdated or mismatched evidence. In contrast, our similarity-guided selection ensures that only semantically aligned prior QA pairs influence the current reasoning, allowing contextual memory to serve as a focused and beneficial signal rather than a source of distraction.

Table 8. **Effect of changing backbone of StreamReady.**

Method	ProReady-QA		StrmB	Vid-MME
	Acc.	ARS	Acc.	Acc.
Oryx-1.5-7B	53.9 (↑ 14.4)	0.64 (↑ 0.38)	63.6 (↑ 15.2)	64.3 (↑ 3.8)
LLaVA-OV 7B	55.2 (↑ 9.8)	0.65 (↑ 0.27)	60.3 (↑ 12.9)	63.8 (↑ 5.6)
Qwen-2-VL 7B	**56.4** (↑ 15)	**0.69** (↑ 0.35)	**63.4** (↑ 18.2)	**65.8** (↑ 2.5)

Table 9. **Effect of Readiness Supervision** using ProReady-QA. ‡ denotes original setup

Methods and Timing Supervision	Acc.	ARS
StreamBridge [39] w/ BCE ‡	53.1	0.60
StreamBridge [39] w/ Contrastive	53.2	0.63
StreamReady w/o timing sup.	56.3	0.58
StreamReady w/ BCE	56.4	0.65
StreamReady w/ Contrastive ‡	**56.4**	**0.69**

Robustness to Different Backbones. Table 8 shows that StreamReady remains highly robust to the choice of 7B-scale MLLM backbone, consistently achieving large improvements on streaming benchmarks such as ProReady-QA and StreamingBench. This consistency indicates that the performance gains stem from our framework's design rather than from any specific backbone architecture. By contrast, the improvements on VideoMME are noticeably smaller across all backbones. This difference highlights a key distinction: existing large MLLMs are already competitive on offline, single-query video reasoning, but they lack the temporal grounding, evidence tracking, and real-time retrieval capabilities required for true streaming understanding. They also lack mechanisms for answer timing based on evidence, which our framework introduces, reflected in the substantial ARS gains seen on ProReady-QA.

Impact of Timing Supervision and Ensuring Fair Comparison. Table 9 shows how different timing signal supervision strategies affect readiness modeling. For StreamReady, moving from no timing signal to BCE-based supervision yields only a modest improvement in ARS, since BCE provides a coarse binary view of readiness and does not meaningfully shape temporal behavior. Our contrastive super-

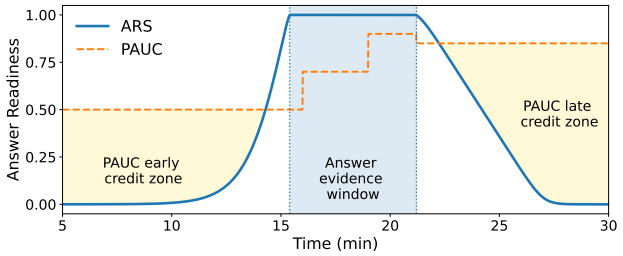

Figure 9. **Temporal characteristics of ARS** in relation to PAUC [43].

vision produces the largest ARS gain by learning an evidence driven readiness window, and trains the <RDY> pathway to recognize when evidence becomes sufficient. Importantly, accuracy remains effectively unchanged across the three variants, indicating that timing supervision influences readiness rather than correctness.

To ensure fairness when comparing with non–timing-aware methods, all offline baselines are triggered using the same confidence threshold as our readiness mechanism. Among existing online approaches, only StreamBridge [39] includes an explicit activation module; however, its design decouples readiness from reasoning by using a separate auxiliary network that does not access the same evidence or memory as the reasoning module. As a result, its readiness predictions are not grounded in the retrieved visual context. Even when trained with our contrastive timing objective (Eq. 8), StreamBridge shows limited ARS improvement and still underperforms our method while incurring higher compute and latency overhead. This shows that StreamReady's gains stem primarily from its unified evidence-aware reasoning and readiness mechanism, rather than its the timing supervision.

Importantly, the pseudo-labels used for training the readiness signal to provide timing supervision are derived from similarity between the *reasoning module's answer representation and visual memory*, rather than direct query–memory matching, thereby reflecting the outcome of reasoning. As a result, supervision reflects the outcome of reasoning rather than low-level visual cues. These labels supervise the readiness pathway to only determine *when* sufficient evidence has accumulated, while the reasoning pathway determines *what* constitutes valid evidence. Since training datasets lack ground-truth evidence windows, we validate pseudo-labels post hoc on ProReady-QA, observing strong alignment with annotated evidence (mean temporal IoU $= 0.87$ over 100 samples).

Robustness to Dense Tasks. Although evaluated in QA settings, the readiness mechanism is not tied to VQA outputs or task-specific supervision. It is trained with task-agnostic pseudo-labels of evidence sufficiency (§3.4), and remains effective even without it (Table 9). Results on ET-Bench (dense video captioning, step localization and captioning, temporal video grounding), demonstrate consistent gains on open-ended generation/dense tasks (Table 10), indicating the readiness mechanism's task-agnostic generalization.

Table 10. **Dense task performance (F1-score) comparison on ET-Bench.**

Method	DVC	SLC	TVG
Qwen2-VL	22.6	13.2	25.3
Dispider	33.8	18.8	36.1
StreamBridge	38.3	22.6	34.3
StreamReady	**43.4**	**24.1**	**36.8**

Effect of Readiness Threshold. Figure 11 shows ARS stability over a broad range of readiness threshold around the default value (0.35), degrading only at extremes due to premature or delayed triggering, indicating model robustness to threshold choice.

Probing Unanswerability. While a fully general readiness model should also identify when a question is unanswerable, the primary focus of this work is to study *when* a model should answer, given that an answer exists, which is consistent with existing streaming benchmarks and allows us to analyze response timing. That said, StreamReady couples readiness with reasoning over accumulated evidence: if a question is never answerable, the memory never contains sufficient query-aware evidence, and the model's readiness signal remains low throughout the stream. To validate this, we probe unanswerability with 30 counterfactual questions across random ProReady-QA videos, and find that StreamReady's readiness score remains low throughout (avg. 0.21), never triggering a response. This reflects its evidence-coupled readiness design that suppresses readiness in the absence of sufficient evidence. ARS also naturally handles such cases ($\tau, t_s, t_e = 0$ Eq. 10, 11), yielding near-zero effective accuracy, even if an answer were produced. Explicit modeling of unanswerability remains a promising future direction.

B.2. ARS Metric Behavior Analysis

Metric Rationale. ARS is designed to reflect how well a model's answers align with the evolving availability of visual evidence, so each component of the formulation is chosen to make timing behavior comparable, stable, and interpretable across diverse videos. The median evidence duration (τ) provides a consistent temporal scale that prevents overly harsh penalties in videos containing short evidence windows and overly lenient ones in long windows. The factor of 2 in EP (Eq. 10) ensures that answers given exactly at the evidence onset ($t_a = t_s$) receive full credit, avoiding the midpoint behavior of the sigmoid function that would

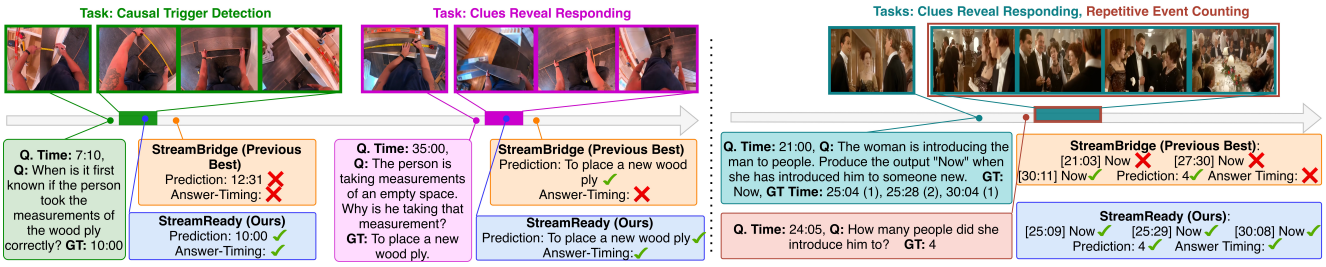

Figure 10. **Qualitative analysis** of readiness-aware streaming understanding on ProReady-QA. StreamReady shows superior performance by consistently giving accurate and on-time answers.

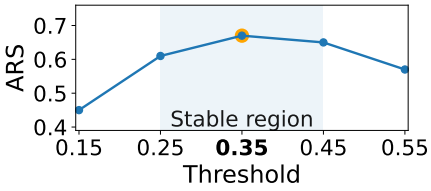

Figure 11. **Effect of readiness threshold (ProReady-QA).**

otherwise mark an on-time answer as partially early with 0.5 penalty. Softmin and softmax operators enforce smooth transitions at window boundaries, preventing abrupt scoring jumps that could unfairly reward or penalize models near start or end of window.

ARS also resolves several practical edge cases. When an answer is given far before the evidence window, EP saturates naturally toward zero, indicating that no part of the response is supported by the visual stream. When $t_s < t_a < t_e$, both EP and LP remain at 1, reflecting that the answer is fully grounded. After delays beyond t_e, LP decays gradually rather than collapsing, which avoids brittle behavior in tasks where evidence dissolves slowly. The formulation also behaves sensibly under atypical annotation cases: if noisy annotations ever produce $t_s > t_e$, the monotonicity of the penalties ensures that ARS does not produce contradictory or negative values. Altogether, the metric components work jointly to provide a consistent and interpretable estimate of answer readiness that aligns with how visual evidence unfolds in streaming video scenarios.

Temporal Characteristics of ARS. Figure 9 contrasts ARS with the timing formulation used in PAUC [43] to highlight how different scoring schemes treat answer timing in streaming settings. PAUC assigns temporal credit using a fixed baseline before the evidence window and then propagates the model's last correctness score after the window ends. This makes it less responsive to early or late answers and less reflective of gradual changes in when evidence becomes supportive. ARS instead models readiness as a continuous, asymmetric curve that rises only when visual evidence becomes valid and declines as that evidence fades. This produces a smoother alignment between the model's response timing and the underlying visual support. The figure highlights these distinctions in temporal behavior and helps clarify why ARS provides a natural fit for evaluating evidence-based readiness in long streaming videos.

Extending to Datasets without Annotated Evidence Window. Although ARS is defined using annotated evidence windows, its underlying requirement is evidence sufficiency, rather than precise temporal boundaries. While explicit windows offer the most reliable signal, sufficiency can also be approximated using native temporal annotations (e.g. action/scene boundary, timestamp) or model-driven cues like answer stabilization or cross-model agreement, enabling readiness-aware evaluation even in benchmarks without annotated evidence windows.

B.3. Qualitative Analysis

In Figure 10, we present a qualitative comparison of StreamReady and StreamBridge on the three tasks (CTD, CRR, REC) of the ProReady-QA benchmark. Across all cases, StreamReady delivers accurate and timely answers because its readiness signal is tied directly to the same evidence used for reasoning through the hierarchical visual memory. In contrast, StreamBridge often misfires because its activation module is decoupled from reasoning, which leads to mistimed outputs, ARS penalties, and answers that drift from the actual visual evidence.

In the CTD example on the *left*, StreamBridge activates only after the causal cue has passed and therefore reasons over the wrong visual moment, producing an incorrect answer and a late penalty. StreamReady detects the cue when it first appears and responds correctly within the evidence window. In the CRR example at the *center*, StreamBridge performs accurate prediction but triggers after the evidence window ends, harming its ARS and effective accuracy. StreamReady tracks the gradual buildup of clues and answers as soon as the evidence becomes sufficient.

The combined REC+CRR example on the *right* shows the most significant failure for StreamBridge. It answers the CRR question prematurely at question time, hallucinating a trigger that does not yet exist and incorrectly count-

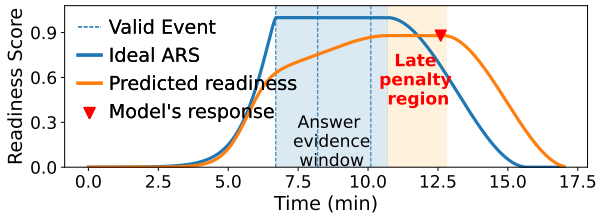

Figure 12. **Readiness failure in counting task.**

ing two individuals during that time. Later, it misses a real event within the evidence window, but still outputs the correct count, producing an answer that is not grounded in the visual sequence; thus incurring ARS penalty and reduced effective accuracy. StreamReady avoids this issue by recognizing that the CRR question is not answerable at question time, waiting for the true clue, and counting only events supported by evidence.

On the other hand, Figure 12 shows a failure case in counting, where the readiness signal remains high beyond evidence window, as the model waits to confirm that no additional relevant event occurs. This reflects the inherent ambiguity of such tasks, leading to delayed responses and a late penalty under ARS.

Overall, these examples illustrate that StreamReady's combined retrieval, reasoning, and readiness design enables precise evidence tracking and on-time answering in timing-sensitive streaming tasks.

C. ProReady-QA Generation Pipeline.

ProReady-QA is constructed from long-form Ego4D and MovieNet videos using a semi-automatic pipeline following [26, 56] to generate proactive QA pairs and answer evidence window annotations. Although the dataset contains fewer source videos than some earlier benchmarks, each video is significantly longer and paired with richer temporal annotations, enabling readiness-aware evaluation under ARS.

Dense Captioning. Each video is divided into 30-second segments, from which 8 uniformly sampled frames are processed by Qwen-2-VL to produce dense captions describing actions, objects, interactions, and spatial context, along with segment timestamps.

Summarization. We then aggregate dense captions over several-minute intervals and summarize them using an Qwen-2, preserving key entities, actions, causal structure, and temporal ordering while avoiding redundancy.

Multi-Turn Proactive QA Generation. To construct future-dependent QA, we take consecutive scene summaries and prompt the VLM to generate a question that a viewer might ask at the end of Scene A and whose answer only becomes knowable in Scene B. This enforces strict temporal ordering and ensures that every question is fundamentally future-dependent. From this pool of valid proactive QA, we then construct multi-turn dialogues for a subset of segments. The VLM is prompted to propose follow-up questions that explicitly reference earlier turns; either the immediately preceding question or one from farther back, while still requiring new future evidence to answer. These multi-turn chains may span different task types.

Answer Evidence Timestamp Annotation. We extract coarse candidate evidence spans using multimodal cues from native annotations (actions, interactions, scene changes, or subtitle segments) to anchor where the answer is likely revealed. Within this restricted window, a frame-wise VLM check identifies the earliest frame where the answer becomes inferable and the latest frame where the supporting cue remains valid. We refine these boundaries using dwell-based smoothing and task-specific rules (e.g., step transitions for SSR, clue sufficiency for CRR, count completion for REC, goal realization for GSD, causal effect visibility for CTD). This process yields evidence windows appropriate for timing-aware evaluation through ARS.

Human Refinement. After the automatic pipeline, annotators review each QA pair and its proposed evidence window. They verify that the answer is not inferable before t_s, that the supporting cue truly disappears at t_e, and that the QA instance is strictly future-dependent. Ambiguous or trivial cases, hallucinated cues, or QA pairs whose evidence does not align with the timestamps are corrected or removed, with replacement added when needed; especially for complex tasks such as CTD and CRR.

Quality Control. We assess annotation reliability on 100 ProReady-QA samples, where two annotators achieve high agreement on evidence windows (mean temporal IoU = 0.85), with model-based answer verifiability rising sharply within these windows (0.35 to 0.82), confirming annotation reliability.

D. Implementation Details

Table 11 shows the training data used for fine-tuning the reasoning module (§3.3) and the readiness mechanism (§3.4), while the visual encoder and language decoder remain frozen. The trainable parameters are trained for 5 epochs with a learning rate of $2e-5$ with a cosine annealing scheduler and AdamW optimizer ($[0.9, 0.999]$). We use $\alpha = 0.985$ for EMA decay factor in Eq. 1, 2. $\lambda_{reg} = 0.1$ in Eq. 8. All videos are sampled at 1 FPS following standard protocol of streaming video understanding and input video frames are resized to 224×224.

E. Future Work

While StreamReady and ProReady-QA advance readiness-aware streaming video understanding, several promising directions remain open for future exploration. Extending the

Table 11. **Dataset Statistics** for training the reasoning module and readiness mechanism

Task	Dataset	#QA Pairs
VQA	MSRVTT-QA [48]	10k
	MSVD-QA [48]	10k
	ActivityNet-QA [53]	32k
	MovieChat-1k [37]	10k

framework beyond a single-view setting to multi-camera or multi-agent environments would allow reasoning over parallel visual threads, where evidence may appear asynchronously across different viewpoints. Enabling multimodal streaming input (e.g. audio) could further enhance readiness estimation in scenes where critical cues are partially non-visual. Another natural extension is to explore readiness-aware behavior in interactive or embodied agent settings, where a model can decide not only when to answer but also when to request more information, ask clarifying questions, or seek an alternative viewpoint. Such capabilities are increasingly important for agentic frameworks, and incorporating readiness into these systems may help them maintain evidence-grounded decision-making in long streaming contexts.