Think-as-You-See: Streaming Chain-of-Thought Reasoning for Large Vision-Language Models

Supplementary Material

A. Details of Streaming CoT Pipeline

A.1. CLIP-Guided Frame ID Alignment

Step 1: Semantic anchoring before resampling. Given a video $\mathcal{V} = \{F_t\}_{t=1}^{T}$ with timestamps $\{\tau_t\}_{t=1}^{T}$ and annotated keyframe captions $\mathcal{C} = \{c_k\}_{k=1}^{K}$, we first compute CLIP embeddings for all frames and captions:

$$\boldsymbol{f}_t = \mathrm{Enc}_{\mathrm{CLIP}}^{\mathrm{img}}(F_t), \qquad \boldsymbol{g}_k = \mathrm{Enc}_{\mathrm{CLIP}}^{\mathrm{text}}(c_k).$$

We utilize cosine similarity throughout the alignment process:

$$\mathrm{sim}(\boldsymbol{a}, \boldsymbol{b}) \;=\; \frac{\boldsymbol{a}^\top \boldsymbol{b}}{\|\boldsymbol{a}\|\,\|\boldsymbol{b}\|}.$$

For each keyframe caption c_k, we identify its most similar frame index:

$$t_k^\star \;=\; \arg\max_{t \in \{1,\dots,T\}} \mathrm{sim}(\boldsymbol{f}_t, \boldsymbol{g}_k),$$

recording the anchor timestamp $\widehat{\tau}_k = \tau_{t_k^\star}$. These anchors serve as semantic locks preserved during subsequent resampling.

Step 2: Timestamp-based resampling at 2 FPS with anchor preservation. Let the target sampling interval be $\Delta = 0.5\,\mathrm{s}$ (2 FPS) and the target grid be $\{\tau'_{t'}\}_{t'=1}^{T'}$ with $\tau'_{t'} = (t'-1)\Delta$. For each target timestamp $\tau'_{t'}$, we select the frame $F_{t'}$ as:

$$F_{t'} \;=\; \begin{cases} F_{t_k^\star}, & \text{if } \tau'_{t'} \in [\widehat{\tau}_k - \epsilon,\ \widehat{\tau}_k + \epsilon] \text{ for some } k, \\ \arg\min\limits_{F_t} |\tau_t - \tau'_{t'}|, & \text{otherwise,} \end{cases}$$

where $\epsilon = 0.1\,\mathrm{s}$ is a tolerance window ensuring every semantic anchor $\widehat{\tau}_k$ snaps to the nearest sampling point. Post-selection, frame indices are renormalized, and clips are truncated to the maximum input duration (30 s).

A.2. Quality Assurance and Temporal Filtering

To ensure generated frame-level trajectories are temporally grounded and semantically reliable, we apply a three-stage filtering process (Algorithm 1). First, we identify question-relevant keyframes via embedding similarity. Second, we prune temporally adjacent captions with redundant semantics to preserve distinct perceptual events. Finally, we format the supervision sequence by assigning `</EOT>` to selected keyframes and `<SKIP>` to others. This yields a temporally sparse but well-aligned target stream, guiding the model to reason only at meaningful moments.

Algorithm 1 Quality Assurance and Temporal Filtering

Require: Question Q_t, keyframe captions $\{c_k\}$
Require: Thresholds $\tau_q = 0.7$, $\tau_{\mathrm{adj}} = 0.9$
Step 1: Question–caption relevance screening
for each caption c_k **do**
  $s_k \leftarrow \mathrm{sim}(e(Q_t), e(c_k))$
end for
$\mathcal{K}_t \leftarrow \{k \mid s_k \geq \tau_q\}$
Step 2: Anti-redundancy temporal de-duplication
Sort $\mathcal{K}_t$ by time
$\mathcal{K}_t^\star \leftarrow [\,]$
for each k in $\mathcal{K}_t$ **do**
  if $\mathcal{K}_t^\star$ is empty **then**
    Append k to $\mathcal{K}_t^\star$
  else
    Let j be last element in $\mathcal{K}_t^\star$
    $s_{j,k} \leftarrow \mathrm{sim}(e(c_j), e(c_k))$
    if $s_{j,k} < \tau_{\mathrm{adj}}$ **then**
      Append k to $\mathcal{K}_t^\star$
    end if
  end if
end for
Step 3: Formatting supervision targets
for each sampled frame index t' **do**
  if $t' \in \mathcal{K}_t^\star$ **then**
    Emit $[R_{t'}]$ $</EOT>$
  else
    Emit `<SKIP>`
  end if
end for

A.3. Practical Notes

- **Embedding normalization.** All embeddings are ℓ_2-normalized prior to similarity computation to stabilize thresholds.
- **Batching.** Frame and caption embeddings are computed in batches to mitigate I/O latency for long videos.
- **Hyperparameters.** Default values are $\Delta = 0.5\,\mathrm{s}$, $\epsilon = 0.1\,\mathrm{s}$, $\tau_q = 0.7$, and $\tau_{\mathrm{adj}} = 0.9$, balancing temporal precision with retention of key semantic content.

A.4. Details of Dataset

The dataset spans 12 video reasoning tasks covering fine-grained event interpretation and high-level semantic understanding. As shown in Figure 8 and Table 3, the task distribution is long-tailed: *Causal Analysis* and *Event Dynamic*

Analysis dominate, while *Ingredient Analysis* and *Behavior Analysis* are less frequent. This reflects the natural prevalence of reasoning behaviors in real-world video content while ensuring broad coverage for multi-step reasoning evaluation.

Temporal structure also varies significantly. Figure 9 illustrates the distribution of keyframe counts, revealing a wide spectrum of temporal sparsity. Some videos contain sparse salient moments, while others feature dense, extended event sequences. This variability is critical for evaluating streaming reasoning, requiring models to adapt to varying event frequencies and accurately identify meaningful visual changes.

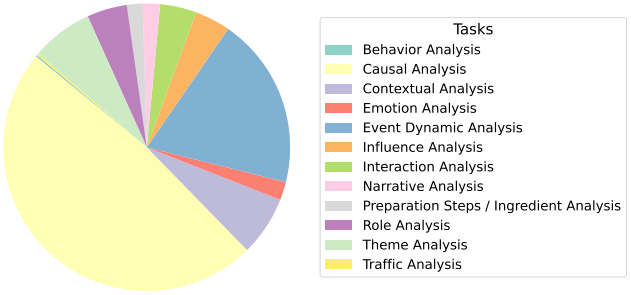

Figure 8. Task distribution in the dataset.

Table 3. Distribution of task categories in training and test sets.

Task	Train Set	Test Set
Causal Analysis	52,566	208
Event Dynamic Analysis	18,675	82
Preparation Steps / Ingredient Analysis	2,252	74
Theme Analysis	6,206	33
Interaction Analysis	4,208	38
Influence Analysis	4,406	45
Role Analysis	4,843	31
Emotion Analysis	1,999	39
Narrative Analysis	1,755	35
Contextual Analysis	6,827	38
Behavior Analysis	227	12
Traffic Analysis	218	14

B. Prompt Details

We present the complete prompts used in our pipeline, including QA construction (Figure 10), CoT inference (Figure 11), and subjective evaluation (Figure 12).

C. Training Details

We train TaYS using a streaming-aware decoder-only objective, where visual and reasoning tokens are interleaved with causal masking. Optimization employs AdamW with cosine decay, mixed-precision (bfloat16), gradient accumulation, activation checkpointing, and DeepSpeed ZeRO-3 for memory efficiency. The vision encoder remains frozen, while the multimodal projector and LLM backbone are fine-tuned. We regulate video token length via pixel-based constraints and train for two epochs with an effective sequence length of 8192 tokens.

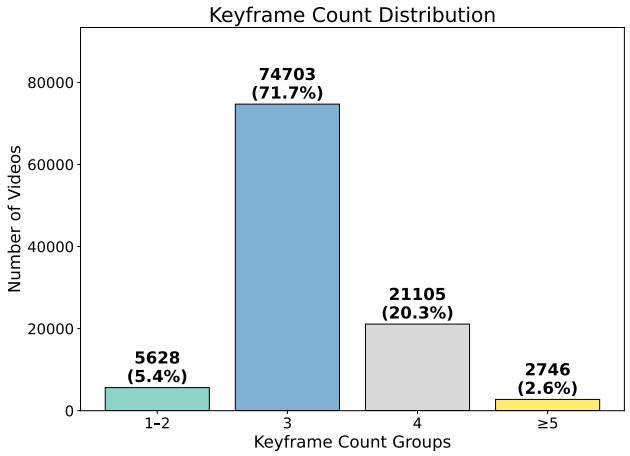

Figure 9. Distribution of keyframe counts per sample.

Prompt: QA Construction

You are a video description normalizer. Rewrite the evidence based on the given question and answer.
Question: {question}
Answer: {answer}
Rules:
1. Each frame must have exactly one sentence.
2. Each sentence starts with "In frame X," and ends with a period.
3. Keep all original frame numbers, in the same order, without adding or removing any.
4. Do not merge multiple frames into one sentence or split a frame into several.
5. Write each frame on a new line.
6. Adjust expressions so the narrative aligns naturally with the QA semantics while keeping the visual meaning unchanged.
Original Evidence:
{evidence}
Rewritten Evidence:

Figure 10. Prompt template for QA construction.

Prompt: CoT Inference

Please imagine this question as a person pondering deeply. The content of frames related to references and issues is used to form a thinking process in chronological order of the events.
Encourage self reflection or verification during the reasoning process.
Please provide a detailed reasoning between the <think> and </think> tags, and then provide your final answer between the <answer> and </answer> tags.

Figure 11. Prompt template for CoT inference.

Prompt: Subjective Evaluation

You are an expert evaluation assistant. Your task is to compare three model outputs based on the given Question and the Ground-Truth Answer.
First, carefully read the Question and the Ground-Truth Answer.
Then evaluate each model output (A, B, and C) across the following four aspects:
Evaluation Criteria (1–10 per dimension)
Logic: Evaluate whether the reasoning is coherent, structured, and follows logically from the question.
(1–2: illogical; 3–4: inconsistent; 5–6: partially logical; 7–8: mostly logical; 9–10: fully logical)
Factuality: Check correctness and absence of factual errors relative to the ground-truth answer.
(1–2: mostly incorrect; 3–4: major errors; 5–6: minor errors; 7–8: highly factual; 9–10: fully factual)
Accuracy: Assess how precisely the model answers the question and matches the ground-truth answer.
(1–2: irrelevant; 3–4: weak alignment; 5–6: partial correctness; 7–8: mostly accurate; 9–10: perfectly accurate)
Conciseness: Evaluate clarity and brevity without unnecessary verbosity.
(1–2: very verbose or incomplete; 3–4: unfocused; 5–6: somewhat concise; 7–8: concise; 9–10: extremely concise and clear)
You should internally consider all four scores for each model to judge overall quality, but DO NOT output the scores.
Instead, provide a final decision on which model output is best overall.
Final Output Instruction
Choose only ONE of the following responses: Best: A; Best: B; Best: C; Best: Tie;

Figure 12. Prompt template for subjective evaluation.

Table 4. Training hyperparameters for TaYS.

Config	Value
input resolution	variable (pixel-constrained)
max token length	8192
vision encoder	frozen
trainable modules	LLM + MLP projector
precision	bfloat16
optimizer	AdamW
learning rate	2×10^{-5}
lr schedule	cosine decay
warmup ratio	0.03
batch size	1 (grad accum = 16)
epochs	2
gradient clipping	1.0
gradient checkpointing	enabled
distributed training	torchrun + ZeRO-3
max video frames	60
video token budget	24K tokens (pixel-based)

D. Evaluation Details

Construction of Test Set. Following the VideoEspresso protocol, we construct the test set with three distractor options per question. Distractors are designed to match the correct answer in contextual relevance and linguistic form while containing explicit factual inaccuracies, ensuring a discriminative evaluation. We apply the same answer-rewriting procedure as in training to maintain consistency.

Objective Evaluation Protocol. For each sample, we evaluate a free-form prediction $\tilde{y}$ against a reference answer $y^\star$ and multiple-choice options $\mathcal{O} = \{o_1, o_2, o_3, o_4\}$, where $o^\star$ is the correct option. We use a semantic similarity function $\mathrm{sim}(\cdot, \cdot)$ with a threshold $\tau = 0.8$.

Stage 1: Reference similarity. We first compute $s_{\mathrm{ref}} = \mathrm{sim}(\tilde{y}, y^\star)$. If $s_{\mathrm{ref}} < \tau$, the prediction is deemed incorrect.

Stage 2: Option discrimination. We compute similarities $s_j = \mathrm{sim}(\tilde{y}, o_j)$ for all options. Let $s_{\mathrm{opt}} = \mathrm{sim}(\tilde{y}, o^\star)$ and $s_{\max}^{\mathrm{neg}} = \max_{o_j \neq o^\star} s_j$. A prediction is correct only if:

$$s_{\mathrm{ref}} \geq \tau, \quad s_{\mathrm{opt}} \geq \tau, \quad \text{and} \quad s_{\mathrm{opt}} > s_{\max}^{\mathrm{neg}}.$$

Algorithm 2 Two-Stage Objective Evaluation

Require: Prediction $\tilde{y}$, reference answer $y^\star$, options $\mathcal{O} = \{o_1, o_2, o_3, o_4\}$, correct option $o^\star \in \mathcal{O}$, similarity function sim, threshold τ

$s_{\mathrm{ref}} \leftarrow \mathrm{sim}(\tilde{y}, y^\star)$
if $s_{\mathrm{ref}} < \tau$ **then**
  return INCORRECT
end if
for each $o_j \in \mathcal{O}$ **do**
  $s_j \leftarrow \mathrm{sim}(\tilde{y}, o_j)$
end for
$s_{\mathrm{opt}} \leftarrow \mathrm{sim}(\tilde{y}, o^\star)$
$s_{\max}^{\mathrm{neg}} \leftarrow \max\{s_j : o_j \in \mathcal{O},\, o_j \neq o^\star\}$
if $s_{\mathrm{opt}} \geq \tau$ **and** $s_{\mathrm{opt}} > s_{\max}^{\mathrm{neg}}$ **then**
  return CORRECT
else
  return INCORRECT
end if

Latency Evaluation Protocol. We quantify real-time performance using two metrics: **(1) Time to First Token (TTFT)**, measuring the interval between the arrival of the first frame and the emission of the first token; **(2) Overall Delay**, measuring the total time to complete reasoning and produce the final answer. All inferences run on identical hardware with token-level timing resolution to ensure fair comparison.