

Scene-VLM: Multimodal Video Scene Segmentation via Vision-Language Models

Nimrod Berman^{1*†} Adam Botach^{2*§‡} Emanuel Ben-Baruch² Shunit Haviv Hakimi²
Asaf Gendler² Ilan Naiman^{1†} Erez Yosef^{3†} Igor Kviatkovsky²

¹Ben-Gurion University ²Amazon Prime Video ³Tel-Aviv University

{bermann,naimani}@post.bgu.ac.il {kabotach,emanbb,havivs,gendlasa,kviat}@amazon.com erez.yo@gmail.com

Abstract

Segmenting long-form videos into semantically coherent scenes is a fundamental task in large-scale video understanding. Existing encoder-based methods are limited by visual-centric biases, classify each shot in isolation without leveraging sequential dependencies, and lack both narrative understanding and explainability. In this paper, we present Scene-VLM, the first fine-tuned vision-language model (VLM) framework for video scene segmentation. Scene-VLM jointly processes visual and textual cues including frames, transcriptions, and optional metadata to enable multimodal reasoning across consecutive shots. The model generates predictions sequentially with causal dependencies among shots and introduces a context-focus window mechanism to ensure sufficient temporal context for each shot-level decision. In addition, we propose a scheme to extract confidence scores from the token-level logits of the VLM, enabling controllable precision-recall trade-offs that were previously limited to encoder-based methods. Furthermore, we demonstrate that our model can be aligned to generate coherent natural-language rationales for its boundary decisions through minimal targeted supervision. Our approach achieves state-of-the-art performance on standard scene segmentation benchmarks. On MovieNet, for example, Scene-VLM yields significant improvements of +6 AP and +13.7 F1 over the previous leading method.

1. Introduction

Video scene segmentation, the task of identifying coherent narrative boundaries within long-form video content, is fundamental to video understanding [5, 15, 16, 19, 25, 26, 30]. Accurate scene boundary detection is crucial for organizing, searching, and understanding video content at scale, enabling applications such as automated structured summa-

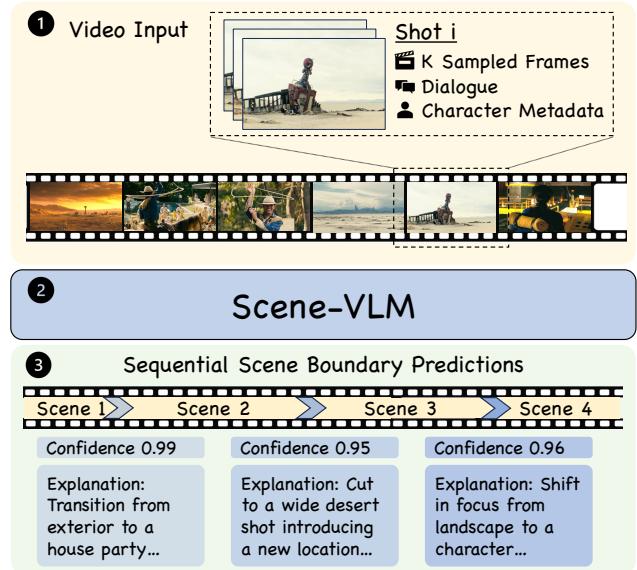


Figure 1. **Video scene segmentation with Scene-VLM.** We present Scene-VLM, the first vision-language model (VLM) framework fine-tuned for video scene segmentation. Scene-VLM jointly processes visual frames, dialogue, and metadata from consecutive shots to sequentially predict scene boundaries with associated confidence scores, and can be aligned to produce coherent post-hoc explanations for its decisions.

rization, semantic retrieval and contextual advertising. Formally, a *scene* is a consecutive sequence of shots sharing semantic coherence in location, time, characters, or narrative theme, where each *shot* is a continuous sequence of frames captured in a single, uninterrupted camera take (see Fig. 1). Despite decades of research, video scene segmentation remains challenging as it requires understanding narrative semantics beyond visual cues, distinguishing meaningful story transitions from superficial visual changes.

Recent state-of-the-art methods have made substantial progress through cross-modal fusion and efficient temporal modeling. BaSSL [19] employs boundary-aware self-supervised pretraining with pseudo-boundaries to learn transition cues. In contrast, TranS4mer [16] combines

*Equal contribution.

†Work done during an Amazon internship.

‡Internship mentor.

§Corresponding author: kabotach@amazon.com.

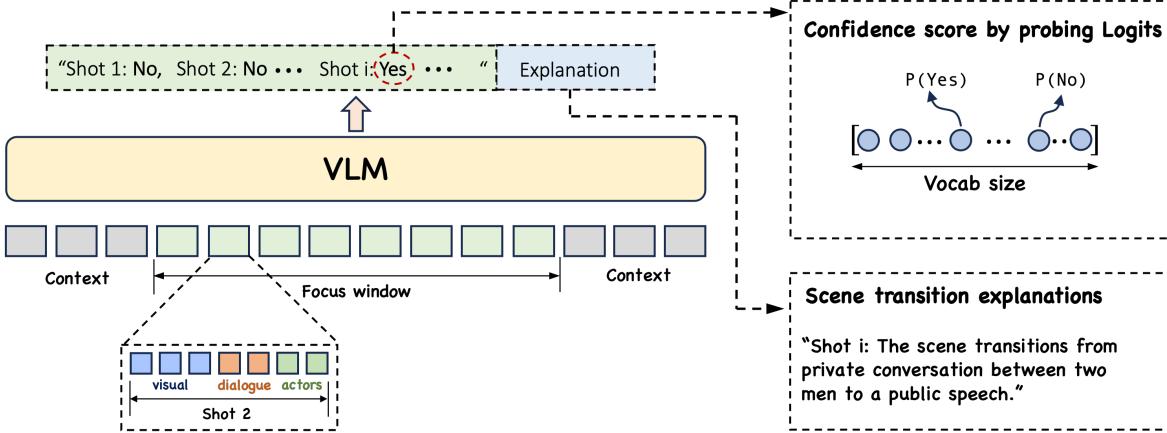


Figure 2. Proposed approach. The VLM receives a sequence of N multimodal shot representations as input. Each shot representation consists of visual frames, dialogue, and optional metadata. *Scene-VLM* processes the shots within a focus window (green) using information from a larger context window (gray), and outputs scene boundary predictions for the shots in focus in the format “Shot i : Yes/No.” For each shot, we compute a confidence score by probing the softmax logits of the “Yes” and “No” tokens and normalize by $P(\text{Yes})/(P(\text{Yes}) + P(\text{No}))$. The model can also be aligned to generate coherent post-hoc explanations for its scene-boundary decisions.

self-attention with state-space layers to efficiently process long-range dependencies, and MEGA [26] aligns video, audio, and text via cross-modal attention. Despite these advances, existing methods share several fundamental limitations. First, they all exhibit visual-centric biases, ignoring or underutilizing non-visual signals such as dialogue and character presence despite their importance for narrative understanding [15]. Second, they all follow a mutual point-wise prediction paradigm, classifying each shot independently within a local temporal window without leveraging causal dependencies across consecutive decisions. Finally, as encoder-based methods, they all offer no insight into *why* a boundary was predicted beyond a confidence score, limiting their usability in human-in-the-loop settings.

Recent advances in vision-language models (VLMs) [1–3, 22, 27, 28] offer a compelling path to overcome these limitations. VLMs unify visual perception and natural language understanding in a single framework, processing visuals alongside text and generating textual responses that enable deeper cross-modal reasoning and explainable predictions. Popular model families such as Qwen-VL [22], LLaMA [2], and GLM [28] demonstrate strong multimodal capabilities, making them viable backbones for structured video understanding tasks.

In this work, we introduce *Scene-VLM*, a fine-tuned VLM framework for video scene segmentation. To the best of our knowledge, this represents the first application of VLMs to this task, marking a paradigm shift from traditional encoder-based approaches to multimodal narrative reasoning. As illustrated in Fig. 2, *Scene-VLM* addresses prior limitations through several key design choices.

First, we introduce a **structured multimodal shot representation** fusing visual frames, speech, and optional metadata, providing the model with narrative context un-

available to visual-centric methods. Second, we replace the common point-wise prediction paradigm with **sequential processing**, where predictions for multiple shots are generated sequentially, making each boundary decision causally inform subsequent ones. Third, to mitigate edge effects from limited context at sequence boundaries, we employ a **context–focus window design**: a broader context window provides temporal padding around a central focus window where predictions are emitted, ensuring every shot has adequate past and future evidence. Fourth, since VLMs do not natively produce confidence scores, we propose a **confidence prediction scheme** that reliably derives confidence scores from token-level logits of the model’s outputs. This enables flexible operating points across precision–recall trade-offs, a capability typically reserved for encoder-based classifiers. Finally, we demonstrate that *Scene-VLM* can be effectively aligned to generate **coherent natural-language rationales** for its boundary predictions through fine-tuning on a small set of annotated explanations.

To conclude, we summarize our contributions as follows:

- We introduce *Scene-VLM*, a *fine-tuned VLM framework for video scene segmentation*, featuring a structured multimodal shot representation, sequential predictions with causal dependencies, and a context–focus window design for comprehensive temporal reasoning.
- We propose a novel *scene boundary confidence prediction scheme* that derives scores from VLM token-level logits, enabling controllable precision–recall trade-offs typically reserved for encoder-based methods.
- We achieve *state-of-the-art results* on MovieNet (+6 AP, +13.7 F1 over the previous leading method), strong zero-shot performance on BBC Planet Earth, and demonstrate adaptability to video chaptering.
- We provide *comprehensive analysis* through ablations

and attention studies, and explore post-hoc explainability by aligning our method to generate natural-language rationales for its boundary predictions.

2. Related Work

Video Scene Segmentation Segmenting long-form videos into semantically coherent scenes is a long-standing problem in video understanding. Early approaches cast it as an unsupervised clustering task, grouping visually similar shots based on low-level descriptors [6]. While label-free, such methods failed to capture narrative-level transitions extending beyond visual continuity. With the emergence of deep learning, hand-crafted features were replaced by CNN-based embeddings [5], and sequential models (e.g., LSTMs) modeled temporal continuity across shots [15]. The MovieNet dataset [15] enabled large-scale supervised learning and multimodal integration of subtitles, speech, and character identities [25, 26].

To further improve representation learning, *ShotCoL* [9] introduced a contrastive self-supervised framework that learns discriminative shot embeddings. Building on this, *BaSSL* [19] proposed a boundary-aware approach that learns transition cues without annotations. However, both operate on short temporal spans and rely mainly on appearance cues, limiting their ability to capture semantic or narrative context. More recently, *TransS4mer* [16] combined self-attention with state-space modeling to efficiently capture long-range dependencies, yet remains an encoder-based classifier without narrative reasoning or explainability. On the multimodal front, *MEGA* [26] fused visual, subtitle, and screenplay features via multimodal alignment and distillation, but depends on tightly aligned metadata and remains constrained by fixed fusion strategies. Despite these advances, most methods still struggle to capture high-level semantics and generalize beyond the cinematic domain.

A related task, video chapter segmentation in web videos [33], defines boundaries semantically without relying on shot units. Models such as *Chapter-LLaMA* [29] leverage large language models (LLMs) over transcripts and visual descriptions to segment long-form content, effectively compensating for the lack of high-level semantics that visual-based methods fail to capture;

however, these models exhibit degraded performance when applied to cinematic materials, likely due to the stronger visual structure of films.

Tab. 1 summarizes the capabilities of prior methods relative to ours. Classical video scene segmentation models do not naturally support sequential predictions or explainability and are not designed for chaptering. Chapter-LLaMA, in contrast, is tailored for chaptering and supports sequential generation and explainability, but it neither performs scene segmentation for cinematic content nor provides confidence scores. Our framework is the only one that (i)

Table 1. **High-level comparison of key capabilities across scene segmentation methods.** Scene-VLM supports all 5 capabilities.

Method	Sequential Prediction	Confidence Scoring	Explainability	Cinematic	Chaptering
ShotCoL	✗	✓	✗	✓	✗
BaSSL	✗	✓	✗	✓	✗
TransS4mer	✗	✓	✗	✓	✗
MEGA	✗	✓	✗	✓	✗
Chapter-LLaMA	✓	✗	✓	✗	✓
Scene-VLM (ours)	✓	✓	✓	✓	✓

performs *both* scene segmentation and chaptering competitively, (ii) makes sequential predictions, (iii) exposes confidence scores, and (iv) can produce natural-language rationales.

Vision-Language Models for Video Understanding

Recent years have witnessed rapid progress in the utilization of vision-language models (VLMs) for various video understanding tasks [10, 12, 18, 31, 32, 34]. However, video processing remains challenging due to internal context-length limitations. To mitigate this, recent works introduce hierarchical representations and efficient temporal modeling to extend reasoning horizons [17, 20, 32]. Among open-source models, Qwen2.5-VL [4] demonstrates strong multimodal reasoning, making it a viable backbone for structured video understanding. Despite this emerging potential, to our knowledge our work is the first to explore VLMs for the task of video scene segmentation.

3. Approach

In this section, we outline our proposed approach for leveraging a vision-language model (VLM) to perform video scene segmentation. An overview of our approach is illustrated in Fig. 2. As depicted, our method involves fine-tuning the VLM to jointly process visual frames, subtitles, and character information in order to identify scene boundaries within a given context window, while associating confidence scores with each shot in a sequential manner.

3.1. VLM for Video Scene Segmentation

We define a context window $C = \{s_i\}_{i=1}^N$ containing N consecutive shot representations. Each shot representation s_i consists of K sampled frames $\{f_{i,k}\}_{k=1}^K$ along with synchronized subtitles and character information associated with the shot. The context C is then provided to the vision-language model \mathcal{M} along with an instruction prompt P , which directs the model to identify shots corresponding to scene boundaries. Formally,

$$Y = \mathcal{M}(C, P), \quad (1)$$

where $Y = \{y_j\}_{j=1}^T$ denotes the sequence of output logits produced by the model, and each $y_j \in \mathbb{R}^{|V|}$ represents the predicted token distribution over the vocabulary V . The instruction prompt guides the model to produce its predictions as a sequence of shot-level entries, each following

the format `shot_id:<id>: Yes/No`. Following prior work [16], we label a shot as positive (`Yes`) if it marks the end of a new scene. An illustrative example of the output format is shown in Fig. 2. This scheme enables sequential inference, in which the prediction for each shot is conditioned on the predictions of preceding shots.

Since each prediction depends on the surrounding temporal context, shots near the boundaries of the context window tend to yield less reliable predictions. To address this, the instruction prompt restricts predictions to a *focus window* centered in the context, ensuring that each evaluated shot has sufficient temporal evidence from both preceding and following shots. In practice, the context window typically comprises 20 consecutive shots, and the focus window includes the central 10.

We fine-tune the VLM on an annotated dataset for video scene segmentation (e.g., [15]), where each training sample consists of a context window provided to the model along with the instruction prompt. The model is trained to generate responses Y that align with the target labels (`Yes/No`) for each shot in the focus window. Training is performed using a next-token prediction loss. Optionally, the model can be further aligned to generate natural-language explanations for its boundary predictions, as described in Section 4.7.

3.2. Computing Soft Predictions

Estimating confidence scores for scene boundary predictions is crucial for controlling the precision–recall trade-off. Unlike encoder-based approaches to scene boundary detection, where prediction scores can be obtained directly from a dedicated classification head, VLMs generate a sequence of textual output tokens in response to a given instruction prompt. Consequently, we propose computing a confidence score for each shot-level decision by probing the logit vectors corresponding to the `Yes/No` tokens.

Specifically, the probability of a token t at position j in the output sequence is defined as

$$p_j(t) = \frac{\exp(y_j[t])}{\sum_{u \in V} \exp(y_j[u])}. \quad (2)$$

For each shot, we identify the position of its verdict entry in the structured output sequence and compute the probabilities of `Yes` (scene boundary) and `No` (continuation). We denote these as $p_i(\text{Yes})$ and $p_i(\text{No})$, representing the positive and negative probabilities for shot i , respectively. The confidence score for shot i is then defined as

$$\text{conf}_i = \frac{p_i(\text{Yes})}{p_i(\text{Yes}) + p_i(\text{No})}. \quad (3)$$

This scheme allows sequential computation of the probability that a given shot contains a scene boundary, conditioned on the multimodal inputs *and* on model predictions for preceding shots. We discuss alternative output formats and additional design considerations in the Appendix.

4. Experiments

We present a comprehensive empirical evaluation of *Scene-VLM*. We begin by describing the datasets and implementation details (Sec. 4.1 and Sec. 4.2). We then establish state-of-the-art performance on standard scene segmentation benchmarks (Sec. 4.3). To better understand the source of these gains, we conduct systematic analyses: Sec. 4.4 examines how the model distributes attention across different input modalities and temporal context, and Sec. 4.5 presents a comprehensive ablation study of the model’s key design choices. In Sec. 4.6 we evaluate generalization to the related video chaptering task. Finally, in Sec. 4.7 we explore aligning our framework to generate coherent verbal explanations for its boundary decisions through targeted fine-tuning on a small set of annotated samples. More details and additional experiments are provided in the Appendix.

4.1. Datasets

MovieNet-318. This dataset is a subset of the MovieNet dataset [15] for scene segmentation in cinematic content. It contains 318 movies with shot-level annotations and scene boundary labels, split into 190 for training, 64 for validation, and 64 for testing. On average, each movie contains around 1000 shots, with each shot annotated with a binary label indicating whether it marks a scene boundary.

BBC Planet Earth. This dataset [5] is a standard out-of-distribution benchmark for scene segmentation, consisting of 10 documentary episodes from the Planet Earth series. Each episode averages 50 minutes in duration, with 670 scenes and 4.9K shots in total. Unlike cinematic content, these episodes feature documentary-style narration, non-fiction pacing, and highly diverse visual domains (wildlife, landscapes, climate footage).

VidChapters-7M. This dataset [33] addresses the task of segmenting web videos into chapters with timestamped titles. It aggregates videos and user-defined annotations from YouTube. Since the full dataset is not publicly released, we follow [29] and use a reproducible subset consisting of 1000 training samples and 300 evaluation videos.

4.2. Implementation Details

We use Qwen2.5-VL-7B [4] as our base model. To construct the multimodal shot representations, we: (1) segment videos into shots using standard methods [8, 21] or use provided shot annotations; (2) extract per-shot transcripts using Whisper [23] or use provided subtitles; and (3) add per-shot metadata (e.g., actor identities) when available. Unless stated otherwise, we use a context window of 20 shots, a focus window of 10 shots, and sample 3 frames per shot in all experiments. In addition, we overlay a small visual identifier (shot-ID marker) on each frame to help the model associate visual content with the corresponding shot references

Table 2. **Results on MovieNet-318.** Scene segmentation performance on the MovieNet-318 dataset.

Method	F1 \uparrow	AP \uparrow
LGSS [25]	—	47.1
ShotCoL [9]	—	53.4
BaSSL [19]	47.0	57.4
MEGA [26]	55.3	58.6
TranS4mer [16]	48.4	60.8
Scene-VLM (ours)	62.1	66.8

in the textual input sequence. Additional training, inference and input related details are provided in the Appendix.

4.3. Scene Segmentation Results

MovieNet-318. We train on MovieNet-318 and evaluate on the test split. Following prior work we report F1 and Average Precision (AP) scores, and compare against leading approaches. As depicted in Tab. 2, *Scene-VLM* establishes a new state of the art on MovieNet-318, substantially outperforming previous work. In particular, we achieve a gain of +6.8 in F1 and +8.2 in AP over MEGA [26], and +13.7 in F1 and +6.0 in AP over TranS4mer [16].

BBC Planet Earth. Following prior work [7, 16, 19], we evaluate *zero-shot* on BBC after training on MovieNet-318 to assess cross-domain generalization. Since prior work does not report F1 on BBC, we report only AP. As seen in Tab. 3, *Scene-VLM* sets a new zero-shot state of the art on BBC, outperforming the previous leading method TranS4mer [16] by +2.2 AP.

Method	AP \uparrow
BaSSL [19]	40.0
TimeSformer [7]	42.2
TranS4mer [16]	43.6
Scene-VLM (ours)	45.8

Table 3. **Results on BBC Planet Earth.** Zero-shot scene segmentation performance on the BBC Planet Earth dataset.

4.4. Attention Analysis

Each output token generated by the VLM, particularly the shot-level verdict token (Yes/No for a given shot), attends to all preceding output tokens as well as to the input tokens provided to the model. In this section, we analyze how each input component contributes to the model’s decisions by examining the attention allocated to the input modalities and the previously generated output tokens.

We denote by A_{ij} the attention weight between an output token i and any preceding token j (from either the input or previously generated outputs). We then compute the contribution of each component by averaging attention values across all layers and attention heads. For example, to obtain aggregate attention between a given shot-prediction token at

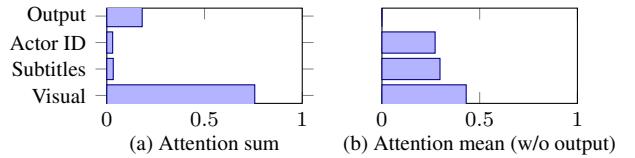


Figure 3. **Attention by modality.** Visualization of attention distribution across input modalities (visual, subtitles, and actor IDs) as well as preceding output shot predictions. (a) Summed attention reveals strong visual dominance and high dependency on prior output tokens. (b) Averaged (length-normalized) attention highlights that subtitles and actor IDs contribute comparably to visual tokens.

position i and the set of visual input tokens \mathcal{V} , we compute $\sum_{j \in \mathcal{V}} A_{ij}$, averaged over all layers and attention heads.

We begin by analyzing the contribution of each component to the model’s predictions by aggregating attention values across all shots for each input modality, and for the preceding output tokens. Fig. 3 (left) shows the aggregated attention distribution across the input modalities: visual, subtitles, and actor ID. As observed, visual tokens receive the largest share of attention, indicating their dominant role in predicting scene transitions. The preceding output tokens also receive substantial attention, highlighting the importance of sequential inference enabled by the method design.

In Fig. 3 (right), we present the attention distribution after normalizing each modality by its token count. For example, for the visual modality, we compute $\frac{1}{|\mathcal{V}|} \sum_{j \in \mathcal{V}} A_{ij}$ to normalize its aggregate attention. This normalization allows us to assess the relative importance of each modality while eliminating biases introduced by differences in token counts. We exclude the output tokens from this normalization since, being few but densely connected, they would otherwise dominate the average. As shown, after normalization, subtitle and actor tokens receive attention comparable to visual tokens, indicating that these modalities provide valuable cues for identifying scene transitions in video.

Next, we analyze how each shot prediction attends to individual shots in the input sequence. We fix a specific shot index and average the attention over multiple samples where a positive scene transition occurs at that index. For each shot, we aggregated attention between the output token corresponding to that shot and the input tokens of each shot, and also examine the relative attention across the visual, subtitle, and actor-ID modalities. In Fig. 4, we show the resulting attention distributions for three output predictions, shots 7, 11, and 15, each averaged over 30 context samples sharing the same positive transition index. Interestingly, as observed particularly for shots 11 (b) and 15 (c), the output token corresponding to the shot prediction attends more strongly to the subsequent shots in the input sequence than to the preceding ones. We hypothesize that this behavior arises because the prediction for a given shot already encodes information about the preced-

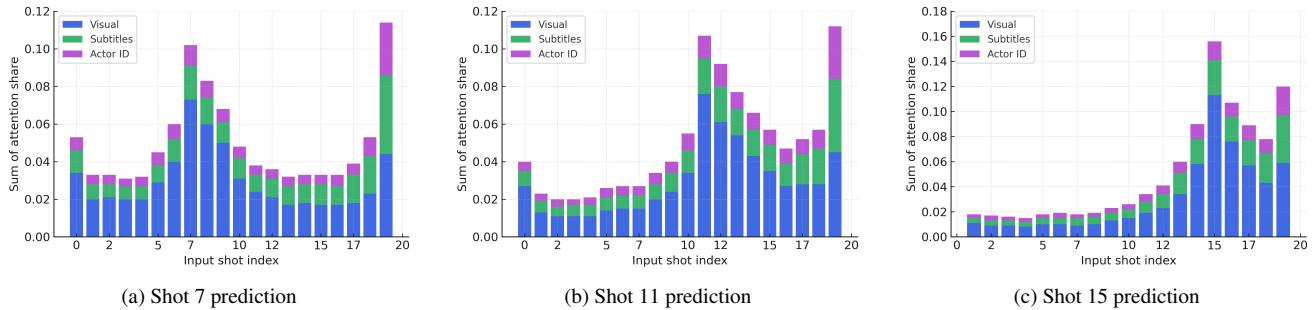


Figure 4. Attention distributions across shots and modalities. Figures show modality-level stacked attention shares for three shot predictions: (a) **Shot 7**, (b) **Shot 11**, and (c) **Shot 15**. Attention is computed between the output token of the corresponding shot prediction and the input tokens for all shots. Each bar represents the relative attention of the visual, subtitle, and actor-ID input modalities per shot.

ing shots through the previously generated output tokens. Consequently, the model allocates less attention to earlier shots it has already “seen” and more to subsequent ones, which provide additional context necessary for identifying the scene transition. This observation supports our findings in Sec. 4.5.2, reinforcing that sequential prediction plays a crucial role. The model appears to *trust* its previous predictions, allocating less attention to shots it has already processed and focusing instead on future inputs that may refine or confirm its ongoing decision.

Finally, we observe a high peak of attention at the first and last input shots. We hypothesize that this behavior helps the model identify the temporal boundaries of the input sequence more easily.

4.5. Ablation Study

We conduct systematic ablations to validate our design choices and quantify the contribution of different components. We examine (1) the relative importance of visual, textual, and metadata inputs, (2) the necessity of the context-focus window design for position-invariant predictions and the impact of different window sizes, (3) the impact of the number of key frames per shot, and (4) model size effects. Ablations are conducted on MovieNet-318.

4.5.1. Input Components Analysis

To understand which input modalities drive performance, we explore the contribution of four components: (1) Visual key frames, (2) shot-ID markers, which provide visual identifiers overlaid on frames, (3) subtitles, and (4) actor-IDs. We perform two complementary ablations: (i) remove one component at a time from the full model, and (ii) keep a single component and remove the rest.

The results shown in Tab. 4 reveal clear performance hierarchies across components. Visual removal causes catastrophic failure ($F1: 62.1 \rightarrow 32.0$), establishing vision as the primary boundary cue. However, other components provide meaningful contributions: Shot-ID removal drops performance by 1.3 F1 points, indicating that temporal anchoring

still matters beyond raw visuals, while subtitle and actor-ID removal each cause approximately 1 point drops, showing these components contribute complementary signals.

The isolation experiments further confirm this hierarchy. Visual-only achieves strong performance (58.6 F1), showing that many scene boundaries have clear visual signatures. In contrast, subtitle-only and actor-only configurations degrade sharply, implying that textual signals alone lack sufficient boundary evidence.

Table 4. Input component ablation. Top section shows removal of individual components; bottom section shows performance with only single components.

Visual	Shot-ID	Subtitles	Actor-ID	F1↑	AP↑
✓	✓	✓	✓	62.1	66.8
✗	✗	✓	✓	32.0	34.7
✓	✗	✓	✓	60.8	64.1
✓	✓	✗	✓	61.1	62.2
✓	✓	✓	✗	61.3	62.0
✓	✗	✗	✗	58.6	61.4
✗	✗	✓	✗	31.5	33.2
✗	✗	✗	✓	24.8	28.6

4.5.2. Context-Focus Window and Sequential Prediction

We next study two aspects of our sequential prediction approach: whether context margins prevent performance degradation at edge positions, and how different context-focus window sizes affect performance while demonstrating the benefits of sequential over point-wise prediction.

Performance Degradation at Sequence Edges. To validate the necessity of our focus mechanism in preventing performance degradation near sequence boundaries, we analyze per-position performance across 10 sequential predictions. We compare two settings: with focus (predicting only for the central 10 shots within a 20-shot context window) versus without focus (predicting for all 10 shots without surrounding context).

Fig. 5 shows the per-position F1 scores for both configurations. *Without the focus mechanism* (red triangles), performance collapses dramatically at sequence boundaries, with edge positions showing severe degradation compared to central positions. *With the focus mechanism* (blue circles), performance remains consistent across all positions, with no outliers beyond 3 standard deviations from the mean. This demonstrates that temporal context margins are essential for position-invariant performance.

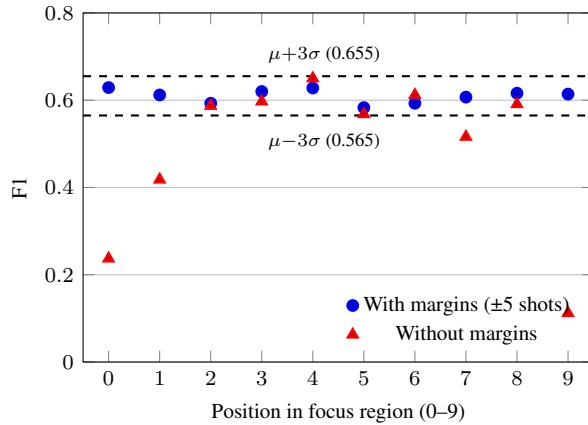


Figure 5. Focus mechanism prevents edge degradation. Performance collapses at boundaries without focus mechanism (red) but remains stable with focus mechanism (blue).

Context & Focus Window Sizes. We systematically ablate context and focus window sizes by testing $l_{\text{context}} \in \{20, 10, 5\}$ and $l_{\text{focus}} \in \{20, 10, 5, 1\}$. Results in Tab. 5 show two consistent trends. First, **sequential prediction matters**: for any context window size, reducing focus to a single shot ($l_{\text{focus}}=1$) consistently degrades performance, demonstrating that the model benefits from causally linked predictions across multiple shots. Second, **longer context helps**: increasing context window size (while keeping margins) reliably improves performance, indicating that broader temporal evidence aids boundary disambiguation. Moreover, in line with our previous discussion, removing temporal margins induces edge degradation within a sequence, reinforcing the need for context padding to achieve position-invariant predictions.

Table 5. Window size ablation. Each column shows context & focus window sizes with corresponding F1 scores.

20 & 20	20 & 10	20 & 5	20 & 1		10 & 10	10 & 5	10 & 1		5 & 5	5 & 1
59.7	62.1	61.9	60.1		58.4	60.9	59.4		55.8	54.9

4.5.3. Number of Frames per Shot

We study the effect of frames-per-shot (K). While higher K can capture more intra-shot dynamics, it inflates token count. Tab. 6 shows results for $K \in \{1, 2, 3\}$. Performance improves modestly but consistently as K increases. The modest gains suggest that for scene segmentation, a single

Table 6. Frames per shot

K	F1	AP
1	61.8	65.3
2	61.9	65.2
3	62.1	66.8

Table 7. Model size

Params \uparrow	F1	AP
1.5B	55.9	58.7
3B	59.6	62.8
7B	62.1	66.8

representative frame suffices for most shots, though additional frames provide complementary evidence, presumably for shots with significant intra-shot motion or visual complexity. See further computational analysis in the Appendix.

4.5.4. Model Size

To understand how model capacity affects performance, we evaluate three model sizes - 1.5B, 3B, and 7B parameters. Tab. 7 shows consistent, monotonic improvements as model size increases: scaling from 1.5B to 3B yields gains of +3.7 F1 and +4.1 AP, while further scaling to 7B adds +2.5 F1 and +4.0 AP. Notably, the gains remain substantial even at the largest scale, suggesting that further scaling may continue to improve performance. These results provide empirical evidence that scene segmentation benefits from increased model capacity, consistent with broader scaling trends observed in vision-language models.

4.6. Adaptation to Video Chaptering

To assess generalization beyond cinematic scene segmentation, we evaluate on the related task of video chaptering [29, 33]. Unlike scene segmentation, this task targets web videos, where chapter boundaries are defined semantically and do not necessarily align with shot boundaries. The task also requires predicting *both* chapter boundary timestamps and a descriptive title for each chapter.

Experimental Setup. We adapt our framework with a minimal change: instead of emitting binary boundary labels, the model outputs boundary times with corresponding titles, following the format hh:mm:ss – Title [29]. We then evaluate on a subset of the VidChapters-7M corpus [29, 33], where content creators provide timestamped chapter titles. The primary baseline is Chapter-LLaMA [29], which uses a LLaMA-3.1 backbone [13]. To isolate methodology from backbone effects, we also compare against a variant of Chapter-LLaMA that uses a Qwen2.5-VL [4] backbone of comparable size, keeping the rest of the pipeline identical. We report F1, temporal IoU (tIoU), SODA, and CIDEr, jointly reflecting boundary accuracy, temporal alignment, and title quality.

Results and Analysis. As shown in Tab. 8, replacing the backbone in Chapter-LLaMA yields a notable performance drop, indicating that a direct backbone substitution is non-trivial. However, under the matched Qwen backbone, our method outperforms the adapted baseline across all metrics. Although the original Chapter-LLaMA with its native

Table 8. Results on Video Chaptering. Under matched backbones *Scene-VLM* outperforms Chapter-LLaMA across all metrics. Chapter-LLaMA with its original LLaMA backbone is listed in gray for reference.

Method	Backbone	F1↑	tIoU↑	SODA↑	CIDEr↑
Chapter-LLaMA	LLaMA 3.1-8B	42.6	70.6	16.4	82.4
Chapter-LLaMA	Qwen2.5-VL-7B	28.4	59.5	10.1	45.5
<i>Scene-VLM</i> (ours)	Qwen2.5-VL-7B	32.2	63.9	10.6	52.2

LLaMA backbone remains strongest in absolute terms, our framework demonstrates clear methodological gains when controlling for the underlying VLM.

We attribute these gains to fundamental differences in how visual information is processed. Chapter-LLaMA is a text-only approach: it first generates text captions from the visual signals, then processes these captions alongside speech transcripts. Our approach instead learns directly from raw visual frames jointly with transcripts, avoiding the intermediate captioning step. We believe this end-to-end multimodal grounding enables our model to discover which visual cues are truly predictive for boundary decisions, rather than relying on caption representations that may lose salient visual information.

4.7. Post-hoc Explainability for Scene Segmentation

Scene boundaries in practical workflows are often reviewed by human editors. Providing natural-language rationales alongside boundary proposals can greatly improve usability, allowing editors to judge whether the model’s rationale aligns with narrative intent. Yet, all prior methods for video scene segmentation offer no such capability. Leveraging the verbal capability of the VLM, we extend *Scene-VLM* to produce concise textual rationales for its boundary decisions. To our knowledge, this is the first demonstration of rationale generation for video scene segmentation.

Our initial approach was simple: we modified the prompt of our fine-tuned model (trained only on boundary detection) to request an explanation for each predicted boundary. However, this naive prompting strategy proved inadequate, as it led to frequent formatting errors and hallucinations, rendering the explanations unreliable for practical use.

Alignment via Minimal Supervision. Since prompting alone proved insufficient, we turned to explicit supervision. However, since no large-scale annotated explanation datasets exist for this task, we instead set to explore whether the model’s generation behavior could be *aligned* toward producing well-structured, grounded explanations using *minimal targeted supervision*. To test this, we curated a small set of 35 human-annotated examples, each pairing a boundary with a short rationale describing the narrative transition (e.g., location, time, characters, dialogue topic). An additional fine-tuning stage on this small set yielded an augmented model, **Scene-VLM + Explain**, capable of producing coherent, structured explanations. Fig. 6 shows this

Table 9. Explainability evaluation. Comparison of explanation quality between Scene-VLM and Scene-VLM + Explain on 30 randomly sampled transitions. The explanation-supervised variant eliminates all formatting errors and hallucinations.

Model	Parsing failures ↓	Hallucinations ↓
Scene-VLM	22 / 30	14 / 30
Scene-VLM + Explain	0 / 30	0 / 30

model’s capability. More examples are in the Appendix.



Figure 6. Scene transition explanation example. Scene-VLM + Explain proposes a boundary in *Lincoln* and provides a brief rationale grounded in visual changes, dialogue, and character presence.

Evaluation. To assess explanation quality, we conduct a small-scale quantitative probe on 30 randomly sampled transitions from four test movies. We evaluate two criteria: (i) **parsing failures**, whether the model produces well-formed, parseable output in the required format, and (ii) **hallucinations**, whether rationales contain factually incorrect details or off-task content. As shown in Tab. 9, the base *Scene-VLM* model (without explanation training) exhibits substantial failure rates. In contrast, Scene-VLM + Explain achieves zero failures on both metrics, demonstrating that minimal targeted supervision is sufficient to align the model toward reliable, well-structured explanations.

5. Conclusion

We introduced *Scene-VLM*, the first VLM-based framework for video scene segmentation. Our approach addresses key limitations of prior methods through a structured multimodal shot representation, sequential predictions based on a context-focus window design and the ability to generate natural-language rationales for boundary deci-

sions. We further propose a confidence prediction scheme that provides flexible precision-recall trade-offs, a capability typically reserved for encoder-based methods. *Scene-VLM* achieves state-of-the-art results on MovieNet and demonstrates strong generalization to BBC Planet Earth and to the video chaptering task. Looking ahead, we aim to leverage reinforcement learning to integrate explicit reasoning into scene predictions, transforming explainability into an integral part of the decision process that both improves accuracy and enhances model interpretability.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2
- [2] Meta AI. Llama 4: The beginning of a new era of natively multimodal ai innovation. Technical report, Meta AI, 2025. Accessed: 2025-11-04. 2
- [3] Anthropic. Claude opus 4 & claude sonnet 4: System card. Technical report, Anthropic, 2025. Accessed: 2025-11-04. 2
- [4] Shuai Bai, Kebin 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*, 2025. 3, 4, 7
- [5] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1199–1202, 2015. 1, 3, 4
- [6] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Shot and scene detection via hierarchical clustering for re-using broadcast video. In *International conference on computer analysis of images and patterns*, pages 801–811. Springer, 2015. 3
- [7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Icmi*, page 4, 2021. 5
- [8] John S Boreczky and Lawrence A Rowe. Comparison of video shot boundary detection techniques. *Journal of Electronic Imaging*, 5(2):122–128, 1996. 4
- [9] 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*, pages 9796–9805, 2021. 3, 5
- [10] Zhe Chen, Jiannan Wu, Wenhui Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 3
- [11] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1
- [12] Tianyu Fang, Hongbo Chen, Liang Ding, Lei Zhu, and Dacheng Tao. Video-llava: Learning video-language alignment with large language models. *arXiv preprint arXiv:2404.06395*, 2024. 3
- [13] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 7
- [14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 1
- [15] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European conference on computer vision*, pages 709–727. Springer, 2020. 1, 2, 3, 4
- [16] 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*, pages 18749–18758, 2023. 1, 3, 4, 5
- [17] Xin Jin, Hao Wang, Qing Zhang, Zhe Liu, Yehui Wang, Lei Ke, Jing Shi, Kai Chen, and Hang Li. Vidgpt: Video generation with frozen language models and adapter-tuning. *arXiv preprint arXiv:2401.12920*, 2024. 3
- [18] Xinyu Li, Qian Chen, Jin Zhang, Xiao Xu, Bohong Jiang, Yulun Zhang, and Zehuan Yuan. Videochat2: Overcoming time mismatches in video-language modeling. *arXiv preprint arXiv:2403.11438*, 2024. 3
- [19] Jonghwan Mun, Minchul Shin, Gunsoo Han, Sangho Lee, Seongsu Ha, Joonseok Lee, and Eun-Sol Kim. Boundary-aware self-supervised learning for video scene segmentation. *arXiv preprint arXiv:2201.05277*, 2022. 1, 3, 5
- [20] Ilan Naiman, Emanuel Ben-Baruch, Oron Anschel, Alon Shoshan, Igor Kviatkovsky, Manoj Aggarwal, and Gerard Medioni. Lv-mae: Learning long video representations through masked-embedding autoencoders, 2025. 3
- [21] Gautam Pal, Dwijen Rudrapaul, Suvojit Acharjee, Ruben Ray, Sayan Chakraborty, and Nilanjan Dey. Video shot boundary detection: a review. In *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2*, pages 119–127. Springer, 2015. 4
- [22] Y. Qin, J. Bai, Z. Duan, R. Zhang, et al. Qwen2.5-vl: A next-generation vision-language model with high-resolution understanding. In *arXiv preprint arXiv:2409.12345*, 2024. 2, 1
- [23] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023. 4
- [24] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020. 1
- [25] Anyi Rao, Lining 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*, pages 10146–10155, 2020. 1, 3, 5
- [26] 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*, pages 23331–23340, 2023. 1, 2, 3, 5
- [27] Gemini Team and Google DeepMind. Gemini 2.5: Pushing the frontier with advanced reasoning, long-context, and multimodal capabilities. Technical report, Google DeepMind, 2025. Accessed: 2025-11-04. 2
- [28] GLM-V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, et al. Glm-4.1v-thinking and glm-4.5v: Towards versatile multimodal reasoning with scalable reinforcement learning. *CoRR*, abs/2507.01006, 2025. Accessed: 2025-11-04. 2
- [29] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Güл Varol. Chapter-llama: Efficient chaptering in hour-long videos with llms. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18947–18958, 2025. 3, 4, 7, 1
- [30] 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*, pages 14021–14030, 2022. 1
- [31] Jialu Wu, Kevin Lin, Haotian Zhang, Junnan Li, and Steven C. H. Hoi. Video-llm: Modeling video with large language models. *arXiv preprint arXiv:2306.02858*, 2023. 3
- [32] Zhen Xue, Wei Li, Jun Zhang, Chengyu Wang, and Xiaodan Huang. Longvila: Scaling long-form video-language understanding with long-context transformers. *arXiv preprint arXiv:2402.11530*, 2024. 3
- [33] Antoine Yang, Arsha Nagrani, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vidchapters-7m: Video chapters at scale. *Advances in Neural Information Processing Systems*, 36: 49428–49444, 2023. 3, 4, 7, 1
- [34] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 3

Scene-VLM: Multimodal Video Scene Segmentation via Vision-Language Models

Supplementary Material

A. Additional Details

A.1. Method

A.1.1. Prompt Structure and Output Format

Fig. 8 illustrates our full prompt structure and the expected output format. The **gray** block contains the *system prompt*, the **green** block provides the *task instructions*, the **blue** block encodes the *per-shot multimodal inputs* (frames, subtitles, actor IDs) in an XML-style layout, and the **purple** block defines the *context-focus scope* (indicating which shots in the context window are also prediction targets). As depicted in the output block, the model generates `shot_id`: Yes/No decisions only for shots in the focus window. On average, a complete prompt contains 7939 words.

A.1.2. Visual Shot-ID Markers

To strengthen the correspondence between visual frames and their textual shot identifiers in the prompt, we overlay a small, high-contrast numerical tag at the top-left corner of each frame, as illustrated in Fig. 7. This visual marker serves two purposes: (1) it explicitly anchors each frame to its corresponding shot ID in the structured text input, reducing ambiguity during multimodal reasoning, and (2) it provides a consistent spatial reference that helps the model track shot boundaries across the temporal sequence. The marker is deliberately positioned to minimize occlusion of semantically relevant content while maintaining high visibility. As demonstrated in our ablation study (Sec. 4.5.1), these markers improve the final results.



Figure 7. **Visual shot-ID markers.** A compact numerical tag is overlaid at the top-left corner of each frame, tightly coupling visual content with the corresponding textual shot IDs referenced in the prompt while preserving semantically relevant content.

A.2. Setup

A.2.1. Datasets

Explainability. To enable aligning our model to generate post-hoc rationales for its predicted scene boundaries (Sec. 4.7), we curate a small supervision set of 35 samples pairing annotated scene boundaries with human-written ex-

planations. Each entry contains the `movie_id`, the inclusive `start_shot` and `end_shot` indices of the scene being concluded, and a free-text `rationale` describing the narrative transition. For example: “*There is a clear scene transition: the narrative shifts from an interview between two women about one woman’s past to a sequence showing her working as a maid and caring for a child, changing place, time, and situation.*”

A.2.2. Metrics

Scene segmentation. Following prior work [16, 26], we report the following standard detection metrics on the MovieNet [15] and BBC [5] datasets:

- **Average Precision (AP):** area under the precision-recall curve.
- **F1:** harmonic mean of precision and recall at an optimized operating point.

Video Chaptering. Following [29, 33], we evaluate chapter segmentation and titling using:

- **Chapter F1:** boundary-detection F1 computed against creator-provided chapter endpoints. Since endpoints are continuous timestamps, matching is based on temporal overlap and averaged over multiple overlap lengths; see [29] for details.
- **tIoU:** temporal Intersection-over-Union between predicted and ground-truth chapters, measuring temporal alignment; the exact protocol follows [29].
- **SODA and CIDEr:** metrics for evaluating semantic alignment of generated titles to ground-truth titles (see [33] for details).

A.2.3. Additional Implementation Details

Frame Processing. Our vision-language model [22] can accept a user defined image size. All frames across datasets are resized to 147×63 pixels. This resolution was selected to balance computational requirements with the preservation of visual details required for training and inference.

Training. All experiments were conducted on a cluster of $8 \times$ A100 (40 GB) GPUs. We fine-tune Qwen2.5-VL-7B on MovieNet ($\sim 29k$ samples) using LoRA [14] (rank 8, $\alpha=16$) for 4 epochs. With mixed precision, FlashAttention [11], and ZeRO-3 [24] sharding (data/optimizer/parameter partitioning), end-to-end fine-tuning on MovieNet-318 completes in approximately 2–4 hours, and on the chaptering task in around 1 hour, depending on I/O and kernel availability.

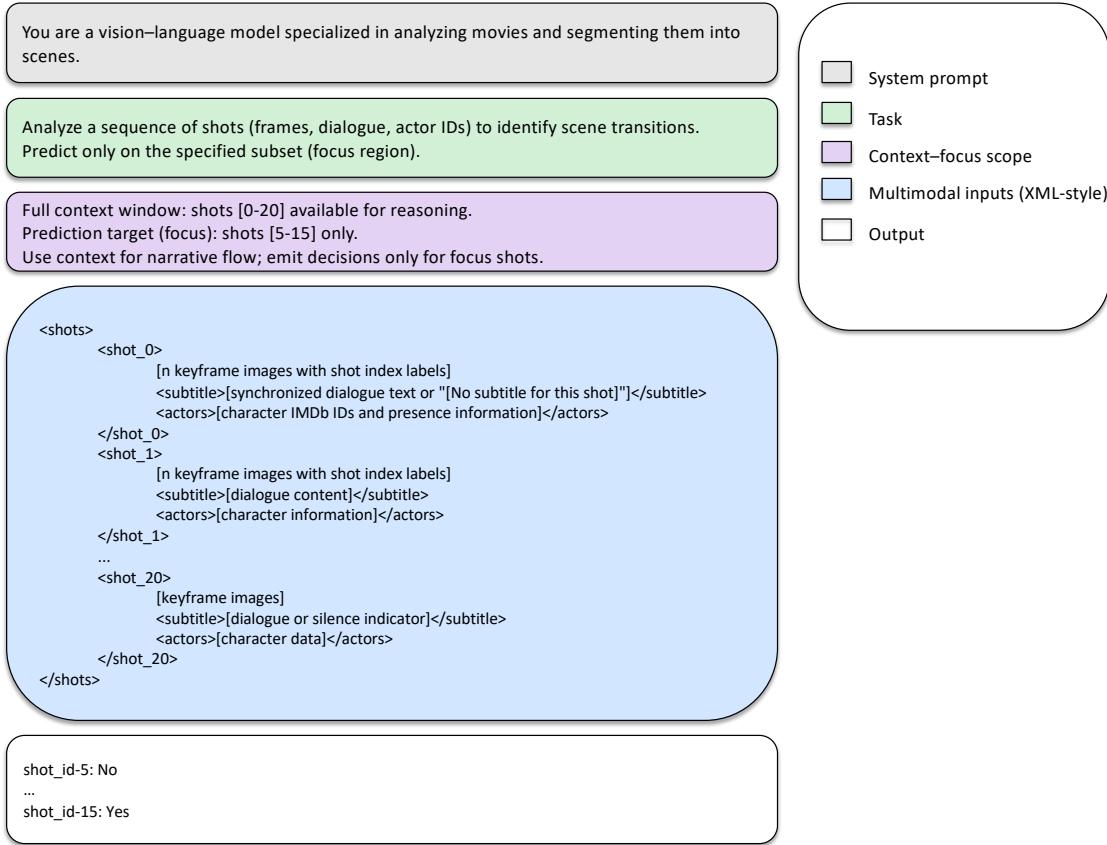


Figure 8. **Full prompt structure and output format.** Illustration of the multimodal prompt structure (instructions, visual frames, subtitles, and metadata) and the model’s structured output with shot-level boundary predictions.

Inference. Evaluation on the MovieNet test split takes approximately 1–2 hours on $8 \times A100$ (40GB) GPUs with data parallelism. Each movie is partitioned into non-overlapping context windows. We run batch-wise sequential decoding per window and then aggregate the outputs to compute the final metrics. This inference procedure applies to both tasks (scene segmentation and video chaptering).

B. Additional Experiments

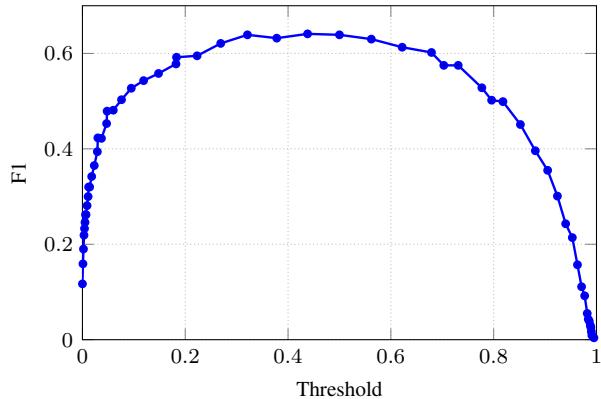
B.1. Alternative Prediction Schemes

Confidence scores are essential for scene segmentation, enabling flexible precision–recall trade-offs across different operating points. However, extracting reliable confidence from VLM outputs is non-trivial, as VLMs produce structured textual responses where confidence must be inferred from token-level probabilities rather than dedicated classification heads. In this section, we compare our proposed approach against two alternative prediction schemes, which differ in output format, confidence estimation capability, ac-

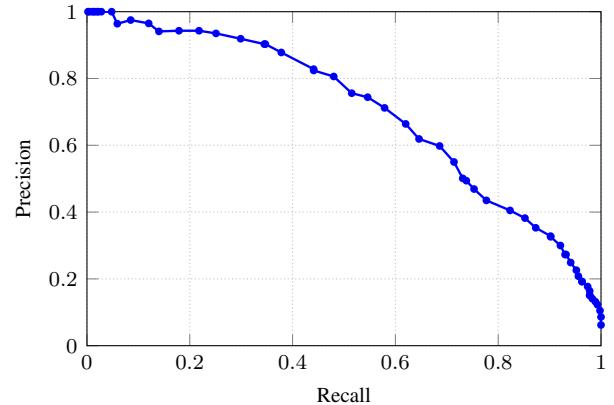
curacy, and computational cost.

Comprehensive scheme. Our proposed scheme (see Sec. 3.2) generates structured outputs in the format `shot_id:<id>: Yes/No` for each shot in the focus window, providing *explicit* predictions for all target shots. Confidence is then computed from the normalized token logits as described in Eq. (3). As shown in Tab. 10, this scheme achieves strong performance (F1: 62.1, AP: 66.8), but requires an average inference time of 87.2 seconds per movie.

Concise scheme. While our Comprehensive scheme is accurate, its inference latency may be prohibitive for latency-sensitive applications. To address this, we explore a more efficient output format where the model emits only Yes tokens for detected boundaries, omitting explicit No predictions for non-boundary shots. However, a subtle question then arises: can we extract confidence scores from this format by probing the Yes token logits? Unfortunately, this approach is fundamentally flawed due to the model’s



(a) **F1 versus decision threshold.** Peak $F1 = 0.641$ at threshold ≈ 0.438 . Broad plateau $\approx 0.62\text{--}0.64$ indicates stable operating range.



(b) **Precision-Recall curve.** Showing a balanced knee. E.g., at $t \approx 0.321$: $P \approx 0.60$, $R \approx 0.69$; at $t \approx 0.50$: $P \approx 0.71$, $R \approx 0.58$.

Figure 9. **F1 and Precision-Recall curves across decision thresholds.** Both curves demonstrate strong operating flexibility for task-dependent tuning.

output structure. When the model predicts a shot ID, it has already committed to marking that shot as a boundary, which means that the subsequent Yes token is obligatory rather than a genuine binary choice. Consequently, $p(\text{Yes} \mid \text{shot_id_predicted}) \approx 1$ by design, making the confidence score uninformative. This contrasts sharply with the Comprehensive scheme, where the Yes/No choice represents a genuine binary decision with meaningful probability mass on both outcomes.

Given this limitation, we evaluate the Concise scheme **without confidence extraction**, using the model’s outputs directly: a shot is classified as a boundary if and only if the model predicts Yes for it. Despite lacking precision-recall flexibility, this variant achieves competitive performance ($F1: 53.4$) with recent methods such as TranS4mer (48.4) and MEGA (55.3), while delivering dramatic speedup over the Comprehensive scheme (10.5s per movie, approximately $8.3\times$ faster). This makes it an attractive option when inference speed is critical and precision-recall control is not required.

Concise scheme with repeated sampling. Given the inability to extract meaningful confidence using the Concise scheme, we investigate whether repeated sampling can provide reliable confidence estimates. Specifically, we perform $m=5$ independent inference runs, draw temperatures uniformly from the interval $[0.5, 1.0]$ for each run, and compute confidence per shot as the proportion of Yes outcomes. Unfortunately, as shown in Tab. 10, this approach fails on both fronts: it is less accurate than Concise without confidence ($F1: 52.6$ vs. 53.4) and even slower than the Comprehensive scheme (105.2s vs. 87.2s), making it strictly worse than both alternatives.

To summarize, the **Comprehensive scheme** is recommended when accuracy and precision-recall control are

paramount, while the **Concise scheme (without confidence)** may be a good alternative when inference speed is the priority and precision-recall control is not strictly required.

Table 10. **Comparison of prediction schemes for scene segmentation.** The Comprehensive scheme achieves the best accuracy and features a confidence prediction capability, while the Concise scheme (without confidence) offers a compelling speed-accuracy trade-off for latency-critical applications.

Prediction Scheme	$F1 \uparrow$	$AP \uparrow$	Avg. time / movie (s) ↓
Concise (without confidence)	53.4	-	10.5
Concise (repeated sampling)	52.6	34.7	105.2
Comprehensive	62.1	66.8	87.2

B.2. Model F1 and Precision-Recall Analysis

To assess our method’s sensitivity to threshold changes, we plot F1 versus decision threshold (Fig. 9a) alongside the corresponding precision-recall curve (Fig. 9b). As depicted, the F1 curve rises sharply from near-zero thresholds and reaches a broad plateau, peaking at $F1 = 0.641$ around threshold 0.438. This plateau ($\approx 0.62\text{--}0.64$ F1 across thresholds $\sim 0.27\text{--}0.56$) demonstrates stable performance with minimal sensitivity to threshold variations. The PR curve exhibits the expected trade-off between precision and recall. For *balanced* operation, a threshold near 0.321 yields $P \approx 0.60$ and $R \approx 0.69$ ($F1 = 0.639$). For *precision-oriented* applications, a threshold near 0.50 yields $P \approx 0.71$ and $R \approx 0.58$ ($F1 = 0.639$). Conversely, recall-oriented scenarios can use lower thresholds (e.g., 0.223–0.269) to push recall above 0.70 with modest precision. To summarize, these curves demonstrate that flexible task-dependent tuning is possible while maintaining robust performance across the F1 plateau.

Table 11. **Computational analysis.** Peak memory, latency, and accuracy at 10 samples. “F” denotes frames per shot. Mean and standard deviation values are computed over five runs.

Method	Memory (GB) ↓	10-sample latency (s) ↓	F1 ↑	AP ↑
Scene-VLM (7B)-3F	18	2.34 ± 0.14	62.1	66.8
Scene-VLM (7B)-1F	16	1.84 ± 0.15	61.8	65.3
Scene-VLM (3B)-3F	9	1.35 ± 0.08	59.6	62.8
Scene-VLM (3B)-1F	7	1.15 ± 0.07	55.7	58.2

B.3. Explainability for Scene Segmentation (Cont.)

We present additional qualitative examples of model-generated rationales for scene-boundary decisions in Fig. 10 and Fig. 11. These examples span both abrupt visual transitions (e.g., title cards) and subtler, socially driven changes (e.g., shifts in location, time, or conversational structure).

B.4. Computational Analysis

In this section, we present a thorough analysis of our models’ computational requirements. We compare four variants, varying the number of frames per shot (1F or 3F) and model size (3B or 7B). For all models, we use a batch size of 1 to enable a fair comparison of memory and runtime. We do not apply model/system optimizations such as weight/activation quantization (e.g., 4-/8-bit) or inference engines with KV-cache optimizations (e.g., vLLM); these could further reduce both latency and memory in future work.

We report peak memory, latency, and accuracy at *10 samples* (i.e., 10 binary boundary decisions) in Tab. 11, using the same configuration as in the main experiments: a context window of 20 shots and a focus window of 10 shots. We repeat the evaluation five times, reporting the mean and standard deviation for wall-clock latency, and the maximum over runs for peak memory. As depicted, reducing frames per shot from 3F to 1F in the 7B model lowers latency from 2.34 s to 1.84 s ($\approx 21\%$) and peak memory from 18 GB to 16 GB, with only a minor accuracy drop (F1/AP: 62.1/66.8 \rightarrow 61.8/65.3); a similar trend holds for the 3B model. The latency reduction stems from the fact that frames dominate the token count of the input, so removing two of three frames per shot shortens the length of the multimodal input sequence and reduces computation. Meanwhile, the small accuracy drop aligns with our results from Sec. 4.5.3, which shows that performance degrades modestly when reducing the number of frames per shot. Intuitively, since shots are segments which typically contain no major visual changes, a single representative frame often preserves most scene-transition-relevant information.

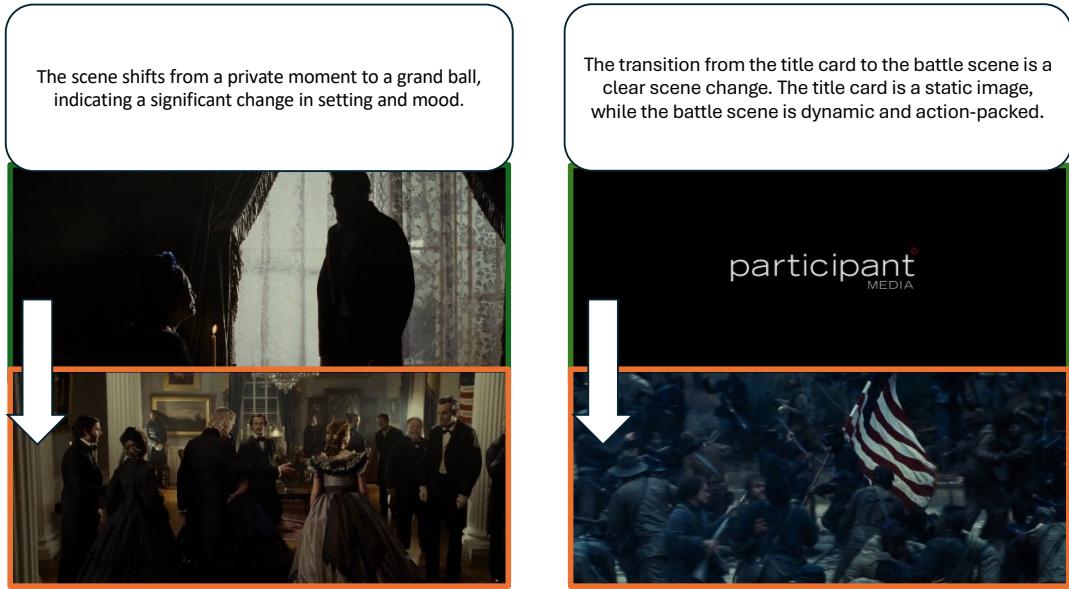


Figure 10. **Example 1.** *Left:* boundary due to a location change. *Right:* transition from a title card to the opening shot.

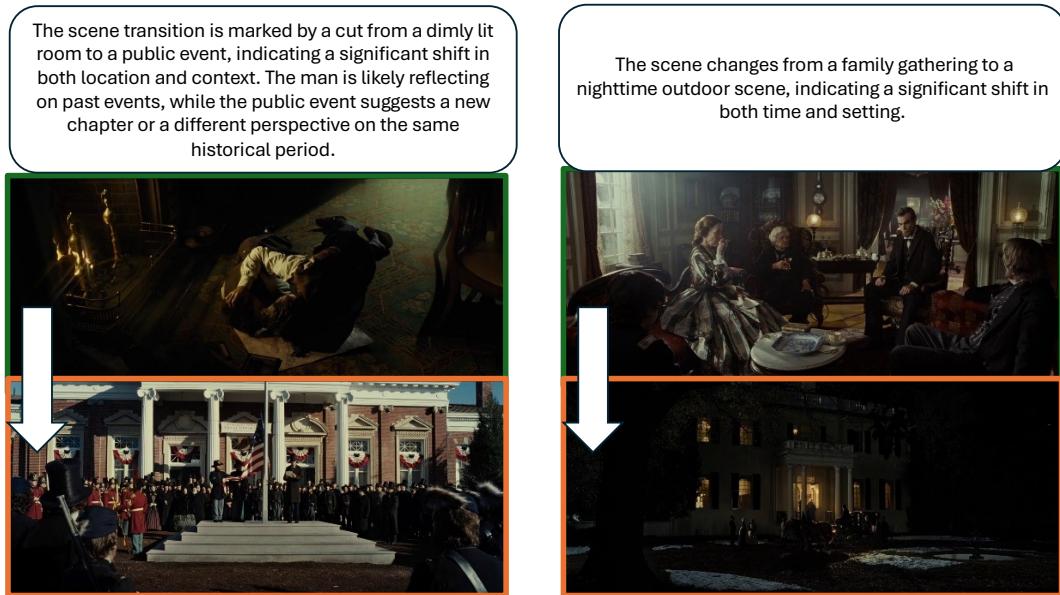


Figure 11. **Example 2.** *Left and right panels:* boundary justified by a joint change in *time* and *place*; the model references visual cues (lighting, background) and/or dialogue context to support the decision.