DAL-CoT: Decision-aligned Latent Reasoning with MLLMs for Video Scene Segmentation

Anonymous authors
Paper under double-blind review

Abstract

Multimodal Large Language Models (MLLMs) have been applied to video scene segmentation, which requires a model to make multiple boundary decisions over a series of continuous video shots. This calls for the ability to aggregate long-range cues from noisy multimodal inputs, where the model must distinguish genuine boundary signals from numerous conflicting distractors, which poses a unique challenge to Chain-of-thought (CoT). Conventional explicit CoT translates continuous and uncertain visual evidence into an autoregressive sequence of definitive statements, where an incorrect commitment to one boundary decision may directly lead to an overconfident yet incorrect answer. To address this, we propose **DAL-CoT**, a decision-aligned latent CoT method that encodes explicit CoT reasoning into the model's hidden states. DAL-CoT assigns each candidate shot an observation latent slot for fine-grained reasoning, preventing premature commitment to boundary decisions. During training, an auxiliary decoder reconstructs the explicit reasoning texts from each latent slot, while at inference it is removed. Extensive experiments demonstrate that DAL-CoT significantly outperforms state-of-the-art approaches while substantially reducing the decoding cost of intermediate reasoning compared with explicit CoT.

1 Introduction

Recently, Multimodal Large Language Models(MLLMs) have demonstrated strong capabilities in the task of video scene segmentation(Berman et al., 2026). Video scene segmentation is the task of identifying coherent narrative boundaries within long-form video content(Rao et al., 2020; Chen et al., 2021). A scene consists of several shots connected by an event, location, or narrative purpose, and a shot is a continuous camera take composed of multiple frames.

However, current MLLM-based methods only provide post-hoc explanations for their predictions, without leveraging the model's reasoning ability to improve segmentation performance. While Chain-of-Thought (CoT) has demonstrated consistent benefits across several tasks(Wei et al., 2022; Fei et al., 2024), its application to video scene segmentation leads to a dramatic metric collapse. A key reason is the **premature boundary commitment** of CoT. Specifically, video scene segmentation requires aggregating long-range cues from noisy multimodal inputs, where the model must distinguish genuine boundary signals from numerous conflicting distractors. However, CoT translates continuous and uncertain visual evidence into an autoregressive sequence of definitive statements, where an incorrect commitment to one boundary decision may directly lead to an overconfident yet incorrect answer, ultimately degrading model performance, as shown in Fig. 1(a).

Recently, Latent CoT (Hao et al., 2025; Shen et al., 2025; Wei et al., 2026) has demonstrated that hidden states are capable of encoding reasoning processes implicitly, making it a promising avenue to address the aforementioned premature commitment problem. Coconut (Hao et al., 2025) feeds the last hidden state of Large Language Models (LLMs) back as the next input embedding, achieving a better accuracy-efficiency trade-off. SIM-CoT (Wei et al., 2026) introduces step-level supervision to implicit reasoning token, fixing the latent instability and semantic homogenization. OneVL (Lu et al., 2026) supervises compact latent tokens with auxiliary decoders, enabling the latent tokens to internalize causal reasoning while matching explicit CoT performance at latency nearly matching direct answer generation. Inspired by OneVL, we consider utilizing an auxiliary decoder to encode explicit CoT into a short sequence of compact latent tokens. However, OneVL stacks several latent

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.

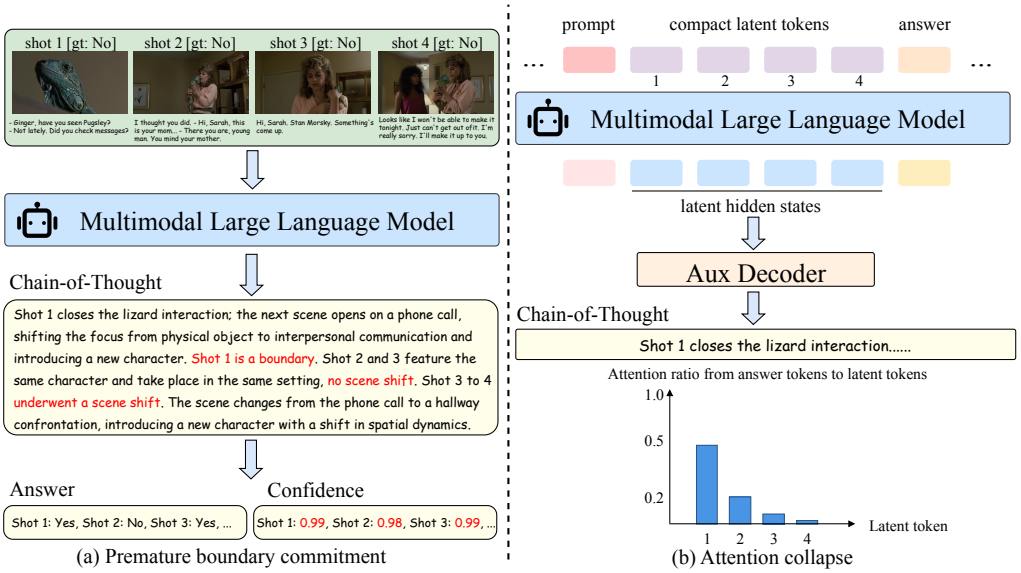


Figure 1: Two failure modes of Chain-of-Thought in video scene segmentation. (a) **Premature boundary commitment**. A new character and the camera position shift do not disrupt the continuity from shot 3 to 4. The model mistakenly treats the distracting cues as decisive cues, leading to an incorrect decision of shot 3 with high confidence. (b)**Attention collapse**. When latent hidden states are fed together into the auxiliary decoder, the model struggles to assign distinct information to each hidden state, ultimately concentrating the vast majority of its attention on the first few latent tokens when generating the final answer.

tokens to jointly encode the reasoning process, which causes most attention to concentrate on the first few tokens while leaving the remaining ones largely unattended, resulting in **attention collapse**, as illustrated in Fig. 1(b).

To address the problem, we propose **DAL-CoT**, a decision-aligned latent Chain-of-Thought method for video scene segmentation. For each boundary decision at shot i, DAL-CoT allocates a dedicated latent token that encodes the transition from shot i to shot $i+1$. Each latent token is aimed to reconstruct only the observation target over the corresponding shot pair, instead of the whole context of all shots. This decision-aligned supervision forces every latent token to attend to its assigned observation region rather than collapsing onto the first few tokens, enabling precise per-decision reasoning. During training, the auxiliary decoder is applied to supervise these targets, while at inference, it is removed. In summary, our contributions are as follows:

Our contributions are:

- We identify two failure modes of Chain-of-Thought in video scene segmentation: **premature boundary commitment** of explicit CoT, where definitive boundary judgments lead to overconfident decisions; and **attention collapse** of latent CoT, where attention is allocated to the first few latent tokens, leaving the remaining ones unattended.

- We propose **DAL-CoT**, a decision-aligned latent Chain-of-Thought method that allocates one dedicated latent token for each shot-pair transition observation, preventing premature boundary commitment while guaranteeing adequate attention allocation to every latent token.

- Extensive experiments demonstrate that DAL-CoT achieves notable accuracy improvements over existing video scene segmentation methods, while running at latency nearly matching direct answer generation during inference.

2 RELATED WORK

2.1 VIDEO SCENE SEGMENTATION

Video scene segmentation is the task of partitioning a video into semantically coherent segments, each representing a continuous narrative unit. Encoder-based methods have developed multimodal local-to-global features (Rao et al., 2020), contrastive shot and scene representations (Chen et al., 2021; Wu et al., 2022), boundary-sensitive pretraining (Mun et al., 2022), long-context temporal models (Islam et al., 2023; Tan et al., 2024), and multimodal or character-aware aggregation (Sadoughi et al., 2023; Tan et al., 2025).

Despite these advances,the above methods struggle to leverage rich textual context (e.g., subtitles, scripts) as primary reasoning input, and their decisions are not accompanied by interpretable rationales. This has motivated a shift toward generative formulations, where scene segmentation is cast as conditional sequence generation conditioned on multimodal inputs. Scene-VLM (Berman et al., 2026) casts scene segmentation as conditional generation from frames, subtitles, and character information. It emits shot-indexed Yes/No decisions and derives confidence from token logits. However, its optional rationales are post-hoc rather than decision-forming, potentially justifying incorrect outputs after the fact. Therefore, we apply CoT to MLLM-based video scene segmentation, allowing reasoning to guide rather than follow the prediction.

2.2 VIDEO CHAIN-OF-THOUGHT

Explicit video CoT organizes visual evidence into generated language. Video-of-Thought (Fei et al., 2024) moves from grounded perception to a high-level answer through structured spatio-temporal reasoning. VIP (Himakunthala et al., 2023) and VideoCoT (Wang et al., 2024) provide frame-oriented and annotated reasoning traces. Agent-of-Thoughts Distillation (Shi et al., 2025) constructs tool-supported traces, and VideoEspresso (Han et al., 2025) emphasizes core-frame supervision for fine-grained reasoning.

However, verbalizing every reasoning step as natural language tokens incutively increases sequence length and inference cost. In contrast, latent CoT replaces some decoded reasoning tokens with continuous computation. Coconut (Hao et al., 2025) feeds hidden states forward as latent thoughts. CODI (Shen et al., 2025) compresses explicit CoT through self-distillation. LaCoT (Sun et al., 2025) treats a visual rationale as a latent variable. SIM-CoT (Wei et al., 2026) uses explicit traces to supervise implicit steps through a training-time decoder. However, common fixed-budget of latent CoT does not create a one-to-one correspondence with a variable set of scene-boundary decisions. Therefore, we consider a decision-aligned latent CoT paradigm for fine-grained reasoning.

3 METHOD

The pipeline of DAL-CoT is shown in Fig. 2. The model receives multimodal inputs comprising frames, subtitles, and characters, and make observations in an implicit way. Each latent hidden state attends to the plot-relevant cues that appear in its corresponding shot pair. And the hidden states can be decoded by an auxiliary decoder into corresponding explicit CoT. This latent reasoning process prevents the premature commitment to boundary decisions caused by explicit CoT, and the auxiliary decoder is removed during inference, resulting in faster inference than explicit CoT.

3.1 PROBLEM FORMULATION

Let a video contain shots $\{s_t\}_{t=1}^{T}$. A sample $C = (s_1, \ldots, s_N)$ contains N continous shots, and a focus sequence $F = (s_{start}, \ldots, s_{end}) \subseteq C$ contains K shots. For each focus shot s_i ($i \in [start, end]$), the model is requested to output `"yes"` or `"no"`, indicating whether s_i is the last shot of its scene in the video. The model receives multimodal inputs for each shot, including visual frames, subtitles, and cast person IDs (PIDs). Conditioned on the multimodal context and user instruction, the model is required to make a boundary decision d_i for each shot s_i to form the answer $\mathbf{a}$:

$$\mathbf{a} = (\texttt{Shot } s_1\texttt{:} \quad d_1, \ldots, \texttt{Shot } s_K\texttt{:} \quad d_K), \qquad d_i \in \{\texttt{Yes}, \texttt{No}\}. \tag{1}$$

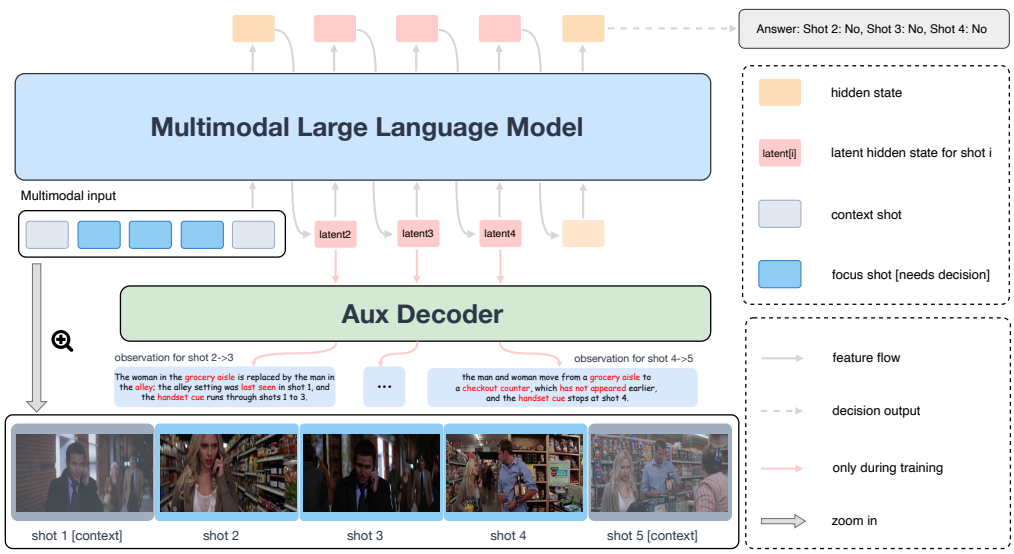


Figure 2: The proposed DAL-CoT pipeline. The model receives multimodal inputs comprising frames, subtitles, and characters, where the focus shots require decisions, and context shots only provide contextual information. The model makes implicit observations in the latent state whether each focus shot is a boundary, and such observations can be decoded by an auxiliary decoder into corresponding explicit natural language. This latent reasoning process prevents the premature commitment to boundary decisions caused by explicit CoT, and the auxiliary decoder is removed during inference, resulting in faster inference than explicit CoT.

For precision–recall trade-off, we extract confidence scores by applying a softmax over the Yes and No token logits at each position:

$$p_i = \frac{\exp(\ell_{i,\texttt{Yes}})}{\exp(\ell_{i,\texttt{Yes}}) + \exp(\ell_{i,\texttt{No}})}. \tag{2}$$

where $\exp(\ell_{i,\texttt{Yes}})$ is the exponential of the logit for the Yes token at position i.

3.2 From explicit CoT to latent reasoning

An explicit pre-answer CoT $\mathbf{r}$ factorizes prediction as

$$P_\theta(\mathbf{r}, \mathbf{a} \mid \mathbf{P}, \mathbf{X}_C) = P_\theta(\mathbf{r} \mid \mathbf{P}, \mathbf{X}_C) P_\theta(\mathbf{a} \mid \mathbf{P}, \mathbf{X}_C, \mathbf{r}), \tag{3}$$

where $\mathbf{P}$ is the instruction, $\mathbf{X}_C$ is the representation of multimodal inputs, and $P_\theta(\cdot)$ denotes the conditional probability distribution induced by the model. The answer $\mathbf{a}$ depends on an autoregressively generated verbal commitment in explicit CoT. In contrast, a latent reasoning process avoids explicit commitment.

To realize latent reasoning, we first insert a short sequence of placeholder tokens between the prompt and the answer, denoted as latent tokens. During training, we extract the final-layer hidden states of these placeholder tokens and feed them into an auxiliary decoder, which is supervised to generate explicit CoT. While during inference, the auxiliary decoder is removed. Moreover, since the latent tokens are fixed, they can be prefilled once and appended to the prompt at inference time. The model then generates the answer directly, preventing the model from premature boundary commitment, while at the same time reducing inference latency.

However, when all hidden states are fed together into the auxiliary decoder for training, the model struggles to assign distinct information to each hidden state, causing it to allocate vast majority of attention to the first few latent tokens when generating the final answer. Therefore, DAL-CoT uses a decision-aligned latent sequence. For K requested decisions, it utilizes exactly K tokens and forces token i to observe the transition $i \to i+1$. Let $\mathbf{Z}(F) = (\mathbf{z}_1, \ldots, \mathbf{z}_K)$ be the latent tokens. The answer distribution is $P_\theta\left(\mathbf{a} \mid \mathbf{P}, \mathbf{X}_C, \mathbf{Z}(F)\right)$ with no explicit reasoning tokens between prompt and answer.

3.3 DECISION-ALIGNED LATENT TOKENS

For a focus shot s_i, the assigned latent tokens are:

$$\mathrm{slot}(i) \equiv \texttt{latent} \parallel \mathrm{str}(i). \tag{4}$$

where a slot denotes several latent tokens aligned with one decision step. For example, $i = 5$ gives `latent5`. Appending a numeric index to "latent" allows each slot to precisely map to its designated shot. We do not add these slots as special tokens, because each film has thousands of shots, making it impractical to register a dedicated special token for each shot number. The complete set of latent tokens is:

$$\mathbf{Z}(F) = \big[\texttt{<|start-latent|>}, \mathrm{slot}(1), \ldots, \mathrm{slot}(K), \texttt{<|end-latent|>}\big]. \tag{5}$$

The number of slots follows the focus range K, which is 10 in our setting. Let $\mathcal{I}_i$ be the full token span of slot i and let $\mathbf{H}^{(-1)}$ be the final backbone hidden sequence. The input fed to the auxiliary decoder is:

$$\mathbf{U}_i = \left[\mathbf{H}^{(0)}_{\mathrm{meta}}; \mathbf{H}^{(-1)}_{\mathcal{I}_i}\right]. \tag{6}$$

where $\mathbf{H}^{(0)}_{\mathrm{meta}}$ denotes the embedding of the multimodal metadata, including frames, subtitles and casts. The auxiliary decoder receives the shared multimodal features of all N shots, and, guided by $\mathrm{slot}(i)$, outputs the CoT for the transition from shot i to shot $i + 1$.

3.4 LOSSES AND THREE-STAGE OPTIMIZATION

For a decoding target with M_i tokens in the auxiliary decoder, the token-normalized reconstruction loss is

$$\mathcal{L}^{\mathrm{dec}}_i = -\frac{1}{M_i} \sum_{m=1}^{M_i} \log P_\psi\left(o^{(m)}_i \mid \mathbf{U}_i, \mathbf{o}^{(<m)}_i\right). \tag{7}$$

where $\mathbf{o}^{(<m)}_i$ is the teacher-forced target prefix before position m, and P_ψ is the next-token distribution produced by the auxiliary decoder with parameters ψ. Normalizing by M_i prevents longer observation targets from receiving greater weight. DAL-CoT averages the token-normalized reconstruction losses over the K decision-aligned CoT targets:

$$\mathcal{L}_{\mathrm{aux}} = \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}^{\mathrm{dec}}_i. \tag{8}$$

The answer loss is weighted next-token cross-entropy:

$$\mathcal{L}_{\mathrm{ans}} = -\frac{\sum_{j \in \mathcal{A}} w_j \log P_\theta\big(a^{(j)} \mid \mathbf{P}, \mathbf{X}_C, \mathbf{Z}(F), \mathbf{a}^{(<j)}\big)}{\sum_{j \in \mathcal{A}} w_j}, \tag{9}$$

where $\mathcal{A}$ contains supervised answer tokens.

Training follows a three-stage curriculum that separates task adaptation from latent-observation supervision. In Stage 0, the model backbone is optimized only with $\mathcal{L}_{\mathrm{ans}}$, without instantiating the auxiliary decoder. This stage first establishes the scene-boundary prediction behavior before introducing the reconstruction objective. In Stage 1, the Stage 0 backbone is frozen, and the newly initialized auxiliary decoder is optimized with $\mathcal{L}_{\mathrm{aux}}$. Training the decoder against a fixed representation space provides a stable initialization without allowing reconstruction gradients to disrupt the backbone. In Stage 2, the backbone and auxiliary decoder are jointly optimized:

$$\mathcal{L} = \mathcal{L}_{\mathrm{ans}} + \lambda_t \mathcal{L}_{\mathrm{aux}}. \tag{10}$$

where the auxiliary coefficient λ_t is gradually increased over the initial joint-training steps, preventing an abrupt auxiliary-gradient spike and enabling stable co-optimization of the answer and reconstruction losses. In this stage, reconstruction process regularizes the decision-aligned latent representations while preserving their predictive utility for boundary detection.

At inference, DAL-CoT constructs decision-aligned latent tokens and generates the answer with the backbone, and the auxiliary decoder is removed. Therefore, inference performs no autoregressive decoding; its method-specific overhead is the prefill cost of the multi-token latent block, which is negligible. When performing case studies or evaluating the model's decoding behavior, the decoding process can be selectively enabled.

Table 1: Comparison with state-of-the-art scene segmentation methods. Best results are in bold.

latent	**backbone**	**MovieNet F1 ↑**	**MovieNet AP ↑**	**BBC AP ↑**
LGSS (Rao et al., 2020)	—	—	47.1	—
ShotCoL (Chen et al., 2021)	—	—	53.4	—
BaSSL (Mun et al., 2022)	—	47.0	57.4	40.0
MEGA (Sadoughi et al., 2023)	—	55.3	58.6	—
TimeSformer (Bertasius et al., 2021)	—	—	—	42.2
TranS4mer (Islam et al., 2023)	—	48.4	60.8	43.6
NeighborNet (Tan et al., 2024)	—	—	64.0	—
Scene-VLM (Berman et al., 2026)	Qwen2.5-VL-7B	62.1	66.8	45.8
Scene-VLM (Berman et al., 2026)	Qwen3-VL-8B	65.3	71.6	51.6
DAL-CoT	Qwen2.5-VL-7B	**62.7**	**68.3**	**47.8**
DAL-CoT	Qwen3-VL-8B	**66.8**	**72.9**	**52.8**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and evaluation protocols. We follow previous works to train and evaluate on **MovieNet-318**, which is a subset of the MovieNet dataset (Huang et al., 2020) for scene segmentation in cinematic content. It has 318 movies, split into 190/64/64 movies for training, validation, and testing, respectively. Each movie contains over 1,000 shots on average with shot-level scene-boundary annotations. Each shot provides three frames, subtitles, and cast identifiers. We report F1 and Average Presicion as evaluation metrics. To evaluate the generalization capability of our method, we conduct zero-shot evaluation on the **BBC Planet Earth** (Baraldi et al., 2015) dataset, which is a 10-episode documentary benchmark with 670 scenes and 4.9K shots.

Compared methods. We compare with representative encoder-based approaches: LGSS (Rao et al., 2020), ShotCoL (Chen et al., 2021), BaSSL (Mun et al., 2022), MEGA (Sadoughi et al., 2023), TranS4mer (Islam et al., 2023), and NeighborNet (Tan et al., 2024). We also compare with the generative Scene-VLM (Berman et al., 2026).

Implementation details. DAL-CoT employs Qwen3-VL-8B-Instruct (Bai et al., 2025a) as the backbone and auxiliary decoder. The model is trained with DeepSpeed ZeRO-3. We optimize with AdamW. Stage 0 trains the backbone for 1 epoch with learning rate 1×10^{-5}; Stage 1 freezes the backbone and trains the auxiliary decoder for 1 epoch at 1×10^{-4}; Stage 2 jointly optimizes both modules for 2 epochs at 1×10^{-5}. The auxiliary-loss weight is linearly warmed from 0 to 1 during the initial Stage-2 steps. At test time, the auxiliary decoder is removed. Following the setting of the previous work (Berman et al., 2026), frames are resized to 147×63. A standard sample uses a 20-shot context window and predicts the central 10-shot focus window.

4.2 COMPARISON WITH STATE-OF-THE-ARTS

Video scene segmentation. Table 1 compares DAL-CoT with established scene-segmentation methods on MovieNet-318 and BBC Planet Earth. Since Scene-VLM is built on Qwen2.5-VL-7B (Bai et al., 2025b), to ensure a fair comparison we evaluate both Scene-VLM and DAL-CoT under matched backbones. When both methods use Qwen2.5-VL-7B, DAL-CoT outperforms Scene-VLM by 0.6 F1 and 1.5 AP on MovieNet-318. And when scaling both methods to Qwen3-VL-8B, DAL-CoT outperforms Scene-VLM by +1.5 F1 and +1.3 AP. These controlled experiments confirm that DAL-CoT's gains are not driven by backbone capacity but by its latent reasoning mechanism. Compared to the previous leading encoder-based method TranS4mer, our gain extends to +18.4 F1 and +12.1 AP, demonstrating that our latent chain-of-thought design brings consistent improvements over both VLM-based and encoder-based baselines.

For cross-domain generalization, we evaluate zero-shot on BBC Planet Earth. And following prior work, we report AP only. DAL-CoT with Qwen3-VL-8B reaches 52.8 AP, surpassing Scene-VLM with the same backbone by 1.2 AP and TranS4mer by 9.2 AP. Under the Qwen2.5-VL-7B backbone,

Table 2: Comparison of different reasoning paradigms on MovieNet-318 under the same backbone.

Method	F1 ↑	AP ↑
Answer-only	65.3	71.6
Post-hoc explanation	63.1	69.1
Observation-only CoT	65.9	72.1
Observation-then-commit CoT	59.0	50.9
DAL-CoT w/o decision-aligned	66.3	72.7
DAL-CoT	**66.8**	**72.9**

DAL-CoT also leads Scene-VLM by 2.0 AP, showing that our method generalizes robustly beyond cinematic training data.

4.3 Ablation Studies

Comparison of reasoning paradigms. To isolate the effect of decision-aligned latent reasoning, we compare our method with five interfaces under a controlled setup. All variants share the Qwen3-VL-8B backbone, multimodal inputs, training set, and two-epoch budget. **Answer-only** emits Yes/No decisions with no reasoning text, matching the Scene-VLM setting. **Post-hoc explanation** generate a rationale after each predicted boundary. **Observation-only CoT** writes a shot-pair observation for each focus shot before the answer, and does not state a boundary decision inside that observation. **Observation-then-commit CoT** follows observation-only CoT, only appending a boundary statement after each observation (e.g., "Shot i is / is not a scene boundary"). Both explicit variants use the observation text that supervises DAL-CoT's auxiliary decoder, placed in the backbone's autoregressive output before the answer. See Appendix for details. **DAL-CoT w/o decision-aligned** keeps latent supervision but gathers every latent token and feeds them jointly to the auxiliary decoder, so no slot is tied to one shot pair. **DAL-CoT** assigns one latent slot to each shot-pair observation.

As shown in Table 2, DAL-CoT reaches 66.8 F1 and 72.9 AP, improving over answer-only by 1.5 F1 and 1.3 AP. Post-hoc explanation scores 63.1 F1 and 69.1 AP, 2.2 F1 and 2.5 AP below answer-only. One possible reason is that post-hoc explanations, despite being produced after the prediction, still tend to inject commitment knowledge into the model, thereby hurting performance. Observation-only CoT scores 65.9 F1 and 72.1 AP, surpassing answer-only by 0.6 in F1 and 0.5 in AP, while observation-then-commit CoT falls further to 59.0 F1 and 50.9 AP, a drop of 6.3 F1 and 20.7 AP from answer-only. This comparison directly demonstrates that in video tasks with complex multimodal cues, incorporating commitment into CoT is more prone to harming model performance, which further highlights the importance of latent CoT. A key factor behind this phenomenon is that premature commitment in CoT causes excessive answer confidence. Even when the answer is wrong, the model still copies the CoT's conclusion verbatim, impairing its reasoning capability. DAL-CoT w/o decision-aligned recovers most of the latent gain, at 66.3 F1 and 72.7 AP. However, joint latent tokens suffer from attention collapse, leaving each slot underutilized and underperforming decision-aligned DAL-CoT, which achieves the best performance among all the reasoning paradigms.

4.4 Computational Cost Analysis

We measure per-sample inference cost for four reasoning interfaces that share the Qwen3-VL-8B backbone: answer-only, post-hoc explanation, observation-then-commit CoT, and DAL-CoT. We report single-stream latency measured on one NVIDIA GPU using batch size 1 and greedy decoding. We time the same 50 samples after three untimed warmup samples. Table 3 reports mean answer-token count and mean wall-clock latency. Answer tokens count every token produced for the response. For DAL-CoT this count includes the latent-slot prefix, which is written by one prefill; the autoregressive column counts only the tokens decoded after that prefix. Relative latency is normalized by answer-only.

DAL-CoT stays within 4% of answer-only latency and gains 1.3 AP. It emits 148.6 answer tokens against 82.8 for answer-only, a 79% increase, while latency moves from 2.75 s to 2.86 s ($1.04\times$). The

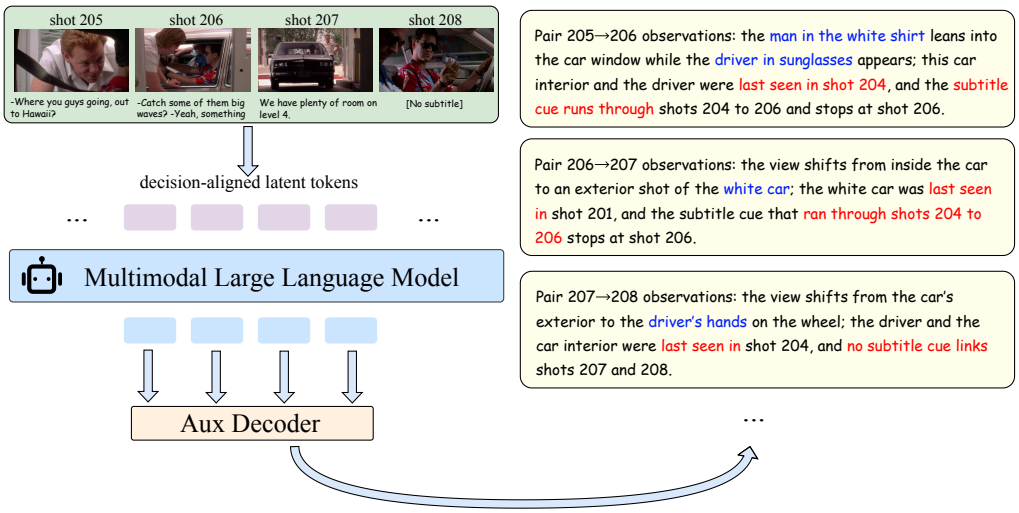


Figure 3: The decoding results of the auxiliary decoder on one sample, showing that each latent slot can decode an explicit CoT for the corresponding shot and capture multimodal cues across visual, subtitle, and character levels.

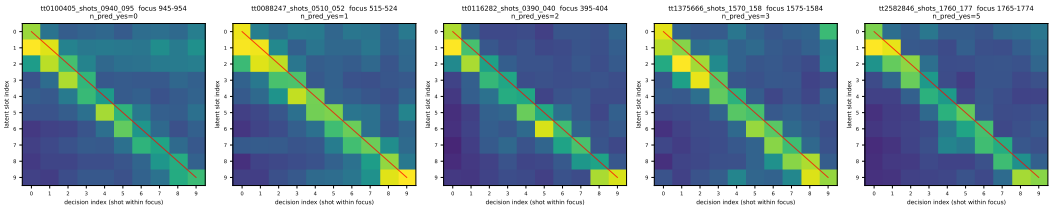


Figure 4: Attention from boundary decisions (rows) to latent slots (columns) for five samples with different number of predicted boundaries. DAL-CoT exhibits a clear moving diagonal rather than concentrating all decisions on the first few latent slots.

added tokens are almost entirely the latent prefix, whose mean length is 61.8. The autoregressive continuation is 86.8 tokens, only 4.0 above answer-only. In contrast, observation-only CoT is slower and less accurate. It generates 847.3 tokens and takes 24.58 s, which is $5.7\times$ the answer tokens and $8.6\times$ the latency of DAL-CoT. These results demonstrate that DAL-CoT matches answer-only latency with an excellent performance, while the auxiliary decoder provides on-demand explicit CoT, unifying efficiency, accuracy, and interpretability in a single framework.

4.5 Decoder Behavior Verification

The auxiliary decoder is removed during standard inference. To check what each decision-aligned slot encodes, we reactivate the auxiliary decoder and decode every slot. The probe covers 100 MovieNet-318 test samples that each contain a 10-shot focus, spanning 64 movies and 1,000 decoded texts. Among these, 1000 slots produce unique decoded texts (100%), and the character length of each decoded text ranges from a minimum of 150 to a mean of 211 and a maximum of 278. Fig. 3 illustrates the decoding results of the auxiliary decoder on one sample, showing that each latent slot can decode an explicit CoT for the corresponding shot and capture multimodal cues across visual, subtitle, and character levels.

Attention visualization. Figure 4 visualizes the attention from each generated boundary decision to the decision-aligned latent slots. As can be seen from the figure, when making each boundary decision, the model assigns the most attention to the corresponding latent slot and the next slot. This

Table 3: Per-sample inference cost on 50 samples. Token counts and latency are averaged over all evaluation samples. Answer tokens include the DAL-CoT latent-slot prefix, while AR tokens denote autoregressive continuation tokens only. Rel. indicates relative latency, defined as the mean latency of a method divided by that of the answer-only baseline. Best AP and lowest latency are in bold.

Method	AP ↑	Ans. tok.	AR tok.	Latency (s)	Rel.
Answer-only	71.6	82.8	82.8	**2.75**	1.00×
Post-hoc explanation	69.1	100.2	100.2	3.35	1.22×
Observation-only CoT	72.1	847.3	847.3	24.58	8.94×
DAL-CoT	**72.9**	148.6	86.8	2.86	1.04×

aligns with the characteristics of video scene segmentation and indicates that the model actually extract cues from both the current shot and the next shot during decision-making.

5 Conclusion

In this paper, we revisit the role of Chain-of-Thought reasoning in MLLM-based video scene segmentation and identify two failure modes that prevent existing reasoning paradigms from transferring to this task. Explicit CoT suffers from premature boundary commitment, which converts continuous and uncertain visual evidence into an autoregressive sequence of definitive statements, leading to an overconfident yet wrong answer. Latent CoT with stacked latent tokens suffers from attention collapse, where the model fails to assign distinct information to each state and concentrates almost all of its attention on the first few tokens when producing the final answer. To address these issues, we propose DAL-CoT, a decision-aligned latent reasoning method for video scene segmentation. Our method allocates one latent slot for each boundary decision, and constrains slot i to observe only the transition from shot i to shot $i + 1$. An auxiliary decoder reconstructs the corresponding explicit observation from each slot during training and is discarded at inference, so that reasoning is internalized into the hidden states rather than verbalized. Experiments show that our method consistently improves over other methods, reaching 66.8 F1 and 72.9 AP on MovieNet-318. Zero-shot evaluation on BBC Planet Earth further shows that the learned decision-aligned representations generalize beyond cinematic content. We leave to future work the extension of decision-aligned latent supervision beyond fixed-length focus windows and video scene segmentation, toward general multi-decision reasoning.

AI use statement

We used a coding assistant to revise this manuscript. The authors specified the method, checked the cited records, and take full responsibility for the final manuscript.

Ethics statement

This work fine-tunes publicly available movie-understanding models using annotations from the MovieNet dataset. The source videos are commercially released films. We do not collect, redistribute, or publicly release any raw video data, video clips, or audiovisual material. All experimental use of video content is limited to feature extraction or inference under fair-use provisions for non-commercial research purposes. We acknowledge that the underlying films are subject to copyright, and this work does not claim any rights over them.

Reproducibility statement

Section 3 defines slot cardinality and order, target scope, auxiliary conditioning, losses, training stages, and the inference boundary. Section 4 distinguishes re-reported published numbers from our measurements and specifies the controls required for each claim. Appendix A records the implemented data and optimization settings, while Appendix **??** defines the diagnostics. We will release the training configurations, evaluation scripts, and target-generation prompts.

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.

Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, pp. 1199–1202, 2015.

Nimrod Berman, Adam Botach, Emanuel Ben-Baruch, Shunit Haviv Hakimi, Asaf Gendler, Ilan Naiman, Erez Yosef, and Igor Kviatkovsky. Scene-VLM: Multimodal video scene segmentation via vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39848–39857, 2026.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Shixing Chen, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. Shot contrastive self-supervised learning for scene boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9796–9805, 2021.

Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne Hsu. Video-of-Thought: Step-by-step video reasoning from perception to cognition. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 13109–13125, 2024.

Songhao Han, Wei Huang, Hairong Shi, Le Zhuo, Xiu Su, Shifeng Zhang, Xu Zhou, Xiaojuan Qi, Yue Liao, and Si Liu. VideoEspresso: A large-scale chain-of-thought dataset for fine-grained video reasoning via core frame selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Vaishnavi Himakunthala, Andy Ouyang, Daniel Rose, Ryan He, Alex Mei, Yujie Lu, Chinmay Sonar, Michael Saxon, and William Yang Wang. Let's think frame by frame with VIP: A video infilling and prediction dataset for evaluating video chain-of-thought. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. MovieNet: A holistic dataset for movie understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Md Mohaiminul Islam, Mahmudul Hasan, Kishan Shamsundar Athrey, Tony Braskich, and Gedas Bertasius. Efficient movie scene detection using state-space transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18749–18758, 2023.

Jinghui Lu, Jiayi Guan, Zhijian Huang, Jinlong Li, Guang Li, Lingdong Kong, Yingyan Li, Han Wang, Shaoqing Xu, Yuechen Luo, et al. Xiaomi OneVL: One-step latent reasoning and planning with vision-language explanation. *arXiv preprint arXiv:2604.18486*, 2026.

Jonghwan Mun, Minchul Shin, Gunsoo Han, Sangho Lee, Seongsu Ha, Joonseok Lee, and Eun-Sol Kim. BaSSL: Boundary-aware self-supervised learning for video scene segmentation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pp. 4027–4043, 2022.

Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Najmeh Sadoughi, Xinyu Li, Avijit Vajpayee, David Fan, Bing Shuai, Hector Santos-Villalobos, Vimal Bhat, and Rohith MV. MEGA: Multimodal alignment aggregation and distillation for cinematic video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23331–23340, 2023.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 677–693, 2025.

Yudi Shi, Shangzhe Di, Qirui Chen, and Weidi Xie. Enhancing Video-LLM reasoning via agent-of-thoughts distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8523–8533, 2025.

Guohao Sun, Hang Hua, Jian Wang, Jiebo Luo, Sohail Dianat, Majid Rabbani, Raghuveer Rao, and Zhiqiang Tao. Latent chain-of-thought for visual reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jiawei Tan, Hongxing Wang, Jiaxin Li, Zhilong Ou, and Zhangbin Qian. Neighbor relations matter in video scene detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18473–18482, 2024.

Jiawei Tan, Hongxing Wang, Kang Dang, Jiaxin Li, and Zhilong Ou. Modality-aware shot relating and comparing for video scene detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 7193–7201, 2025.

Yan Wang, Yawen Zeng, Jingsheng Zheng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. VideoCoT: A video chain-of-thought dataset with active annotation tool. In *Proceedings of the Workshop on Advances in Language and Vision Research (ALVR)*, 2024.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. SIM-CoT: Supervised implicit chain-of-thought. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Haoqian Wu, Keyu Chen, Yanan Luo, Ruizhi Qiao, Bo Ren, Haozhe Liu, Weicheng Xie, and Linlin Shen. Scene consistency representation learning for video scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14021–14030, 2022.

A Implementation details

Both the backbone and auxiliary decoder are Qwen3-VL-8B-Instruct (Bai et al., 2025a). Training uses AdamW. Stage 0 runs for 1epoch at learning rate 1×10^{-5}. Stage 1 runs for 1 epoch at 1×10^{-4} and updates the auxiliary decoder while the backbone remains frozen. Stage 2 runs for 2 epochs at 1×10^{-5} and jointly updates the language backbone and auxiliary decoder. Its auxiliary coefficient warms linearly from 0 to 1 over the first 40 Stage-2 steps and then remains 1; the independent learning-rate scheduler uses a 0.03 warmup ratio. Answer-label tokens have weight 30, structural tokens have weight 0.1, and other supervised tokens retain the default weight.

Table 4: Staged training versus direct Stage 2 optimization on MovieNet-318. Loss start and loss end are the logged training objective at the first and last step.

Schedule	Stage	Loss start	Loss end	F1 ↑	AP ↑
Three-stage	1	15.75	2.08	–	–
Three-stage	2	3.57	1.67	**66.8**	**72.9**
Direct Stage 2	2	19.46	2.23	65.4	71.6

Table 5: Auxiliary decoder capacity. Left: Stage-1 reconstruction loss averaged over blocks of 20 optimizer steps, with the backbone frozen and every other setting matched. Right: end-to-end MovieNet-318 scores after the full curriculum.

Steps	4B dec.	8B dec.	Decoder	F1 ↑	AP ↑
1–20	2.1793	2.1004	Qwen3-VL-4B	65.5	71.2
21–40	1.0557	0.9521	Qwen3-VL-8B	**66.8**	**72.9**
41–60	0.9248	**0.7148**			
61–80	0.9267	**0.6948**			
81–100	0.9008	**0.6990**			

B STAGED TRAINING

Section 3.4 adapts the backbone with the answer loss, then fits the auxiliary decoder on a frozen backbone, and only afterwards optimizes the two together. To test whether this staged training strategy is required, we compare the full three-stage curriculum with a run that enters Stage 2 directly from the pretrained weights. Table 4 reports the logged training loss at the first and last step of each stage, together with MovieNet-318 F1 and AP.

For three-stage training, stage 1 lowers the loss from 15.75 to 2.08 before joint optimization begins. Stage 2 then moves from 3.57 to 1.67. In contrast, directly entering Stage 2 from the pretrained backbone starts at 19.46, about $5.5\times$ the three-stage curriculum's Stage 2 start, and ends near 2.23 with slight variations in the late training stage, above the curriculum's endpoint of 1.67. This higher loss lines up with weaker boundary predictions: direct Stage 2 reaches 65.4 F1 and 71.6 AP, 1.4 F1 and 1.3 AP below the three-stage model. A further sign that the curriculum transfers rather than merely warms up the decoder is that the Stage-2 reconstruction loss starts at 0.68, continuing the 0.72 reached at the end of Stage 1, instead of jumping back to the Stage-1 initial value.

C AUXILIARY DECODER CAPACITY

The auxiliary decoder exists only during training, so it is natural to ask whether a smaller decoder would do. We replace the 8B auxiliary decoder with Qwen3-VL-4B and keep everything else fixed: the same Stage-0 backbone checkpoint, the same learning rate, the same global batch size, the same slot form, and the same data.

Reconstruction is measurably harder for the 4B decoder. Table 5 bins the Stage-1 reconstruction loss in blocks of 20 optimizer steps. Over the first 40 steps the two decoders are nearly indistinguishable: both cross 2.0 at step 8 and 1.2 at step 11, so the early descent reflects the easily learned part of the target format rather than decoder capacity. The curves separate after step 40. The 4B decoder flattens between 0.90 and 0.93 and shows no further trend across the remaining three bins, while the 8B decoder continues to about 0.70 and stabilizes there; over the last 20 steps the 4B loss is 22% higher. In other words, the 4B decoder reaches its capacity ceiling on this reconstruction task at around step 40, and the 8B decoder does not. The larger decoder is also slower, 22.2 s per step against 17.5 s, a 27% increase in Stage-1 training time.

The harder reconstruction shows up in the final metrics. Carried through the full curriculum, the 4B decoder gives 65.5 F1 and 71.2 AP, 1.3 F1 and 1.7 AP below the 8B decoder. What it loses is the part of the benefit that depends on the observation text being reconstructible from the slot.

D PROMPT AND TARGET CONSTRUCTION

The user prompt states the context and focus range. The requested assistant output contains only an `<answer>` block with one comma-separated line of `Shot k: Yes/No` entries covering every focus shot.

For observation slot i, the teacher target describes visual content, dialogue, characters, actions, and location changes across $f_i \rightarrow f_i{+}1$. Its supervised header uses the form `Pair 5→6 observations:`. Targets are generated offline and stored.

E CONSTRUCTION OF THE EXPLICIT COT

The auxiliary decoder is supervised with explicit CoT text, so the shape of that text determines what a latent slot is asked to hold. This appendix explains why we write the explicit CoT as pair-wise observation instead of as reasoning toward a verdict, and how the corpus was produced and checked.

Why observation rather than deliberation. The failure mode of Section 3.2 is premature boundary commitment: once a rationale has written down a verdict for a transition, the answer token tends to copy it, and a wrong commitment becomes a confident error. Any CoT that reasons toward a verdict inherits this risk, so the supervision itself must stop short of the verdict.

The second consideration is the structure of the evidence. A shot transition in a film offers many features that can serve as a boundary cue, and in the general case they disagree. The image may cut to a different location while the dialogue runs uninterrupted across the cut; the camera may change angle and framing while the characters remain in the same room; a cast member may leave the frame without leaving the scene; a subtitle line may be aligned to both sides of a cut as an artifact of its time span rather than as continuity. A single conclusion sentence per transition cannot represent this disagreement: it has to pick one cue and discard the rest. An observation, by contrast, can record that the location changed and that the dialogue continued, and leave the resolution to the model.

We therefore supervise *observational* CoT. For each shot pair the teacher retrieves and states cues, and states nothing else. Each line covers exactly one pair, but its pointers are allowed, and required, to reach across the entire provided context rather than only the two shots being described.

Teacher and prompt. Targets are generated offline by Qwen3-VL-235B-A22B-Instruct, served locally. The teacher is label-blind: the boundary instruction of the task prompt is explicitly overridden, so nothing in the teacher's context states or asks for a verdict. The output format is fixed to exactly one line per transition, each opening with `Pair N→N+1 observations:`, at most two sentences. Figure 5 reproduces the labeling prompt verbatim, including the pointer rules.

Quality control. Target quality is checked in three steps.

Programmatic checks. A validator applies sixteen design checks in three groups, structure, leakage, and pointer form, plus four further checks added after an error analysis found that the first sixteen validate the form of a pointer but never its informativeness, so a well-formed and entirely vacuous pointer passes them. Many of the properties that matter here are decidable by script, and shot indices are the clearest case: one check requires the pair indices of the ten lines to equal the focus range exactly, another requires every cited shot number to lie inside the 20-shot window, and a third requires a recurrence pointer to name a shot strictly earlier than $N{+}1$, since $N{+}1$ is the shot being described and cannot be where something was seen earlier. The leakage group rejects any line containing a verdict word or a `Shot N: Yes/No` pattern. The pointer group rejects bare temporal words without a shot number, cue spans that only restate the pair, cue spans without an end, and a cue cited again after it was declared to stop.

Model self-check. The same validator runs inside the labeling loop rather than only as a post-hoc filter. When a generated block fails, the specific errors and the geometry of the example are fed back to the teacher as a corrective reminder and the block is regenerated exactly once; a block that fails again is left to the programmatic filter.

Human sampling. To complement automatic verification, we manually inspect a random subset of CoT lines whose claims are factual (e.g., numerical values, entity names, causal relations) rather

ROLE

You write observation notes for a shot-transition study. You are an observer, not a judge.

INPUT

20 consecutive movie shots (indices 0–19), each with 3 keyframes, subtitles, and cast pids.

TASK

Describe the 10 transitions between consecutive focus shots: 5→6, 6→7, 7→8, 8→9, 9→10, 10→11, 11→12, 12→13, 13→14, 14→15.

WHAT EACH LINE MUST COVER

Every claim carries explicit shot numbers, because the evidence linking two shots usually lies outside the pair.

1. Local difference — what changes between shot N and N+1 in location, cast, action, framing, and visible head count.

2. Recurrence pointer — where the setting of shot N+1 was already visible within shots 0..N.

3. Cast pointer — give the pid string and state whether every pid in shot N+1 already appears somewhere in shots 0..N-1.

4. Dialogue pointer — report where a run of speech starts and ends.

5. Cue pointer — include ONLY when a cue reaches outside the pair: handset/radio/pager, an eyeline plus what is being looked at, a monochrome or faded image, or a continuing action. Write the span plus its end, where the end is the LAST shot still showing the cue.

HARD CONSTRAINTS

- Output exactly 10 lines, nothing else.
- One line per transition, ascending, each beginning `Pair N→N+1 observations:` .
- Each line is 1–2 sentences.
- Plain observable facts only. Never state or imply a decision.

Now output the 10 observation lines.

Figure 5: The labeling prompt given to the teacher model, reproduced verbatim. The numbered items are the pointer rules that force each observation line to cite concrete shot numbers outside its own pair; the hard constraints fix the output format and forbid every word that would state a boundary decision.

than purely formal. We draw 100 lines per corpus for human review. This sample is excluded from the automatic pass/fail decision; instead, it serves as an independent estimate of how often a syntactically well-formed claim in the CoT is also factually correct.

Table 6 reports the final state of the corpus used in this paper. Every example carries at least one informative pointer, observation lines average 36.9 words, and no example reuses a cue after declaring that it stopped.

Table 6: Programmatic quality control of the observational-CoT corpus. Pointer coverage is the fraction of observation lines carrying at least one concrete shot pointer, averaged over examples. Pointer contradiction is the rate of cue intervals that conflict across pairs of the same example.

Quantity	Value
Examples checked	29,038
Examples passing every check	29,014
Pass rate	99.92%
Pointer coverage (mean)	1.000
Words per observation line (mean)	36.9
Degenerate cue spans	0.0%
Examples reusing a cue after its declared stop	0
Human-reviewed factuality sample	100

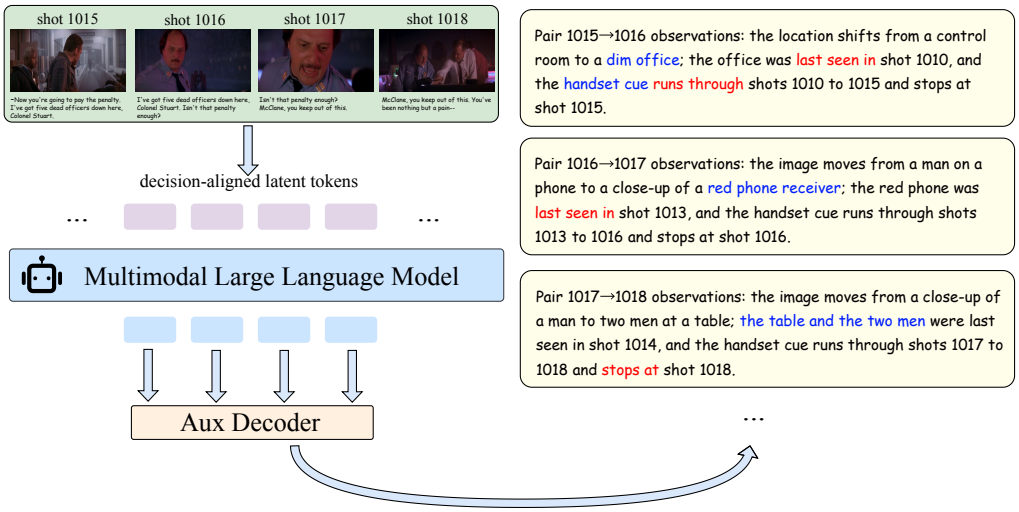


Figure 6: More decoding results of the auxiliary decoder on one sample, showing that each latent slot can decode an explicit CoT for the corresponding shot and capture multimodal cues across visual, subtitle, and character levels.