Research

TransVLM: A Vision-Language Framework and Benchmark for Detecting Any Shot Transitions

HeyGen Research, The University of Melbourne

Abstract

Traditional Shot Boundary Detection (SBD) inherently struggles with complex transitions by formulating the task around isolated cut points, frequently yielding corrupted video shots. We address this fundamental limitation by formalizing the Shot Transition Detection (STD) task. Rather than searching for ambiguous points, STD explicitly detects the continuous temporal segments of transitions. To tackle this, we propose TransVLM, a Vision-Language Model (VLM) framework for STD. Unlike regular VLMs that predominantly rely on spatial semantics and struggle with fine-grained inter-shot dynamics, our method explicitly injects optical flow as a critical motion prior at the input stage. Through a simple yet effective feature-fusion strategy, TransVLM directly processes concatenated color and motion representations, significantly enhancing its temporal awareness without incurring any additional visual token overhead on the language backbone. To overcome the severe class imbalance in public data, we design a scalable data engine to synthesize diverse transition videos for robust training, alongside a comprehensive benchmark for STD. Extensive experiments demonstrate that TransVLM achieves superior overall performance, outperforming traditional heuristic methods, specialized spatiotemporal networks, and top-tier VLMs.

Contents

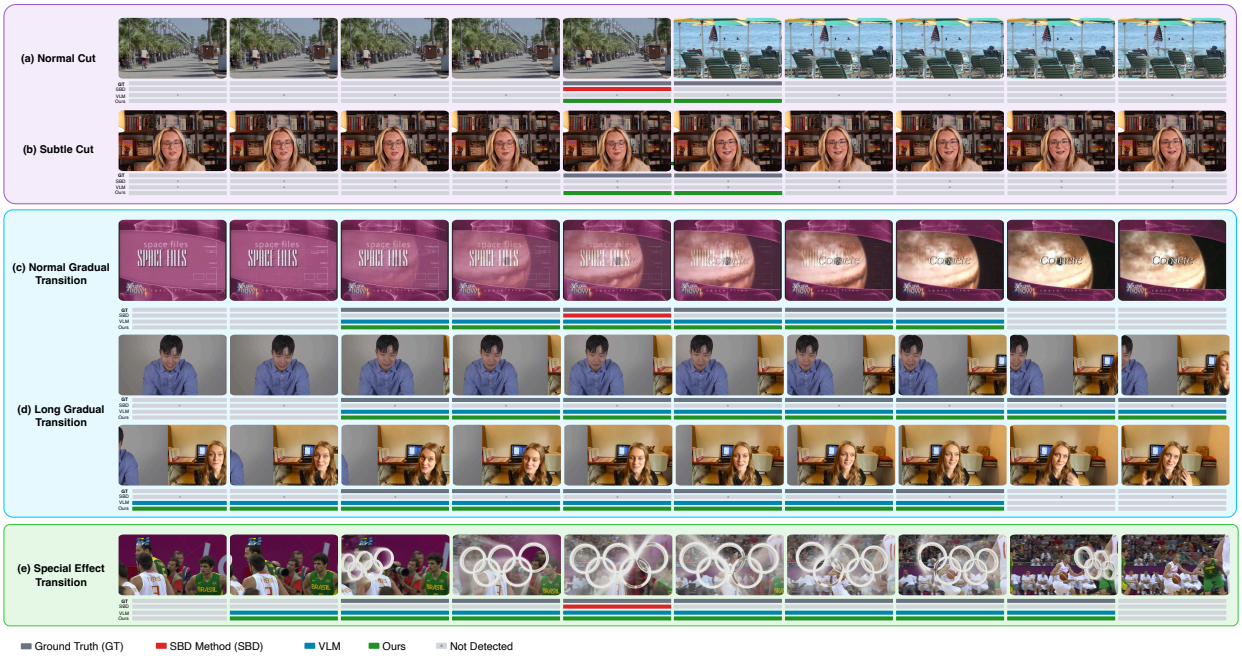

Figure 1 Limitations of existing shot transition detection methods. Predicted transitions are denoted by colored lines: gray for ground truth, **red** for a state-of-the-art SBD method (AutoShot [41]), **blue** for a general-purpose top-tier VLM (Qwen3-VL [3]), and **green** for our proposed TransVLM. While SBD models excel at detecting normal cuts (a) but fail on gradual and special transitions (c, d, e), VLMs can perceive gradual and special transitions but miss normal cuts. Neither of them can detect the subtle cuts (b). In contrast, TransVLM robustly detects all these transitions.

1 Introduction

Precisely detecting the start and end timestamps of shot transitions is a critical prerequisite for modern video processing. By identifying these accurate timestamps, models are shielded from the noise of inter-shot transitions, making this task a foundational pillar for a wide range of downstream applications. In video retrieval, processing clean shots strictly prevents the semantic blending of distinct scenes, thereby ensuring robust cross-modal matching performance [4, 18]. In video understanding tasks [15, 22, 30, 32, 34], such as action recognition and dense captioning, accurate shot transition timestamps guarantee that models learn coherent spatiotemporal dynamics without being disrupted by abrupt visual shifts. Most importantly, in the recent surge of text-to-video (T2V) generation [8, 9, 12, 17, 23, 37], detecting shot transitions is strictly necessary for data curation and label preparation. If the transition information in the training labels is insufficient or inaccurate, it actively causes generative models to produce unintended shot transitions in the generated videos.

To achieve this, two primary paradigms currently exist: traditional Shot Boundary Detection (SBD) methods and Vision-Language Models (VLMs). However, in practice, both exhibit critical flaws. We observe that while existing SBD methods perform adequately on normal abrupt cuts (Figure 1(a)), their performance degrades significantly when encountering complex transitions. For instance, when processing gradual transitions (e.g., dissolves, fades, and wipes), traditional SBD methods often fail to determine precise boundaries, resulting in extracted shots contaminated by dirty transitional frames (Figure 1(c)). Furthermore, these methods frequently miss subtle cuts (cuts with minor content changes), long gradual transitions, and special effect transitions (Figure 1(b, d, e)). Consequently, these corrupted shots severely degrade downstream processing. Conversely, our preliminary experiments reveal that while VLMs exhibit promising performance on complex gradual and special effect transitions (Figure 1(c, d, e)), they struggle significantly with abrupt cuts (Figure 1(a, b)).

To fundamentally resolve these limitations, we reformulate the traditional SBD paradigm and propose the Shot

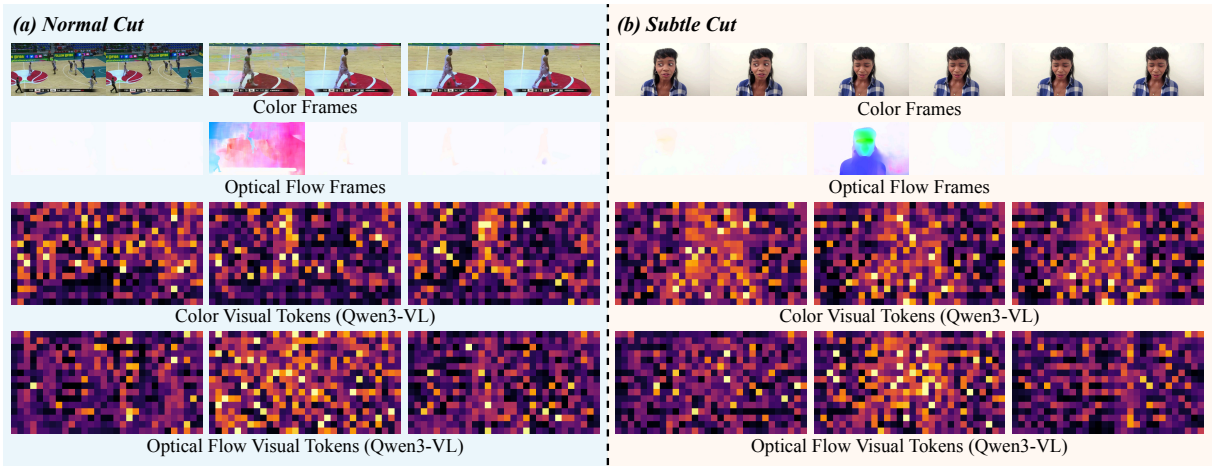

Figure 2 Visual token visualization for color vs. optical flow. By capturing inter-frame changes, optical flow serves as a robust motion prior. For both (a) normal and (b) subtle cuts, the visual tokens derived from optical flow exhibit significantly sharper response contrast at the transition frames compared to standard color tokens. Visualizations are extracted from Qwen3-VL [3] (temporal downsampling stride of 2).

Transition Detection (STD) task. Instead of focusing solely on content variations between adjacent frames like conventional SBD, STD emphasizes the intrinsic spatiotemporal patterns of the transitions connecting different shots. This includes capturing both content and motion dynamics across shots, rather than merely focusing on frame-to-frame content differences. Our detection targets logically shift from isolated cut points to continuous shot transition segments. To standardize evaluation in this newly formalized task, we construct a large-scale, high-quality STD benchmark with multi-dimensional metrics.

To address the respective defects of SBD methods and VLMs on the STD task, we propose the TransVLM framework. We attribute the suboptimal VLM performance on normal cuts to the sparse, low-frame-rate inputs (e.g., 5 FPS) conventionally required to prevent out-of-memory errors and severe inference latency. Such aggressive temporal downsampling makes detecting instantaneous cuts highly challenging. To overcome this, we introduce a temporal sliding-window strategy. By partitioning the video stream into fixed-length, overlapping segments for sequential inference and subsequently merging the local outputs into continuous global predictions, we enable the VLM to reliably focus on and detect fast-changing cuts without exceeding memory constraints.

Furthermore, for subtle cuts, the failure stems from the VLM's inherent bias toward static spatial semantics rather than fine-grained inter-shot motion dynamics. We discover that explicitly providing optical flow is exceptionally effective at capturing these missing motion priors. As illustrated in Figure 2, for both normal and subtle cuts, the temporal variations at the exact transition frames—evident in both the visual frames and the final visual tokens processed by the language model—are significantly more distinguishable in the optical flow modality than in standard color frames. To seamlessly inject this optical flow into the network without inflating the computational burden, we design a simple yet effective feature-fusion strategy, which integrates motion representations while strictly maintaining the same sequence length of visual tokens. Finally, to mitigate the critical scarcity of annotated transitions, our specially designed data engine is leveraged to automatically synthesize massive, high-quality training data.

Our main contributions are summarized as follows:

- We reformulate the traditional SBD paradigm and propose the novel Shot Transition Detection (STD) task, alongside a comprehensive benchmark equipped with multi-dimensional metrics specifically tailored for STD.
- We propose TransVLM, an efficient vision-language framework that explicitly integrates optical flow as a motion prior. Our feature-fusion strategy significantly enhances the capability of detecting abrupt

cuts without incurring any additional visual token overhead.

- We design a scalable data engine capable of automatically synthesizing diverse transition videos, providing high-quality supervision and enabling robust VLM training for the STD task.
- Extensive experiments demonstrate that TransVLM achieves superior overall performance on the proposed STD benchmark, substantially outperforming traditional heuristic methods, specialized spatiotemporal networks, and top-tier general-purpose VLMs.

2 Related Work

Shot Boundary Detection. Traditional SBD approaches [1, 14] rely on heuristic algorithms (e.g., PySceneDetect [11], ECR [36]) that compute low-level visual differences. While computationally inexpensive, they are highly sensitive to local illumination changes and fast motion, leading to frequent false negatives. Recent deep learning architectures [29, 33, 35], such as the TransNet series [24, 25] and AutoShot [41], utilize Convolutional Neural Networks (CNNs) to significantly improve detection performance. However, by formulating the task as a frame-wise binary classification, these methods exclusively target isolated cut points. Consequently, their discrete probability outputs falter on complex gradual transitions, frequently yielding corrupted shots containing dirty frames.

Video Understanding. Recent advancements in video understanding have evolved from specialized spatiotemporal architectures (e.g., I3D [10], SlowFast [13], and VideoMAE [30]) to powerful general-purpose Vision-Language Models (VLMs), such as Video-LLaVA [16], the Qwen-VL series [2, 31], and Gemini [27]. While their robust cognitive capabilities are theoretically well-suited for modeling content and motion variations across shots, directly applying off-the-shelf VLMs to the STD task yields imprecise transition segments. This failure stems from two core limitations. First, VLMs inherently prioritize static spatial semantics over fine-grained inter-shot dynamics. Second, their reliance on sparse, low-frame-rate inputs to prevent memory overload critically restricts their ability to detect instantaneous cuts.

3 Task Formulation

Given a video V consisting of N frames, traditional SBD approaches formulate the task as a frame-wise binary classification problem. The output is typically represented as a sequence of frame-level probabilities, $\mathcal{P}_{frame} = \{p_1, p_2, \ldots, p_N\}$, where each p_i indicates the likelihood of the i-th frame being a shot boundary. Boundaries are subsequently extracted by applying a predefined heuristic threshold. However, these SBD models predominantly focus on detecting isolated cut points while entirely neglecting the intrinsic spatiotemporal patterns of the transitions. Consequently, when processing complex transitions (e.g., dissolves and wipes), the predicted probabilities for transitional frames often lack salient contrast compared to intra-shot frames. This ambiguity makes it nearly impossible to establish a robust threshold capable of cleanly separating transition segments from pure video shots.

In contrast, we formally define our proposed STD as a transition detection task. Rather than assigning independent probability scores to individual frames, STD requires the model to explicitly detect the complete transition segments, thereby holistically capturing both content and motion dynamics across shots. Each transition segment is formally defined as a temporal tuple comprising its precise start and end times. Thus, the prediction is formulated as a set of discrete temporal segments, $\mathcal{P}_{segment} = \{(s_1, e_1), (s_2, e_2), \ldots, (s_M, e_M)\}$, where s_i and e_i denote the start and end timestamps of the i-th transition, respectively. This representation naturally unifies the definition of abrupt cuts (where $s_i \approx e_i$) and gradual or special effect transitions (where $s_i < e_i$).

Crucially, this segment-level formulation offers two distinct advantages. First, the tuple-based format can be effortlessly serialized into natural language tokens, seamlessly aligning the STD task with VLMs. Second, by explicitly forcing the model to predict the continuous temporal span of a transition between shots rather than isolated cut points of a shot, this formulation serves as a strong inductive bias. It actively encourages the model to capture the intrinsic spatiotemporal patterns of the transition dynamics themselves, rather than over-fitting to ambiguous boundary frames.

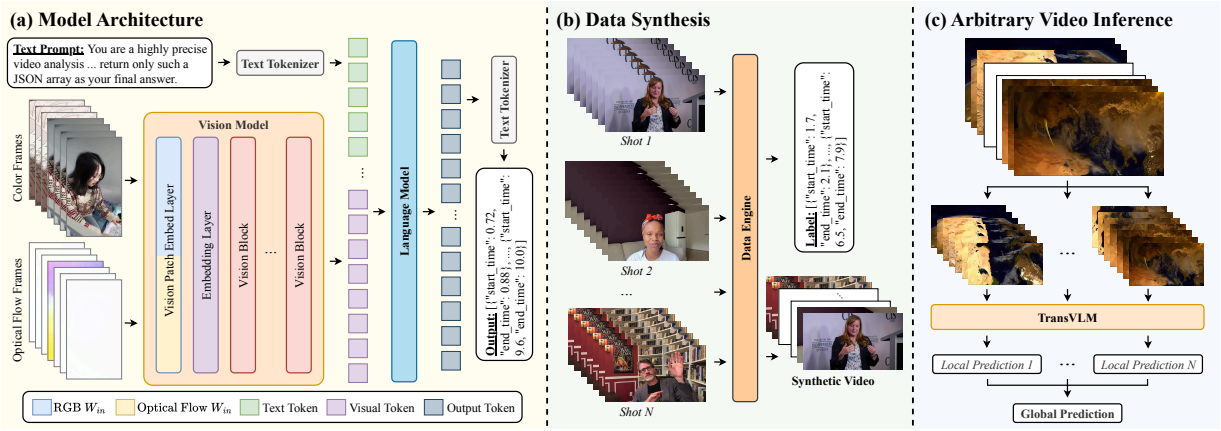

Figure 3 TransVLM Framework. Our proposed framework comprises three core components. **(a) Model Architecture:** We explicitly inject optical flow as a motion prior via a parameter-efficient strategy. By exclusively expanding the input projection weights (W_{in}) of the Vision Patch Embed Layer, the model directly processes concatenated frames of color and optical flow. Crucially, this extracts joint appearance-motion representations without inflating the visual token sequence length, thereby incurring zero additional computational burden on the language model. **(b) Data Synthesis:** Given an arbitrary sequence of clean shots, our scalable data engine automatically synthesizes videos with diverse transitions, simultaneously generating precisely aligned segment-level JSON labels for model training. **(c) Arbitrary Video Inference:** To process videos of unconstrained duration without causing memory overflow, we employ a temporal sliding-window strategy. The input stream is partitioned into overlapping segments to generate local predictions, which are subsequently aggregated into a continuous global prediction via temporal Non-Maximum Suppression (NMS).

4 TransVLM Framework

In this section, we detail the proposed TransVLM framework for STD task. We first introduce the overall model architecture, specifically focusing on the feature-fusion mechanism for integrating optical flow and the zero-padding weight initialization strategy (Section 4.1). Subsequently, we elaborate on the data engine, the quality-aware mixed sampling strategy, and the arbitrary video inference pipeline (Section 4.2).

4.1 Model Architecture

Existing Vision-Language Models (VLMs) generally exhibit suboptimal performance on the STD task. To overcome these limitations, we propose TransVLM, a novel multimodal framework tailored for precise temporal transition detection. TransVLM explicitly integrates motion priors via a parameter-efficient feature-fusion mechanism without inflating the visual token sequence length.

While VLMs possess the robust cognitive capacity theoretically required for complex video tasks, their pre-trained vision encoders are inherently biased towards high-level semantics and shot content. However, fine-grained inter-shot motion changes are crucial for the STD task. To leverage this information, we explicitly inject optical flow as a motion prior into the VLM. Since Qwen3-VL [3] is one of the top-tier open-source VLMs, we employ it as our base backbone for TransVLM. Theoretically, our modifications are model-agnostic and can be applied to any VLM architecture.

As depicted in Figure 3(a), the input to TransVLM consists of both standard color frames and optical flow frames. These two modalities are fused at the input stage and processed by the modified vision model. The resulting visual tokens, alongside the tokenized text prompt instructing the model to output transition segments as a structured JSON array, are fed into the language model to predict the exact start and end timestamps.

Feature-Fusion for Motion Prior. To inject a strong motion prior into the model, we explicitly extract optical flow between consecutive frames. For an optimal balance between computational efficiency and estimation quality, we utilize NeuFlow v2 [38–40] as our optical flow extractor. To align the spatial distribution of the

Table 1 Details of the TransVLM Training Dataset. Our comprehensive training dataset comprises existing public data (row 1-4), high-fidelity manual annotated data (row 5), highly scalable synthesized data (row 6) and all training data (row 7).

Dataset	Domain	Label Quality	Total Videos		Total Transitions		Transition Types		
			Count	Dur. (h)	Count	Dur. (s)	Cut	Normal	Long
MovieShots2 (Train) [21]	Movies	Very High	5,222	226.17	152,191	6,347.6	152,191	0	0
SportsShot (Train) [20]	Sports	Very High	720	27.87	13,295	2,731.2	9,930	2,958	407
ClipShots (Train) [26]	Web Videos	High	5,360	319.42	158,825	24,931.9	129,544	24,727	4,554
AutoShot (Train) [41]	Short Videos	Medium	654	7.36	5,275	425.1	4,606	647	22
Internal Data	Web Videos	Very High	2,254	23.00	7,765	5,633.3	918	5,905	942
Synthetic Data	Web Videos	Very High	218,993	959.92	353,198	604,068.9	40,487	112,158	200,553
STD Training Data	**Diverse**	**Mixed**	**233,203**	**1,563.74**	**690,549**	**644,138.0**	**337,676**	**146,395**	**206,478**

motion data with the pre-trained visual distribution expected by the VLM, the raw two-dimensional optical flow fields (d_x, d_y) are mapped into a standardized three-channel color visualization format [5], treating the flow magnitude and orientation as color saturation and hue.

Let $\mathbf{I}_{\text{color}} \in \mathbb{R}^{H\times W\times 3}$ denote the original video frame and $\mathbf{I}_{\text{Flow}} \in \mathbb{R}^{H\times W\times 3}$ denote the corresponding visualized optical flow frame, where H and W represent the spatial height and width, respectively. We propose a data-level feature-fusion strategy by concatenating these modalities strictly along the channel dimension:

$$\mathbf{I}_{\text{Fused}} = \text{Concat}(\mathbf{I}_{\text{color}}, \mathbf{I}_{\text{Flow}}) \in \mathbb{R}^{H\times W\times 6}. \tag{1}$$

To process this fused tensor, we only modify the first layer of the vision model (i.e., the Vision Patch Embedding layer in Qwen3-VL), symmetrically expanding its input channels from 3 to 6. Crucially, because the output channel dimension of this embedding layer remains unchanged, the resulting sequence length of the visual tokens is strictly identical to that of a standard 3-channel input. This simple yet highly effective design ensures that the integration of the motion prior introduces zero additional computational overhead to the subsequent vision blocks and the language model.

Weight Initialization for Stable Fine-tuning. Directly expanding the input channels of the pre-trained Vision Patch Embedding layer invalidates its original weight tensor dimensions, shifting from $(C, 3, D_k, H_k, W_k)$ to $(C, 6, D_k, H_k, W_k)$, where D_k, H_k, and W_k denote the depth, height, and width of the 3D convolutional kernel, respectively. A naive random initialization or uniform scaling of the newly added optical flow channels risks generating massive noisy gradients during early training epochs, potentially leading to catastrophic forgetting of the highly optimized spatial representations learned during the VLM's pre-training phase.

To guarantee optimization stability, we adopt a strict zero-padding initialization strategy. Let $\mathbf{W}_{\text{in}} \in \mathbb{R}^{C\times 3\times D_k\times H_k\times W_k}$ denote the pre-trained weights of the original 3-channel convolutional layer. The extended weights $\mathbf{W}_{\text{in}}^{'}$ are initialized as follows:

$$\mathbf{W}_{\text{in}}^{'} = \text{Concat}(\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{in}}^{0}) \in \mathbb{R}^{C\times 6\times D_k\times H_k\times W_k}, \tag{2}$$

where $\mathbf{W}_{\text{in}}^{0}$ is a zero-initialized tensor of the exact same shape as $\mathbf{W}_{\text{in}}$. This ensures that at the first training step, the model behaves identically to the pre-trained VLM, allowing it to progressively and safely learn to attend to the newly introduced motion prior without disrupting existing spatial knowledge.

4.2 Model Training and Inference

To effectively optimize the TransVLM for the continuous transition detection task, we establish a robust training and inference pipeline designed to overcome inherent data and memory constraints.

Data Synthesis and Quality-Aware Sampling. Training a VLM requires massive diverse video data with high-quality segment-level annotations. Because relying solely on legacy public datasets is insufficient due to their inherent noise and label ambiguity, we construct a scalable data engine (Figure 3(b)) that automatically synthesizes continuous videos with random transitions and precise start/end labels. To ensure robust generalization, we

Table 2 Details of proposed STD Benchmark. Unlike SBD datasets that provide ambiguous shot boundaries, our benchmark (row 8) re-annotates existing public datasets (row 1-6) and introduces synthetic data (row 7) to evaluate different types of transitions. This equips all 5,215 videos with **segment-level** transition labels. Transition types are categorized by duration: Cut (< 0.1s), Normal (≤ 1s), and Long (> 1s).

Dataset	Domain	Original Label	Total Videos		Total Transitions		Transition Types		
			Count	Dur. (h)	Count	Dur. (s)	Cut	Normal	Long
RAI [7]	TV Shows	Point	10	1.64	1,036	304.4	757	188	91
BBC [6]	Documentaries	Point	11	9.00	4,943	703.7	4,255	582	106
AutoShot (Test) [41]	Short Videos	Point	200	2.01	2,065	545.0	1,008	1,004	53
ClipShots (Test) [26]	Web Videos	Point	500	32.85	6,923	1,548.1	4,798	1,830	295
MovieShots2 (Test) [21]	Movies	Point	282	20.72	14,767	9,566.4	13,436	710	621
SportsShot (Val) [20]	Sports	Point	240	9.37	5,045	944.4	3,899	1,064	82
STD Synthesis Data	Web Videos	Segment	3,972	24.67	10,460	18,249.3	3,593	1,615	5,252
STD Benchmark	**Diverse**	**Segment**	**5,215**	**100.26**	**45,239**	**31,861.3**	**31,746**	**6,993**	**6,500**

aggregate this synthetic data with reformatted public datasets. Crucially, to prevent inconsistent, noisy public labels from dominating gradient updates, we implement a quality-aware mixed sampling strategy. During training, the probability of sampling a batch from a specific dataset is strictly weighted by its manually assessed quality tier, ensuring stable optimization.

Arbitrary Video Inference. Because the model is explicitly trained on short video clips, directly inputting arbitrarily long videos would cause out-of-distribution errors and severe memory overflow. To maintain training-inference consistency, we employ a temporal sliding-window strategy. The video stream is partitioned into overlapping temporal windows, generating local segment-level predictions for each window. Finally, we resolve redundant or conflicting transition predictions within the overlapping regions by merging them via Non-Maximum Suppression (NMS), seamlessly yielding a robust, continuous global output.

5 Benchmark

To standardize the evaluation of the newly proposed Shot Transition Detection (STD) task, we establish a comprehensive benchmark comprising customized a large-scale, high-quality test dataset and multi-dimensional evaluation metrics.

5.1 Benchmark Construction

Existing public datasets primarily provide noisy annotations based on isolated cut points, which are fundamentally inadequate for the segment-level requirements of the STD task. Therefore, we conducted a massive manual re-annotation process. Expert annotators meticulously reviewed the original videos and corrected the exact start and end boundaries across multiple public datasets, strictly eliminating boundary ambiguity and annotation noise.

Combined with our scalable data engine, we constructed a comprehensive evaluation benchmark encompassing 5,215 videos (approximately 100.3 hours) from diverse domains. As summarized in Table 2, the benchmark contains 45,239 transitions. To rigorously evaluate performance across different dynamics, we categorize them by duration: abrupt cuts (< 0.1s), normal transitions (≤ 1s), and long transitions (> 1s). This accurate, segment-level ground-truth data enables robust generalization evaluation for STD models.

5.2 Evaluation Metrics

Traditional Shot Boundary Detection (SBD) predominantly evaluates isolated cut points independently. We argue that this point-centric evaluation is fundamentally incompatible with the STD transition detection task, which strictly requires detecting the continuous temporal span of a transition as a unified tuple.

To this end, we propose a rigorous evaluation suite comprising segment-level and frame-level metrics. Recognizing the inherent subjective ambiguity when human annotators define the exact boundaries of gradual

transitions, we introduce a temporal tolerance τ (ranging from 0.0 to 0.5 seconds) to expand the boundaries of ground-truth and predicted segments. We evaluate the metrics across various τ values and calculate their mean, analogous to mean Average Precision (mAP).

Segment-Level F_1. This metric evaluates model's instance retrieval capability. A predicted segment matches a ground-truth segment if their temporal intersection is strictly positive. Overlapping predictions are mathematically resolved via a Greedy Matching algorithm prioritizing the largest intersection duration.

Frame-Level F_1. While segment-level metrics treat all transitions equally, frame-level metrics explicitly measure the temporal coverage fidelity in the continuous time domain. Since our model outputs the start and end timestamps of a transition, we multiply these predictions by the frames per second (FPS) to obtain the exact frames for evaluation. This metric severely penalizes models that correctly detect a transition's existence but drastically misjudge its temporal span.

Absolute Boundary Error (ABE). To purely quantify the detection precision of the boundaries, we calculate the ABE for all successfully matched segment pairs. It measures the average absolute temporal offset (in seconds) between the predicted and ground-truth boundaries.

Real-Time Factor (RTF). To evaluate computational efficiency, we employ the RTF metric, defined as the ratio of the total inference time to the total duration of the processed video. An RTF strictly less than 1 indicates faster-than-real-time processing capabilities.

6 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of the proposed TransVLM. Leveraging the newly established STD benchmark, we comprehensively compare our model against three distinct paradigms of baselines: traditional heuristic approaches [11], specialized deep learning networks [24, 41], and top-tier open- and closed-source Vision-Language Models (VLMs) [2, 27]. Empirical results consistently demonstrate the superiority of TransVLM on the continuous transition detection task.

6.1 Experimental Setup

Implementation Details. To strike an optimal balance between detection performance and inference speed, we build TransVLM upon the pre-trained Qwen3-VL [3] 4B Instruct model. As demonstrated in Table 3, relying on larger or "Thinking" model variants severely degrades inference speed without proportional performance gains. Furthermore, because the STD task primarily demands temporal pattern recognition rather than complex reasoning, massive parameter counts are unnecessary. Conversely, the 2B Instruct model lacks the fundamental capacity for this task (consistently generating meaningless string outputs rather than valid temporal segments). Therefore, the 4B Instruct model serves as the ideal foundational backbone.

Optical flow representations are extracted using the NeuFlow v2 [38–40] architecture. The modified Vision Patch Embedding layer is initialized strictly following the zero-padding weight initialization strategy detailed in Section 4.1. We optimize the model using the recently open-sourced VeOmni [19] training framework. The network is optimized for 5,000 steps using the AdamW optimizer with a peak learning rate of 1.0×10^{-5} and a cosine learning rate decay schedule. The per-device batch size is set to 4. All training experiments are conducted across 8 NVIDIA H100 (80GB) GPUs, yielding an effective global batch size of 32. For the sliding-window inference, we strictly adhere to a window size of 10 seconds with a temporal stride of 9 seconds. Powered by FFmpeg [28], our engine supports the generation of 59 distinct transition effects (see Appendix S2 for details). For the quality-aware mixed sampling strategy, the sampling probabilities for Very High, High, and Medium tiers are 0.7, 0.2, and 0.1, respectively.

Baselines and Fair Comparison. We benchmark TransVLM against three distinct categories of representative methods: (1) *Heuristic algorithms*, represented by the widely adopted PySceneDetect [11]; (2) *Specialized deep learning networks*, including the current state-of-the-art SBD models TransNetV2 [24] and AutoShot [41]; and (3) *General-purpose foundational VLMs*, specifically evaluating the Qwen3-VL [3] and Gemini [27] series. To guarantee a strictly fair and rigorous comparison, the predictions generated by SBD baselines are systematically converted to transition segments through simple temporal subtraction. All comprehensive

Table 3 Quantitative comparison on the STD Benchmark. We report both mean segment-level and frame-level metrics (Precision, Recall, and F_1) across the defined temporal tolerance values to comprehensively evaluate the detection performance. Performance is measured by the Absolute Boundary Error (ABE) in seconds. Inference efficiency is reported via the Real-Time Factor (RTF). The evaluations are conducted on both public and synthetic datasets. The best results are highlighted in **bold**, and the second-best are underlined. Our proposed TransVLM outperforms state-of-the-art SBD methods and general VLMs. P means Precision and R means Recall.

	Public Data							Synthetic Data							
	Segment			Frame				Segment			Frame				
Methods	P	R	F1	P	R	F1	ABE	P	R	F1	P	R	F1	ABE	RTF
PySceneDetect [11]															
- Adaptive	0.656	0.716	0.684	0.645	0.372	0.457	1.93	**0.924**	0.172	0.290	0.922	0.036	0.069	0.28	0.09
- Content	0.627	0.759	0.686	0.603	0.383	0.453	1.08	0.909	0.128	0.225	**0.968**	0.029	0.056	0.61	0.09
- Hash	0.566	0.778	0.654	0.549	0.416	0.457	2.11	0.880	0.118	0.208	0.940	0.027	0.052	0.85	0.09
- Hist	0.452	0.741	0.559	0.419	0.398	0.394	1.19	0.577	0.379	0.455	0.762	0.117	0.197	1.04	0.09
- Threshold	0.364	0.031	0.057	0.306	0.014	0.027	3.08	0.773	0.039	0.074	0.749	0.008	0.016	1.46	0.09
TransNetV2 [24]	0.727	0.780	0.752	**0.731**	0.427	0.528	1.87	0.275	0.149	0.194	0.417	0.034	0.063	0.61	0.07
AutoShot [41]	0.707	0.804	0.751	0.709	0.440	0.532	1.78	0.379	0.248	0.299	0.530	0.058	0.102	0.51	**0.03**
Gemini Series [27]															
- 2.5 Pro	0.558	0.527	0.542	0.453	0.361	0.401	3.62	0.338	0.851	0.465	0.638	0.760	0.686	0.82	0.81
- 3 Pro	0.527	0.573	0.549	0.469	0.343	0.393	2.18	0.479	0.768	0.588	0.711	0.482	0.568	0.88	1.32
Qwen3-VL Instruct Series [3]															
- 4B	0.235	0.088	0.124	0.134	0.174	0.148	55.00	0.717	0.306	0.428	0.458	0.618	0.525	8.88	0.31
- 8B	0.222	0.297	0.246	0.132	0.307	0.184	34.39	0.586	0.597	0.591	0.741	0.204	0.315	1.60	0.34
- 32B	0.309	0.473	0.370	0.214	0.279	0.241	3.96	0.895	0.623	0.735	0.911	0.361	0.516	1.35	0.98
- 30B-A3B(MoE)	0.218	0.300	0.242	0.171	0.222	0.192	9.61	0.806	0.593	0.683	0.749	0.355	0.482	1.86	0.35
Qwen3-VL Thinking Series [3]															
- 4B	0.449	0.079	0.134	0.265	0.051	0.086	1.50	0.839	0.218	0.346	0.748	0.087	0.156	1.68	1.03
- 8B	0.450	0.144	0.217	0.297	0.094	0.143	4.25	0.854	0.389	0.534	0.923	0.147	0.252	1.29	1.31
- 32B	0.403	0.260	0.315	0.308	0.160	0.209	1.09	0.900	0.608	0.726	0.943	0.323	0.479	1.16	3.30
- 30B-A3B(MoE)	0.374	0.217	0.275	0.291	0.127	0.175	**0.98**	0.890	0.610	0.724	0.915	0.331	0.485	1.18	0.92
TransVLM(Ours)	**0.762**	**0.806**	**0.783**	0.574	**0.562**	**0.568**	1.58	0.908	**0.882**	**0.895**	0.946	**0.930**	**0.938**	**0.11**	0.50

evaluations are standardized on our newly established STD benchmark. To balance fairness and evaluation efficiency, we set the video FPS to 5 when evaluating general VLMs.

6.2 Quantitative Experiments

The comprehensive quantitative results evaluating TransVLM against SBD approaches and top-tier foundational VLMs are summarized in Table 3. We follow the official calling guidance for each method. Overall, our proposed framework establishes a new state-of-the-art across both public and synthetic datasets, consistently dominating the primary F_1 metrics. See Appendix S5 for details.

Performance on Public Data. On the public datasets, TransVLM significantly outperforms all baselines in accurately detecting transition instances and their temporal coverage. Specifically, it achieves the highest segment-level F_1 (78.3%) and frame-level F_1 (56.8%), surpassing highly specialized deep learning models such as AutoShot (75.1% and 53.2%, respectively). Our Absolute Boundary Error (ABE) of 1.58s is highly competitive. This marginal variance from the absolute lowest score is largely attributable to the inherent limitations of public training datasets, which severely suffer from noisy, subjective, and ambiguous human annotations (detailed examples and discussions are provided in the Appendix S4). Consequently, these subjective inconsistencies inevitably cap the measurable upper bound of precise boundary detection on public test sets.

Notably, on public data, traditional SBD methods significantly outperform general VLMs. Because transitions in these public datasets are predominantly abrupt cuts, this observation corroborates our hypothesis regarding VLM deficiencies: VLMs possess insufficient perception of fine-grained inter-shot motion changes, and their reliance on sparse, low-frame-rate inputs severely restricts their ability to detect instantaneous abrupt cuts.

Table 4 Ablation Study of TransVLM. The ablations include: (1) *Training*: identifying the importance of our training process; (2) *Data Composition*: isolating the impact of our data engine by removing either public or synthetic data; (3) *Motion Prior*: removing the optical flow input; (4) *Fusion Strategy*: substituting our feature fusion strategy by inputting color and optical flow frames separately (*w/o feature fusion*), which drastically increases RTF; and (5) *Initialization*: replacing our zero-padding initialization with a naive channel duplication (*duplicate weight*). The results demonstrate that our full TransVLM configuration achieves the optimal balance between detection performance (highest overall F1) and inference efficiency (RTF). Best results are in **bold** and second-best are underlined. P means Precision and R means Recall.

	Real-world Data							Synthetic Data							
	Segment			Frame				Segment			Frame				
Methods	P	R	F1	P	R	F1	ABE	P	R	F1	P	R	F1	ABE	RTF
w/o training	0.545	0.448	0.487	0.265	0.501	0.343	7.630	0.809	0.478	0.599	0.701	0.304	0.424	2.085	0.76
w/o real-world data	0.756	0.287	0.416	0.330	0.223	0.264	1.700	0.962	0.894	0.927	0.957	0.926	**0.941**	0.114	0.51
w/o synthetic data	0.755	0.706	0.730	0.587	0.483	0.530	1.849	**0.972**	0.572	0.720	0.857	0.656	0.743	0.455	**0.50**
w/o optical flow	**0.793**	0.617	0.694	0.574	0.435	0.495	**1.299**	0.946	0.783	0.857	0.954	0.838	0.892	0.136	0.69
w/o feature fusion	0.700	0.782	0.739	**0.609**	0.518	0.557	1.921	0.931	**0.931**	**0.931**	**0.957**	0.925	0.941	0.120	1.29
duplicate weight	0.767	0.575	0.655	0.573	0.419	0.484	1.786	0.958	0.830	0.889	0.949	0.909	0.928	0.143	0.69
TransVLM(Ours)	0.762	**0.806**	**0.783**	0.574	**0.562**	**0.568**	1.576	0.908	0.882	0.895	0.946	**0.930**	0.938	**0.113**	0.50

Performance on Synthetic Data. Unlike public data, the synthetic data purposefully contains a massive proportion of challenging scenarios, including subtle cuts, complex gradual transitions, and prolonged transitions spanning multiple seconds. Here, TransVLM demonstrates absolute dominance, achieving an unprecedented segment-level F_1 of 89.5%, a frame-level F_1 of 93.8%, and an astonishingly low mABE of just 0.11s.

When confronted with these diverse dynamics, SBD methods experience a catastrophic performance drop. For instance, AutoShot's frame-level F_1 plummets to 10.2%, exposing its fundamental bias towards simple, abrupt cuts. Interestingly, on the synthetic data, general VLMs generally outperform SBD methods. This further validates our critique of SBD limitations: SBD methods focus heavily on searching for ambiguous isolated cut points rather than capturing the intrinsic spatiotemporal patterns of complete transition segments. However, while general-purpose VLMs exhibit some inherent cognitive capacity to perceive gradual transitions, they drastically fail to accurately detect abrupt cuts, resulting in elevated mABE scores. TransVLM is the only framework that seamlessly and robustly unifies the detection of all transition types.

Inference Efficiency. Beyond state-of-the-art detection performance, TransVLM maintains highly practical inference efficiency. As reported in Table 3, our model achieves a Real-Time Factor (RTF) of 0.50. An RTF strictly less than 1.0 signifies faster-than-real-time processing capabilities (i.e., processing one second of video takes only 0.5 seconds). While lightweight heuristic algorithms and heavily down-sampled CNNs (e.g., AutoShot at RTF 0.03) naturally execute faster due to their simplistic architectures, their fundamental inability to parse complex, gradual transitions renders them inadequate for modern high-quality video generation pipelines. TransVLM strikes the optimal trade-off, delivering unprecedented temporal detection precision without compromising real-world deployment viability.

6.3 Ablation Studies

To validate the contribution of each component in TransVLM, we conduct comprehensive ablation studies. The quantitative results across both public and synthetic datasets are detailed in Table 4. See Appendix S6 for details.

Ablation on the Training Process. In order to identify the importance of our training process, we evaluate the pre-trained model directly using our arbitrary video inference strategy as a baseline. The results show that without subsequent fine-tuning, the general VLM performs poorly across all dimensions (e.g., yielding only 48.7% segment-level F_1 on public data). However, compared to the direct application of FPS=5 shown in Table 3, utilizing our sliding-window inference strategy yields noticeable improvements across most metrics. This underscores that while off-the-shelf VLMs lack the fundamental zero-shot capability for precise transition detection, our inference strategy is structurally beneficial.

Ablation on the Data Composition. We first evaluate the necessity of our fine-tuning paradigm and the constructed

datasets by isolating the effects of our mixed-data training strategy. Training exclusively on synthetic data (*w/o public data*) achieves excellent results on the synthetic test set but suffers a severe domain shift when evaluated on public videos, plummeting to 41.6% segment-level F_1. Conversely, training solely on public datasets (*w/o synthetic data*) yields suboptimal generalization on complex synthetic transitions. This stark contrast highlights a significant domain gap in transition distributions (as previously foreshadowed in Table 2). By implementing a quality-aware mixed training strategy, TransVLM successfully bridges this gap, leveraging the vast diversity of the synthetic data while robustly anchoring its spatial understanding to public visual distributions, achieving an optimal segment-level F_1 of 78.3% on public data.

Ablation on the Motion Prior. Gradual transitions are inherently defined by inter-frame pixel dynamics. When the optical flow input is entirely removed (*w/o optical flow*), the model is forced to rely solely on color spatial semantics. This causes a significant performance degradation, dropping the public segment-level F_1 from 78.3% to 69.4%, and the synthetic frame-level F_1 from 93.8% to 89.2%. This validates our core hypothesis: explicitly injecting optical flow effectively compensates for VLM's inherent insensitivity to low-level temporal motion cues.

Ablation on the Fusion Strategy. To demonstrate the efficiency of our simple feature-fusion mechanism, we evaluate an alternative (*w/o feature fusion*), where the color and optical flow frames are processed as two separate visual inputs via different vision encoders to the VLM. While this uncompressed paradigm achieves highly competitive performance (e.g., 94.1% frame-level F_1 on synthetic data), it brutally inflates the sequence length of visual tokens. Consequently, the self-attention computational overhead skyrockets, deteriorating the Real-Time Factor (RTF) from a highly efficient 0.50 to a sluggish 1.29, rendering it entirely unsuitable for real-time video processing. Our simple feature-fusion modification perfectly balances temporal sensitivity with strict inference efficiency.

Ablation on the Zero-Padding Initialization. Finally, we ablate our mathematically stable zero-padding initialization strategy by substituting it with a naive channel duplication method (*duplicate weight*), where the pre-trained color weights are copied and scaled for the optical flow channels. As evidenced by the sharp drop in public segment-level F_1 (78.3% $\rightarrow$ 65.5%), forcefully introducing large, uncalibrated motion gradients during the initial fine-tuning epochs severely destabilizes the optimization process. It causes catastrophic forgetting of the VLM's pre-trained spatial representations. Our zero-padding strategy effectively neutralizes this risk, ensuring stable convergence.

7 Conclusion

In this work, we address the fundamental limitations of traditional Shot Boundary Detection (SBD) by formulating the Shot Transition Detection (STD) task. We propose TransVLM, a Vision-Language Model (VLM) framework that explicitly integrates optical flow as a motion prior alongside standard color frames via a parameter-efficient feature-fusion strategy. This design significantly enhances the model's temporal awareness for detecting diverse transitions without incurring any additional visual token overhead. Furthermore, to overcome the severe class imbalance and annotation noise in public data, we develop a scalable data engine for robust VLM optimization and establish a comprehensive STD benchmark rigorously equipped with multi-dimensional metrics with temporal tolerance. Extensive experiments demonstrate that TransVLM achieves superior overall performance, substantially outperforming traditional heuristic methods, specialized networks, and top-tier VLMs.

References

[1] Sadiq H Abdulhussain, Abd Rahman Ramli, M Iqbal Saripan, Basheera M Mahmmod, Syed Abdul Rahman Al-Haddad, and Wissam A Jassim. Methods and challenges in shot boundary detection: a review. *Entropy*, 20(4): 214, 2018.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization. *Text Reading, and Beyond*, 2(1):1, 2023.

[3] 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*, 2025.

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021.

[5] Simon Baker, Daniel Scharstein, James P Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31, 2011.

[6] 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.

[7] 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.

[8] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[9] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

[10] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[11] Brandon Castellano. Pyscenedetect: Video scene cut detection and analysis tool. *Version 0.6. 0*, 2014. URL https://github.com/Breakthrough/PySceneDetect.

[12] Zhixuan Chu, Lei Zhang, Yichen Sun, Siqiao Xue, Zhibo Wang, Zhan Qin, and Kui Ren. Sora detector: A unified hallucination detection for large text-to-video models. *arXiv preprint arXiv:2405.04180*, 2024.

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[14] Tejaswini Kar, Priyadarshi Kanungo, Sachi Nandan Mohanty, Sven Groppe, and Jinghua Groppe. Video shot-boundary detection: issues, challenges and solutions. *Artificial Intelligence Review*, 57(4):104, 2024.

[15] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

[16] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 5971–5984, 2024.

[17] Do Xuan Long, Xingchen Wan, Hootan Nakhost, Chen-Yu Lee, Tomas Pfister, and Sercan Ö Arık. Vista: A test-time self-improving video generation agent. *arXiv preprint arXiv:2510.15831*, 2025.

[18] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022.

[19] Qianli Ma, Yaowei Zheng, Zhelun Shi, Zhongkai Zhao, Bin Jia, Ziyue Huang, Zhiqi Lin, Youjie Li, Jiacheng Yang, Yanghua Peng, et al. Veomni: Scaling any modality model training with model-centric distributed recipe zoo. *arXiv preprint arXiv:2508.02317*, 2025.

[20] Multimedia Computing Group, Nanjing University (MCG-NJU). SportsShot: A fine-grained dataset for shot segmentation in multiple sports. https://codalab.lisn.upsaclay.fr/competitions/20982, 2024. Dataset hosted on CodaLab.

[21] 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*, pages 10146–10155, 2020.

[22] Mohammadreza Reza Salehi, Jae Sung Park, Aditya Kusupati, Ranjay Krishna, Yejin Choi, Hanna Hajishirzi, and Ali Farhadi. Actionatlas: A videoqa benchmark for domain-specialized action recognition. *Advances in Neural Information Processing Systems*, 37:137372–137402, 2024.

[23] Hanwen Shen, Jiajie Lu, Yupeng Cao, and Xiaonan Yang. Enhancing scene transition awareness in video generation via post-training. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 706–721, 2025.

[24] Tomás Soucek and Jakub Lokoc. Transnet v2: An effective deep network architecture for fast shot transition detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11218–11221, 2024.

[25] Tomáš Souček, Jaroslav Moravec, and Jakub Lokoč. Transnet: A deep network for fast detection of common shot transitions. *arXiv preprint arXiv:1906.03363*, 2019.

[26] Shitao Tang, Litong Feng, Zhanghui Kuang, Yimin Chen, and Wei Zhang. Fast video shot transition localization with deep structured models. In *Asian Conference on Computer Vision*, pages 577–592. Springer, 2018.

[27] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[28] Suramya Tomar. Converting video formats with ffmpeg. *Linux journal*, 2006(146):10, 2006.

[29] Wenjing Tong, Li Song, Xiaokang Yang, Hui Qu, and Rong Xie. Cnn-based shot boundary detection and video annotation. In *2015 IEEE international symposium on broadband multimedia systems and broadcasting*, pages 1–5. IEEE, 2015.

[30] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

[31] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[32] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

[33] Lifang Wu, Shuai Zhang, Meng Jian, Zhe Lu, and Dong Wang. Two stage shot boundary detection via feature fusion and spatial-temporal convolutional neural networks. *IEEE Access*, 7:77268–77276, 2019.

[34] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6787–6800, 2021.

[35] Jingwei Xu, Li Song, and Rong Xie. Shot boundary detection using convolutional neural networks. In *2016 Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2016.

[36] Ramin Zabih, Justin Miller, and Kevin Mai. A feature-based algorithm for detecting and classifying scene breaks. In *Proceedings of the third ACM international conference on Multimedia*, pages 189–200, 1995.

[37] Yifu Zhang, Hao Yang, Yuqi Zhang, Yifei Hu, Fengda Zhu, Chuang Lin, Xiaofeng Mei, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Waver: Wave your way to lifelike video generation. *arXiv preprint arXiv:2508.15761*, 2025.

[38] Zhiyong Zhang, Aniket Gupta, Huaizu Jiang, and Hanumant Singh. Neuflow v2: High-efficiency optical flow estimation on edge devices. *arXiv preprint arXiv:2408.10161*, 6:12, 2024.

[39] Zhiyong Zhang, Huaizu Jiang, and Hanumant Singh. Neuflow: Real-time, high-accuracy optical flow estimation on robots using edge devices. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5048–5055. IEEE, 2024.

[40] Zhiyong Zhang, Aniket Gupta, Huaizu Jiang, and Hanumant Singh. Neuflow-v2: Push high-efficiency optical flow to the limit. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2479–2485. IEEE, 2025.

[41] Wentao Zhu, Yufang Huang, Xiufeng Xie, Wenxian Liu, Jincan Deng, Debing Zhang, Zhangyang Wang, and Ji Liu. Autoshot: A short video dataset and state-of-the-art shot boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2238–2247, 2023.

S Appendix

S.1 Extended Details of Motivation

Detailed Transition Definitions and SBD Limitations. A video shot is defined as a continuous sequence of frames captured seamlessly by a single, uninterrupted camera. It serves as the fundamental, structurally indivisible unit of a video, wherein all intra-shot frames share consistent visual content and coherent motion. Conversely, a transition is the temporal sequence bridging two adjacent shots. Its duration can vary drastically, ranging from an instantaneous abrupt cut to a prolonged gradual progression. When traditional SBD methods [1, 11, 14, 24, 25, 29, 33, 35, 36, 41] process gradual transitions (e.g., dissolves, fades, and wipes), they often fail to determine precise boundaries, resulting in extracted shots contaminated by dirty transitional frames. Furthermore, these methods frequently miss subtle cuts (cuts with minor content changes), long gradual transitions, and special effect transitions. By explicitly detecting these continuous shot transition segments in our formulated STD task, pure and clean shots can be efficiently extracted through simple temporal subtraction.

Benchmark Class Imbalance and Data Engine Synthesis. During the construction of our benchmark, we meticulously re-annotated widely used public data (e.g., RAI [7], BBC [6], AutoShot [41], and MovieShots [21]). However, through random sampling analysis, we observed that these public data suffer from severe class imbalance. They are heavily dominated by normal cuts ($\sim$80%), with only a marginal presence of gradual transitions ($\sim$15%) and exceptionally few long transitions ($\sim$5%). This skewed distribution fundamentally hinders a comprehensive evaluation. Leveraging our scalable data engine, we generated diverse synthetic data comprising normal and subtle cuts ($\sim$30%), gradual transitions ($\sim$20%), and prolonged gradual or special effect transitions ($\sim$50%), thereby enabling a rigorous evaluation across all diverse transition dynamics.

Multi-dimensional Evaluation Protocol. Recognizing that traditional SBD metrics provide an incomplete assessment for the STD task, we establish a multi-dimensional evaluation protocol. Specifically, we introduce segment-level F_1 to evaluate the overall performance of transition detection, frame-level F_1 to measure overall temporal coverage, mean Absolute Boundary Error (ABE) to analyze boundary offsets, and Real-Time Rate (RTR) to evaluate inference speed. To account for the inherent ambiguity in human annotation, our metrics explicitly incorporate a temporal tolerance applied exclusively to the segment annotations and model predictions, ensuring a robust and equitable benchmarking process.

Analysis of Visual Tokens for Optical Flow. To validate the efficacy of introducing explicit motion priors, we visualize the feature maps of the visual tokens extracted immediately after the vision model and prior to the language model. For a given video segment, the optical flow of the current frame is computed relative to its preceding frame. To generate the feature maps, we apply a max-pooling operation along the feature channel dimension of the visual tokens. The feature tensors are then visualized using a sequential colormap, where lower values are rendered in darker color (approaching black) and higher activations in brighter color (approaching yellow). It is worth noting that the pre-trained vision model of Qwen3-VL intrinsically employs a temporal downsampling stride of 2; thus, every two consecutive raw video frames are compressed into a single temporal visual token representation.

As illustrated in Figure S.1 and S.2, for abrupt shot cut—encompassing both standard hard cuts and highly challenging subtle cuts—pure optical flow modalities distinctly capture the inter-frame motions. Consequently, the visual tokens derived from optical flow exhibit significantly stronger response at cut frames when compared to the tokenized representations of pure RGB frames.

In our proposed TransVLM, the RGB and optical flow modalities are fused at the data level when fed to the vision model. Because we employ a zero-padding initialization strategy to guarantee training stability and preserve pre-trained knowledge, the macroscopic pattern of TransVLM's visual tokens visually resembles that of the pure RGB tokens. However, upon closer inspection (e.g., Case (c) in Figure S.1), the fused feature maps at the cut frames demonstrate a sharper contrast against adjacent frames than their RGB-only counterparts. This enhancement provides compelling evidence that the injected motion prior could augment the VLM's sensitivity to fine-grained inter-frame motions without corrupting its foundational spatial understanding.

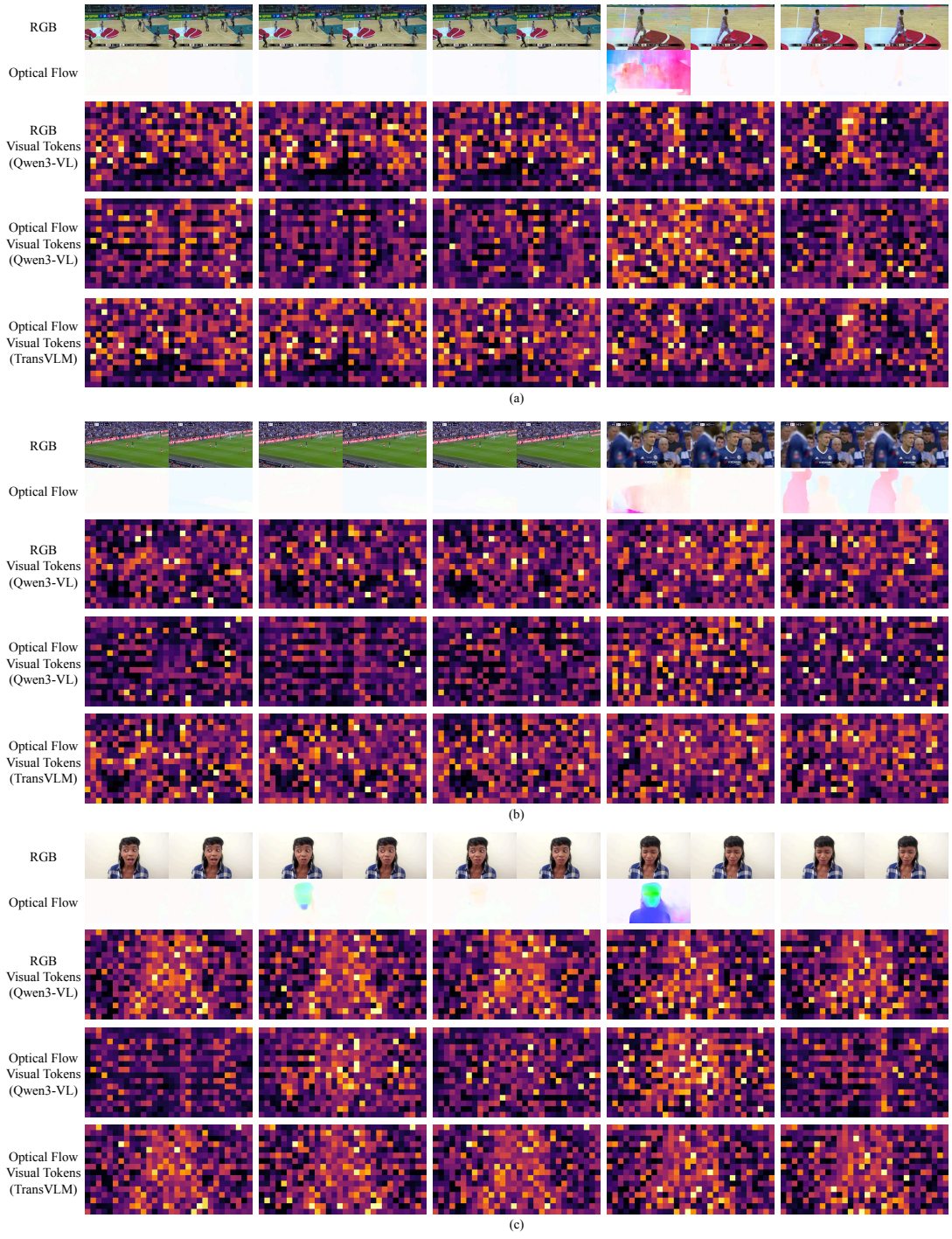

Figure S.1 Feature map visualizations of visual tokens extracted by vision model (Part A). The feature maps are obtained via channel-wise max pooling, with brighter yellow indicating higher magnitudes. While TransVLM's fused tokens generally resemble RGB representations due to zero-padding initialization, they exhibit distinctly sharper temporal contrast at transition boundaries, proving the enhanced sensitivity brought by the optical flow motion prior.

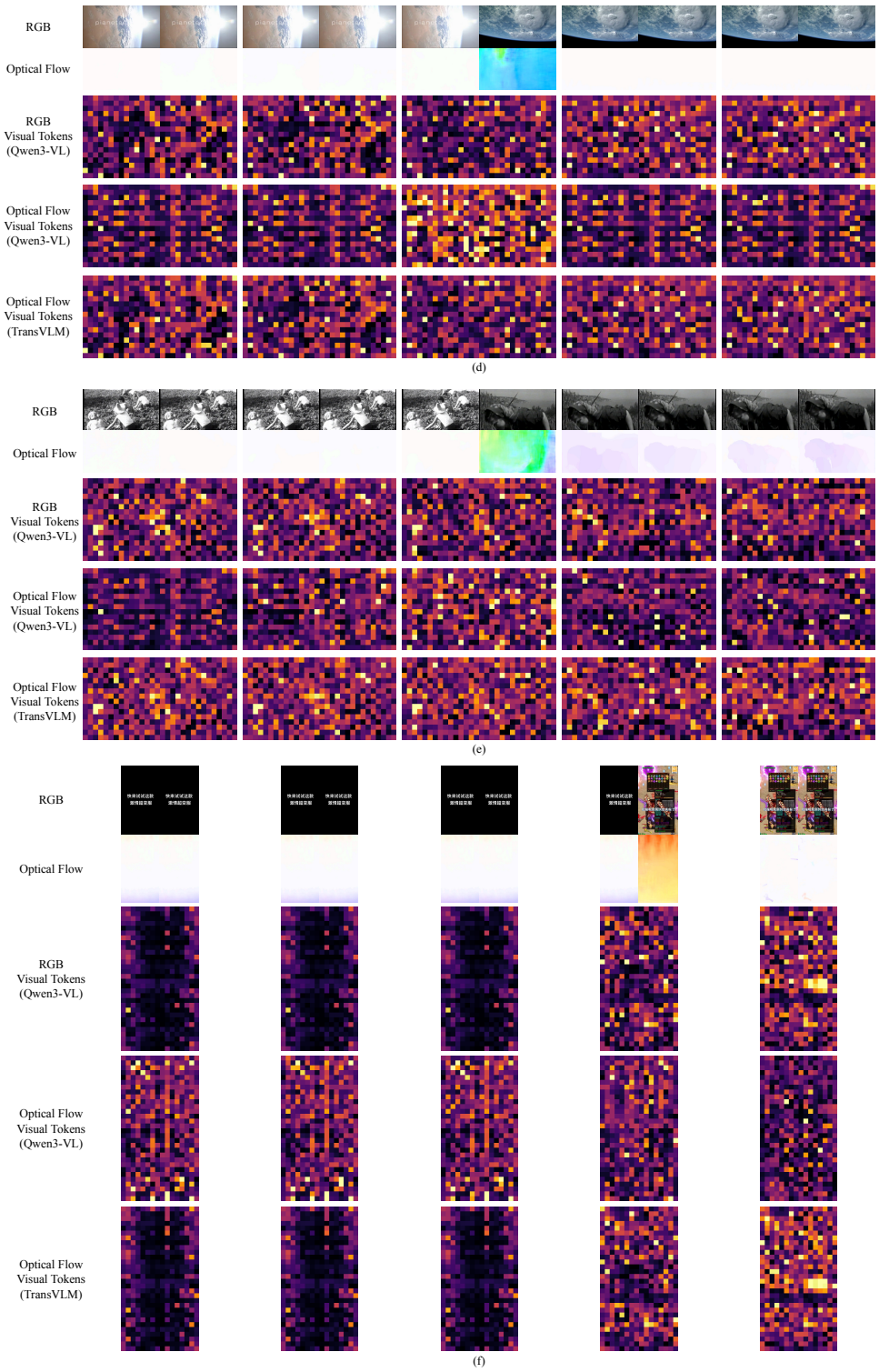

Figure S.2 Feature map visualizations of visual tokens extracted by vision model (Part B). The feature maps are obtained via channel-wise max pooling, with brighter yellow indicating higher magnitudes. While TransVLM's fused tokens generally resemble RGB representations due to zero-padding initialization, they exhibit distinctly sharper temporal contrast at transition boundaries, proving the enhanced sensitivity brought by the optical flow motion prior.

S.2 Extended Details of Benchmark and Data Engine

S.2.1 Detailed Metric Formulations

For both segment-level and frame-level evaluations in the STD task, we compute Precision (P), Recall (R), and the F_1-score using the standard formulations:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2 \times P \times R}{P + R}. \tag{S.1}$$

For **Segment-Level metrics**, True Positives (TP) correspond to the total count of successfully matched predicted segments; False Positives (FP) represent the count of unmatched predictions; and False Negatives (FN) denote the count of ground-truths that fail to match with any prediction.

For **Frame-Level metrics**, TP is defined as the total number of frames within the intersections across all successfully matched prediction and ground-truth segment pairs. FP is the sum of the frames across all predicted segments minus the TP frames. Symmetrically, FN is the sum of the frames across all ground-truth segments minus the TP frames.

To quantify the precision of the boundary localization, the mean Absolute Boundary Error (ABE) is formally calculated as:

$$ABE = \frac{1}{2|TP|} \sum_{(P,G) \in TP} (|g_s - p_s| + |g_e - p_e|), \tag{S.2}$$

where (p_s, p_e) and (g_s, g_e) represent the start and end timestamps of the predicted boundaries and the ground-truth boundaries, respectively.

To critically evaluate the computational efficiency of the models, we report the Real-Time Factor (RTF). It is mathematically defined as the ratio of the total inference time required by the model to the total temporal duration of the processed video:

$$RTF = \frac{T_{\text{inference}}}{T_{\text{video}}}, \tag{S.3}$$

where $T_{\text{inference}}$ denotes the absolute processing time (in seconds) and T_{video} represents the actual duration of the input video sequence (in seconds). An RTF strictly less than 1.0 signifies that the model achieves faster-than-real-time processing, which is a highly desirable property for deploying the transition detection task in practical, large-scale video pipelines.

S.2.2 Benchmark Dataset Distribution

To ensure robust generalization evaluation, the 5,215 videos constituting our STD benchmark are carefully sourced from highly diverse domains. Based on the total temporal duration, the benchmark comprises television shows (RAI [7], 1.6%), documentaries (BBC [6], 9.0%), short mobile videos (AutoShot [41], 2.0%), web videos (ClipShots [26], 32.8%), movies (MovieShots2 [21], 20.7%), sports broadcasting (SportsShot [20], 9.3%), and our synthesized dataset (STD-Synth, 24.6%). The synthesized dataset systematically injects varying transition patterns, including abrupt cuts, normal fades, and prolonged transitions, into continuous pure shots to systematically test model robustness.

S.2.3 Strict Re-annotation Quality Control

Because existing public SBD datasets primarily provide noisy annotations centered on isolated cut points, they are fundamentally inadequate for the segment-level requirements of the continuous transition detection task. To meet these rigorous demands, we conducted a massive manual re-annotation process. Expert annotators meticulously reviewed the original videos and corrected the exact start and end boundaries of the transitions across all utilized public datasets (RAI [21], BBC [6], AutoShot [41], ClipShots [26], MovieShots2 [21], and SportsShot [20]). This strict quality control process establishes a highly accurate, segment-level ground-truth reference tailored specifically for comprehensively evaluating STD models.

S.2.4 Data Engine Based on FFmpeg [28]

To construct a robust and highly scalable training pipeline for the Shot Transition Detection (STD) task, we developed an automated data synthesis engine powered by FFmpeg. This engine seamlessly concatenates discrete pure video shots while injecting diverse, randomized temporal transition dynamics, overcoming the severe class imbalance and annotation noise inherent in legacy public data.

Transition Synthesis Logic. Given a sequential pool of pure video shots, the data engine automatically synthesizes continuous video streams through a rigorous randomized sampling mechanism. For any two adjacent shots, the engine defines a maximum allowable transition duration, denoted as $\mathcal{T}_{max}$. This upper bound is dynamically constrained by the actual temporal lengths of the incoming and outgoing adjacent shots to strictly prevent temporal boundary overflow.

Subsequently, the engine explicitly samples a continuous transition duration $d \sim \mathcal{U}(0, \mathcal{T}_{max})$ and randomly uniformly selects a specific transition effect from a comprehensive pool of 59 distinct transition types. Crucially, if the abruptly transitioning *"cut"* type is selected, the transition duration d is strictly overridden to 0. Once the parameters are sampled, the engine utilizes FFmpeg's advanced rendering pipeline to synthesize the fused video clip. Concurrently, it accurately calculates the precise start and end timestamps of the injected transition segment, automatically exporting these segment-level temporal tuples into a structured JSON label file. This paradigm inherently guarantees the generation of 100% noise-free, accurate segment-level ground truth required for robust Vision-Language Model (VLM) training.

Supported Transition Types. To ensure the trained model robustly generalizes across diverse temporal and spatial transition patterns, our data engine supports the synthesis of 59 distinct transition effects. The exhaustive list, along with their spatiotemporal definitions, is provided below:

1. **cut**: An instantaneous, abrupt switch between shots with exactly zero temporal duration.
2. **fade**: A standard crossfade where the outgoing shot smoothly decreases in opacity while the incoming shot increases.
3. **wipeleft**: A linear spatial wipe transitioning across the frame from right to left.
4. **wiperight**: A linear spatial wipe transitioning from left to right.
5. **wipeup**: A vertical linear wipe progressing from the bottom edge to the top edge.
6. **wipedown**: A vertical linear wipe progressing from the top edge to the bottom edge.
7. **slideleft**: Translates the incoming shot globally over the outgoing shot from right to left.
8. **slideright**: Translates the incoming shot horizontally from left to right.
9. **slideup**: Translates the incoming shot vertically upwards from the bottom edge.
10. **slidedown**: Translates the incoming shot vertically downwards from the top edge.
11. **circlecrop**: Reveals the incoming shot through an expanding circular spatial mask originating from the center.
12. **rectcrop**: Reveals the incoming shot through a symmetrically expanding rectangular mask.
13. **distance**: Blends pixels temporally based on spatial distance calculations between the two frames.
14. **fadeblack**: Fades the outgoing shot completely to a solid black frame before fading into the incoming shot.
15. **fadewhite**: Fades the outgoing shot completely to a solid white frame before fading into the incoming shot.
16. **radial**: A circular, clock-like rotational sweep mask revealing the incoming shot.
17. **smoothleft**: A fluid, smoothed sliding motion of the incoming shot from right to left with easing dynamics.

18. **smoothright**: A fluid, smoothed sliding motion from left to right.
19. **smoothup**: A fluid, smoothed vertical sliding motion from bottom to top.
20. **smoothdown**: A fluid, smoothed vertical sliding motion from top to bottom.
21. **circleopen**: Expands a circular aperture from the frame center to transition shots.
22. **circleclose**: Contracts a circular aperture to hide the outgoing shot, revealing the underlying incoming shot.
23. **vertopen**: Opens a vertical slit outward from the center horizontally to reveal the incoming shot.
24. **vertclose**: Closes a vertical slit inward horizontally to transition between shots.
25. **horzopen**: Opens a horizontal slit outward vertically from the center.
26. **horzclose**: Closes a horizontal slit inward vertically.
27. **dissolve**: A specialized pixel-level cross-dissolve that smooths the structural blending of overlapping frames.
28. **pixelize**: Applies a highly blocky, pixelation filter that interpolates spatial frequencies between the two shots.
29. **diagtl**: A diagonal wipe originating strictly from the top-left corner.
30. **diagtr**: A diagonal wipe originating strictly from the top-right corner.
31. **diagbl**: A diagonal wipe originating strictly from the bottom-left corner.
32. **diagbr**: A diagonal wipe originating strictly from the bottom-right corner.
33. **hlslice**: Interleaves horizontal slices sliding inward from the left boundary.
34. **hrslice**: Interleaves horizontal slices sliding inward from the right boundary.
35. **vuslice**: Interleaves vertical slices sliding inward from the top boundary.
36. **vdslice**: Interleaves vertical slices sliding inward from the bottom boundary.
37. **hblur**: Applies a heavy horizontal motion blur simultaneously during the temporal crossfade.
38. **fadegrays**: Desaturates the outgoing shot to grayscale before temporally fading into the incoming color shot.
39. **wipetl**: A corner-based linear wipe progressing towards the top-left.
40. **wipetr**: A corner-based linear wipe progressing towards the top-right.
41. **wipebl**: A corner-based linear wipe progressing towards the bottom-left.
42. **wipebr**: A corner-based linear wipe progressing towards the bottom-right.
43. **squeezeh**: Horizontally compresses and squeezes the outgoing shot to reveal the incoming one.
44. **squeezev**: Vertically compresses and squeezes the outgoing shot.
45. **zoomin**: Scales and zooms into the center of the outgoing shot while fading into the incoming shot.
46. **fadefast**: A non-linear, temporally accelerated fade transition.
47. **fadeslow**: A non-linear, temporally decelerated (eased) fade transition.
48. **hlwind**: A horizontal wind-like pixel dispersion effect sweeping from the left.
49. **hrwind**: A horizontal wind-like pixel dispersion effect sweeping from the right.
50. **vuwind**: A vertical wind-like pixel dispersion effect sweeping upwards.

51. **vdwind**: A vertical wind-like pixel dispersion effect sweeping downwards.
52. **coverleft**: The incoming shot completely covers the outgoing shot by moving leftwards.
53. **coverright**: The incoming shot covers the outgoing shot by moving rightwards.
54. **coverup**: The incoming shot covers the outgoing shot by moving upwards.
55. **coverdown**: The incoming shot covers the outgoing shot by moving downwards.
56. **revealleft**: The outgoing shot continuously slides left to reveal the underlying incoming shot.
57. **revealright**: The outgoing shot slides right to reveal the incoming shot.
58. **revealup**: The outgoing shot slides up to reveal the incoming shot.
59. **revealdown**: The outgoing shot slides down to reveal the incoming shot.

S.3 Extended Details of Training and Inference

S.3.1 Training Dataset Statistics and Processing

As detailed in Table 1 of the main text, our finalized training dataset comprises over 233,000 videos totaling approximately 1,563 hours. It features more than 690,000 explicitly annotated transitions across diverse temporal dynamics (e.g., abrupt cuts, normal gradual transitions, and prolonged transitions). To maintain stable memory usage and a consistent temporal context during the VLM optimization process, all video inputs sourced from both the data engine and the public datasets are uniformly chunked into clips of ≤ 10 seconds and sampled at a fixed rate of 25 FPS.

S.3.2 Quality-Aware Sampling Tiers

To implement the quality-aware mixed sampling strategy, we categorized all constituent datasets into discrete quality tiers (e.g., Very High, High, Medium) based on extensive manual sampling evaluations. This hierarchical categorization ensures that datasets with highly accurate, segment-level annotations have a proportionally higher probability of being sampled in each training batch compared to those containing noisier legacy annotations.

S.3.3 Sliding-Window Inference Parameters

During the arbitrary video inference phase, a given video is continuously partitioned into overlapping temporal windows. We mathematically define the window size as W and the temporal stride as S, where $S \leq W$. This explicitly enforces a temporal overlap of $W - S$ between adjacent windows (in practice, $W = 10$s and $S = 9$s, leaving a 1-second overlap). For each local window, the model generates textual outputs representing the start and end timestamps of any detected transitions relative to that specific window's timeline. These local timestamps are subsequently projected back onto the global video timeline before we apply temporal Non-Maximum Suppression (NMS) to merge overlapping segments.

S.3.4 Detailed Training Configurations

To effectively optimize TransVLM for the continuous Shot Transition Detection (STD) task, we implemented our training pipeline based on the open-source VeOmni framework. The training details and specific hyperparameter configurations are comprehensively described below:

Optimization and Learning Rate. The model is initialized with the Qwen3-VL-4B-Instruct weights. Critically, we leave the Vision Transformer (ViT) entirely unfrozen (`freeze_vit: false`) to enable full-parameter fine-tuning, allowing the vision encoder to smoothly adapt to the newly fused color and optical flow modalities utilizing our zero-padding initialization strategy. We optimize the network using the AdamW optimizer with a maximum gradient norm clipped at 1.0. The peak learning rate is strictly set to 1.0×10^{-5}, modulated by a cosine learning rate decay schedule to ensure stable convergence.

(a) 1.mp4 in RAI dataset. Gradual Transition at (20.40, 21.84) which is not in the label.

(b) 1.mp4 in RAI dataset. Gradual Transition at (23.80, 25.80) which is not in the label.

(c) 13335596295.mp4 in AutoShot dataset (test split). Subtle Cut at (43.00, 43.04) which is not in the label.

(d) 12580139534.mp4 in AutoShot dataset (train split). Cut at (34.80, 34.84) which is not in the label.

(e) -96lap9jYibdZbCgpB8pBw__.mp4 in ClipShots dataset (train split - only gradual). Cut at (2.08, 2.12) which is not in the label.

Figure S.3 Visualizations of omitted transitions in legacy public datasets (Part A). We uniformly sample 5 or 10 frames around the exact temporal locations of the transitions. Despite the clear visual evidence of shot changes, these transitions are entirely absent from the original ground-truth labels.

(f) -96lap9jYibdZbCgpB8pBw__.mp4 in ClipShots dataset (train split - only gradual). Cut at (10.44, 10.48) which is not in the label.

(g) -96lap9jYibdZbCgpB8pBw__.mp4 in ClipShots dataset (train split - only gradual). Cut at (11.12, 11.16) which is not in the label.

(h) -96lap9jYibdZbCgpB8pBw__.mp4 in ClipShots dataset (train split - only gradual). Cut at (11.44, 11.48) which is not in the label.

(i) -96lap9jYibdZbCgpB8pBw__.mp4 in ClipShots dataset (train split - only gradual). Cut at (12.08, 12.12) which is not in the label.

(j) -96lap9jYibdZbCgpB8pBw__.mp4 in ClipShots dataset (train split - only gradual). Cut at (12.52, 12.56) which is not in the label.

(k) task_0083_12.mp4 in MovieShots2 dataset (test split). Cut at (0.28, 0.32) which is not in the label.

(l) task_0083_12.mp4 in MovieShots2 dataset (test split). Cut at (1.12, 1.16) which is not in the label.

(m) task_0083_12.mp4 in MovieShots2 dataset (test split). Cut at (3.40, 3.44) which is not in the label.

(n) task_0083_12.mp4 in MovieShots2 dataset (test split). Cut at (7.80, 7.84) which is not in the label.

Figure S.4 Visualizations of omitted transitions in legacy public datasets (Part B). We uniformly sample 5 or 10 frames around the exact temporal locations of the transitions. Despite the clear visual evidence of shot changes, these transitions are entirely absent from the original ground-truth labels.

Batching and Sequence Length. The maximum sequence length during training is deliberately set to 16,384 tokens. This extended context window comfortably accommodates the dense visual tokens extracted from the maximum 10.0-second training clips, alongside the customized text prompts. The training is conducted across 8 NVIDIA H100 (80GB) GPUs. We assign a micro-batch size of 4 per device, yielding an effective global batch size of 32. The model is fine-tuned for a maximum of 500 optimization steps.

Memory and Computational Efficiency. Given the intensive memory demands of processing high-frame-rate multimodal inputs, we integrate several advanced memory-optimization mechanisms. Specifically, we utilize Fully Sharded Data Parallel 2 (FSDP2) for distributed training, coupled with standard gradient checkpointing. To accelerate the attention computation, `flash_attention_2` is explicitly adopted. Furthermore, we enable padding-free training relying on position IDs (`rmpad_with_pos_ids: true`) combined with a dynamic batch size buffer, which drastically minimizes redundant computations on zero-padded tokens.

Prompt Formatting. To align the VLM's generative output with the strict segment-level evaluation requirements of the STD task, our customized user prompt explicitly instructs the model to output the detected transition information exclusively as a single, raw JSON array of temporal tuples. The generation of markdown wrappers (e.g., "‘`json`) or any auxiliary explanatory text is strictly prohibited during the supervised fine-tuning phase.

S.4 Bad Annotations in Public Datasets

During our data inspection and re-annotation process, we identified a general issue regarding the annotation fidelity of public datasets designed for SBD task, spanning both training and testing splits. This detrimental issue is the omission of valid transition boundaries—i.e., some shot transitions are completely unrecorded in the original ground-truth labels. This inherently high false-negative rate in the annotations significantly compromises the reliability of any evaluation conducted solely on these datasets, as models correctly identifying these unannotated transitions would be unfairly penalized. **However, given the large scale of the training data, exhaustively re-annotating every single video is practically infeasible. Consequently, the inevitable presence of these residual noisy annotations in the training splits may introduce adverse gradients and negatively impact the model optimization process.** This practical limitation further underscores the necessity of the quality-aware mixed sampling strategy we implemented during training.

To demonstrate this critical flaw, Figure S.3 and S.4 presents a representative selection of omitted transitions randomly discovered during our manual review of the public datasets. For optimal visual clarity, we uniformly sample 5 or 10 continuous frames within the immediate temporal neighborhood of each missed transition. As unequivocally shown in the visualization, these sequences contain distinct and visually obvious shot changes (both abrupt cuts and gradual transitions) that were entirely overlooked by the original human annotators. This further validates the absolute necessity of our meticulously re-annotated TBD Benchmark.

S.5 More Details of Quantitative Experiments

In this section, we provide an exhaustive breakdown of our quantitative evaluations. The following figures illustrate the performance curves of various methods as a function of the temporal tolerance τ. These evaluations are systematically conducted across different aggregated data domains (public data S.5, synthetic data S.6, and the overall benchmark S.7). As explicitly demonstrated by the plotted curves, our proposed TransVLM consistently achieves superior overall performance. Regardless of the strictness of the temporal tolerance, our framework robustly dominates the primary evaluation metrics—specifically the segment-level F_1 and frame-level F_1 scores—further validating its absolute effectiveness and stability on the continuous transition detection task.

The horizontal axis represents τ, while the vertical axis represents the value of selected metric.

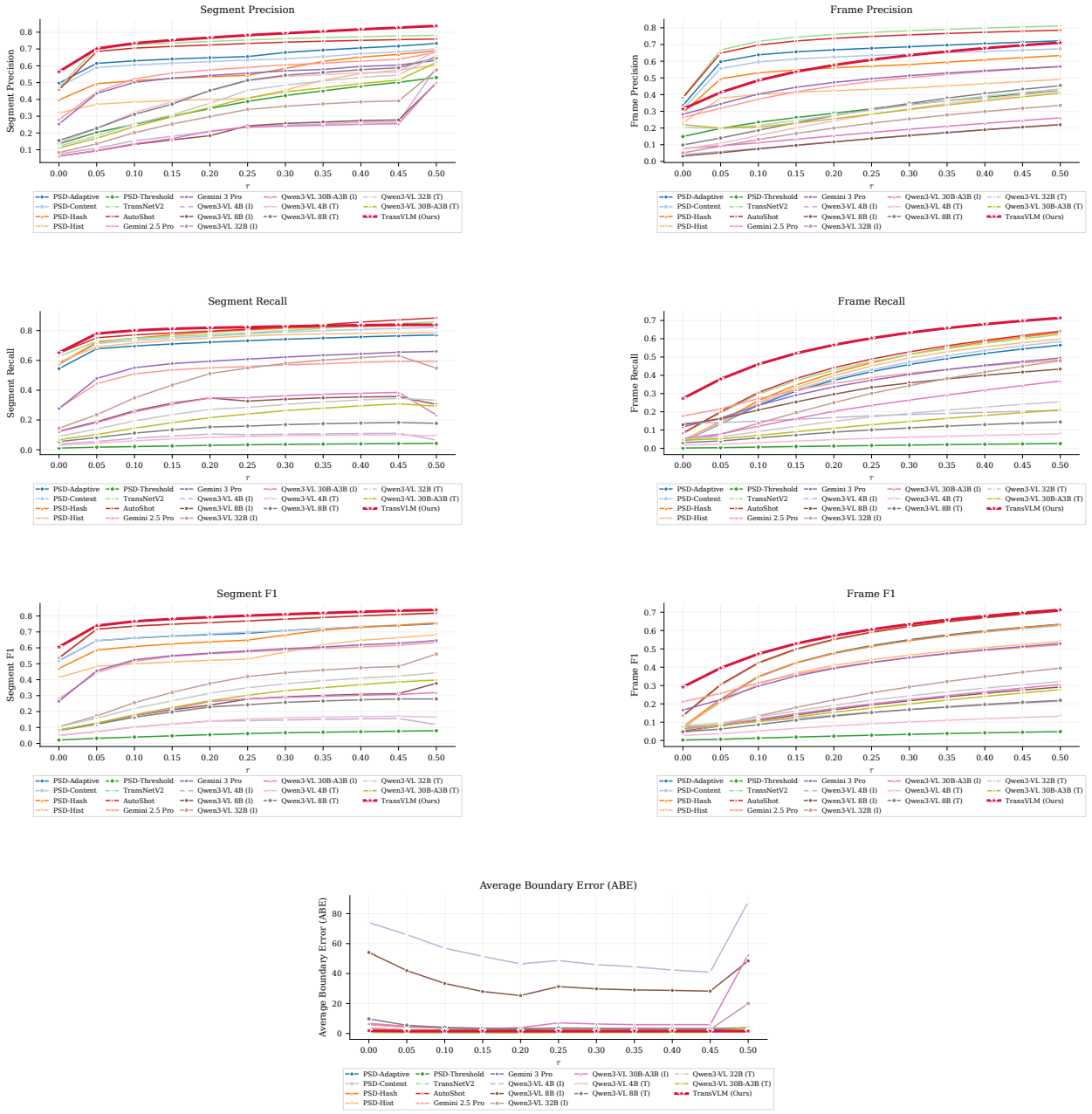

Figure S.5 Quantitative comparison visualization on all public data.

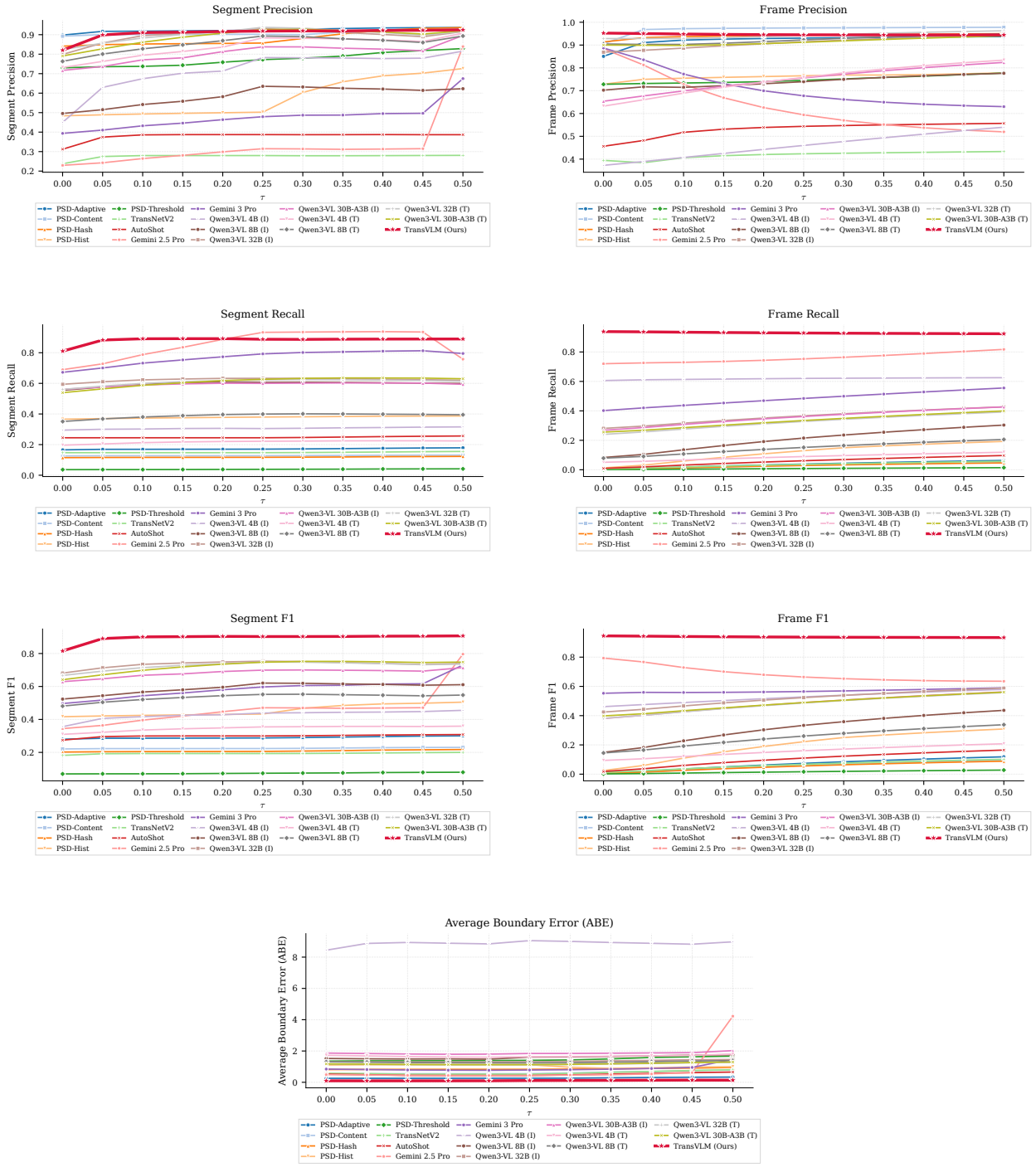

Figure S.6 Quantitative comparison visualization on all synthetic data.

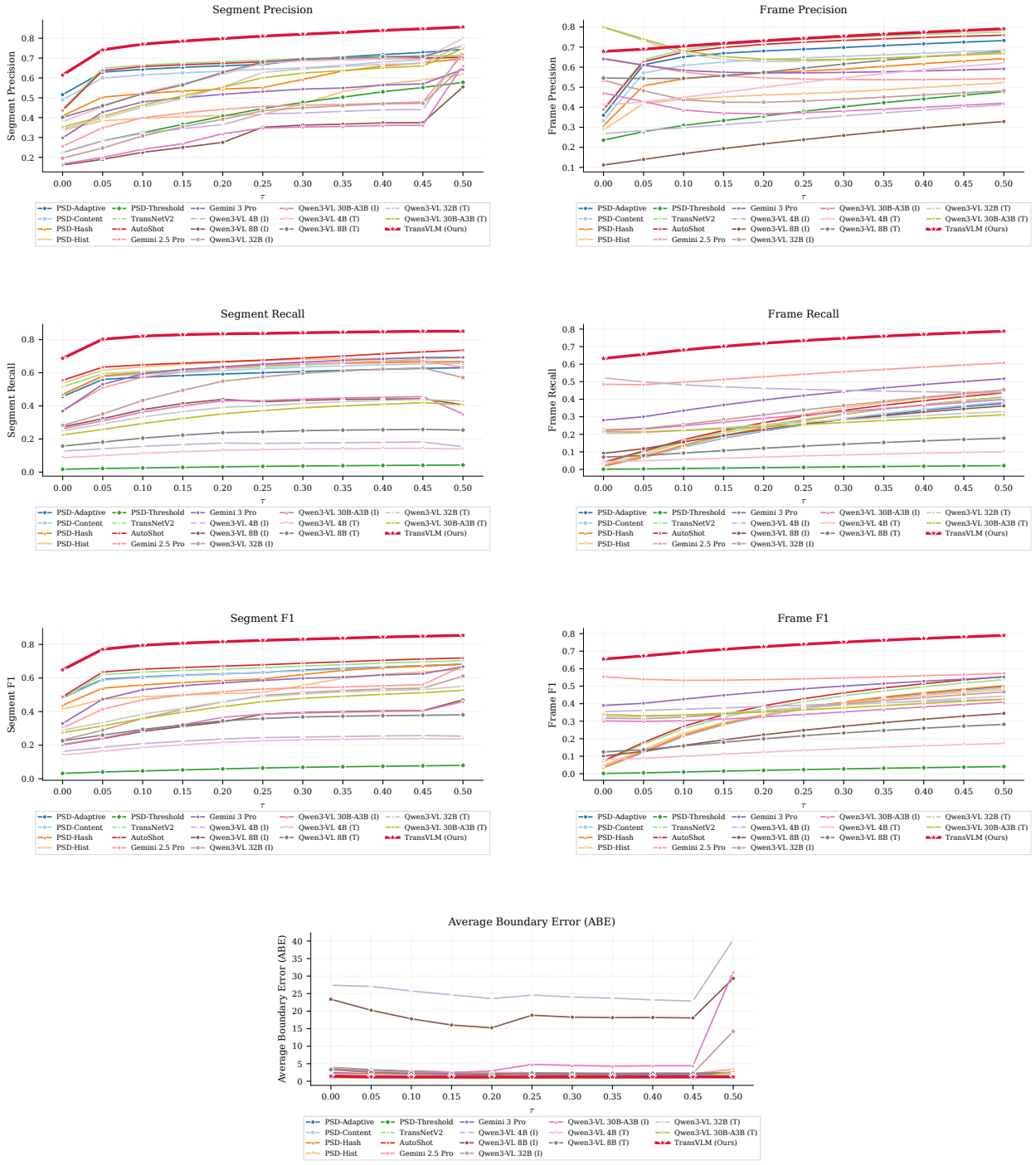

Figure S.7 Quantitative comparison visualization on all data.

S.6 Extended Details of Ablation Studies

In this section, we provide an exhaustive breakdown of our comprehensive ablation studies to further validate the individual contributions of each component within TransVLM. The following figures illustrate the performance curves of various methods as a function of the temporal tolerance τ. These evaluations are systematically conducted across different aggregated data domains (public data S.8, synthetic data S.9, and the overall benchmark S.10). As explicitly demonstrated by the plotted curves, our proposed TransVLM consistently achieves superior balance between public and synthetic data.

The horizontal axis represents τ, while the vertical axis represents the value of selected metric.

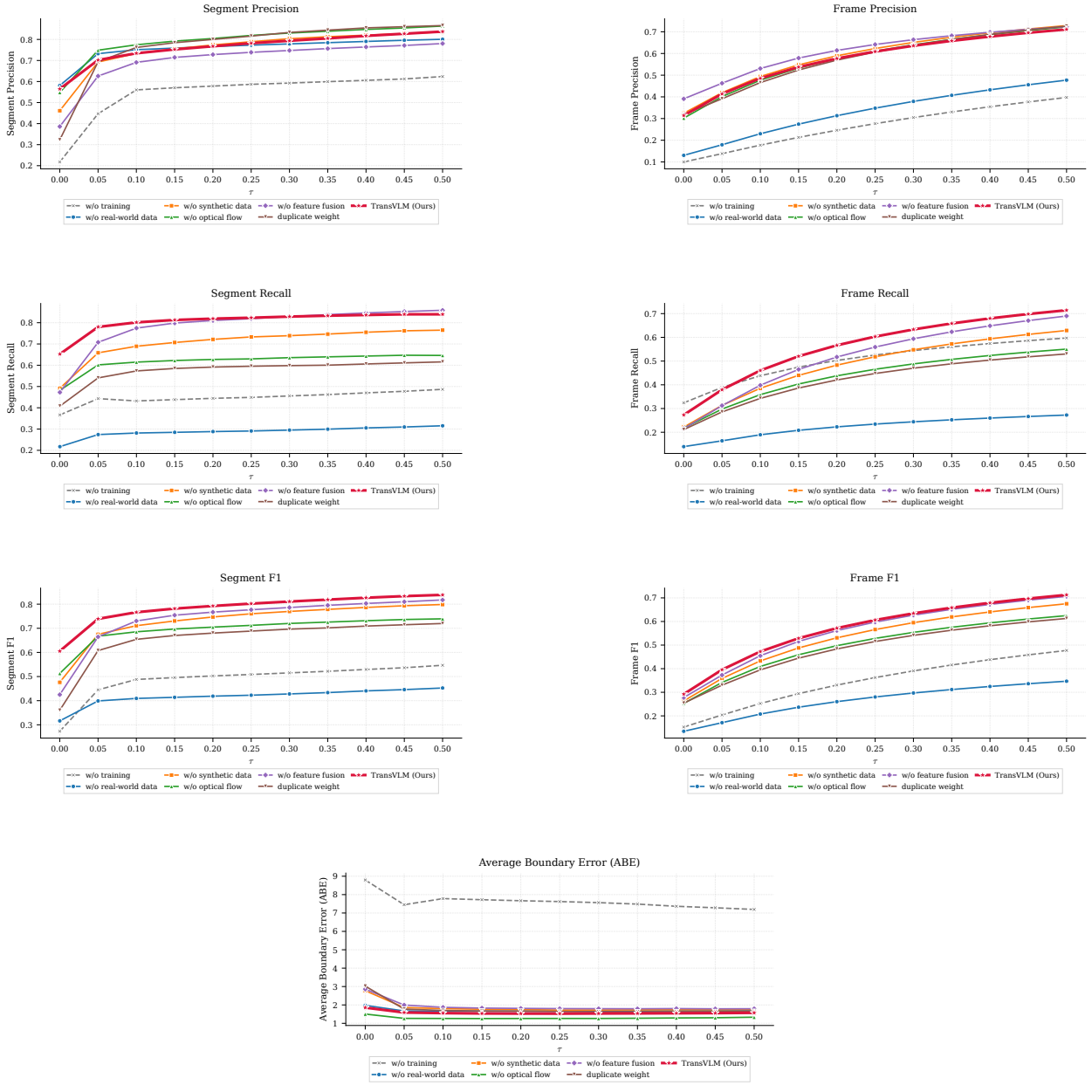

Figure S.8 Ablation study visualization on all public data.

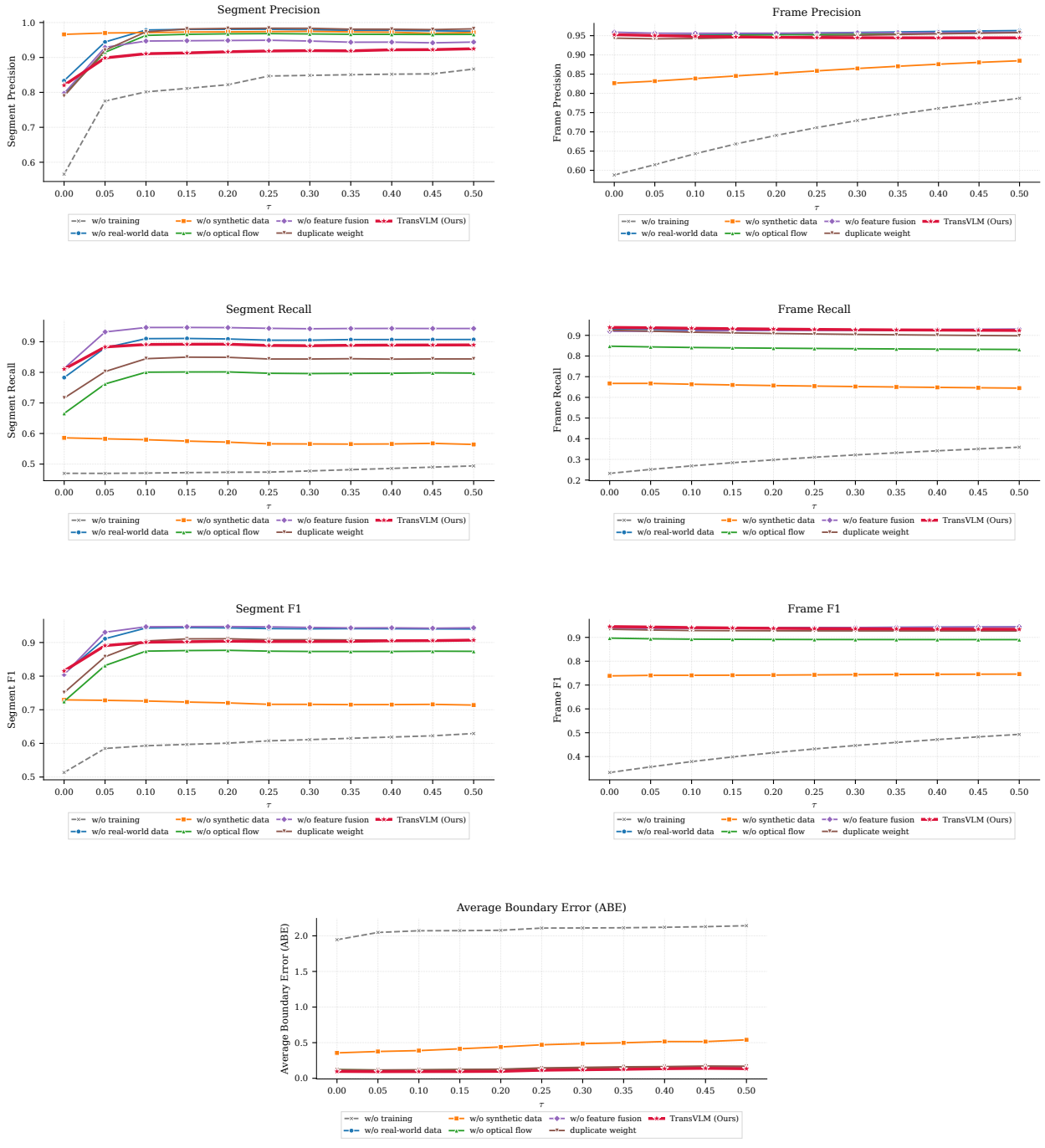

Figure S.9 Ablation study visualization on all synthetic data.

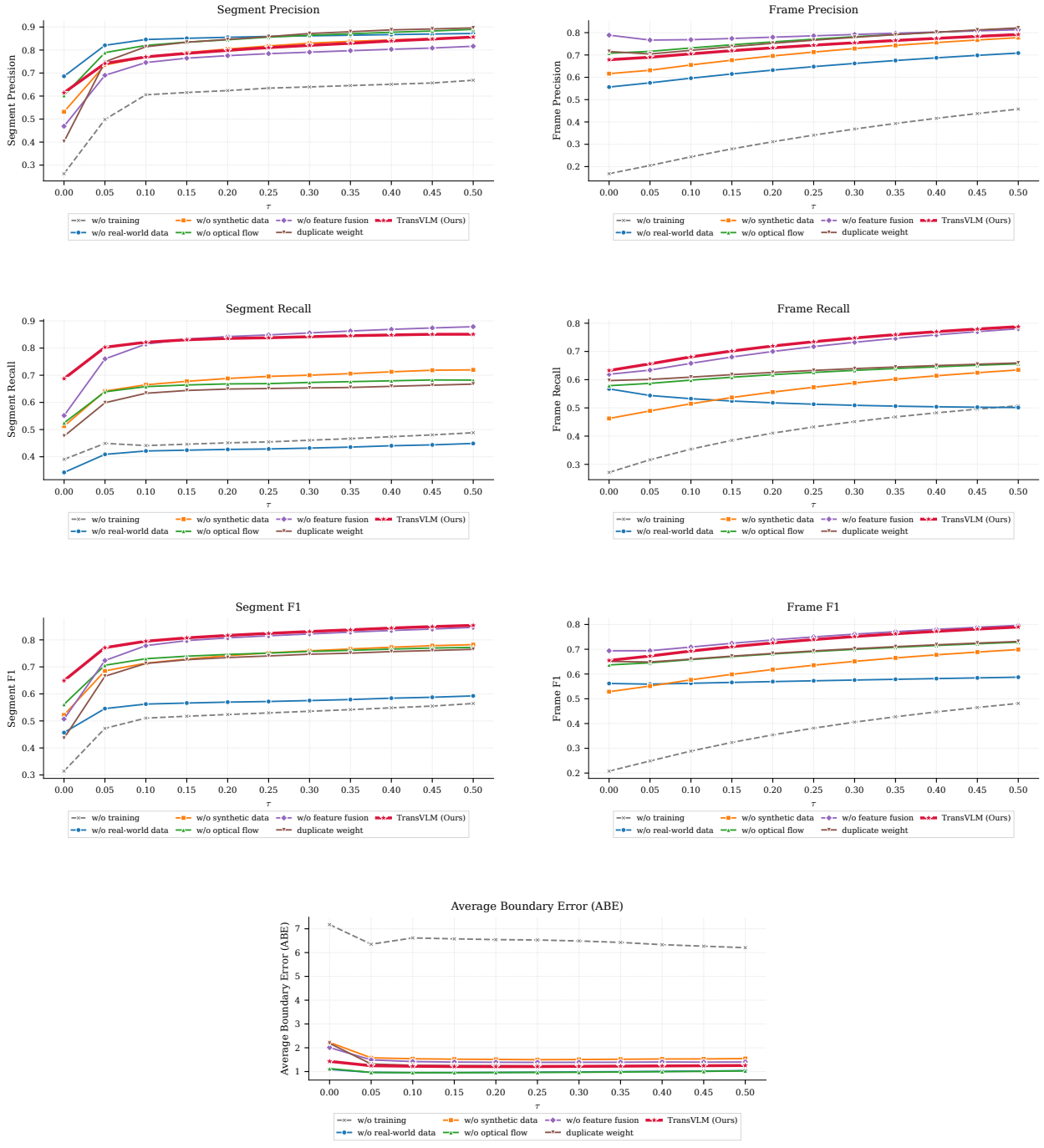

Figure S.10 Ablation study visualization on all data.