OmniShotCut: Holistic Relational Shot Boundary Detection with Shot-Query Transformer

Boyang Wang[1], **Guangyi Xu**[1], **Jiahui Zhang**[1], **Zhipeng Tang**[2], **Zezhou Cheng**[1]
[1]University of Virginia [2]University of Massachusetts Amherst

Project Page: Omni-Shot-Cut.github.io

Abstract

Shot Boundary Detection (SBD) aims to automatically identify shot changes and divide a video into coherent shots. While SBD was widely studied in the literature, existing methods often produce non-interpretable boundaries on transitions, miss subtle yet harmful discontinuities, and rely on noisy, low-diversity annotations and outdated benchmarks. To alleviate these limitations, we propose **OmniShotCut** to formulate SBD as structured relational prediction, jointly estimating shot ranges with *intra*-shot relations and *inter*-shot relations, by a shot query-based dense video Transformer. To avoid imprecise manual labeling, we adopt a fully synthetic transition synthesis pipeline that automatically reproduces major transition families with precise boundaries and parameterized variants. We also introduce OmniShotCutBench, a modern wide-domain benchmark enabling holistic and diagnostic evaluation. Experiments on the benchmarks demonstrate the effectiveness and generality of our method.

1 Introduction

Modern video production is inherently compositional, where multiple shots are assembled through editing operations rather than captured in a single continuous take. The transitions between these shots follow artistic principles, spanning from abrupt hard cuts and jump cuts to gradual effects such as dissolves, fades, wipes, etc. To understand the structural composition of such edited videos, it is necessary to identify the most atomic temporal units, a group of frames that form a coherent shot. This task is known as Shot Boundary Detection (SBD).

Shot Boundary Detection [44, 30, 31, 34] has long been regarded as a well-established problem in video understanding. However, despite its apparent maturity, progress in this area has stagnated. We revisit SBD from the perspective of its downstream applications and ask: has the problem truly been well defined and solved, and is it addressed in the most efficient and scalable manner? We argue that current SBD pipelines remain limited along several practical axes.

First, the predicted shots lack interpretability, as it is unclear whether a predicted boundary corresponds to a scene or an editing transition (see Fig. 1a). For each detected shot, the output should not be limited to a simple temporal range, but should also include higher-level structural information that better supports downstream applications. For instance, in video generation [36, 43, 2], transition may be less critical, and clean vanilla shot segments are often preferred. To this end, we introduce ***intra*-shot** relation classification as the outputs of the model. The intra-label characterizes the shot itself, indicating whether it is a vanilla segment or a specific transition type.

Second, previous SBD models fail to detect subtle yet harmful discontinuities (i.e., sudden jumps) that negatively affect downstream tasks. Sudden jump (see Fig. 1b) introduces excessive abrupt motion or texture change in two consecutive frames, which exert negative influence on motion tracking [18], video segmentation [13], latent video compression [2, 43, 36], and more downstream tasks. To this end, we introduce ***inter*-shot** relation classification. The inter-label captures its relationship with the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

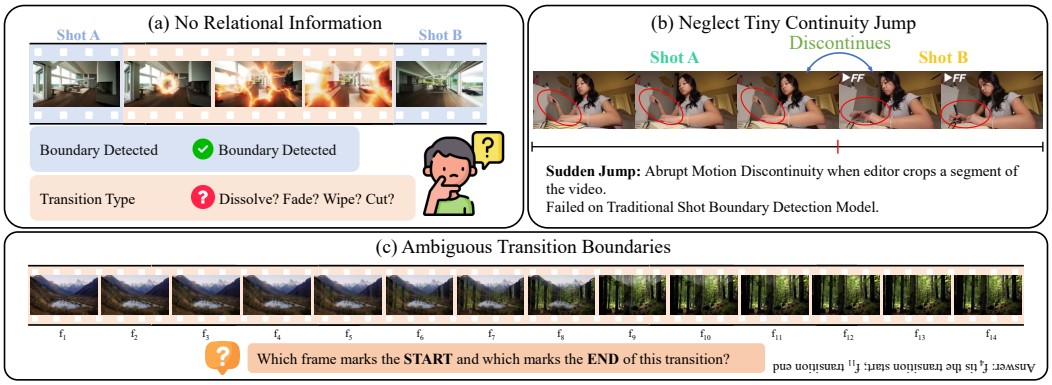


Figure 1: **Limitations of traditional Shot Boundary Detection models.** (a) Detected Shots are hard to interpret: predicted boundaries lack explicit transition semantics; (b) Sudden Jump is under-modeled and often missed; (c) Human annotations are unreliable for gradual transitions with subtle start/end frames.

preceding shot, modeling the cross-shot continuity relationship. Further, existing state-of-the-art SBD models, such as TransNetV2 [30] and AutoShot [44], rely on 3D CNN architectures that are not well-suited for our richer formulation. Instead, we design a shot query-based Transformer architecture that jointly optimizes all objectives through shared hidden states, enabling unified modeling of temporal shot range prediction and relational understanding.

Third, human-supervised annotation struggles to localize subtle changes accurately (see Fig. 1c). Prior works [5, 6, 4] heavily rely on manually annotated real-world data for shot boundary detection. However, transition labeling is highly labor-intensive and inherently imprecise. In particular, humans struggle to accurately localize subtle boundaries, such as the exact start and end frames of fading or dissolve effects, where minor changes in illumination and transparency are difficult to perceive. As a result, manual annotation is not well-suited for fine-grained transition modeling. More importantly, transition effects are in fact generated by video editing software (e.g., Apple iMovie or Adobe professional editing suites). Instead of investing costly human effort in reverse annotation, we propose a forward generation strategy that programmatically reproduces transitions (Fig. 2), which covers 9 main types and 30 subtypes, and yields hundreds of variations by sweeping controllable parameters in directions, edges, intensity, layout, and more. This methodology enables the construction of a synthetic training dataset with precise transition ranges, while covering rare yet realistic cases (e.g., mosaic, puzzle, cube, doorway) that are underrepresented in existing datasets. Furthermore, naively stitching together unrelated videos does not reflect real-world editing patterns. To address this, we leverage a self-supervised learning method to group semantically similar videos from our million-scale video clips pool, thereby simulating more realistic transition contexts.

Fourth, existing benchmarks contain noisy annotations, outdated and narrow domain video sources, and miss the focus of the sudden jump, which fail to reflect the diversity and complexity of modern video content. To close this gap, we introduce **OmniShotCutBench**, a contemporary SBD benchmark built from wide-domain, highly complex sources, with intra- and inter-shot relational labels and, importantly, a confidence scoring system to account for human judgment uncertainty. We hope OmniShotCutBench can offer a more holistic and diagnostic evaluation for modern Shot Boundary Detection.

In summary, our contributions are as follows:

- We reformulate Shot Boundary Detection by enriching each shot with both intra-shot and inter-shot relational information, moving beyond simple temporal range prediction.
- We propose a shot query-based Transformer architecture that jointly optimizes range prediction and relational classification within a unified hidden state.
- We introduce a fully synthetic pipeline with novel self-supervised clustering that groups similar but not identical clips and provides precise, diverse labels without manual annotation.
- We introduce a new benchmark for modern shot boundary detection that captures diverse contemporary transition patterns and sudden jumps, where our model consistently outperforms existing methods across multiple evaluation dimensions.



Figure 2: **Main Transition Types.** We consider a diverse and comprehensive set of video shot transitions that are largely underexplored in prior shot boundary detection works [44, 30, 31]. This figure illustrates several representative transition types, including dissolve, fade, wipe, slide, doorway, zoom, sudden jump, and hard cut. Each example shows the temporal progression from Shot A to Shot B with the transition region highlighted. We skip the Pushing effect demo here. More subtypes are provided in the supplementary.

2 Related Works

Shot Boundary Detection. Shot Boundary Detection [17] operates on native, full video sequence inputs without frame downsampling. It requires frame-level precision to localize each boundary. This inherently demands high-density temporal inputs for the model. Traditional approaches, such as PySceneDetect [9] and Koala-36M [38], primarily rely on handcrafted low-level features (e.g., color histogram differences or structural similarity) to detect abrupt transitions. These methods are sensitive to illumination changes and often struggle to capture higher-level semantic consistency across frames. Deep learning-based methods have since become dominant, including DeepSBD [11], ClipShots [39], TransNetV2 [30], and AutoShot [44]. They employ 3D CNNs to detect transition intervals. To handle long sequences efficiently, these models often downsample spatial resolution aggressively (e.g., to 48×27) to reduce computational cost. To evaluate SBD, even though benchmarks like BBC [5], RAI [6], and IACC3 [4] are popularly used with corresponding training datasets, many shot cut labels lack a clear definition and neglect transitions and subtle motion discontinuity cases like sudden jumps. In addition, these datasets are primarily derived from legacy broadcast footage and do not reflect the diversity of modern video domains.

Downstream Applications. Shot Boundary Detection has become increasingly important in dataset curation for internet-scale in-the-wild videos. In data curation, massive raw long-form videos must be segmented into temporally coherent clips without abrupt changes. The temporal consistency of each clip is critical for training downstream models that demand continuous sequences of images, like the video generation task. In state-of-the-art video generation, videos are encoded by a temporal VAE [43, 36], where multiple frames share the same token. If sudden jumps are not accurately detected, consecutive frames may exhibit drastic spatial shifts (e.g., a subject abruptly moving from left to right), significantly hindering temporal compression and modeling. Moreover, shot boundary detection models are widely used to filter in-the-wild online videos and have been incorporated into numerous recent datasets and benchmark works [21, 22] as the key component in curation preprocessing. In the scene segmentation task, MovieNet [14] provides a dataset that is first cropped by SBD, and then the following works, like LGSS [28], BaSSL [25], ShotCOL [10], and Scene-VLM [7], apply downsampled frames from the predicted shot boundaries to detect scene boundaries. This growing reliance further underscores the need for a more precise, scalable, and transition-aware shot boundary detection model.

Synthetic Data. While supervised training on manually annotated pairs remains a standard approach in computer vision, several domains have noted the difficulty of collecting precisely aligned data in certain tasks. As a result, synthetic data generation strategies [24] have been increasingly adopted, leveraging programmable forward transformations to automatically construct large-scale labeled

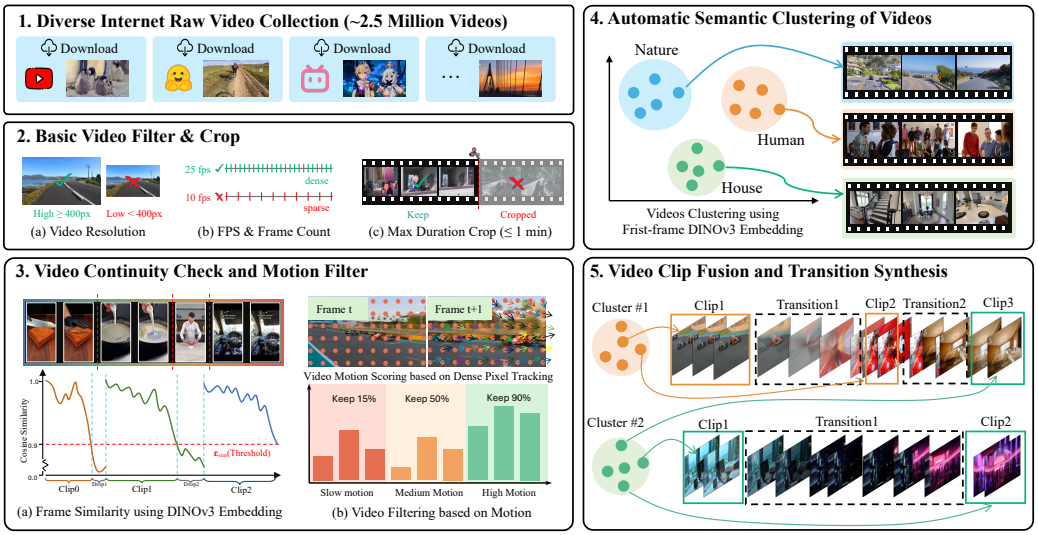


Figure 3: **Large-scale transition source video curation.** (1) We collect ∼2.5M raw videos from diverse Internet sources. (2) Videos are filtered based on resolution, frame rate, and duration constraints. (3) Temporal continuity and motion strength are verified using frame-level semantic similarity and dense motion tracking. (4) Remaining videos are automatically clustered using the SSL data curation method [35] to group semantically similar videos. (5) Finally, video clips in the same and different clusters are fused to synthesize large-scale shot boundary detection training datasets.

datasets. A representative example arises in low-level vision [40, 16, 37], where perfectly pixel-aligned degraded inputs and original high-quality pairs are rarely available in real-world settings. Instead, their degraded images or videos are synthesized via degradation to high-quality ground-truth sources. Similarly, prior research [39] in image forensics and editing detection has leveraged scripted pipelines to automatically synthesize Photoshop-manipulated images. For transition modeling, where effects are typically synthesized by editing software, a programmatic synthesis strategy is therefore a principled and effective solution. Though previous works like TransnetV2 [30] and DeepSBD [11] mix real data with synthesized hard-cut and dissolve, most transition effects remain underexplored, and we extend to dozens of transitions available. More importantly, we explore the extent to which purely synthetic training data can push the limits of synthetic supervision.

3 Method

Problem Formulation. We define the problem as empowering traditional Shot Boundary Detection within an end-to-end model such that it not only predicts the temporal ranges of each shot, but also outputs the intra-relation classification of the shot itself and the inter-relation classification with respect to the previous shot. For intra-relation, we include 8 major categories, which includes vanilla General video, Dissolve, Wipe, Push, Slide, Zoom, Fade, and Doorway. For inter-relation, we classify whether the boundary corresponds to a Transition, a Hard Cut, or a Sudden Jump.

3.1 Automatic Video Transition Synthesis

To synthetically construct shot transitions, a clean video source pool is crucial. Our curation pipeline is shown in Fig. 3. We first apply basic parameter filtering on duration, resolution, frames-per-second (fps), and aspect ratio to all collected in-the-wild video sources and crop them into segments with a maximum duration of 1 minute. To conservatively extract fluent video segments that contain no abrupt content change, inspired by motion evaluation of VBench [15], we encode frames sampled at a constant interval into DINO [29] embeddings, and compute the cosine similarity between consecutive embeddings. If the cosine similarity is higher than the threshold ε_{sim}, this indicates that the two frames are semantically consistent and do not exhibit an abrupt hard cut or transitions changing. We continue this process until the similarity below ε_{sim}, at which point the clip is terminated. And then we refresh the cache and start finding the next video clip. Empirically, we observe that this approach

is effective at identifying fading dark frames and dissolve-induced transparent frames, cropping out early when transitions start happening. Although this approach cannot guarantee that every extracted clip is completely clean, in subsequent synthetic transition generation, training with a large proportion of correctly labeled data can effectively mitigate the influence of noisy labels.

In this paper, identifying the *Sudden Jump* is a critical task. Sudden Jump typically arise during video editing where a short period of segments is manually cropped, resulting in abrupt discontinuities, where the video cannot be regarded as a fluent shot. Thus, we believe that sudden jump is aligned with the purpose of the shot boundary detection task and should be solved in this domain. To incorporate into our shot boundary detection framework, we explicitly estimate the motion strength during the data curation stage. This enables us to select clips with medium-level motion intensity (neither too fast nor too slow) as suitable sources for constructing sudden jump samples. We estimate motion strength using the CoTracker3 [18] model, which provides dense tracking points with configurable grid density. By measuring the displacement magnitude of tracked points across frames and averaging over all frames, we obtain an overall motion strength score for each video. Further, another functionality of motion strength information is that it can be helpful to increase video clip pool complexity. This is because we observe that a large portion of raw clips exhibit small motion magnitudes; therefore, we filter out these slow-motion cases with weak dynamic patterns.

Once each video clip is properly curated, we want to group **similar but not identical** videos into the same cluster, such that the sources before and after a synthetic transition can be sampled with high similarity to better simulate real-world video. We adopt the DINO representation and perform Self-Supervised Learning-based (SSL) clustering following the methodology of [35]. For each video clip, we extract the DINO embedding of its first frame, directly reusing the embeddings computed during the earlier curation stage. And then, we apply a semantic deduplication [1] paradigm to filter instances whose cosine similarity is less than the threshold ε_{dup} in each cluster. This avoids near-duplicate videos when we are using large-scale random videos from the internet. Finally, we apply hierarchical K-means [35] clustering to group semantically similar embeddings. Each cluster represents a collection of videos sharing similar semantic content, like indoor scenes, vehicles, housing, mountains, etc. The SSL method does not directly tell us what the content is, but it can ensure that the contents are perceptually similar and strongly related.

Synthetic Transition Composition. After the curation stage, we obtain a large collection of clean video clips. We then randomly choose videos from this pool and stitch clips together with diverse synthetic transitions (as shown in Fig. 2 and Fig. 7). In the synthesis, most video clips are sampled from the same SSL-grouped cluster pools as previous videos, while we also allow cross-cluster selection to reflect the unpredictability of real-world video editing (see Fig. 3 (5)). For sudden jump cases, we restrict the video source selection to middle motion strength. Excessive camera or object motion often induces large structural changes, making it difficult to distinguish from hard cuts, whereas some small motion cases (e.g., static talk-show scenarios) yield changes that are barely perceptible even to humans. The synthesis settings are detailed in the supplementary.

3.2 Shot Query-based Dense Video Transformer

As shown in Fig. 4, we propose a Shot Query-based end-to-end video Transformer model, which is composed of the image encoder, Transformer encoder, and Transformer decoder. We start from the image-based DETR [8] object detection Transformer model and introduce critical modifications for our task. The input consists of video frames of length F, height H, and width W, forming a tensor in $\mathbb{R}^{F\times H\times W\times C}$. The video is first encoded by ResNet [12] as an image encoder in a frame-by-frame manner.

The encoded per-frame features are fed into the Transformer encoder. We flatten the spatial and temporal dimensions into a single dimension, resulting in a feature map of size $\mathbb{R}^{d\times(F\cdot H\cdot W)}$, where d is the hidden state dimension. Each Transformer encoder layer is composed of multi-head self-attention. Since our input shifts from images to videos, the tokens in the Transformer remain permutation-invariant by design, we need to extend the 2D position embedding to 3D position embedding, introducing additional positional information along the temporal dimension. Specifically, following the approach of VisTR [41], we generalize the cumulative spatial coordinates (x, y) to 3D (t, x, y) and apply sinusoidal embeddings along the temporal and spatial axes, enabling the Transformer to model joint spatiotemporal relationships in video inputs. The 3D position embedding

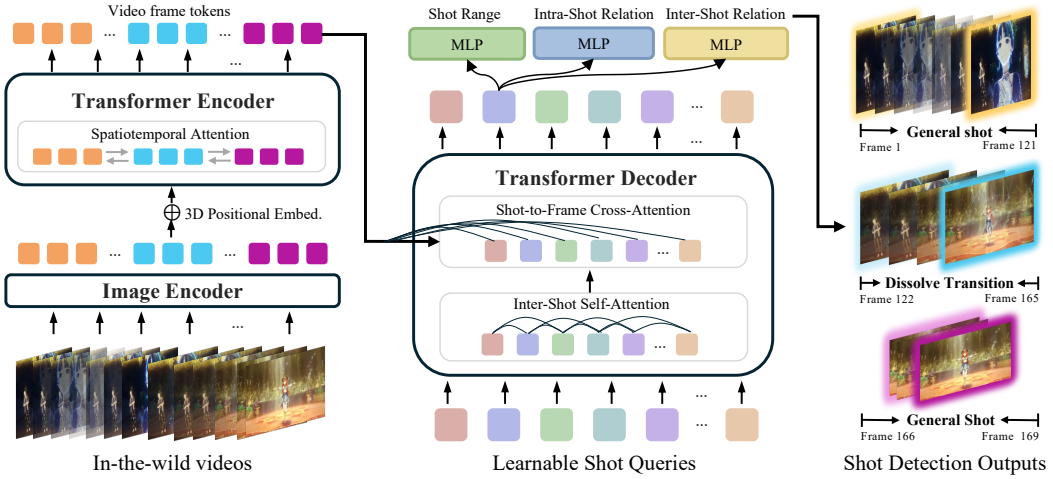


Figure 4: **Shot Query-based Dense Video Transformer.** Frame tokens from input videos are encoded using a spatiotemporal Transformer encoder with 3D positional embedding. Learnable shot queries in the decoder interact with frame features through cross-attention to predict shot range, intra-shot relation, and inter-shot relation.

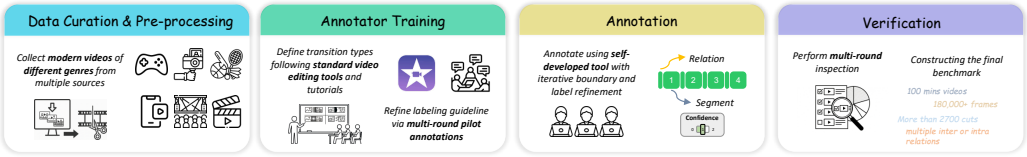
will also be flattened to $\mathbb{R}^{d\times(F\cdot H\cdot W)}$ and then added with the flattened video tokens before entering the Transformer encoder.

The input to the Transformer decoder is a fixed-length set of trainable embeddings, referred to as *shot queries*. At a high level, each shot query serves as a shot prediction slot, aggregating shot-specific evidence from the video sources into a compact hidden state for decoding. The entire Transformer decoder consists of multiple decoder layers. In each layer, the shot queries first undergo self-attention, followed by cross-attention with the tokens $\mathbb{R}^{d\times(F\cdot H\cdot W)}$ produced by the Transformer encoder. The number of input shot queries is fixed. At the output stage, shot queries will predict a dedicated termination token to explicitly indicate the end of shot prediction when it reaches the last shot. All queries after the termination token are discarded, and only the preceding ones are considered valid predictions.

Each shot query on the output of our Transformer decoder is passed through three heads: a range head, an intra-relation head, and an inter-relation head. Directly adopting DETR-style [20] formulation by replacing bounding box prediction with an L_1 + 1D GIoU regression loss results in a suboptimal learning objective for temporal range prediction (check Sec. 4.3). Regression over normalized continuous coordinates is inherently ill-suited for accurate frame-level boundary localization across long sequences, where even a one-frame deviation at a hard-cut transition constitutes a significant error for the SBD task. To address this limitation, we reformulate range prediction as a discrete classification problem over frame indices, which provides improved localization precision and more stable optimization. Specifically, the range head predicts the index of the last frame of each shot p^{end}, formulated as a classification problem where the number of classes equals the total number of frames. As shots in SBD are consecutive and non-overlapping, the start of each shot is implicitly defined by the end of the previous one, with the first shot starting at frame 0. This classification formulation for the range prediction does not require post-processing of heuristic thresholding like prior SBD methods [30, 44]. Consequently, Hungarian matching [19] is no longer required. We retain auxiliary supervision at intermediate decoder layers to facilitate stable optimization.

3.3 Evaluation Benchmark

We introduce OmniShotCutBench, a modern shot boundary detection benchmark designed to comprehensively evaluate models' performance on versatile transitions from modern internet video sources. Each range label is paired with a confidence score. This is because we recognize that human perception is inherently insensitive to subtle transition variations, particularly in transparent dissolve and fading effects. OmniShotCutBench is constructed entirely from real edited videos and is disjoint from all synthetic training sources. No synthetic transitions generated by our pipeline are included in the benchmark. The construction pipeline and statistics comparison is shown in Fig. 5.



Dataset	Statistics				Annotations			Content
	# Videos	Duration (min)	Avg Len (s)	# Cuts	Transition Type	Inter-shot Relation	Confidence	Genre Diversity
BBC	11	550	6.57	5,000	✗	✗	✗	Low
RAI	10	100	5.65	1,060	✗	✗	✗	Low
AutoShot	853	560	2.59	11,606	✗	✗	✗	Medium
OmniShotCut Bench	114	114	2.09	3,091	✓	✓	✓	High

Figure 5: OmniShotCutBench Pipeline and Data Statistics.

We collect diverse video sources with modern video editing techniques from the topics of vlog, anime, movie, concert, documentary, monitor recording, game, sports, etc. We randomly truncate the videos to one minute (or shorter) and standardize all videos to 480p resolution at 30 FPS to ensure consistent temporal precision, which is critical for accurate shot boundary localization. In total, we curate 114 videos, which is 114 minutes of diverse, high-quality, and representative video sources.

To ensure high-quality annotations, following other datasets & benchmark works [22, 23], we mimic their high-standard curation paradigm, as shown in Fig. 5. All annotators first studied several professional video editing tutorials online and learning video editing applications like iMovie. Annotators were required to review these materials prior to labeling to establish a clear understanding of transition taxonomy and visual characteristics. We then conducted multiple rounds of pilot annotations to align labeling criteria and ensure consistency across annotators. During the final annotation phase, ambiguous or contentious cases were systematically documented and resolved to maintain annotation quality and consistency. Analysis and visualization of our benchmark are provided in the supplementary material. We will open-source this benchmark for future research.

4 Experiment

4.1 Implementation Details

We train our model with 8 Nvidia A100 GPUs for 50 epochs. Since training resolution is only 128x96, we choose the smallest pretrained ResNet18 [12] as the image encoder to maintain more spatial tokens after the encoder. We set 3 Transformer encoders and 6 Transformer decoders for our model, which is 34.49M parameters, out of which 11.17M is the ResNet backbone. We set 24 fixed learnable shot query tokens as the input to the Transformer decoder, corresponding to the maximum number of shots observed in a 100-frame training clip. The learning rate for the ResNet backbone is set to $1e-5$, and the Transformer encoder and decoder learning rate is $1e-4$. Total batch size across all GPUs is 64. We randomly crop 100 frames of the full video source for the training, following the 100-frame input window of TransNet V2 [30] and AutoShot [44]. In the training, we do several online augmentations. This includes horizontal and vertical flip, color jittering, blurring, Gaussian and Poisson noises, and compression artifacts [37]. More implementation details are provided in the supplementary.

4.2 Experiment Results

Though our model jointly outputs the relational label, our capability is mainly inherited from shot boundary detection. To evaluate traditional shot boundary detection, we choose to compare with mainstream baselines in the literature, which include non-learning-based method PySceneDetect [9], and previous state-of-the-art learning-based methods by 3D CNNs, TransNet V2 [30], and AutoShot [44].

The evaluation is done on OmniShotCutBench and legacy benchmark BBC Planet Earth Documentation [5]. Our evaluation considers traditional shot range precision, recall, and F1 metrics, which are based on the ShotBench in Cosmos [2]. We use their default tolerance of 2 frames. Further, we specialize in the analysis of the transition of the IoU and sudden jump hit rate. For transition IoU, we select the GT shots that are labeled with a transition label, and find the closest prediction

Table 1: Quantitative comparison with existing shot boundary detection methods. We report results on the OmniShotCut Bench and the legacy BBC benchmark [5]. For our benchmark, we additionally evaluate transition localization, sudden jump detection, and relation classification. All metrics are higher the better. The best is **highlighted**.

Method	OmniShotCut Bench							BBC Bench		
	Trans. IoU	Sudden Jump Hit Rate	Shot Range			Relation Acc.		Shot Range		
			Precision	Recall	F1	Intra	Inter	Precision	Recall	F1
PySceneDetect	0.183	0.417	0.833	0.689	0.755	-	-	0.893	0.884	0.889
TransNet V2	0.193	0.262	**0.913**	0.735	0.814	-	-	0.983	0.951	0.967
AutoShot	0.253	0.455	0.849	0.783	0.815	-	-	**0.984**	0.922	0.952
Ours	**0.644**	**0.759**	0.904	**0.859**	**0.881**	**0.962**	**0.827**	0.967	**0.974**	**0.971**

results to calculate the IoU, applying our human label confidence to dynamically adjust the tolerance range. For sudden jump hit rate, we identify all ground-truth inter-relation labels corresponding to sudden jumps and measure the proportion of them at which a shot cut is predicted at the same frame index (i.e., the recall of sudden jumps). A zero tolerance is applied, as the transition is expected to occur instantaneously. False positive cuts are penalized by the range precision, which counts every predicted boundary. We further evaluate intra- and inter-relation classification accuracy, which is the number of correct classifications divided by the total number of shots. For each ground-truth shot, the predicted shot with the highest IoU is selected and used for classification comparison. All metrics count across all videos in the benchmark, instead of the average per video. This is because the number of cuts is unbalanced in different videos.

Tab. 1 reports the quantitative results. Traditional shot boundary detection methods such as PySceneDetect [9], TransNetV2 [30], and AutoShot [44] achieve reasonable performance on overall range-based metrics, with F1 scores between 0.75 and 0.82. However, they exhibit clear limitations in transition localization and sudden jump detection. In particular, transition IoU remains low (0.18–0.25), indicating that predicted boundaries are often only roughly aligned with the true transition ranges. Sudden jump hit rate is also limited, suggesting difficulty in reliably detecting instantaneous discontinuities. In contrast, our method significantly improves transition localization, achieving a transition IoU of 0.644, substantially outperforming all baselines. It also achieves the best range F1 score of 0.881. Moreover, our framework enables structured relation prediction, reaching 0.962 intra-shot accuracy and 0.827 inter-shot accuracy, which are not supported by prior methods. In the legacy BBC benchmark [5] comparison, we applied an overlap window of 10 between processing each clip and merging transition labels to match traditional hard-cut only output, and we achieved the best range recall and F1. The visual result is available in the supplementary.

4.3 Ablation Study

In this section, we conduct ablation studies to examine the impact of key components and hyperparameter influence on our performance. Results are summarized in Tab. 2.

For the first study, we examine whether the DETR-style range regression objective, L_1 + 1D GIoU [8], is preferable to our proposed formulation, which converts boundary estimation to discrete classification. In our model, we directly predict discrete boundary labels like classification. For the L_1 + 1D GIoU variant, the model outputs are passed through a sigmoid to map into $[0, 1]$ range, and then scaled by the maximum prediction length (100 frames in our training). Default 2D GIoU is changed to a 1D version, based on [20]. As shown in Tab. 2 (a), L_1 + 1D GIoU can slightly improve transition IoU, but it degrades under stricter criteria, such as zero-tolerance sudden jump hit rate and range F1. We attribute this drop to the inherent difficulty of regression losses in resolving the last 1-2 frames precisely, which is crucial for abrupt hard-cut and sudden-jump boundaries.

In Tab. 2 (b), we study the influence of sampling transition sources from the same SSL-curated clusters [29, 35] versus purely random selection. In our base setting, we follow [35] and sample transition sources from the same cluster with 75% probability. Hereby, we construct a variant where all clips are selected at random. *Pure random sampling consistently degrades performance across all metrics.* We hypothesize that semantically aligned clips produce more challenging synthesized

Table 2: Ablation studies on key design choices. The best result within each ablation group is **highlighted** in bold.

(a) Loss Design

Loss	Trans. IoU	Sudden Jump Hit Rate	Range F1	Intra / Inter Acc.
Classification	0.644	**0.759**	**0.881**	**0.962 / 0.827**
L_1 + 1D GIoU	**0.718**	0.717	0.857	0.956 / 0.809

(b) SSL Data Clustering

Setting	Trans. IoU	Sudden Jump Hit Rate	Range F1	Intra / Inter Acc.
With	**0.644**	**0.759**	**0.881**	**0.962 / 0.827**
Without	0.551	0.664	0.861	0.957 / 0.722

(c) Short Dense Cuts Distribution

Setting	Trans. IoU	Sudden Jump Hit Rate	Range F1	Intra / Inter Acc.
With	**0.644**	**0.759**	**0.881**	**0.962 / 0.827**
Without	0.610	0.509	0.831	0.955 / 0.790

(d) Encoder Quantity

Quantity	Trans. IoU	Sudden Jump Hit Rate	Range F1	Intra / Inter Acc.
3 enc.	**0.644**	0.759	**0.881**	**0.962 / 0.827**
6 enc.	0.598	**0.774**	0.879	0.955 / 0.815

(e) Training Resolution

Resolution	Trans. IoU	Sudden Jump Hit Rate	Range F1	Intra / Inter Acc.
64×96	0.582	0.699	0.877	0.959 / 0.818
96×128	**0.644**	**0.759**	**0.881**	**0.962 / 0.827**
128×160	0.584	0.723	0.867	0.954 / 0.794

(f) Number of Shot Queries

Number	Trans. IoU	Sudden Jump Hit Rate	Range F1	Intra / Inter Acc.
12	0.556	0.735	0.876	0.953 / **0.833**
24	**0.644**	**0.759**	**0.881**	**0.962** / 0.827
48	0.590	0.753	0.873	0.959 / 0.811

transitions during training, requiring the model to leverage fine-grained temporal and structural cues to identify subtle content changes. In contrast, randomly paired clips often exhibit large semantic discrepancies, making the discrimination task comparatively trivial.

Next, we aim to study whether modeling continuous dense hard cuts and the sudden jump during the data synthesis is beneficial. Under purely random synthesis, fewer than 0.005% of generated samples contain more than five consecutive hard cuts, despite such patterns being common in real-world videos. Therefore, in our base setting, we enforce continuous hard cuts in 25% of the synthesized videos. Removing this strategy leads to consistent performance degradation across all metrics, as shown in Tab. 2 (c). These results suggest that understanding real-world shot distributions and designing synthesis pipelines that better reflect such distributions are important for effective model training.

For the architectural ablations in Tab. 2 (d)–(f), the base configuration provides the most balanced overall performance. Increasing the number of encoder layers improves sudden jump hit rate, but leads to noticeable degradation in most other metrics. A training resolution of 96×128 gives the best resolution trade-off. For the number of shot queries, using 24 queries yields the strongest overall results, achieving the best sudden jump hit rate and range F1, whereas 12 and 48 queries only improve transition IoU and inter-label accuracy, respectively.

5 Conclusion

In this paper, we present **OmniShotCut**, reformulating shot boundary detection with explicit *intra*-shot and *inter*-shot relations by a shot query-based Transformer framework. To overcome the limitations of manual annotation, we develop a fully synthetic data synthesis pipeline that automatically produces diverse transition effects with precise temporal supervision, and curate a modern complex shot boundary detection benchmark. Experiments demonstrate that our approach achieves superior performance across multiple benchmarks and evaluation metrics. Our results suggest that fully synthetic supervision provides a scalable and effective paradigm for next-generation shot boundary detection datasets. Moreover, our insights into intra- and inter-shot relations may further benefit downstream applications that require more accurate and explainable shot boundary detection.

Limitations and Future Work. Despite the strong performance, more sophisticated artistic and semantically dynamic transitions may require additional modeling beyond our current synthetic parameterization. In particular, capturing complex cinematic transition patterns could benefit from

large-scale industry-level transition template collections, which are not publicly available. Exploring such resources remains an interesting direction for future work.

Acknowledgments and Disclosure of Funding

The authors acknowledge the Adobe Research Gift, the University of Virginia Research Computing and Data Analytics Center, Advanced Micro Devices AI and HPC Cluster Program, Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, and National Artificial Intelligence Research Resource (NAIRR) Pilot for computational resources, including the Anvil supercomputer (National Science Foundation award OAC 2005632) at Purdue University and the Delta and DeltaAI advanced computing resources (National Science Foundation award OAC 2005572). The authors also acknowledge the NVIDIA Academic Grant Program Award for providing the GPU resources used in this work.

References

[1] Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

[2] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[3] Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, et al. Llava-onevision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*, 2026.

[4] George Awad, Asad A Butt, Jonathan Fiscus, David Joy, Andrew Delgado, Willie Mcclinton, Martial Michel, Alan F Smeaton, Yvette Graham, Wessel Kraaij, et al. Trecvid 2017: evaluating ad-hoc and instance video search, events detection, video captioning, and hyperlinking. In *TREC video retrieval evaluation (TRECVID)*, 2017.

[5] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1199–1202, 2015.

[6] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Shot and scene detection via hierarchical clustering for re-using broadcast video. In *International conference on computer analysis of images and patterns*, pages 801–811. Springer, 2015.

[7] Nimrod Berman, Adam Botach, Emanuel Ben-Baruch, Shunit Haviv Hakimi, Asaf Gendler, Ilan Naiman, Erez Yosef, and Igor Kviatkovsky. Scene-vlm: Multimodal video scene segmentation via vision-language models. *arXiv preprint arXiv:2512.21778*, 2025.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[9] Brandon Castellano. Pyscenedetect: Python and opencv-based scene cut/transition detection program & library. `https://github.com/Breakthrough/PySceneDetect`, 2025. Software.

[10] Shixing Chen, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. Shot contrastive self-supervised learning for scene boundary detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9796–9805, 2021.

[11] Ahmed Hassanien, Mohamed Elgharib, Ahmed Selim, Sung-Ho Bae, Mohamed Hefeeda, and Wojciech Matusik. Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks. *arXiv preprint arXiv:1705.03281*, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Hengrui Hu, Kaining Ying, and Henghui Ding. Segment anything across shots: A method and benchmark. *arXiv preprint arXiv:2511.13715*, 2025.

[14] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European conference on computer vision*, pages 709–727. Springer, 2020.

[15] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[16] Mehran Jeelani, Noshaba Cheema, Klaus Illgner-Fehns, Philipp Slusallek, Sunil Jaiswal, et al. Expanding synthetic real-world degradations for blind video super resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1199–1208, 2023.

[17] 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.

[18] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6013–6022, 2025.

[19] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[20] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021.

[21] Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, et al. Sekai: A video dataset towards world exploration. *arXiv preprint arXiv:2506.15675*, 2025.

[22] Zhiqiu Lin, Siyuan Cen, Daniel Jiang, Jay Karhade, Hewei Wang, Chancharik Mitra, Tiffany Ling, Yuhan Huang, Sifan Liu, Mingyu Chen, et al. Towards understanding camera motions in any video. *arXiv preprint arXiv:2504.15376*, 2025.

[23] Hongbo Liu, Jingwen He, Yi Jin, Dian Zheng, Yuhao Dong, Fan Zhang, Ziqi Huang, Yinan He, Yangguang Li, Weichao Chen, et al. Shotbench: Expert-level cinematic understanding in vision-language models. *arXiv preprint arXiv:2506.21356*, 2025.

[24] Alhassan Mumuni, Fuseini Mumuni, and Nana Kobina Gerrar. A survey of synthetic data augmentation methods in machine vision. *Machine Intelligence Research*, 21(5):831–869, 2024.

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

[26] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

[27] Zhenglin Pan. Sakuga-42m dataset: Scaling up cartoon research. *arXiv preprint arXiv:2405.07425*, 2024.

[28] 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.

[29] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[30] 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.

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

[32] Mohammad Reza Taesiri, Finlay Macklon, and Cor-Paul Bezemer. Clip meets gamephysics: Towards bug identification in gameplay videos using zero-shot transfer learning. In *Proceedings of the 19th International Conference on Mining Software Repositories*, pages 270–281, 2022.

[33] Zhiyu Tan, Xiaomeng Yang, Luozheng Qin, and Hao Li. Vidgen-1m: A large-scale dataset for text-to-video generation. *arXiv preprint arXiv:2408.02629*, 2024.

[34] 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.

[35] Huy V Vo, Vasil Khalidov, Timothée Darcet, Théo Moutakanni, Nikita Smetanin, Marc Szafraniec, Hugo Touvron, Camille Couprie, Maxime Oquab, Armand Joulin, et al. Automatic data curation for self-supervised learning: A clustering-based approach. *arXiv preprint arXiv:2405.15613*, 2024.

[36] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[37] Boyang Wang, Bowen Liu, Shiyu Liu, and Fengyu Yang. Vcisr: Blind single image super-resolution with video compression synthetic data. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 4302–4312, 2024.

[38] Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, et al. Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8428–8437, 2025.

[39] Sheng-Yu Wang, Oliver Wang, Andrew Owens, Richard Zhang, and Alexei A Efros. Detecting photoshopped faces by scripting photoshop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10072–10081, 2019.

[40] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021.

[41] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8741–8750, 2021.

[42] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[43] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[44] 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.

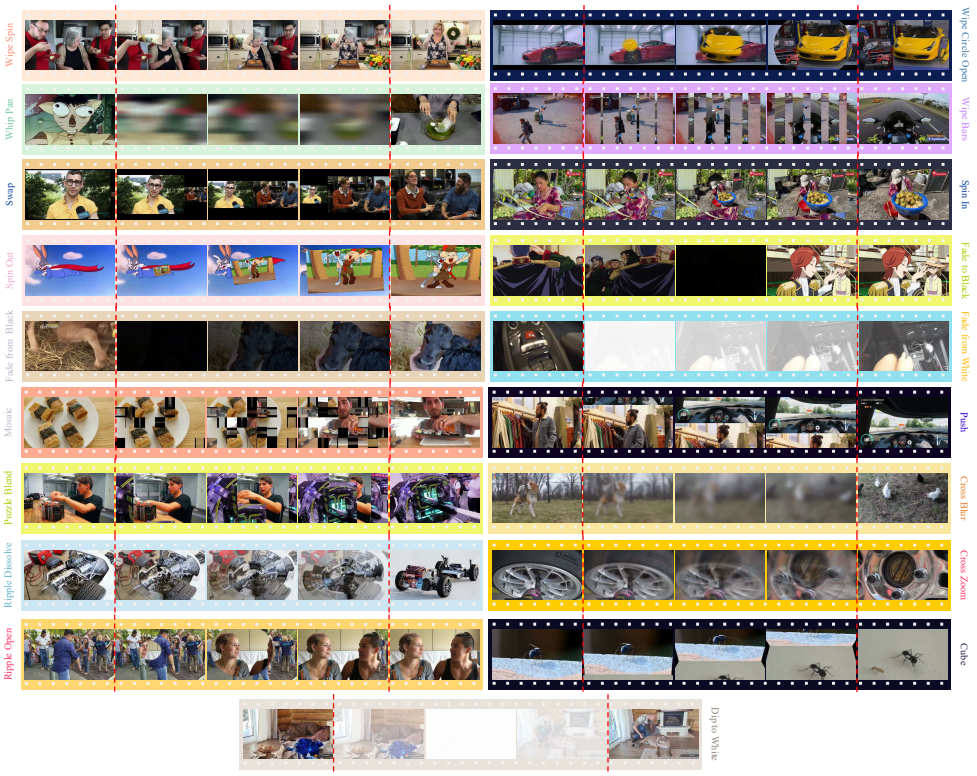


Figure 6: **Full Transition Types**. We complement the transition types listed in the main paper section.

Supplementary Material

This supplementary material provides more implementation and technical details and additional qualitative visualization to complement the main manuscript. In Sec. A, we present full transition genre types. In Sec. B, we present more information about the transition synthesis parameter setting. In Sec. C, we present more implementation details for the model training and setting. In Sec. D, we compare with general-purpose VLMs. In Sec. E, we present the benchmark annotation GUI and details. In Sec. F, we present the visual comparisons between different models.

A Full Transition Genre Types

In the main paper, we provide the main transition type visualization. However, this is still not all the types we consider. We consider numerous subtypes in transition and classify them into different categories based on the pattern. The visualization is shown in Fig. 6.

We consider a diverse taxonomy of editing transitions covering both common and fine-grained variants. Specifically, the transition set includes Dissolve transitions (Transparent Dissolve, Cross-Blur Dissolve, and Ripple Dissolve); Wipe transitions (Unidirectional Wipe, Diagonal Wipe, Circular Wipe, Bar Wipe, Ripple Wipe, Page-Curl Wipe, and Mosaic Wipe); Push transitions (Unidirectional Push and Puzzle Push); Slide transitions (Horizontal Slide, Whip-Pan Slide, and Cube Slide); Zoom transitions (Zoom In/Out, Spin In/Out, Cross Zoom, and Swap Zoom); Fade transitions (Fade to Black, Fade to White, Fade from Black, Fade from White, Dip to Black, and Dip to White); and Doorway transitions (Doorway Open).

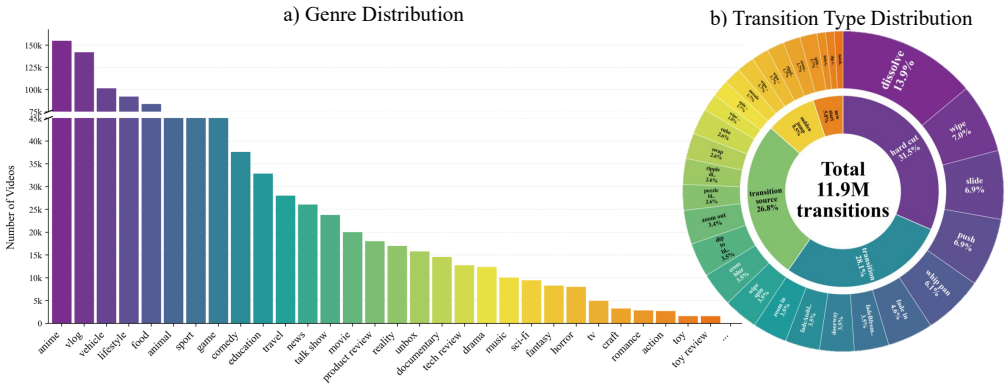


Figure 7: **Left:** Our curation pipeline scales to internet-scale, wide-domain video collection and yields ∼1.5M curated clip source for transition synthesis; genres are annotated by Qwen3 [42]. **Right:** Transition statistics of the synthetic corpus. The inner ring shows the inter-shot relation distribution, and the outer ring breaks down all of the main and sub-transition types that our pipeline can synthesize. In total, we synthesize 11.9M transitions for training.

B Transition Synthesis Details

In this section, we share our transition synthesis details, which are the key to our training dataset construction. A preview can be found in Fig. 7.

In the curation stage, our video sources are mostly coming from existing video datasets on Huggingface, which include datasets like OpenVid [26], VidGen [33], Sakuga [27], GamePhysics [32], and several publicly accessible sources. we set continuous similarity threshold ε_{sim} to 0.9 and deduplication threshold ε_{dup} to 0.05. The motion tracking [18] is sampled per 3 frames at 256x320 resolution. The number of SSL clusters [35] is set to 27,000, where we use DINOv3 [29] ViT large variant. We discard cluster sizes that are less than 5 videos to avoid reusing the same video source.

We construct our synthetic transition training data via a fully parameterized pipeline. The number of clips per video is sampled from a Poisson distribution with $\lambda = 7.0$ and constrained to $[1, 28]$. Clip durations are sampled from Gaussian distributions of $\mathcal{N}(2.8, 1.6^2)$ seconds. 75% of clips are selected from the same DINOv3 [29] cluster to maintain semantic coherence. For sudden-jump cases, we crop [24, 40] frames, and their valid source videos should be those with motion strength in the [25, 60] percentile range, sorted from slowest to fastest. We further assign 25% of the synthesis to be extremely short and dense composition, where we generate continuous 28 videos that are within the duration of $[0.15, 1.0]$ seconds for each clip. For the offline augmentation, we apply 5% on adding subtitle text, and 7.5% on the lighting variations. In total, we create 300K synthetic videos used for training, where each of them contains at least 240 frames at 24 fps. The quantity of synthesis videos can be infinite, but we set it to 300K videos as a reasonable range.

The distribution of each transition is different. We sample 35% for the hard cut. In the dissolve type transition, 9.4% is distributed for the transparent dissolve, 2.4% for the cross-blur dissolve, and 1.8% for the ripple dissolve. In the wipe type transition, 4.7% is distributed evenly for the vanilla wipe in up, down, bottom, and right directions, 2.4% for spin wipe, 2.4% for circle open and close wipe, 1.2% for the bar wipe, 1.2% for the ripple wipe, and 1.2% for the mosaic wipe. In the push-type transition, 4.7% is distributed evenly for the vanilla push in up, down, bottom, and right directions, and 1.8% for the puzzle blending push. In the slide-type transition, 4.7% is distributed evenly for the vanilla slide, 4.1% for the whip pan, and 1.8% for the cube slide. In the zoom-type transition, 2.4% is distributed for the zoom in, 2.4% for the zoom out, 2.4% for the spin in and out, 1.2% for the cross zoom, and 1.8% for the swap. In the fading-type transition, 2.9% is distributed for fading the first source to black or white screen, 2.9% for fading the second source from black or white screen, and 2.9% for the dip fading effect. In the doorway transition, 2.9% is distributed.

For all transition types, we carefully control as many parameters as possible to ensure consistency and precise manipulation. This yields a large set of explicit control philosophies spanning (i) discrete mode switches (e.g., transition direction, hard vs. soft edges, constant vs. linear smoothing), (ii) temporal controls (including the start time, duration, and the speed curve of the transition over time), (iii) spatial controls (anchor locations and margins for added text, effect centers for zoom/ripple, grid

resolution for mosaic, doorway seam orientation), and (iv) intensity controls (blurring range and curve shape, zoom magnitude and sampling density, lighting gains/gamma/contrast and color wash/spotlight strength, feather widths for soft boundaries). Additionally, we control content-level factors such as text selection and layout (wrapping, line count, spacing), while dropping near-duplicate frames on the edge of transition phases. Overall, our transition synthesis pipeline defines a reproducible distribution over diverse transitions with fine-grained, interpretable parameters that can be tuned. To implement the transition, we apply LLM as an auxiliary tool, but humans do the final check to ensure that the transition is correct.

C More Implementation Details

For the model training, we optimize a weighted sum of three classification losses:

$$\mathcal{L} = \lambda_{\text{range}}\,\mathcal{L}_{\text{range}} + \lambda_{\text{intra}}\,\mathcal{L}_{\text{intra}} + \lambda_{\text{inter}}\,\mathcal{L}_{\text{inter}}, \tag{1}$$

where

$$\mathcal{L}_{\text{range}} = \frac{1}{N}\sum_{i=1}^{N} \text{CE}\big(p_i^{\text{end}},\ y_i^{\text{end}}\big)\,, \tag{2}$$

$$\mathcal{L}_{\text{intra}} = \frac{1}{N}\sum_{i=1}^{N} \text{CE}\big(p_i^{\text{intra}},\ y_i^{\text{intra}}\big)\,, \tag{3}$$

$$\mathcal{L}_{\text{inter}} = \frac{1}{N}\sum_{i=1}^{N} \text{CE}\big(p_i^{\text{inter}},\ y_i^{\text{inter}}\big)\,. \tag{4}$$

In the experiment, $\lambda_{\text{range}}, \lambda_{\text{intra}}, \lambda_{\text{inter}}$ is set to 5, 1, 1.

D Comparison with General-Purpose VLMs

We prompt each VLM to output shot-boundary timestamps on the 114-video OmniShotCutBench, using each model's recommended frame sampling (2 fps for Qwen3-VL-8B [42]; native codec input for LLaVA-OneVision-2-8B [3]), convert the predicted times to frame indices, and evaluate with the same metrics as Tab. 1. Both VLMs reach only about 0.13 Range F1, showing that general-purpose VLMs are not suited to frame-accurate temporal grounding and motivating a task-specific dense Transformer.

Table 3: Comparison with general-purpose VLMs on OmniShotCutBench.

Method	Trans. IoU	Sudden Jump Hit Rate	Range P	Range R	Range F1
Qwen3-VL-8B	0.2396	0.0238	0.1379	0.1283	0.1329
LLaVA-OneVision-2-8B	0.1995	0.0119	0.1533	0.1145	0.1311
Ours	**0.6438**	**0.7589**	**0.9045**	**0.8586**	**0.8809**

E Benchmark Annotation Details

As shown in Fig. 8, we develop an annotation tool for our OmniShotCut bench. This tool helps us swiftly locate the boundaries on the per-frame level and label dense transition cases for long video instances. To facilitate efficient annotation, we implement several useful features. A floating window dynamically displays the current segment's labels (type and confidence) or the relation label, allowing annotators to quickly verify existing annotations. In addition, the tool supports multi-selection, enabling annotators to label multiple segments or relations simultaneously with a single action. An auto-save mechanism is also integrated to automatically store labeling progress and prevent data loss.

Certain transitions of interest in our benchmark, such as sudden jumps, dissolves, and fades, often involve subtle frame-level changes that require careful inspection. To address this, we introduce an

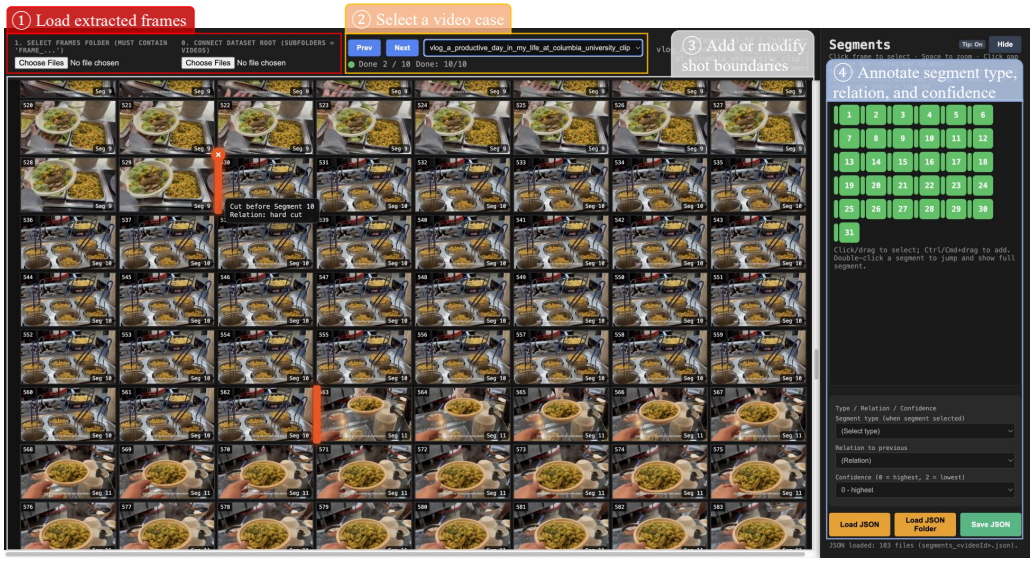


Figure 8: **Benchmark Annotation Tool**. Annotators first load extracted frames and select a video case. Shot boundaries are created by clicking between frames along the timeline. The right panel provides an overview of segments and enables labeling of type, relation, and confidence. Additional features, including multi-selection, auto-save, and frame-level inspection, facilitate efficient dataset construction.

open-image inspection mode (see Fig. 9). Annotators can open a frame by double-clicking or pressing the space bar, which displays the frame along with its associated type, confidence, and relation annotations. Using the left and right arrow keys, annotators can navigate through frames sequentially, effectively previewing the sequence as a short video clip. These features together provide an efficient and user-friendly platform for constructing and verifying our benchmark dataset. A preview of the videos in the benchmark is shown in Fig. 10.

F Visual Comparisons

The visual comparison result is shown in Fig. 11. As we can see, our model succeeded in the fading and dissolve transition as well as sudden jump detection. Baseline models like TransNet V2 [30] and AutoShot [44] failed, where they usually choose the start frame in the middle of the dissolve or fading effects. This confusing first frame is not friendly for the downstream applications like video generation, where they demand a clear first frame source for the image-to-video generation. Our result aligns with the ground-truth labels. Further, these baselines cannot realize the sudden jump, which lacks sensitivity to the subtle changes.

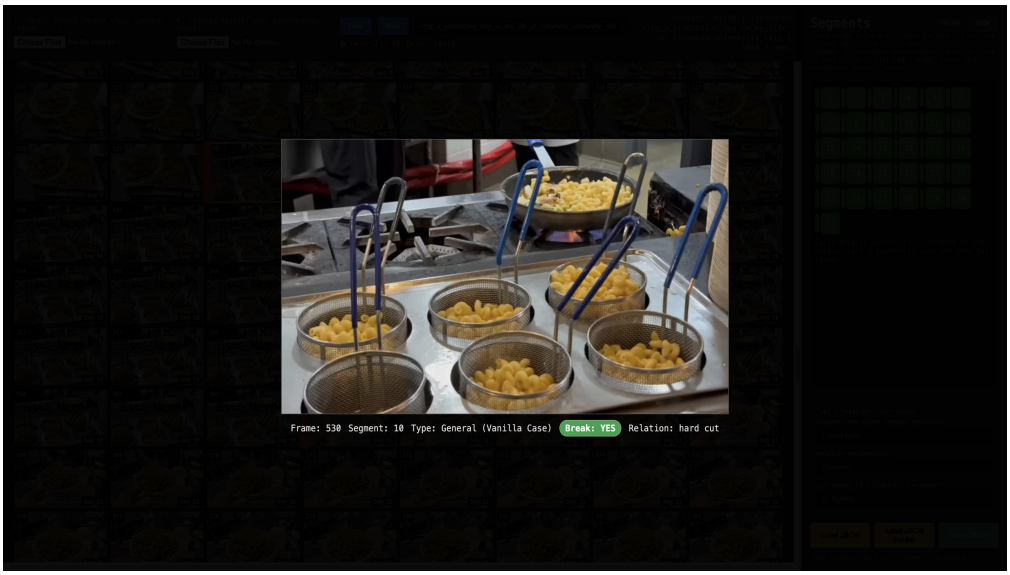


Figure 9: **Annotation Tool Open-Image Inspection Mode**. The Inspection mode shows the high resolution and labeling details for the annotators to define subtle transition changes accurately. It can be played as videos fluently by pressing the buttons to check gradual transitions frame-by-frame.

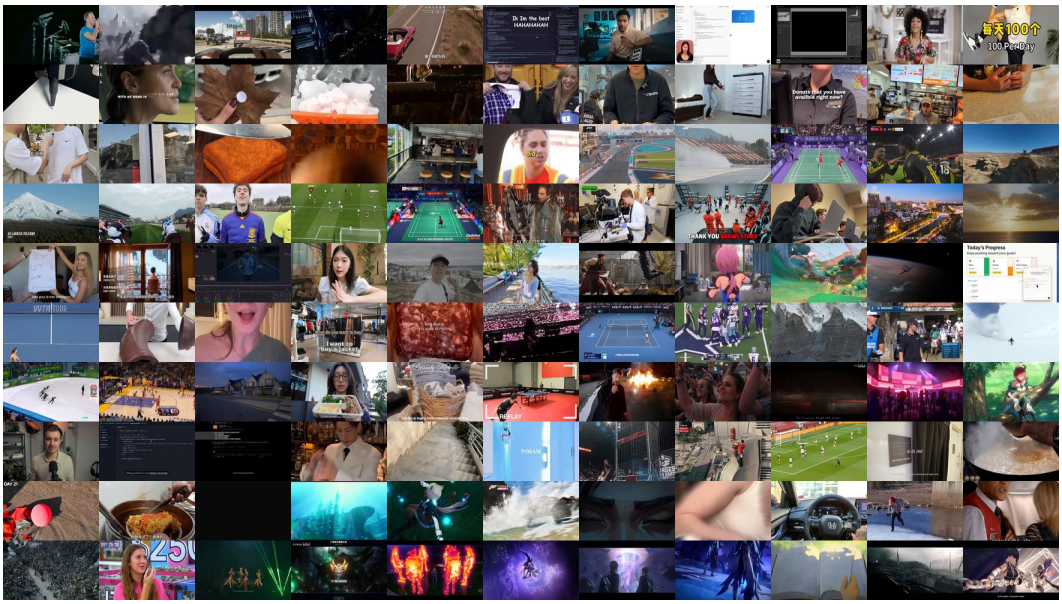


Figure 10: **OmniShotCut Bench Sample Images**. Our benchmark covers diverse topics spanning lifestyle, sports, entertainment, anime, game, unboxing, vlog, shorts, tutorials, urban scenes, screen-based media, and so on.

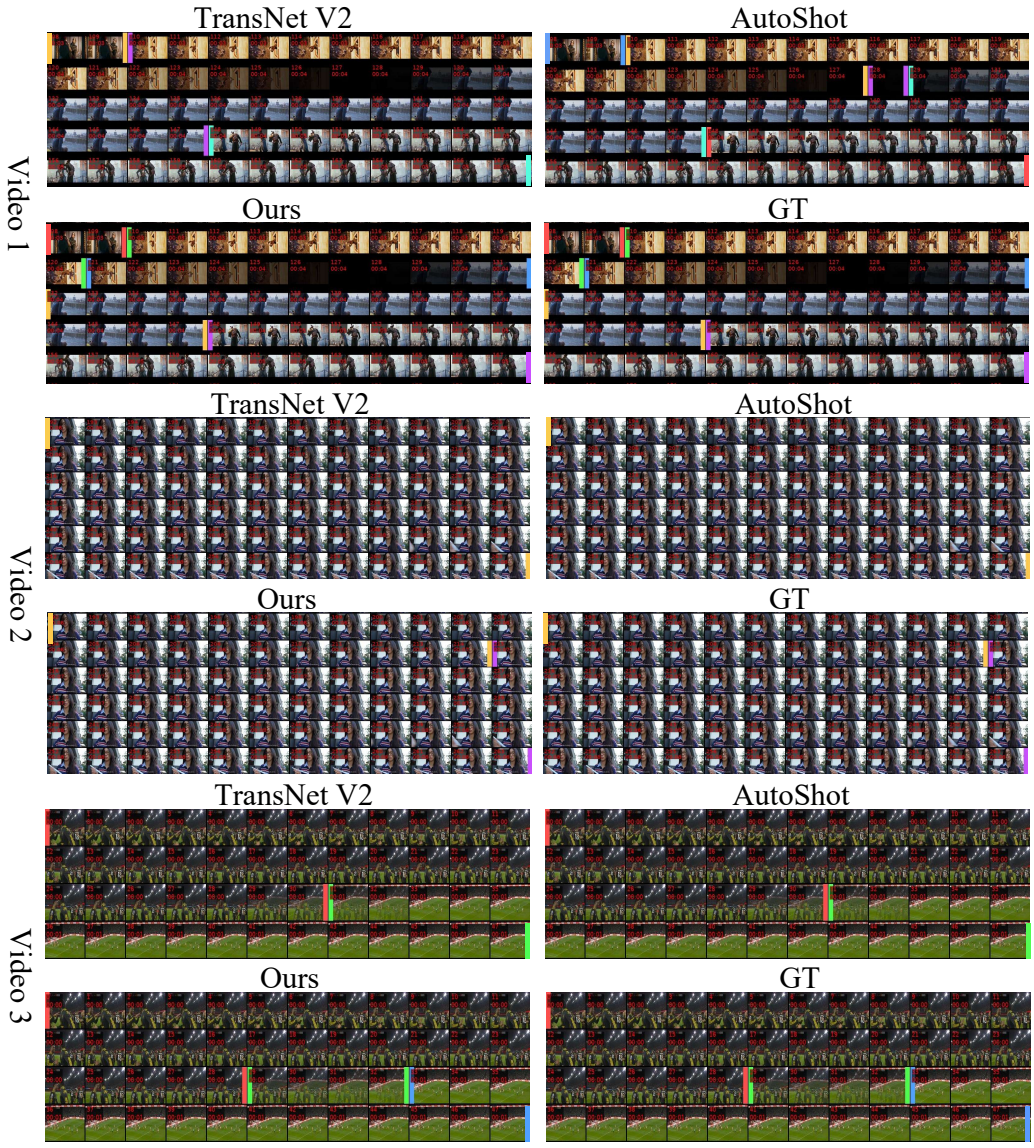


Figure 11: **Shot Boundary Detection Qualitative Comparisons**. We compare TransNet V2 [30], AutoShot [44], and ours on Fading (video 1), Sudden Jump (video 2), and Dissolve (video 3). Each vertical bar with the same color denotes the start and end of a clip cut by the model. **Zoom in for the best view.**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes. The main claims in the abstract and introduction accurately reflect the paper's contributions: a shot-query based formulation for holistic shot boundary and transition relation detection, supported by the proposed model, label design, synthetic data construction, and experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mentioned our limitation in the synthetic data curation in the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: We are not theory paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide all the hyperparameters that are needed for the training and the resources we use.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The inference code and model weights are publicly available at `https://github.com/UVA-Computer-Vision-Lab/OmniShotCut`; the benchmark and the training code will be released. In addition, the paper provides the dataset construction procedure, evaluation protocol, public data sources, and implementation details needed to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: This is provided in the experiment section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars because the evaluations are performed on fixed test sets with deterministic metrics, and the reported numbers are aggregate results over all videos.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We say in the experiment section that we use 8GPU Nvidia A100 for 2 days for all the experiments on 50 epochs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

Justification: The paper discusses positive impacts such as improving video understanding, editing, indexing, and dataset curation, as well as potential risks from large-scale video analysis or misuse in surveillance-style applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk dual-use models such as large generative models or scraped sensitive datasets; therefore, no additional release safeguards are required.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly credit the creators and original owners of the code, datasets, models, and other assets used in the paper, and we explicitly mention and respect their licenses and terms of use where applicable.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We properly credit the creators and original owners of the code, datasets, models, and other assets used in the paper, and we explicitly mention and respect their licenses and terms of use where applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not have a human study

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: We do not have

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs were used only as an auxiliary component in the synthetic data curation process.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.