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

Ce Chen[1,2], Yi Ren[2], Yuanming Li[2], Viktor Goriachko[2], Zhenhui Ye[2], Zujin Guo[2,3], Zhibin Hong[2], and Mingming Gong[1]

[1] University of Melbourne, Parkville VIC 3010, Australia
ce.chen@student.unimelb.edu.au
mingming.gong@unimelb.edu.au
[2] HeyGen, Los Angeles CA 90094, USA*
{yi.ren, yuanming.li, viktor.goriachko, zhenhui.ye, charly}@heygen.com
[3] Nanyang Technological University, Singapore 639798, Singapore
zujin001@e.ntu.edu.sg

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.

Keywords: Shot Transition Detection · Vision-Language Models · Video Benchmark

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.

* Ce Chen and Zujin Guo did this work during their internship at HeyGen.

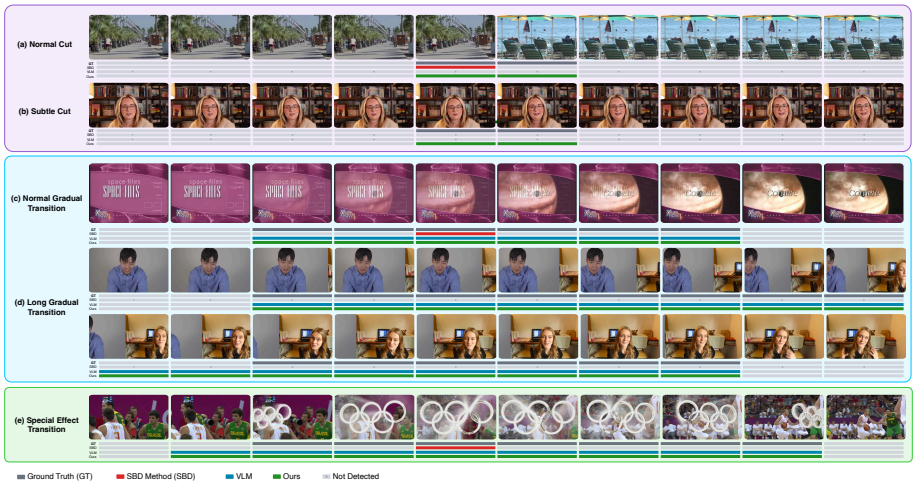

Fig. 1: Limitations of existing shot transition detection methods. Predicted transitions are denoted by colored lines: **gray** for ground truth, **red** for a state-of-the-art SBD method (AutoShot [50]), **blue** for a general-purpose top-tier VLM (Qwen3-VL [3]), and **green** for our proposed TransVLM. 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.

In video retrieval, processing clean shots strictly prevents the semantic blending of distinct scenes, thereby ensuring robust cross-modal matching performance [4, 22]. In video understanding tasks [16, 27, 36, 40, 42], 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 generation tasks [8, 9, 12, 14, 17, 21, 28, 37, 38, 46], 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 process-

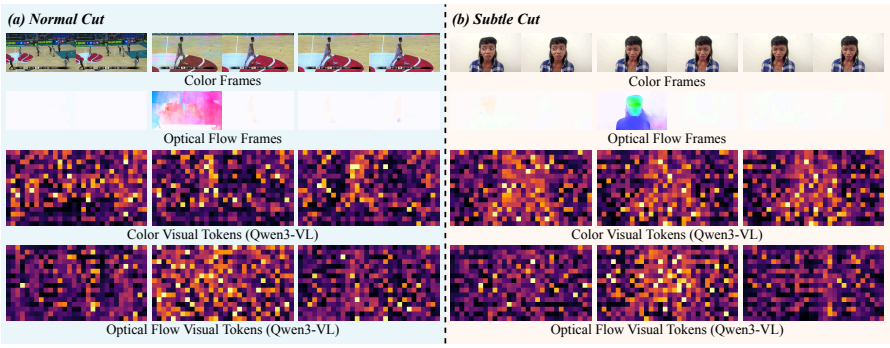

Fig. 2: Visual token visualization for color vs. optical flow. By capturing inter-frame changes, optical flow serves as a robust motion prior. For both (a) normal and (b) subtle cuts, the visual tokens derived from optical flow exhibit significantly sharper response contrast at the transition frames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

ing. 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 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 [1,15] rely on heuristic algorithms (e.g., PySceneDetect [11], ECR [44]) 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 [35, 41, 43], such as the TransNet series [29, 30] and AutoShot [50], 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.

Temporal Action Localization. Temporal Action Localization (TAL) shares a structurally similar formulation with STD, whose goal is to predict start and end intervals of target events in untrimmed videos. Boundary-based proposal networks such as BSN [20], BMN [19], and BSN++ [31] obtain precise temporal boundaries from frame-level features, while transformer-based detectors such as ActionFormer [45] and recent scalable grounding frameworks such as SnAG [24] improve this paradigm with stronger long-range modeling. However, TAL targets

action classes with high intra-segment visual coherence, whereas STD targets transitions where content is discontinuous across shots.

Video Understanding. Recent advancements in video understanding have evolved from specialized spatiotemporal architectures (e.g., I3D [10], SlowFast [13], and VideoMAE [36]) to powerful general-purpose Vision-Language Models (VLMs), such as Video-LLaVA [18], the Qwen-VL series [2,39], and Gemini [33]. 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 spatiotem-

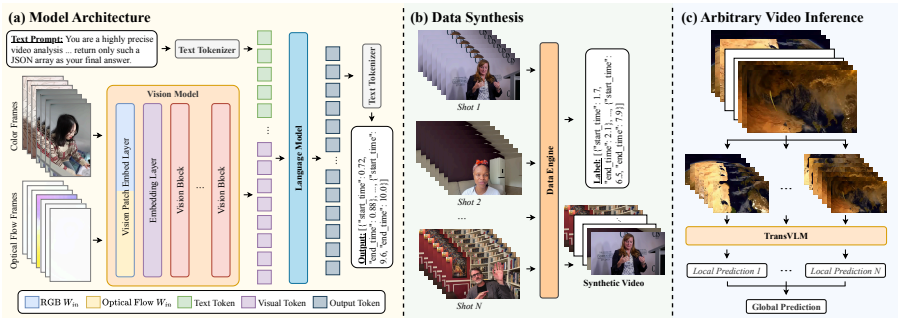

Fig. 3: TransVLM Framework. Our proposed framework comprises three core components. **(a) Model Architecture:** We explicitly inject optical flow as a motion prior via a parameter-efficient strategy. By exclusively expanding the input projection weights (W_{in}) of the Vision Patch Embed Layer, the model 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).

poral patterns of the transition dynamics themselves, rather than over-fitting to ambiguous boundary frames.

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. In principle, our modification is architecture-independent and can be ported to any VLM with a patch-embedding visual input.

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 [47–49] as our optical flow extractor. To align the spatial distribution of the 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.

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) [26]	Movies	Very High	5,222	226.17	152,191	6,347.6	152,191	0	0
SportsShot (Train) [25]	Sports	Very High	720	27.87	13,295	2,731.2	9,930	2,958	407
ClipShots (Train) [32]	Web Videos	High	5,360	319.42	158,825	24,931.9	129,544	24,727	4,554
AutoShot (Train) [50]	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**

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 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 pre-

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) [50]	Short Videos	Point	200	2.01	2,065	545.0	1,008	1,004	53
ClipShots (Test) [32]	Web Videos	Point	500	32.85	6,923	1,548.1	4,798	1,830	295
MovieShots2 (Test) [26]	Movies	Point	282	20.72	14,767	9,566.4	13,436	710	621
SportsShot (Val) [25]	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**

dictions 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. Please refer to the Supplementary Material for more details.

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). Details for all metrics are provided in the Supplementary Material.

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 [29, 50], and top-tier open- and closed-source Vision-Language Models (VLMs) [2, 33]. 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 and Thinking variants yield

Table 3: Quantitative comparison on the STD Benchmark. We report segment-, frame-, and IoU-based Precision (P), Recall (R), and F_1 averaged over temporal-tolerance values, and Absolute Boundary Error (ABE) in seconds. IoU is averaged over thresholds $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. **FPS** is the inference frame rate. **RTF** is the end-to-end Real-Time Factor including NeuFlow v2 optical-flow extraction for TransVLM. **Params** is the total parameter count. **M-JSON (%)** is the malformed-JSON rate of VLM outputs. **Bold**: best. underline: second-best. "–": not applicable.

		Public Data										**Synthetic Data**												
		Segment			Frame			IoU				Segment			Frame			IoU						M-JSON
Methods	FPS	P	R	F1	P	R	F1	P	R	F1	ABE	P	R	F1	P	R	F1	P	R	F1	ABE	RTF	Params	(%)
PySceneDetect [11]																								
- Adaptive	25	0.656	0.716	0.684	0.645	0.372	0.457	**0.174**	0.191	0.182	1.93	**0.924**	0.172	0.290	0.922	0.036	0.069	**0.684**	0.126	0.213	0.28	0.09	–	–
- Content	25	0.627	0.759	0.686	0.603	0.383	0.453	0.159	0.191	0.173	1.08	0.909	0.128	0.225	**0.968**	0.029	0.056	0.234	0.033	0.058	0.61	0.09	–	–
- Hash	25	0.566	0.778	0.654	0.549	0.416	0.457	0.133	0.194	0.158	2.11	0.880	0.118	0.208	0.940	0.027	0.052	0.237	0.032	0.057	0.85	0.09	–	–
- Hist	25	0.452	0.741	0.559	0.419	0.398	0.394	0.099	0.183	0.129	1.19	0.577	0.379	0.455	0.762	0.117	0.197	0.108	0.082	0.093	1.04	0.09	–	–
- Threshold	25	0.364	0.031	0.057	0.306	0.014	0.027	0.014	0.001	0.002	3.08	0.773	0.039	0.074	0.749	0.008	0.016	0.013	0.000	0.001	1.46	0.09	–	–
TransNetV2 [29]																								
- Official	25	0.727	0.780	0.752	**0.731**	0.427	0.528	0.173	0.200	0.186	1.87	0.275	0.149	0.194	0.417	0.034	0.063	0.044	0.028	0.034	0.61	0.07	7.6M	–
- w/ DH	25	0.251	**0.865**	0.389	0.227	0.625	0.333	0.066	**0.282**	0.106	1.63	0.487	0.862	0.622	0.610	0.589	0.598	0.131	0.340	0.189	1.00	0.07	8.1M	–
AutoShot [50]	25	0.707	0.804	0.751	0.709	0.440	0.532	0.157	0.221	0.184	1.78	0.379	0.248	0.299	0.530	0.058	0.102	0.088	0.069	0.077	0.51	0.03	14.3M	–
ActionFormer [45]	25	0.390	0.568	0.462	0.368	**0.795**	0.498	0.153	0.208	0.176	**0.34**	0.360	0.758	0.488	0.619	**0.986**	0.760	0.343	**0.766**	0.474	0.99	**0.02**	29.2M	–
Gemini Series [33]																								
- 2.5 Pro	5	0.558	0.527	0.542	0.453	0.361	0.401	0.105	0.105	0.105	3.62	0.338	0.851	0.465	0.638	0.760	0.686	0.155	0.465	0.233	0.82	0.81	–	0.10
- 3 Pro	5	0.527	0.573	0.549	0.469	0.343	0.393	0.091	0.100	0.096	2.18	0.479	0.768	0.588	0.711	0.482	0.568	0.171	0.292	0.216	0.88	1.32	–	0.29
Qwen3-VL Instruct Series [3]																								
- 4B	5	0.235	0.088	0.124	0.134	0.174	0.148	0.011	0.006	0.008	55.00	0.717	0.306	0.428	0.458	0.618	0.525	0.083	0.060	0.070	8.88	0.31	4.4B	4.96
- 8B	5	0.222	0.297	0.246	0.132	0.307	0.184	0.009	0.016	0.012	34.39	0.586	0.597	0.591	0.741	0.204	0.315	0.032	0.035	0.033	1.60	0.34	8.8B	3.10
- 32B	5	0.309	0.473	0.370	0.214	0.279	0.241	0.012	0.020	0.015	3.96	0.895	0.623	0.735	0.911	0.361	0.516	0.243	0.197	0.218	1.35	0.98	33.4B	0.38
- 30B-A3B(MoE)	5	0.218	0.300	0.242	0.171	0.222	0.192	0.009	0.017	0.011	9.61	0.806	0.593	0.683	0.749	0.355	0.482	0.153	0.120	0.134	1.86	0.35	31.1B	1.63
Qwen3-VL Thinking Series [3]																								
- 4B	5	0.449	0.079	0.134	0.265	0.051	0.086	0.018	0.004	0.006	1.50	0.839	0.218	0.346	0.748	0.087	0.156	0.088	0.024	0.037	1.68	1.03	4.4B	38.53
- 8B	5	0.450	0.144	0.217	0.297	0.094	0.143	0.020	0.008	0.011	4.25	0.854	0.389	0.534	0.923	0.147	0.252	0.080	0.037	0.050	1.29	1.31	8.8B	22.34
- 32B	5	0.403	0.260	0.315	0.308	0.160	0.209	0.022	0.016	0.019	1.09	0.900	0.608	0.726	0.943	0.323	0.479	0.180	0.122	0.146	1.16	3.30	33.4B	4.54
- 30B-A3B(MoE)	5	0.374	0.217	0.275	0.291	0.127	0.175	0.019	0.012	0.015	0.98	0.890	0.610	0.724	0.915	0.331	0.485	0.168	0.115	0.136	1.18	0.92	31.1B	1.23
TransVLM (Ours)																								
- 4B	5	0.454	0.692	0.548	0.478	0.613	0.527	0.075	0.147	0.099	1.97	0.783	0.704	0.741	0.860	0.838	0.849	0.473	0.434	0.453	0.34	0.36	4.8B	0.59
- 4B	25	**0.762**	0.806	**0.783**	0.574	0.562	**0.568**	0.166	0.221	**0.189**	1.58	0.908	**0.882**	**0.895**	0.946	0.930	**0.938**	0.670	0.662	**0.666**	**0.11**	0.72	4.8B	**0.03**

malformed-JSON rates of 54% and 65%, respectively, making them nearly unusable. Therefore, the 4B Instruct model serves as the ideal foundational backbone.

Optical flow representations are extracted using the NeuFlow v2 [47–49] 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 [23] 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 [34], our engine supports the generation of 59 distinct transition effects (see Supplementary 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 [29] and AutoShot [50]; (3) *General-purpose foundational VLMs*, specifically evaluating the Qwen3-VL [3] and Gemini [33] series; and (4) *Temporal Action Localiza-*

tion (TAL) methods, represented by ActionFormer [45] trained on our STD training data. To guarantee a strictly fair and rigorous comparison, SBD predictions are converted to transition segments by pairing adjacent boundary points into intervals (see Supplementary S3). For reference, under the traditional boundary-based metric TransNetV2 and AutoShot achieve F_1 of 0.67/0.42 and 0.66/0.49 on public/synthetic data, respectively. SBD and TAL baselines are evaluated at their native 25 FPS, while general VLMs are evaluated at 5 FPS to balance fairness and efficiency. All comprehensive evaluations are standardized on our newly established STD benchmark.

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 Supplementary 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 Supplementary 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. Despite adopting a similar segment-prediction formulation, TAL methods such as ActionFormer [45] attain competitive boundary precision but lag substantially in segment-level F_1, indicating that boundary regression alone cannot disambiguate transitions without robust visual-semantic understanding. TransNetV2 (w/ DH) similarly inflates recall (0.865) at the cost of precision (F_1 0.389). Even our 5-FPS variant outperforms all VLMs under matched FPS, confirming that the gains stem from architectural design rather than higher FPS.

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

Table 4: Ablation Study of TransVLM. Metrics, column conventions, and bold/underline notations follow Table 3. The ablations are: (1) *Sliding Window* (*w/o sliding window*, causing visual-token overload); (2) *Training* (off-the-shelf VLM with sliding window only); (3) *Data Composition* (*w/o public/synthetic data*); (4) *Motion Prior* (*w/o optical flow*); (5) *Fusion Strategy* (*w/o feature fusion*, two-encoder paradigm); and (6) *Initialization* (*duplicate weight*).

		Public Data										**Synthetic Data**											
		Segment			Frame			IoU				Segment			Frame			IoU					M-JSON
Methods	FPS	P	R	F1	P	R	F1	P	R	F1	ABE	P	R	F1	P	R	F1	P	R	F1	ABE	RTF	Params	(%)
w/o sliding window	25	0.450	0.676	0.540	0.443	**0.589**	0.502	0.137	0.106	0.120	3.31	0.883	0.846	0.864	0.883	0.924	0.903	0.213	0.525	0.303	0.63	0.89	4.8B	4.35
w/o training	25	0.545	0.448	0.487	0.265	0.501	0.343	0.047	0.079	0.059	7.63	0.809	0.478	0.599	0.701	0.304	0.424	0.113	0.093	0.102	2.08	0.76	4.4B	2.13
w/o public data	25	0.756	0.287	0.416	0.330	0.223	0.264	0.151	0.056	0.082	1.70	0.962	0.894	0.927	**0.957**	0.926	**0.941**	**0.696**	0.654	**0.674**	**0.11**	0.73	4.8B	**0.01**
w/o synthetic data	25	0.755	0.706	0.730	0.587	0.483	0.530	0.150	0.161	0.155	1.85	**0.972**	0.572	0.720	0.857	0.656	0.743	0.614	0.372	0.464	0.46	0.72	4.4B	0.06
w/o optical flow	25	**0.793**	0.617	0.694	0.574	0.435	0.495	**0.175**	0.154	0.164	**1.30**	0.946	0.783	0.857	0.954	0.838	0.892	0.663	0.555	0.604	0.14	**0.69**	4.8B	0.05
w/o feature fusion	25	0.700	0.782	0.739	**0.609**	0.518	0.557	0.146	0.178	0.160	1.92	0.931	**0.931**	**0.931**	**0.957**	0.925	**0.941**	0.636	0.649	0.643	0.12	1.51	4.4B	0.03
duplicate weight	25	0.767	0.575	0.655	0.573	0.419	0.484	0.109	0.137	0.121	1.79	0.958	0.830	0.889	0.949	0.909	0.928	0.647	0.587	0.615	0.14	0.91	4.8B	0.83
TransVLM (Ours)	25	0.762	**0.806**	**0.783**	0.574	0.562	**0.568**	0.166	**0.221**	**0.189**	1.58	0.908	0.882	0.895	0.946	**0.930**	0.938	0.670	**0.662**	0.666	**0.11**	0.72	4.8B	0.03

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 an end-to-end Real-Time Factor (RTF) of 0.72 (0.22 from NeuFlow v2 plus 0.50 from the VLM forward pass). An RTF strictly less than 1.0 signifies faster-than-real-time processing (i.e., processing one second of video takes only 0.72 seconds). While lightweight heuristic algorithms and heavily downsampled 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 Supplementary S6 for details.

Ablation on the Sliding Window. Removing the temporal sliding-window inference (*w/o sliding window*) forces the entire video into a single forward pass,

severely overloading the visual-token budget. As a result, the malformed-JSON rate spikes from 0.03% to 4.35%, and the public segment-level F_1 drops from 78.3% to 54.0%. This validates that the sliding-window strategy is essential for scaling our framework to arbitrary-length videos.

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 end-to-end RTF from a highly efficient 0.72 to a sluggish 1.51, 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 theoretically stable zero-padding initialization strategy by substituting it with a naive channel duplication method (*duplicate weight*), where the pre-trained weights are

copied and scaled for the optical flow channels. As evidenced by the sharp drop in public segment-level F_1 ($78.3\% \rightarrow 65.5\%$), introducing large gradients during the initial fine-tuning epochs severely makes the optimization process unstable. It causes severe forgetting of the pre-trained spatial representations. Our zero-padding strategy effectively mitigates 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.

Limitations and Future Work. While our channel-level feature fusion adds limited overhead to the VLM, TransVLM still relies on an external optical-flow extractor, which may bottleneck ultra-low-latency streaming. Deriving motion priors natively from ViT temporal-attention differences is a promising direction. Although the chosen Qwen3-VL-4B is the smallest backbone that reliably produces well-formed temporal outputs, future work may explore smaller VLMs with constrained decoding or dedicated timestamp tokens. Finally, our framework outputs only the start and end timestamp of each transition. Extending the framework with transition-type classification (e.g., cut, dissolve, wipe) ability would provide richer signals for downstream video editing.

Code and Benchmark Availability. Related code, model weights and benchmark will be released at `https://github.com/chence17/TransVLM` after internal review. Project page: `https://chence17.github.io/TransVLM/`.

Acknowledgements

Ce Chen is partially supported by the Melbourne Research Scholarship and the Rowden White Scholarship from the University of Melbourne. Ce Chen gratefully acknowledges the late Sir Alfred Edward Rowden White, whose donation established the latter. Computing resources and proprietary datasets are provided by HeyGen, with engineering support from the HeyGen team.

References

1. Abdulhussain, S.H., Ramli, A.R., Saripan, M.I., Mahmmod, B.M., Al-Haddad, S.A.R., Jassim, W.A.: Methods and challenges in shot boundary detection: a review. Entropy **20**(4), 214 (2018)
2. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966 (2023)
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
4. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: 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. pp. 1728–1738 (2021)
5. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International journal of computer vision **92**(1), 1–31 (2011)
6. Baraldi, L., Grana, C., Cucchiara, R.: A deep siamese network for scene detection in broadcast videos. In: Proceedings of the 23rd ACM international conference on Multimedia. pp. 1199–1202 (2015)
7. Baraldi, L., Grana, C., Cucchiara, R.: Shot and scene detection via hierarchical clustering for re-using broadcast video. In: International conference on computer analysis of images and patterns. pp. 801–811. Springer (2015)
8. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
9. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., et al.: Video generation models as world simulators. OpenAI Blog **1**(8), 1 (2024)
10. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
11. Castellano, B.: Pyscenedetect: Video scene cut detection and analysis tool. Version 0.6. 0 (2014), `https://github.com/Breakthrough/PySceneDetect`, accessed: 2026-06-25
12. Chu, Z., Zhang, L., Sun, Y., Xue, S., Wang, Z., Qin, Z., Ren, K.: Sora detector: A unified hallucination detection for large text-to-video models. arXiv preprint arXiv:2405.04180 (2024)
13. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6202–6211 (2019)
14. Guo, Z., Ye, Z., Ren, Y., Li, Y., Chen, C., Hong, Z., Loy, C.C.: Generate your talking avatar from video reference. arXiv preprint arXiv:2604.27918 (2026)
15. Kar, T., Kanungo, P., Mohanty, S.N., Groppe, S., Groppe, J.: Video shot-boundary detection: issues, challenges and solutions. Artificial Intelligence Review **57**(4), 104 (2024)
16. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J., Zhang, H., Stoica, I.: Efficient memory management for large language model serving with pagedattention. In: Proceedings of the 29th symposium on operating systems principles. pp. 611–626 (2023)

17. Liang, B., Chen, C., Lin, D., Somov, I., Zhao, J., Yuan, J., Zhang, J., Huang, J., Nolte, N., Haqiqi, P., et al.: Avatar v: Scaling video-reference avatar video generation. arXiv preprint arXiv:2606.13872 (2026)
18. Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. In: Proceedings of the 2024 conference on empirical methods in natural language processing. pp. 5971–5984 (2024)
19. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: Bmn: Boundary-matching network for temporal action proposal generation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3889–3898 (2019)
20. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
21. Long, D.X., Wan, X., Nakhost, H., Lee, C.Y., Pfister, T., Arık, S.Ö.: Vista: A test-time self-improving video generation agent. arXiv preprint arXiv:2510.15831 (2025)
22. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. Neurocomputing **508**, 293–304 (2022)
23. Ma, Q., Zheng, Y., Shi, Z., Zhao, Z., Jia, B., Huang, Z., Lin, Z., Li, Y., Yang, J., Peng, Y., et al.: Veomni: Scaling any modality model training with model-centric distributed recipe zoo. arXiv preprint arXiv:2508.02317 (2025)
24. Mu, F., Mo, S., Li, Y.: Snag: Scalable and accurate video grounding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18930–18940 (2024)
25. 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, accessed 2026-06-25
26. Rao, A., Xu, L., Xiong, Y., Xu, G., Huang, Q., Zhou, B., Lin, D.: A local-to-global approach to multi-modal movie scene segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10146–10155 (2020)
27. Salehi, M.R., Park, J.S., Kusupati, A., Krishna, R., Choi, Y., Hajishirzi, H., Farhadi, A.: Actionatlas: A videoqa benchmark for domain-specialized action recognition. Advances in Neural Information Processing Systems **37**, 137372–137402 (2024)
28. Shen, H., Lu, J., Cao, Y., Yang, X.: 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. pp. 706–721 (2025)
29. Soucek, T., Lokoc, J.: Transnet v2: An effective deep network architecture for fast shot transition detection. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 11218–11221 (2024)
30. Souček, T., Moravec, J., Lokoč, J.: Transnet: A deep network for fast detection of common shot transitions. arXiv preprint arXiv:1906.03363 (2019)
31. Su, H., Gan, W., Wu, W., Qiao, Y., Yan, J.: Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 2602–2610 (2021)

32. Tang, S., Feng, L., Kuang, Z., Chen, Y., Zhang, W.: Fast video shot transition localization with deep structured models. In: Asian Conference on Computer Vision. pp. 577–592. Springer (2018)
33. Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
34. Tomar, S.: Converting video formats with ffmpeg. Linux journal **2006**(146), 10 (2006)
35. Tong, W., Song, L., Yang, X., Qu, H., Xie, R.: Cnn-based shot boundary detection and video annotation. In: 2015 IEEE international symposium on broadband multimedia systems and broadcasting. pp. 1–5. IEEE (2015)
36. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Advances in neural information processing systems **35**, 10078–10093 (2022)
37. Wan, Z., Chen, C., Lin, R., Huang, J., Chen, T., Xu, Y., Liu, T., Gong, M.: Mobile-vton: High-fidelity on-device virtual try-on. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 38081–38090 (2026)
38. Wan, Z., Xu, Y., Hu, D., Cheng, W., Chen, T., Wang, Z., Liu, F., Liu, T., Gong, M.: Mft-viton: High-fidelity virtual try-on with minimal input via a mask-free transformer-diffusion model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1985–1994 (2025)
39. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
40. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., et al.: Internvideo: General video foundation models via generative and discriminative learning. arXiv preprint arXiv:2212.03191 (2022)
41. Wu, L., Zhang, S., Jian, M., Lu, Z., Wang, D.: Two stage shot boundary detection via feature fusion and spatial-temporal convolutional neural networks. IEEE Access **7**, 77268–77276 (2019)
42. Xu, H., Ghosh, G., Huang, P.Y., Okhonko, D., Aghajanyan, A., Metze, F., Zettlemoyer, L., Feichtenhofer, C.: Videoclip: Contrastive pre-training for zero-shot video-text understanding. In: Proceedings of the 2021 conference on empirical methods in natural language processing. pp. 6787–6800 (2021)
43. Xu, J., Song, L., Xie, R.: Shot boundary detection using convolutional neural networks. In: 2016 Visual Communications and Image Processing (VCIP). pp. 1–4. IEEE (2016)
44. Zabih, R., Miller, J., Mai, K.: A feature-based algorithm for detecting and classifying scene breaks. In: Proceedings of the third ACM international conference on Multimedia. pp. 189–200 (1995)
45. Zhang, C.L., Wu, J., Li, Y.: Actionformer: Localizing moments of actions with transformers. In: European Conference on Computer Vision. pp. 492–510. Springer (2022)
46. Zhang, Y., Yang, H., Zhang, Y., Hu, Y., Zhu, F., Lin, C., Mei, X., Jiang, Y., Peng, B., Yuan, Z.: Waver: Wave your way to lifelike video generation. arXiv preprint arXiv:2508.15761 (2025)
47. Zhang, Z., Gupta, A., Jiang, H., Singh, H.: Neuflow v2: High-efficiency optical flow estimation on edge devices. arXiv preprint arXiv:2408.10161 **6**, 12 (2024)
48. Zhang, Z., Gupta, A., Jiang, H., Singh, H.: Neuflow-v2: Push high-efficiency optical flow to the limit. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2479–2485. IEEE (2025)

49. Zhang, Z., Jiang, H., Singh, H.: 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). pp. 5048–5055. IEEE (2024)
50. Zhu, W., Huang, Y., Xie, X., Liu, W., Deng, J., Zhang, D., Wang, Z., Liu, J.: 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. pp. 2238–2247 (2023)