Temporal Perceiver: A General Architecture for Arbitrary Boundary Detection

Jing Tan ©, Yuhong Wang ©, Gangshan Wu ©, Member, IEEE, and Limin Wang ©, Member, IEEE

Abstract—Generic Boundary Detection (GBD) aims at locating the general boundaries that divide videos into semantically coherent and taxonomy-free units, and could serve as an important preprocessing step for long-form video understanding. Previous works often separately handle these different types of generic boundaries with specific designs of deep networks from simple CNN to LSTM. Instead, in this paper, we present Temporal Perceiver, a general architecture with Transformer, offering a unified solution to the detection of arbitrary generic boundaries, ranging from shot-level, event-level, to scene-level GBDs. Our core design is to introduce a small set of latent feature queries as anchors to compress the redundant video input into a fixed dimension via cross-attention blocks. Thanks to this fixed number of latent units, it reduces the quadratic complexity of attention operation to a linear form of input frames. Specifically, to explicitly leverage the temporal structure of videos, we construct two types of latent feature queries: boundary queries and context queries, which handle the semantic incoherence and coherence accordingly. Moreover, to guide the learning of latent feature queries, we propose an alignment loss on the cross-attention maps to explicitly encourage the boundary queries to attend on the top boundary candidates. Finally, we present a sparse detection head on the compressed representation, and directly output the final boundary detection results without any post-processing module. We test our Temporal Perceiver on a variety of GBD benchmarks. Our method obtains the state-of-the-art results on all benchmarks with RGB single-stream features: SoccerNet-v2 (81.9 percent average-mAP), Kinetics-GEBD (86.0 percent average-f1), TAPOS (73.2 percent average-f1), MovieScenes (51.9 percent AP and 53.1 percent $M_{ext}$) and MovieNet (53.3 percent AP and 53.2 percent $M_{ext}$), demonstrating the generalization ability of our Temporal Perceiver. To further pursue a general GBD model, we combine various tasks to train a class-agnostic Temporal perceiver and evaluate its performance across all benchmarks. Results show that the class-agnostic Perceiver achieves comparable detection accuracy but better generalization ability compared to dataset-specific counterparts.

Index Terms—Feature compression, general perception, generic boundary detection, latent feature query, long-form video understanding, query-based detection, temporal modeling.

I. INTRODUCTION

V

ideo content analysis [1], [2], [3], [4], [5], [6], [7], [8], [9] is a fundamental and important topic in computer vision due to the drastic growth of videos captured and shared online. Mainstream video understanding focuses on shot-form videos for action recognition [10], [11], [12], [13], [14], [15], action detection [16], [17], [18], [19], [20], [21], video retrieval [22], [23], [24], and video grounding [25], [26], [27], [28], [29]. Long-form videos, such as surveillance videos, movies, and recordings of sport events, contain hours of content with rich semantics. These long-form videos still remain under-explored in previous literature due to the deficit in temporal segmentation of long videos to different levels. To facilitate long-form video understanding, we perform research to bridge the gap between long-form and short-form video understanding, by segmenting long videos into a series of shorter, meaningful pieces, as a basic pre-processing technique.

We study the problem of generic boundary detection (GBD), which aims to localize its temporal location in long-form videos. The core concept of this task is generic boundary [30], which is a specific kind of temporal boundary that emerges only from semantic incoherence. Different from the well-studied action instance boundaries of limited target classes [31], [32], [33], [34], [35], generic boundaries are not specific to any pre-defined semantic category and can indicate the temporal structure of videos without any semantic bias. The concept was explored in [30] in regard to event-level generic boundary. We extend this study to arbitrary conditions including shot-level boundary in soccer matches [36] and scene-level boundary in movies [37]. Fig. 1 provides examples of class-agnostic generic boundaries from shot-level, event-level to scene-level instances. To detect such diverse temporal boundaries, different levels of information are required for capturing temporal structure and context at different scales.

Current research on generic boundary detection is separately studied in different tasks based on their boundary definition and granularity. For example, camera shot segmentation [36] targets at shot-level boundaries Fig. 1(a) that are rapid transitions between camera shots. Generic event boundary detection (GBED) [30] aims to locate event-level generic boundaries Fig. 1(b) which are moments where the action/subject/environment changes. Scene segmentation [37] tries to detect scene-level generic boundaries Fig. 1(c), which are transits between movie scenes, indicating the high-level plot twists. Previous methods [30], [38], [39], [40] mainly focused on building informative feature representations carefully designed...
used CNNs in local temporal window and resorted to LSTM to address event boundary. Scene boundary is determined by holistic scene understanding and resorted to LSTM to capture long-range temporal context for scene boundary. Moreover, to produce the final boundary detection, these methods often resorted to sophisticated post-processing techniques to remove duplicate false positives. The complicated design and post-processing is highly correlated with specific boundary type and prevent them from generalizing well to different types of generic boundary detection.

Instead, we argue that in spite of the difference in boundary granularity, these tasks share the similar semantic structure and pose similar requirement. Therefore, a natural question arises: whether we can address these different kinds of generic boundary detection in a general and unified architecture? To this end, in this paper, we unify the detection of different kinds of boundaries within a simple sparse detection framework. We introduce the **Temporal Perceiver**, a general and efficient framework designed to detect arbitrary kinds of boundaries using a single Transformer-based architecture. Attention modules are flexible architectural building blocks, but also require computational cost quadratically with the input dimension. Based on the key observation that there exists high redundancy among frames in each segment, we figure out to compress the long-form video before applying attention operations. Specifically, our core contribution is to introduce a small set of latent units as anchors to adaptively compress the long video sequence into a fixed dimension. These latent units form a cross-attention block to squeeze the original video input into a latent feature space with fixed dimension. This step reduces the computational cost of attention operations to be linear complexity of temporal duration. The compression rate varies according to different levels of redundancy in each benchmark. We show in experiments that the input features can be compressed into \( \sim 50\% \) for most benchmarks and the largest compression is up to 68% for SoccerNet-v2.

To improve the compression effectiveness of latent units, we propose two specific designs on their form and training strategy, respectively. Naturally, videos can be viewed a series of semantically coherent segments divided by generic boundaries. A good compression for GBD is expected to only skip these redundant frames inside the segments yet still retain the informative ones around the boundaries. First, to meet this requirement, our latent units are explicitly composed of boundary queries and context queries. The boundary queries aim to aggregate the important semantic incoherence regions across boundaries in a one-on-one manner, while the context queries are learned to cluster the coherence regions into a few semantic centers to suppress video redundancy. In practice, we compute an array of coherence scores on every temporal location to characterize the coherence structure. Second, to utilize this coherence score for effective compression, we propose a new alignment loss to guide the training on cross-attention maps and make this compression process more informative. This novel loss will endow our Temporal Perceiver with the awareness of video-specific temporal structure during latent unit learning, which turns out to be helpful for speeding up model convergence as well as boosting detection performance. Finally, we place a transformer decoder on top to perform sparse detection of generic boundaries with a set of learnable proposals. Due to the nature of sparse detection and strict matching criteria during training, our method is free of complicated post-processing and can be generalized to different types of GBD.

We perform experiments on five popular benchmarks across the shot-level, event-level and scene-level GBD tasks: SoccerNet-v2 [36], Kinetics-GBD [30], TAPOS [41], MoviScenes [39] and MovieNet [37], to verify the effectiveness of Temporal Perceiver for arbitrary generic boundary detection. In practice, Temporal Perceiver employs image encoder ResNet50 [42] as backbone network by default for fair comparison with previous literature. Our method achieves the state-of-the-art performance on all benchmarks with remarkable improvements with RGB inputs only. Additionally, we test the performance of Temporal Perceiver with the IG-65M [43] pre-trained video encoder CSN [44] as backbone on Kinetics-GBD val set to demonstrate our performance upper-bound. We also visualize qualitative results for better understanding of our models. We hope these quantitative results can provide more insightful analysis on our modeling details of Temporal Perceiver.

In summary, our contributions are summarized as follows: 1) We present **Temporal Perceiver**, a general architecture to tackle generic boundary detection in long-form videos, providing a unified solution to the detection of the arbitrary boundary with Transformer. 2) To tackle the temporal redundancy of long videos and reduce the model complexity, we introduce a small set of latent queries for feature compression via cross-attention mechanism. 3) To improve the compression effectiveness of latent units, we devise customized designs on their form and training strategy, respectively. 4) Extensive experiments demonstrate that our method outperforms the existing state-of-the-art methods with RGB features only on shot-level, event-level, and scene-level generic boundary benchmarks, demonstrating its generalization ability on GBD.

**II. RELATED WORK**

In this section, we first discuss previous works in generic boundary detection, with a particular focus on shot-level, event-level and scene-level boundary detection. After that, we mention...
several related Visual Transformer literature that tackle the quadratic encoder complexity.

A. Generic Boundary Detection

Generic boundary detection is termed for unified detection of arbitrary temporal generic boundaries. In this research, we particularly study GBD at shot-level (also termed as camera shot segmentation), event-level (generic event boundary detection, GEBD), and scene-level (movie scene segmentation).

Previous literature used specific hand-crafted techniques to solve different types of boundaries. Methods from Scikit [38] and PySceneDetect [45] libraries achieved superior performance in shot boundary detection based on frame variations in color or intensity. Previous GEBD methods [30], [31], [41], [46], [47], [48], [49] focused on building representations specifically designed for event-level boundaries and exploited dense prediction paradigm with post-processing. Temporal Convolution Network (TCN) [46], [47] used a two-layer temporal convolution network to densely predict the confidence score for each frame. PC [30] explored the local temporal dependency by modeling past and feature features at each temporal location for final dense confidence estimation. These methods used CNNs in local temporal windows to address event boundary and depended heavily on post-processing to remove duplicates. Scene segmentation attracts academic attention from both supervised [39], [50], [51], [52] and unsupervised [40], [53], [54], [55], [56], [57] domain. The pioneering method LGSS [39] resorted to LSTM to capture long-range temporal context for scene boundary detection and manually designed a dynamic programming algorithm for post-processing. No previous work has addressed these boundaries in a single and unified framework. In contrast, our temporal perceives presents a general solution with global view to handle these different types of boundaries without any specific design.

B. Visual Transformers

Inspired by the great advances of Transformer architecture [58] in NLP field [59], researchers have explored the application of such architecture in visual tasks [18], [60], [61], [62], [63], [64], [65]. DETR [60] was the first to successfully utilize a Transformer encoder-decoder structure in object detection task and achieves on par performance with its highly optimized CNN-based counterparts. Despite its success, DETR suffered from the quadratic memory complexity in encoder self-attention layers and therefore had limitations with large input sequences. Deformable DETR [66] replaced the global and dense self-attention in encoder with a set of sparsely sampled reference points and deformable attention mechanism. PnF-DETR [67] performed differentiable sparse sampling on input sequences and reduced the input length to Transformer encoder. RTD [18] extended the idea of DETR to the video domain for temporal action localization with several important improvements. In our Temporal Perceiver, we present a new architecture composed of two transformer decoders. Our first decoder is to re-purpose the Transformer decoder as an encoder for temporal feature compression in video domain. This decoder-as-encoder structure squeezes temporal redundant input into a small set of feature queries for subsequent processing. Our second decoder is similar to the work of DETR and RTD, but designed for a different task (i.e. generic boundary detection).

Recently, Jaegle et al. proposed the Perceiver [68] and Perceiver IO [69] architecture, which formulates prediction tasks across modalities with a general “read-process-write” perception paradigm. They read large inputs into small latent dimension through cross attention blocks, process the latent queries with stacked self-attention blocks, and outputs prediction results with cross attention blocks. In fact, we developed our method independently in June 2021, but lately found our idea similar to the Perceiver architecture in compressing large input to small latent space. However, our Temporal Perceiver still has several important differences with Perceiver or Perceiver IO. First, our Temporal Perceiver deals with a temporal detection problem in videos using a sparse detection head, while Perceiver and Perceiver IO handles with classification and dense prediction problems, respectively. More importantly, we explicitly leverage the temporal structure for the design of latent units and their training strategy. Our explicit alignment loss will help to speed up the convergence of latent units and contribute to a better detection performance. Finally, in our Temporal Perceiver, the specific design of network architecture is different. The same number of layers of self-attention and cross-attention blocks are coupled and stacked alternatively to compress video features, while Perceiver and Perceiver IO use less cross-attention blocks and more self-attention blocks in separate reading and processing steps, respectively. We discuss the difference in details between our method and the Perceivers in Section III-C3.

III. Method

A. Overview

We propose the Temporal Perceiver based on Transformers, to efficiently locate arbitrary generic boundaries in long-form videos. Given an untrimmed video X, Temporal Perceiver generates a set of generic boundary proposals \( \Psi = \{ t_n \}_{n=1}^{N_d} \) to locate the generic boundaries \( \hat{\Psi} = \{ \hat{t}_n \}_{n=1}^{N_d} \) in video X, with \( N_d \) and \( N_d \) denote the number of detected boundary and boundary groundtruths, respectively.

The main structure of Temporal Perceiver is depicted in Fig. 2. We take the frame-level \( \text{RGB features} F \in \mathbb{R}^{N \times C} \) and \( \text{coherence scores} S \in \mathbb{R}^{N \times 1} \) from backbone network as input, where \( N \) is the number of frames and \( C \) is the number of feature channels. A cross-attention based Transformer \( \text{encoder} E \) and a set of learnable \( \text{feature queries} (\text{latent units}) Q_e \in \mathbb{R}^{M \times C} \), distill and transform the input sequence into a latent feature space of reduced dimension \( F \in \mathbb{R}^{M \times C} \), where \( M \) is the number of latent units. The latent units are less than the input frames \( (M < N) \) and each unit acts as an anchor to cluster the temporal information in a global view. These latent units could be supervised by an explicit alignment loss. Based on this compressed representation, we use another Transformer \( \text{decoder} D \) and a set of learnable \( \text{propal queries} (\text{latent units}) Q_d \in \mathbb{R}^{N_d \times C} \) to directly produce generic boundary representations. Finally, feed forward networks (FFNs) are used as localization head
Fig. 2. The overall pipeline of Temporal Perceiver. The input to our model is video features and an array of coherence scores, both of length $N$. Added with sine positional encoding, the input video features are shuffled and ranked in the descending order of coherence score. Feature queries compress the input feature into latent dimension $M$ in the encoder cross-attention. An alignment loss is enforced on the cross-attention map to make sure that $K$ boundary queries preserve top-$K$ boundary features and $M - K$ context queries cluster context features into $M - K$ semantic centers. After the feature compression, the decoder with FFNs (Feed-forward Networks) employs $N_p$ proposal queries to decode boundary location and confidence from the compressed latent sequence.

Fig. 3. An illustration of the backbone module. The backbone are trained on benchmarks with dense boundary supervision. Video features are extracted after the max pooling on each snippet. The coherence scores are predicted from the final linear layer and scoring head to give the final predictions. Our Temporal Perceiver yields a simple and end-to-end temporal detection framework without any post-processing technique, which makes no specific assumption about the boundary type and can be generalized to arbitrary generic boundary detection (GBD).

B. Feature Encoding and Coherence Score

As illustrated in Fig. 3, we adopt ResNet50 [42] pre-trained on ImageNet [70] as backbone to extract appearance features $F = \{f_i\}_{i=1}^N$ and coherence scores $S = \{s_i\}_{i=1}^N$ from RGB frames. Specifically, we first densely sample snippets from videos. Each snippet contains $k$ frames and is sampled at stride $\tau$. Given a snippet $i$, we concatenate its $k$ feature vectors encoded by backbone, and then perform temporal convolution, max pooling and linear layer upon the feature vectors. The output feature $f_i$ is extracted after max-pooling layer. The final linear layer outputs a classification score $s_i$ to represent the probability that its center frame $i$ is boundary. This classification score is the coherence score used in Temporal Perceiver. The backbone network is trained with the dense boundary groundtruth on the benchmarks in advance.

C. Temporal Perceiver

The Temporal Perceiver architecture is formulated as an encoder-decoder framework and built upon the Transformer. The encoder, which is a re-purposed Transformer decoder of $L_{enc}$ layers, takes in a set of feature queries and compresses the video feature into a reduced sequence of latent representations. The decoder, which is a vanilla Transformer decoder of $L_{dec}$ layers, directly output final predictions with a set of proposal queries from the compressed feature. The input features of length $N$ are down-scaled into fixed length $M$ via the cross-attention operations in encoder ($M < N$). Compared with the original Transformer encoder of fully self-attention, our Temporal Perceiver complexity is reduced from $O(N^2)$ to $O(NM + M^2)$, resulting in linear complexity of $N$ with a relatively small $M$.

1) Encoder: Temporal Compression Via Latent Units: Videos can be divided into two kinds of regions in time, namely boundary regions and context regions (i.e., within each segment). Boundary region refers to the temporal interval around the generic boundary with gradual temporal transition, while context region refers to more central regions within coherent segments. For better localization performance, our main focus is on boundary regions. Context regions are highly redundant and could be compressed into shorter sequences for efficiency. To this end, we build two types of feature queries (boundary
queries and context queries) to attend on semantic incoherent (boundary) and coherent (context) regions accordingly.

Specifically, we distinguish the boundary region and the context region based on dense coherence scores $S$ obtained from backbone. These coherence scores are estimates of the likelihood that the corresponding location is a generic boundary. We sort the input features according to the descending order of coherence scores and obtain the sorted permutation $\pi$. The top $K$ locations with highest coherence scores are taken as boundary regions

$$F_b = \{f_{\pi_n} | n = 1, \ldots, K\}. \quad (1)$$

The rest as context regions $F_c = F \sim F_b$. We allocate $M$ latent feature queries to $K$ boundary queries $Q_b$ and $M - K$ context queries $Q_c$. These latent queries $Q_c = [Q_b, Q_c]$ are learnable embeddings shared among all videos in the dataset. They are randomly initialized and trained jointly with network weights. Self-attention in encoder models the global dependency for feature queries, adding pair-wise feature interactions throughout the compression. Through cross-attention layers, $K$ boundary queries $Q_b$ handle the features of boundary regions in a one-on-one manner, and $M - K$ context queries $Q_c$ flexibly cluster the vast context regions into a few contextual centers. The overall encoder of Temporal Perceiver is as follows:

Encoder $\mathcal{E}: \hat{F} =$

Cross_attention([F_b, F_c], Self_attention(Q_c)), \quad (2)

where $\hat{F} \in \mathbb{R}^{M \times C}$ is the compressed video representation for subsequent sparse detection in decoder.

Alignment Constraint on Cross-attention: To guide the boundary feature queries to attend the actual video boundaries, we incorporate alignment constraint on its cross-attention maps. As we have re-organized the feature sequence according to its coherence score in the ranking step (in (1)), we could simply introduce an identity matrix based alignment loss on the last cross-attention map of boundary queries. In this sense, each boundary query is enforced to only attend one of top $K$ locations with highest coherence score. For context queries, we allow them to flexibly attend context information without any constraints. With this new training strategy, we find our feature queries can speed up the model convergence and generate more stable compressed feature representations corresponding to both boundaries and context.

2) Decoder: Sparse Detection Via Proposal Queries: We follow the direct proposal generation pipeline in [18], [60] to generate boundary prediction with Transformer decoders and a set of learnable proposal queries. Throughout the stacked self-attention layers and cross-attention layers, the pair-wise dependencies are modeled for proposals via self-attention to avoid duplicates, and the full sequence of reduced features are attended to via cross-attention to generate proposal embedding. Specifically, the sparse detection is defined as follows:

Decoder $\mathcal{D}: \Phi =$

Cross_attention($\hat{F}$, Self_attention($Q_d$)), \quad (3)

where $\Phi \in \mathbb{R}^{N_p \times C}$ is the decoded representations for all proposal queries.

The final predictions are decoded from proposal query embedding via a two-branch head architecture. The localization head uses a three-layer MLP to predict the boundary location, and the classification head uses a fully-connected layer followed by sigmoid function to get confidence scores that effectively distinguish boundaries.

3) Discussion With Perceiver and Perceiver IO: In spirit, our work is similar to the recent Perceiver [68] and Perceiver IO [69] architectures. Although we both utilize latent queries for input compression, our method is different from the Perceivers in several important aspects.

First, the basic processing scheme and pipeline is different. Both Perceiver and Perceiver IO follow the general “read-process-write” scheme by designing separate compression and processing modules. In this decoupled architecture, it uses a symmetric architecture where a cross-attention block is equipped with multiple self-attention block. Instead, our Temporal Perceiver employs a coupled and progressive “compression-process” scheme by stacking multiple layers of encoders, each of which is compose of a cross-attention and self-attention block. Our Temporal Perceiver aims to gradually compress and transform the original video feature sequence in a joint manner. Therefore, the number of cross-attention and self-attention blocks in our Temporal Perceiver is the same. We analyze that this coupled and progressive compression and transformation process can enhance its effectiveness and also allows for deep supervision to guide the this process as discussed next.

Second, the training paradigm and loss is different. Both Perceiver and Perceiver IO fail to utilize the explicit loss to guide the training of latent units. They solely rely on the final target supervision such as classification and prediction loss. Yet, this blind learning process might lead to insufficient latent unit usage and inferior performance. Instead, we carefully consider the specific property of video data and the GBD task, and propose customized latent unit form and training strategy. In particular, we explicitly divide the latent queries into two kinds (boundary and context) to handle the temporal redundancy of input video data. Meanwhile, we introduce a novel alignment loss on cross attention maps to guide the training of boundary queries, while Perceiver and Perceiver IO do not employ any auxiliary supervision. Ablative experiments show that the alignment loss helps to speed up the model convergence as well as improve model performance.

Finally, the target problem and domain is different. Both Perceiver and Perceiver IO deal with the general classification and dense prediction in space. These tasks often require a classification head to generate a label or a dense prediction head to generate pixel-wise labels. Instead, our Temporal Perceiver handle the temporal detection problem in videos. We need to integrate our Temporal Perceiver with a sparse detection head to directly regress the location of generic boundary. In general, the detection task is more challenging than the classification and dense prediction task due to the large variance in its detected targets. Overall, direct application of Perceiver architecture is not sufficient for good boundary detection and our customized...
designs are effective with non-trivial performance gains in general generic boundary detection.

D. Training

First, we divide videos features into fixed-length sequences via a sliding-window manner: short videos are zero-padded and long videos are divided into non-overlapping segments. Then, we filter out those windows with no groundtruth for training.

1) Label Assignment: Following the practice of [60], we assign positive labels to predictions with groundtruths in a strict matching scheme. For video input \( X \), the groundtruth set is denoted \( \Psi = \{ t_n \}_{n=1}^{N_{\Psi}} \) and the prediction set is denoted \( \Psi = \{ \hat{t}_n \}_{n=1}^{N_{\Psi}} \). The bipartite matcher searches for an optimal one-on-one matching between the two sets. The matcher minimizes the cost function via hungarian algorithm to seek the optimal matching \( \sigma(\cdot) \), then assign positive labels to those predictions that are matched with groundtruths. The cost function is defined as:

\[
C = \sum_{n: \sigma(n) \neq \emptyset} \alpha_{loc} \cdot |t_n - \hat{t}_{\sigma(n)}| - \alpha_{cls} \cdot p_n, \tag{4}
\]

where \( p_n \) denotes the boundary confidence score of proposal query \( n \), \( \sigma(\cdot) \) denotes a permutation for predictions to match with the groundtruths. \( \alpha_{cls} \) and \( \alpha_{loc} \) denote coefficients for classification error and localization error.

2) Loss Functions: Conventionally, we define a localization loss and a classification loss to supervise the final predictions. The localization loss is formulated with \( L_1 \) loss on predicted boundary point \( t_n \) and its matched groundtruth \( t_{\sigma(n)} \):

\[
L_{loc} = \frac{1}{N_{p, pos} \sum_{n: \sigma(n) \neq \emptyset}} L_1(t_n, \hat{t}_{\sigma(n)}). \tag{5}
\]

The classification loss is defined with cross-entropy loss:

\[
L_{cls} = -\frac{1}{N_p} \sum_{n: \sigma(n)} (\hat{t}_{\sigma(n)} \log(p_n) + (1 - \hat{t}_{\sigma(n)}) \log(1 - p_n)), \tag{6}
\]

\( \hat{t}_{\sigma(n)} \) denotes the matched binary groundtruth label for proposal query \( n \).

To enforce additional constraint on feature compression in order to preserve the important boundary information, we add a diagonal alignment loss \( L_{align} \) on cross-attention maps:

\[
L_{align} = -\log \frac{1}{K} \sum_{m=1}^{K} A_{m,m}, \tag{7}
\]

where \( A \) denotes the last-layer cross-attention map. It is noted that we use auxiliary loss with classification loss and localization loss, but not with the alignment loss.

The overall loss function \( \mathcal{L} \) writes as follows:

\[
\mathcal{L} = \alpha_{loc} \cdot L_{loc} + \alpha_{cls} \cdot L_{cls} + \alpha_{align} \cdot L_{align}, \tag{8}
\]

where \( \alpha_{align} \) is the alignment loss coefficient. We use the same set of classification and localization coefficients for the matching cost function and the loss function.
TAPOS [41] contains 16,294 valid instances from Olympics sport videos. All instances are split into train, validation and test sets, of sizes 13094, 1790, 1763, respectively. We trim the action instances as input video and detect the sub-action boundaries within the action. The benchmark is re-furnished for generic event boundary detection by Shou et al. [30]. Our model is evaluated on the validation set.

Scene-Level Boundary Dataset: MovieScenes [39] is scene segmentation dataset containing 297 hours of 150 movies. Previous works divide the dataset into a 100-video training set, a 20-video validation set and a 30-video test set. MovieNet [37] dataset expands upon MovieScenes and becomes another popular benchmark for scene segmentation. It contains 42 K annotated scene segments from 318 movies, and it is divided into training, validation and testing sets with 190, 64 and 64 movies respectively. We compare with previous works on MovieScenes and report ablation results on MovieNet validation set.

B. Implementation Details.

We adopt ResNet50 [42] for feature extraction, the sampling stride \( \tau = 1 \) for SoccerNet-v2, \( \tau = 3 \) for Kinetics-GEBD and TAPOS. Due to copyright, MovieNet only provides three frames from each shot, so the actual temporal strides between frames can be huge. We strategically take all frames into account and set the sampling stride \( \tau = 1 \), to mitigate large regression errors caused by large input stride. The local window size \( k \) for frame-level feature extraction is 9 for SoccerNet-v2, 10 for Kinetics-GEBD and TAPOS, 12 for MovieNet.

We process input videos via a sliding-window mechanism for all datasets, the window size \( N \) for SoccerNet-v2, Kinetics-GEBD and TAPOS is set to 100 and the window size for MovieNet is set to 30. Short videos are zero-padded to window size. We don’t overlap windows, so the overlap ratio is 1 for both training and inference. The refined feature length \( M \) is set to 32, 60, 60, 15 for SoccerNet-v2, Kinetics-GEBD, TAPOS and MovieNet; the number of boundary queries and context queries is 20 and 12 for SoccerNet-v2, 48 and 12 for Kinetics-GEBD and TAPOS, 10 and 5 for MovieNet. \( L_{\text{enc}} = 6 \) for Kinetics-GEBD and TAPOS, 3 for MovieNet and SoccerNet. \( L_{\text{dec}} = 6 \) for all benchmarks.

The loss parameters \( \alpha_{\text{cls}}, \alpha_{\text{loc}} \) and \( \alpha_{\text{align}} \) are set to 2, 1, 1 respectively. We use AdamW as optimizer. The learning rate is set to 2e-4 and the batch size is 64. The confidence threshold \( \gamma \) for final submissions is set to 0.9 for SoccerNet-v2, Kinetics-GEBD and TAPOS, 0.7 for MovieNet.

C. Comparison With the State of the Art

1) Shot Boundary Detection. Evaluation Metrics: We follow [36] to use mAP metric with tolerance \( \delta \) of 1 s for evaluation. A predicted boundary is positive if it falls within the given tolerance \( \delta \) of a ground-truth timestamp.

Comparisons with the state of the art: We train Temporal Perceiver on the training set, validate on the validation set to select the best model, then report the results on the testing set. As Table I illustrates, Temporal Perceiver outperforms the previous state-of-the-art method by 3.4%, generalizing well to shot-level generic boundary detection. Our model outperforms all previous methods on fading transitions with a large margin of 9.4% and achieves competitive results on abrupt and logo transitions, thanks to the adaptive receptive field of attention mechanism.

2) Event Boundary Detection. Evaluation metrics: We use f1 score under different Relative Distance thresholds for quality measurement. Relative Distance is the relative distance between the predicted and groundtruth boundary timestamps, divided by the duration of the corresponding video. Given a fixed Relative Distance, we use it as threshold to determine whether a boundary prediction is correct, then the f1 metrics can be computed. We mainly compare f1 score with 0.05 relative threshold and average f1 score.

Comparisons with the state of the art: In Table II, we compare the results of ResNet50-based Temporal Perceiver with the state-of-the-art methods on Kinetics-GEBD and TAPOS for event-level boundary detection. The results show that our method comfortably outperforms the state-of-the-art method especially with smaller Rel.Dis. threshold (less error tolerance). We also provide results of Temporal Perceiver with CSN backbone on Kinetics-GEBD to show the performance upper-bound of our model. This also demonstrates that our model produces more precise and accurate boundary predictions than previous methods both with image encoder and video encoder backbone.

3) Scene Boundary Detection. Evaluation metrics: Following [39], we use Average Precision (AP) and \( M_{\text{iou}} \) to evaluate the quality of detected scene boundaries, where \( M_{\text{iou}} \) calculates the weighted sum of intersection-over-union of a detected scene with respect to its distance to the closest ground-truth scene.

Comparisons with the state of the art: In Table III, we compare the results of Temporal Perceiver with the state-of-the-art methods on MovieScenes. The results of previous literature is quoted from [39]. However, the exact data-split scheme in [39] is not provided. Thus, we apply a 10-fold cross-validation on the dataset and report the mean and standard deviation of AP and \( M_{\text{iou}} \).

To compute Average Precision, we construct Gaussian distributions centered at our sparsely-predicted boundary locations to approximate dense score sequences for each video. AP is
reported based on the approximated dense scores. Temporal Perceiver improves the AP by 4.8% and $M_{iso}$ by 4.3% over previous SOTA, using visual modality only. The remarkable improvement demonstrates the generalization ability and effectiveness of our model on generic boundary detection under various granularity.

4) Coherence Scores and Optical Flow Input: In Tables I, II, and III, we have included the evaluation results of the coherence module for each benchmark to show the quality of coherence scores. Results show that Temporal Perceiver significantly improves performance over the coherence scores across all benchmarks.

As Optical Flow is shown to improve temporal boundary detection systems in the past literature, we also incorporated this modality along with RGB input for shot-level and event-level boundary detection in Table I and Table II. Unfortunately, we were unable to provide MovieNet results with optical flow input due to the benchmark’s lack of temporal consistent frames, which prevents the extraction of optical flow frames. Our findings in Tables I and II suggest that adding optical flow is generally beneficial for GBD. However, we also observed that the gains from optical flow become less obvious when RGB frames alone are sufficient to achieve high accuracy in boundary detection, as is the case for shot boundary detection on SoccerNet-v2 and for event boundary detection with CSN backbone on Kinetics-GEBD.

D. Efficiency Analysis

We analyze Temporal Perceiver’s advantage in efficiency in terms of Shot Per Second (SPS) in inference and FLOPs of feature encoders on MovieNet. We compare our model with the previous state-of-the-art method LGSS and a Transformer baseline method. The results are reported on one RTX 2080-Ti GPU. We take three random movies selected from MovieNet validation set (2275 shots in total) as input. Table IV shows this.

### Table II: Comparison With the Event-Level State-of-The-Art Methods in Terms of F1@Rel.Dis on Kinetics-GEBD Validation Set and TAPOS Validation Set

| Benchmark | Method | Backbone | f1 @ Rel.Dis. |
|-----------|--------|----------|---------------|
| Kinetics-GEBD | BMN [31] | ResNet50 | 18.6 |
| | BMN-StartEnd [31] | ResNet50 | 49.1 |
| | TCN-ATAPO [46], [47] | ResNet50 | 46.4 |
| | TCN [46], [47] | ResNet50 | 58.5 |
| | PC [30] | ResNet50 | 62.5 |
| | CVPR'22 LOVEU winner [74]| CNS | 85.4 |
| TAPOS | Coherence Scores | ResNet50 | 66.8 |
| | Temporal Perceiver | ResNet50 | 74.8 |
| | Temporal Perceiver* | ResNet50 | 77.4 |
| | Coherence Scores | CNS | 76.2 |
| | Temporal Perceiver | CNS | 82.2 |
| | Temporal Perceiver* | CNS | 82.7 |
| | TCN [46], [47] | ResNet50 | 23.7 |
| | TransParser [41] | BNNInception | 28.9 |
| | PC [30] | ResNet50 | 52.2 |
| | Coherence Scores | ResNet50 | 51.6 |
| | Temporal Perceiver | ResNet50 | 55.2 |
| | Temporal Perceiver* | ResNet50 | 60.1 |

### Table III: Comparison With the Scene-Level State-of-The-Art Methods in Terms of AP and $M_{iso}$ on MovieScenes and MovieNet

| Method | Modality | AP (†) | $M_{iso}$ (†) |
|--------|----------|--------|---------------|
| GraphCut [54] | Visual | 14.1 | 29.7 |
| SCSSA [55] | Visual | 14.7 | 30.5 |
| DP [56] | Visual | 15.5 | 32.0 |
| Grouping [50] | Visual | 17.6 | 33.1 |
| StoryGraph [57] | Visual | 25.1 | 35.7 |
| SiamSec [51] | Visual | 28.1 | 36.0 |
| LGSS [39] | Visual Audio Actor Action | 47.1 | 48.8 |
| Temporal Perceiver | Visual | 51.9±2.4 | 53.1±1.5 |
| Coherence Scores* | Visual | 51.9 | 51.6 |
| Temporal Perceiver* | Visual | 53.3 | 53.2 |

### Table IV: Efficiency Comparison With Previous Scene-Level GBD Method

| Method | ShotPS(†) | GFLOPs (‡) |
|--------|-----------|------------|
| LGSS [39] | 60.7 | 8141 |
| Vanilla Transformer | 371.1 | 66 |
| Temporal Perceiver | 377.9 | 49 |

The backbone network is excluded for all models in run-time.
that our model is more efficient than LGSS and the vanilla Transformer baseline due to effective feature compression. Free of post-processing modules, Temporal Perceiver infers 6 times faster with much less FLOPs than LGSS.

E. Ablation Study

Comparison with Transformer-based baselines: To achieve fair comparison and demonstrate the effectiveness of our model over Transformer architecture, we implement two baseline models: 1) Sparse Transformer with a 2x downsample to reduce features before encoding and 2) PerceiverIO with cross-attention blocks for read, self-attention blocks for feature compression processing, and cross-attention blocks for decoding to predict dense boundaries. As shown in Table V, feature compression via learned latent is a more effective method than 2x down-sampling, as the Sparse Transformer is outperformed by PerceiverIO and Temporal Perceiver. Both without alignment loss, PerceiverIO proves to be a strong competitor that achieves similar results with Temporal Perceiver on Kinetics-GEBD and outperforms Temporal Perceiver on MovieNet. Added with the alignment loss, which is specifically designed for the boundary-context queries in Temporal Perceiver, our model generally outperforms PerceiverIO on Kinetics-GEBD and MovieNet. Note that PerceiverIO as a dense prediction model inherently performs better under dense metric than sparse model, which explains its 0.5% advantage over Temporal Perceiver. The alignment loss also boosts performance for PerceiverIO in terms of average f1 on Kinetics-GEBD and $M_{iou}$ on MovieNet. These results indicate the effectiveness of the alignment loss and the overall superiority of Temporal Perceiver for temporal boundary detection.

Study on input temporal length: The input temporal feature length $N$ determines the global temporal receptive field in feature compression, therefore we study the impact of increased temporal field to detection performance in Table VI. The experiments are conducted based on the default choices of $N$ for each benchmark, and we learn that increasing $N$ by 2× or 3× results in severe performance decrease. Decreasing the input length is also harmful to the detection performance. We analyze that boundary detection do not need very large temporal receptive field and increased window size could introduce more noise.

E. Ablation Study

Comparison with Transformer-based baselines: To achieve fair comparison and demonstrate the effectiveness of our model over Transformer architecture, we implement two baseline models: 1) Sparse Transformer with a 2x downsample to reduce features before encoding and 2) PerceiverIO with cross-attention blocks for read, self-attention blocks for feature compression processing, and cross-attention blocks for decoding to predict dense boundaries. As shown in Table V, feature compression via learned latent is a more effective method than 2x down-sampling, as the Sparse Transformer is outperformed by PerceiverIO and Temporal Perceiver. Both without alignment loss, PerceiverIO proves to be a strong competitor that achieves similar results with Temporal Perceiver on Kinetics-GEBD and outperforms Temporal Perceiver on MovieNet. Added with the alignment loss, which is specifically designed for the boundary-context queries in Temporal Perceiver, our model generally outperforms PerceiverIO on Kinetics-GEBD and MovieNet. Note that PerceiverIO as a dense prediction model inherently performs better under dense metric than sparse model, which explains its 0.5% advantage over Temporal Perceiver. The alignment loss also boosts performance for PerceiverIO in terms of average f1 on Kinetics-GEBD and $M_{iou}$ on MovieNet. These results indicate the effectiveness of the alignment loss and the overall superiority of Temporal Perceiver for temporal boundary detection.

Study on input temporal length: The input temporal feature length $N$ determines the global temporal receptive field in feature compression, therefore we study the impact of increased temporal field to detection performance in Table VI. The experiments are conducted based on the default choices of $N$ for each benchmark, and we learn that increasing $N$ by 2× or 3× results in severe performance decrease. Decreasing the input length is also harmful to the detection performance. We analyze that boundary detection do not need very large temporal receptive field and increased window size could introduce more noise.

Study on feature compression: With latent queries and cross-attention mechanism, we reduce the temporal redundancy and model complexity via feature compression. Fig. 4(a) shows the AP and encoder GFLOPs on MovieNet under different settings of $M$. The point size and the captions represent encoder GFLOPs. The AP performance reaches the peak at $M = 0.5N$, indicating the sweet spot for compression without severe information loss. It is interesting that lower compression also performs worse. We argue that while high compression could miss important content, at low compression the video redundancy would distort the quality of compressed features, making $M = 0.5N$ a perfect solution for balance and efficiency. Compared with the zero-compression ($M = N$) setting, our design improves the performance by 1.34% in AP and reduces the FLOPs by half.

Study on varying $N$ and $M$. Apart from studying different $N$ and $M$ separately on MovieNet, we also plot the model’s performance under different $(N, M)$ pairs to see the model’s sensitivity to the two parameters. In Fig. 4(b) is a performance plot of varying $N$, $M$ for Kinetics-GEBD. The results indicate that the model is generally not sensitive to $M$ for Kinetics-GEBD, and we have chosen $M = 0.6N$ at $N = 100$ as a default value based on empirical results. Most Kinetics videos last 10 seconds and are encoded at 24 ∼ 30 fps, which results in video feature sequences of length 100 for most cases. Hence, there is a noticeable performance difference between $N = 50$ and $N = 100$, but the model becomes less sensitive to $N$ as $N \geq 100$. 

| Table V | Quantitative Comparison With Transformer-Based and Perceiver-Based Baselines |
|---|---|---|---|
| Method | Feature Compression | Alignment Loss | MovieNet | Kinetics-GEBD |
| | | | AP | $M_{iou}$ | h@0.05 | avg f1 |
| Sparse Transformer | 2x down-sample | ✓ | 46.2 | 50.4 | 70.4 | 81.7 |
| PerceiverIO | ✓ | 53.8 | 52.2 | 72.0 | 83.6 |
| PerceiverIO | ✓ | 53.6 | 52.6 | 71.8 | 85.8 |
| Temporal Perceiver | ✓ | 50.6 | 51.9 | 73.7 | 83.3 |
| Temporal Perceiver | ✓ | 53.3 | 53.2 | 74.8 | 86.0 |

| Table VI | Ablative Experiments of Temporal Perceiver on Different Choices of Alignment Implementations, Input Temporal Length and Ranking Orders |
|---|---|---|---|---|---|---|
| Benchmarks | Metrics | Alignment | Input Length | Ranking Order | TP |
| | | None | Concat | 0.5N | 2N | 3N | by gt | by temp. order | rand. #1 | rand. #2 | rand. #3 | rand. avg |
| MovieNet | AP | 50.6 | 49.6 | 52.2 | 47.2 | 19.2 | 52.7 | 52.4 | 41.8 | 41.6 | 42.3 | 41.9 |
| | $M_{iou}$ | 51.9 | 52.2 | 52.0 | 51.7 | 38.4 | 52.8 | 52.5 | 47.3 | 47 | 47.3 | 47.2 |
| Kinetics-GEBD | f1@0.05 | 72.6 | 73.5 | 69.4 | 74.5 | 33.8 | 74.6 | 74.4 | 73.2 | 73.9 | 71.7 | 73.0 | 74.8 |
| | avg-f1 | 83.4 | 82.2 | 82.8 | 85.6 | 49.7 | 84.4 | 85.2 | 84.2 | 84 | 83.7 | 84.0 | 86.0

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Study on query construction: With specific latent query construction, we exploit the semantic structure of long-form videos and compress features into stable latent queries. The ratio of boundary and context queries in latent query construction needs consideration. We test different number of boundary queries $K$ with fixed compression rate on Kinetics-GEBD and MovieNet. Table VII shows that the detection performance is sub-optimal when only one type of latent query is used in compression. Compared to using only boundary queries ($K = 0$), Temporal Perceiver improves performance by 2.2% under $f_1@0.05$ and 2.6% under average $f_1$ for Kinetics-GEBD, by 1.1% in $M_{iou}$ and nearly 3% AP for MovieNet. We argue that when there are few boundary queries, the compressed features lack discriminative boundary information, explaining the weak performance. The performance is also weak at $K = M$ without context queries, demonstrating the complementary contributions of context information. Generally, it is observed that boundary information benefits the detection performance more than context information. We report with the setting $K = \frac{2}{3}M$ for the best performance based on empirical results.

Study on alignment loss: The alignment loss is proposed to guarantee the respective compression scheme for boundary and context queries. With the additional alignment supervision, the $f_1@0.05$ and avg-$f_1$ performance increases by 2.2% and 2.6% on Kinetics-GEBD benchmark, and the $M_{iou}$ and AP performance increases by 1.3% and 2.7% on MovieNet. In addition, we plot the convergence curve on Kinetics-GEBD in Fig. 4(c), the results show that the model converges faster with the alignment loss and improves $f_1$ score.

In order to build stable boundary-context representations, the boundary queries attend to boundary features in a one-on-one manner via alignment supervision on the last layer of encoder cross-attention. For the design simplicity, one may ask: why not directly use boundary features to replace learned boundary queries? In Table VI, We compare the results of the alignment loss guided model and another variant model that discards the boundary queries and directly concatenates the boundary features with learned context centers. On both benchmarks, the performance of the direct concatenation variant is weaker than the alignment-guided model. We contend that boundary queries aggregate crucial context into feature embedding via layers of cross-attention (before the last layer) and self-attention.

Study on Ranking Orders: To facilitate alignment loss implementation, we devise a permutation $\pi$ for input frame features to rank them according to the descending order of coherence scores, and use identity matrix on the last cross-attention layer to align boundary frames and queries. We study the importance of ranking via exploring different ranking orders $\pi$, described as follows. (a) ranking by learned coherence scores (default in TP) is to construct a permutation $\pi$ based on coarse predictions of the boundary likeliness at each temporal location. These scores are smooth and consistent boundary estimations at both training and inference. (b) ranking by groundtruth-based scores is to construct a permutation $\pi$ based on groundtruth during training. As the groundtruth encodes hard boundary positions that cannot be used for ranking, we devise a set of gaussian kernels centered on each gt boundary to give smoother scores. The only drawback is that the practice also creates gap for training and inference as inference samples do not have access to the groundtruth-based scores. For fair comparison, we construct $\pi$ based on learned coherence scores (as in (a)) for testing. (c) ranking by temporal order indicates a permutation that computes $\pi(n) = n$ for each input frame. (d) random ranking is to construct a random permutation for each input frame. We train and test the random ranking model for three times, and report the result for each run with the average result.

The results are presented in Table VI. Note that, in this ranking order ablation study, the alignment constraint on cross-attention maps will not be adjusted according to the ranking order.
Therefore, different ranking orders will define different sets of frames to align with boundary queries. Our findings indicate that ranking by original temporal order and random orders will result in performance decrease, especially for random rankings, which validates the effectiveness of boundary frame-query alignment. Meanwhile, we notice that ranking by groundtruth would lead to inconsistency between training and inference and ultimately harm performance.

F. Class-Agnostic Perceiver

To further pursue a general boundary detection model, we have developed a class-agnostic perceiver that joins Kinetics-GBD, TAPOS and SoccerNet for training and testing. In our implementation, we use a shared and single model for different tasks and datasets. Specifically, the encoder and decoder layers are shared among all tasks and datasets. Only data pre-processing, latent queries and proposal queries are tailored to each dataset and task to indicate distinct data distribution, compression level and decoding prior for each dataset. During training, we employ balanced batch sampling to select the same amount of samples from each datasets.

We did not include MovieNet in the training process due to its frame rate inconsistency with other datasets. Instead, we test the model’s generalization ability on MovieNet with both the class-agnostic and class-specific models. To generalize the class-agnostic model to MovieNet, specific latent queries and proposal queries are initiated for the unseen dataset. Then, we follow the locked tuning protocol \[75\] to freeze the encoder-decoder layers and only finetune the latent and proposal queries for MovieNet for generalization results.

The results are presented in Table VIII. Class-agnostic TP detects boundaries with comparable accuracy to the task-specific TP across all three datasets, while also achieving better generalization on MovieNet with a margin of 1.8% \(M_{iou}\) and 1.4% AP compared with the task-specific TP.

V. MODEL VISUALIZATION

In this section, we provide qualitative results for event-level and scene-level boundary detection. In addition, comprehensive visualizations for coherence scores and encoder cross-attention are provided to facilitate the understanding of how Temporal Perceiver works. We specifically illustrate how the coherence scores instructs the compression for scene-level and event-level
Fig. 6. The visualization of coherence scores on MovieNet (scene-level). We show the raw frames from seven scenes in a movie before the compression with the x-axis showing the shot number (row 1), large redundancy are present in especially scene 6. The groundtruth boundaries of this movie (row 2) further demonstrate the large redundancy in movies. Then, we show the predicted coherence scores for the seven scenes achieves high recall for boundary capture in comparison with the groundtruth (row 3), serving as good prior for feature compression.

(a) The visualization of coherence scores on MovieNet (scene-level). We show the raw frames from seven scenes in a movie before the compression with the x-axis showing the shot number (row 1), large redundancy are present in especially scene 6. The groundtruth boundaries of this movie (row 2) further demonstrate the large redundancy in movies. Then, we show the predicted coherence scores for the seven scenes achieves high recall for boundary capture in comparison with the groundtruth (row 3), serving as good prior for feature compression.

(b) The visualization of coherence scores on Kinetics-GEBD (event-level). The video records the event of a woman doing laundry, as shown in raw frames (row 1). The temporal redundancy occurs mainly in clip 3 and 4. Coherence scores (row 3) accurately detects potential boundaries and the redundant clips are clustered into semantics centers. Additionally, boundary regions decided by top-k highest coherence scores are indicated with dark blocks in row 4.

Fig. 6. The visualization of coherence scores on (a) MovieNet for Scene-level GBD; (b) Kinetics-GEBD for Event-level GBD.

Fig. 7. The Visualization of encoder cross-attention from Layer 1 to Layer 6. The x-axis indicates the temporal ranking order in input sequence, and the y-axis indicates the sequence in latent queries. The boundary queries and features are shown with green bars along axes, and the context queries and features are indicated with red bars. The obvious diagonal pattern observed in the last cross-attention shows efficient preservation of boundary features by boundary queries.
boundary detection. Moreover, we look inside the encoder cross-attention maps for feature compression and show how the alignment loss is enforced on cross-attention activation maps.

A. Qualitative Evaluation

In Fig. 5(a), we provide the qualitative results on a randomly selected segment from the validation set of MovieNet. We visualize the groundtruth boundaries and the predictions from our model and previous SOTA method LGSS. The LGSS results are reproduced on MovieNet with multi-modal input, whereas our method takes single-stream RGB features only. Compared with LGSS results, our method does not produce duplicates around groundtruth or other within-scene false positives and is able to detect the generic boundary with accurate localization.

To demonstrate the model generalization ability, in Fig. 5(b) we also provide the qualitative results on a random video sample from Kinetics-GEBD validation set. Compared to the previous state-of-the-art method PC, the predictions of Temporal Perceiver are more accurate, with less relative distance to groundtruth and handles the relatively intense within-event motion in the third event with fewer false positives.

B. Visualization of Coherence Scores

We perform feature compression on boundary and context regions and distinguish these regions based on the coherence estimation scores. Fig. 6 illustrates coherence scores in feature compression process with scene-level and event-level video to demonstrate its effectiveness with different levels of video redundancy. In Fig. 6(a), the coherence scores are visualized in row 3. Darker color represents bigger likeliness of frames to cover incoherence regions. In the visualization of video content and groundtruth, we observe larger temporal redundancy for movie clips, especially in scene 6. The coherence scores predict higher likeliness scores near groundtruth boundaries and remain low in scene 6. As a result, the redundant frames in scene 6 are not marked as boundaries and would be clustered into few contextual centers, achieving redundancy compression.

The coherence scores for event-level testing sample is shown in Fig. 6(b). Compared to scene-level movie clips, although daily event videos are less redundant in temporal aspect, Temporal Perceiver still works for event boundary detection. We observe that similar to feature compression with scene-level sample, the coherence scores (row 3) give coarse yet reliable estimation of boundary regions. We additionally marks the selected boundary regions for event-level sample in row 4. It is observed that all groundtruths are covered in these selected boundary regions. Meanwhile, the temporal redundancy between generic boundaries, such as the repeated back-and-forth movement of pouring detergent in the fourth clip, are excluded and would be clustered into a few centers.

C. Visualization of Encoder Cross-Attention Maps

Fig. 7 is the visualization of cross-attention blocks in all 6 layers of encoder on a randomly selected video of Kinetics-GEBD. In the first 2 layers, the latent queries aggregate information from both the boundary and context regions as an initialization. In the third to fifth layer, latent queries attend to parts of boundary regions and context regions. Beam patterns appear at different parts for each layer without overlap. Our analysis in Fig. 8 reveals that, in the selected sample, the vertical beams are representative frames in both boundary and context regions. With the vertical beams, the model can locate important keyframes from a slow-changing video sequence and effectively filter out redundant frames. In the last layer, guided by the diagonal alignment loss, most of the boundary queries aggregate corresponding boundary features in a diagonal-alike pattern, with a few out-of-line attention on neighboring features. The context queries are free of additional supervision and learn to cluster the contextual centers within the context region.

VI. CONCLUSION

In this paper, we have presented Temporal Perceiver, a general architecture for generic boundary detection. It offers an effective pipeline with Transformer architecture and provides a unified framework for the sparse detection of arbitrary generic boundaries. The core contribution is to utilize cross-attention blocks and a small set of latent queries to squeeze redundant video input into a fixed dimension, reducing encoder complexity to linear-level. Moreover, we construct the latent units with boundary and context queries to pattern semantic incoherent boundary features and coherent context features. Furthermore, to facilitate training, we introduce a new alignment loss on encoder cross-attention maps for feature-query alignment. Finally, the sparse detection paradigm of transformer decoder allows our model to be free of post-processing, resulting in a more efficient framework. Experiments show that Temporal Perceiver achieves state-of-the-art results on benchmarks of shot-level, event-level, and scene-level generic boundaries, demonstrating the generalization ability of our model to handle arbitrary generic boundaries.

REFERENCES

[1] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new model and the kinetics dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 4724–4733.
[2] L. Wang et al., “Temporal segment networks: Towards good practices for deep action recognition,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 20–36.
[3] Z. Liu et al., “Video swin transformer,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 3192–3201.
[4] Z. Tong, Y. Song, J. Wang, and L. Wang, “VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training,” 2022, arXiv:2203.12602.
[5] G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 813–824.

[6] Y. Li, Z. Wang, L. Wang, and G. Wu, “Actions as moving points,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 68–84.

[7] J. J. R. Desai, J. C. Niebles, “Detecting human-object relationships in videos,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 8086–8096.

[8] N. Bai et al., “Home action genome: Cooperative compositional action understanding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 11 184–11 193.

[9] J. J. R. Krishna, L. Fei-Fei, and J. C. Niebles, “Action genome: Actions as compositions of spatio-temporal scene graphs,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10 233–10 244.

[10] G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 6201–6210.

[11] Y. Zhi, Z. Tong, L. Wang, and G. Wu, “MSGSampler: An explainable sampling strategy for video action recognition,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 12 149–12 150.

[12] C. Lin et al., “Learning salient boundary feature for anchor-free temporal action localization,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 3320–3329.

[13] T. Lin, X. Shi, J. Zhang, B. Kang, and L. Wang, “TEA: Temporal excitation and aggregation for action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 906–915.

[14] L. Wang, T. Tong, B. Ji, and G. Wu, “TDN: Temporal difference networks for efficient action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 13 598–13 607.

[15] Y. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, “Rethinking the faster R-CNN architecture for temporal action localization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 1130–1139.

[16] R. Yu et al., “Learning salient boundary feature for anchor-free temporal action localization,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 3320–3329.

[17] J. Tan, J. Tang, L. Wang, and G. Wu, “Relaxed transformer decoders for direct action proposal generation,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 13 526–13 535.

[18] J. Zhao et al., “TubeR: Tubelet transformer for video action detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 237–240.

[19] K. Baraldi, C. Grana, and R. Cucchiara, “A deep siamese network for scene understanding,” in Proc. IEEE Conf. Comput. Vis., 2019, pp. 5551–5560.

[20] B. Castellano, “Pycenedetect: Video scene cut detection and analysis tool,” 2014. [Online]. Available: https://github.com/Breakthrough/PySceneDetect

[21] L. Baraldi, C. Grana, and R. Cucchiara, “A deep siamese network for scene understanding,” in Proc. IEEE Conf. Comput. Vis., 2019, pp. 12 046–12 055.

[22] L. Wang, W. Li, W. Li, and L. V. Gool, “Appearance-and-relation networks for video classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 1430–1439.

[23] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, “TubeDETR: Spatio-temporal video grounding with transformers,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 16 442–16 453.
[55] V. Chasani, A. Likas, and N. P. Galatsanos, “Scene detection in videos using shot clustering and sequence alignment,” *IEEE Trans. Multimed.*, vol. 11, no. 1, pp. 89–100, Jan. 2009.

[56] B. Han and W. Wu, “Video scene segmentation using a novel boundary evaluation criterion and dynamic programming,” in *Proc. IEEE Int. Conf. Multimedia Expo*, 2011, pp. 1–6.

[57] M. Tapaswi, M. Bäuml, and R. Stiefelhagen, “StoryGraphs: Visualizing character interactions as a timeline,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 827–834.

[58] A. Vaswani et al., “Attention is all you need,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[59] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics - Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[60] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[61] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in *Proc. Int. Conf. Learn. Representations*, 2021.

[62] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 10 012–10 022.

[63] Z. Liu et al., “Video swin transformer,” 2021, arXiv:2106.13230.

[64] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10 347–10 357.

[65] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, “VViT: A video vision transformer,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 6836–6846.

[66] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETR: Deformable transformers for end-to-end object detection,” in *Proc. Int. Conf. Learn. Representations*, 2021.

[67] T. Wang, L. Yuan, Y. Chen, J. Feng, and S. Yan, “PnP-DETR: Towards efficient visual analysis with transformers,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 4661–4670.

[68] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, “Perceiver: General perception with iterative attention,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4651–4664.

[69] A. Jaegle et al., “Perceiver IO: A general architecture for structured inputs & outputs,” 2021, arXiv:2107.14795.

[70] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[71] W. Kay et al., “The kinetics human action video dataset,” 2017, arXiv:1705.06950.

[72] A. Cioppa et al., “A context-aware loss function for action spotting in soccer videos,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13 123–13 133.

[73] T. Souček and J. Lokoč, “TransNet V2: An effective deep network architecture for fast shot transition detection,” 2020, arXiv:2008.04838.

[74] D. Hong, X. Ma, X. Wang, C. Li, Y. Wang, and L. Wen, “SC-Transformer++: Structured context transformer for generic event boundary detection,” 2022, arXiv:2206.12634.

[75] X. Zhai et al., “LiT: Zero-shot transfer with locked-image text tuning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18 102–18 112.

Jing Tan received the BSc degree from Nanjing University, Nanjing, China, in 2020. She is currently working toward the MS degree with the Department of Computer Science and Technology, Nanjing University. Her research interests include computer vision and deep learning, with a focus on video understanding and temporal action detection.

Yuhong Wang received the BSc degree from Nanjing University, Nanjing, China, in 2020. She is currently working toward the MS degree with the Department of Computer Science and Technology, Nanjing University. Her research interests include computer vision, deep learning, and video understanding.

Gangshan Wu (Member, IEEE) received the BSc, MS, and PhD degrees from the Department of Computer Science and Technology, Nanjing University, Nanjing, China, in 1988, 1991, and 2000 respectively. He is currently a professor with the Department of Computer Science and Technology, Nanjing University. His current research interests include computer vision, multimedia content analysis, multimedia information retrieval, digital museum, and large-scale volumetric data processing.

Limin Wang (Member, IEEE) received the BSc degree from Nanjing University, Nanjing, China, in 2011, and the PhD degree from the Chinese University of Hong Kong, Hong Kong, in 2015. From 2015 to 2018, he was a postdoctoral researcher with the Computer Vision Laboratory, ETH Zurich. He is currently a professor with the Department of Computer Science and Technology, Nanjing University. His research interests include computer vision and deep learning. He was the first runner-up at the ImageNet Large Scale Visual Recognition Challenge 2015 in scene recognition, and the winner at the ActivityNet Large Scale Activity Recognition Challenge 2016 in video classification. He has served as a senior PC or Area chair for CVPR, ICCV, NeurIPS, AAAI, IJCAI, WACV.