GateHUB: Gated History Unit with Background Suppression for Online Action Detection

Junwen Chen*‡ Gaurav Mittal*† Ye Yu† Yu Kong‡ Mei Chen†
†Microsoft ‡Rochester Institute of Technology
{gaurav.mittal, yu.ye, mei.chen}@microsoft.com {jc1088,yu.kong}@rit.edu

Abstract

*Online action detection is the task of predicting the action as soon as it happens in a streaming video. A major challenge is that the model does not have access to the future and has to solely rely on the history, i.e., the frames observed so far, to make predictions. It is therefore important to accentuate parts of the history that are more informative to the prediction of the current frame. We present GateHUB, **Gate**d **H**istory **U**nit with **B**ackground Suppression, that comprises a novel position-guided gated cross-attention mechanism to enhance or suppress parts of the history as per how informative they are for current frame prediction. GateHUB further proposes Future-augmented History (FaH) to make history features more informative by using subsequently observed frames when available. In a single unified framework, GateHUB integrates the transformer's ability of long-range temporal modeling and the recurrent model's capacity to selectively encode relevant information. GateHUB also introduces a background suppression objective to further mitigate false positive background frames that closely resemble the action frames. Extensive validation on three benchmark datasets, THUMOS, TVSeries, and HDD, demonstrates that GateHUB significantly outperforms all existing methods and is also more efficient than the existing best work. Furthermore, a flow-free version of GateHUB is able to achieve higher or close accuracy at 2.8× higher frame rate compared to all existing methods that require both RGB and optical flow information for prediction.*

1. Introduction

Online action detection is the task to predict actions in a streaming video as they unfold [12]. It is critical to applications including autonomous driving, public safety, virtual and augmented reality. Unlike action detection in the offline setting, where the entire untrimmed video is observable at any given moment, a major challenge for online action detection is that the predictions are solely based on observations of history without access to video frames in the future. The model needs to build a causal reasoning of the present in correlation to what happened hitherto, and as efficiently as possible for the online setting.

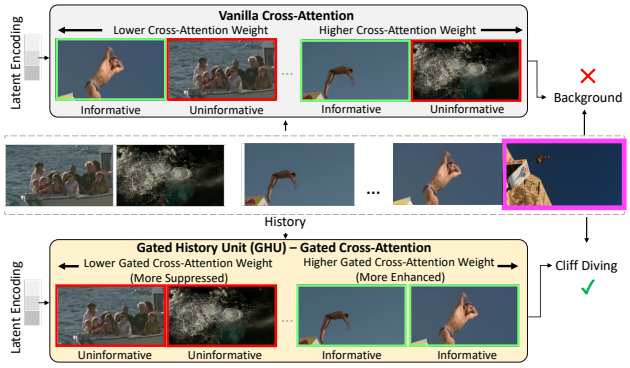

Figure 1. We show an example video stream (middle row) where the current frame (magenta) contains *Cliff Diving* action. Weights from vanilla cross-attention (top row) do not correlate with how informative each history frame is to current frame prediction, leading to incorrect prediction of *Background*. Our novel Gated History Unit (GHU) (bottom row) calibrates cross-attention weights using gating scores to enhance history frames that are informative to current frame prediction (green) and suppress uninformative ones (red), leading to accurate prediction of *Cliff Diving*.

Prior work for online action detection [14, 15, 19, 35, 48, 52] include recurrent-based LSTMs [21] and GRUs [9] that are prone to forgetting informative history as sequential frame processing is ineffective in preserving long-range interactions. Emerging methods [46, 49] employ transformers [42] to mitigate this by encoding sequential frames in parallel via self-attention. Some improve model efficiency by using cross-attention [23, 49] to compress the video sequence into a fixed-sized latent encoding for prediction.

Fig. 1 shows an example video stream (middle row) where the latest (current) frame contains *Cliff Diving* action. It is worth noting that, as commonly observed in video sequences, not every history frame is informative for current frame prediction (*e.g.* frames showing people cheering

*Authors with equal contribution.
This work was done as Junwen Chen's internship project at Microsoft.

or camera panning in Fig. 1). Existing transformer-based approaches [49] use vanilla cross-attention to learn attention weights for history frames that determine their contribution to the current frame prediction. Such attention weights do not correlate with how informative each history frame is to current frame prediction. As shown in Fig. 1 (top row), when history frames are ordered from lower to higher cross-attention weights for vanilla cross-attention, frames that are informative for current frame prediction may have lower weights while uninformative frames may have higher weights, leading to incorrect current frame prediction. Another common challenge for existing methods is false positive prediction for background frames that closely resemble action frames (*e.g*. pre-shot routine before golf swing). Existing methods also do not leverage that although future frames are not available for the current frame prediction, subsequently observed frames that are future to the history can be leveraged to enhance history encoding, which in return improves current frame prediction.

To address the above limitations, we propose GateHUB, **Gate**d **H**istory **U**nit with **B**ackground suppression. GateHUB comprises a novel Gated History Unit (GHU), a position-guided gated cross-attention module that enhances informative history while suppressing uninformative frames via gated cross-attention (as shown in Fig. 1, bottom row). GHU enables GateHUB to encode more informative history into the latent encoding to better predict for current frame. GHU combines the benefit of an LSTM-inspired gating mechanism to filter uninformative history with the transformer's ability to effectively learn from long sequences.

GateHUB leverages *future frames for history* by introducing Future-augmented History (FaH). FaH extracts features for a history frame using its future, *i.e*. the subsequently *observed* frames. This makes a history frame aware of its future and helps it to be more informative for current frame prediction. To tackle the common false positives in prior art, GateHUB proposes a novel background suppression objective that has different treatments for low-confident action and background predictions. These novel approaches enable GateHUB to outperform all existing methods on common benchmark datasets THUMOS [22], TVseries [12], and HDD [36]. Keeping model efficiency in mind for the online setting, we also validate that GateHUB is more efficient than the existing best method [49] while being more accurate. Moreover, our proposed optical flow-free variant is $2.8\times$ faster than all existing methods that require both RGB and optical flow data with higher or close accuracy. To summarize, our main contributions are:

1. Gated History Unit (GHU), a novel position-guided gated cross-attention that explicitly enhances or suppresses parts of video history as per how informative they are to predicting action for the current frame.
2. Future-augmented History (FaH) to extract features for a history frame using its subsequently observed frames to enhance history encoding.
3. A background suppression objective to mitigate the false positive prediction of background frames that closely resemble the action frames.
4. GateHUB is more accurate than all existing methods and is also more efficient than the existing best work. Moreover, our proposed optical flow-free model is $2.8\times$ faster compared to all existing methods that require both RGB and optical flow information while achieving higher or close accuracy.

2. Related Work

Online Action Detection. Previous methods for online action detection include use 3D ConvNet [12], reinforcement learning [17], recurrent networks [14, 19, 35, 35, 48, 52] and more recently, transformers [46, 49]. The primary challenge in leveraging history is that for long untrimmed videos, its length becomes intractably long over time. To make it computationally feasible, some [14, 19, 35, 46] make the online prediction conditioned only on the most recent frames spanning less than a minute. This way the history beyond this duration that might be informative to current frame predictions is left unused. TRN [48] mitigates this by the hidden state in LSTMs [21] to memorize the entire history during inference. But LSTM limits its ability to model long-range temporal interactions. More recently, [49] proposes to scale transformers to the history spanning longer duration. However, not every history frame is informative and useful. [49] lacks the forgetting mechanism of LSTM to filter uninformative history which causes it to encode uninformative history into the encoding leading to incorrect predictions. Our Gated History Unit (GHU) and Future-augmented History (FaH) combine the benefits of LSTM's selective encoding and transformer's long range modeling to leverage long-duration history more informatively to outperform all previous methods.

Transformers for Video Understanding. Transformers can achieve superior performance on video understanding tasks by effectively modeling the spatio-temporal context via attention. Most of the previous transformer-based methods [1, 3, 16, 33] focus on action recognition in trimmed videos [6] (videos spanning few seconds) due to the quadratic complexity w.r.t. video length. Untrimmed videos have a longer duration from a few minutes to hours and contain frames with irrelevant actions (labeled as background). Temporal action localization (TAL) [4, 18, 27, 30, 38, 39, 47, 51, 53, 54] and temporal action proposal generation (TAP) [27, 28, 40] are two fundamental tasks in untrimmed video understanding. AGT [32] proposes activity graph transformer for TAL based on DETR [5]. TAPG

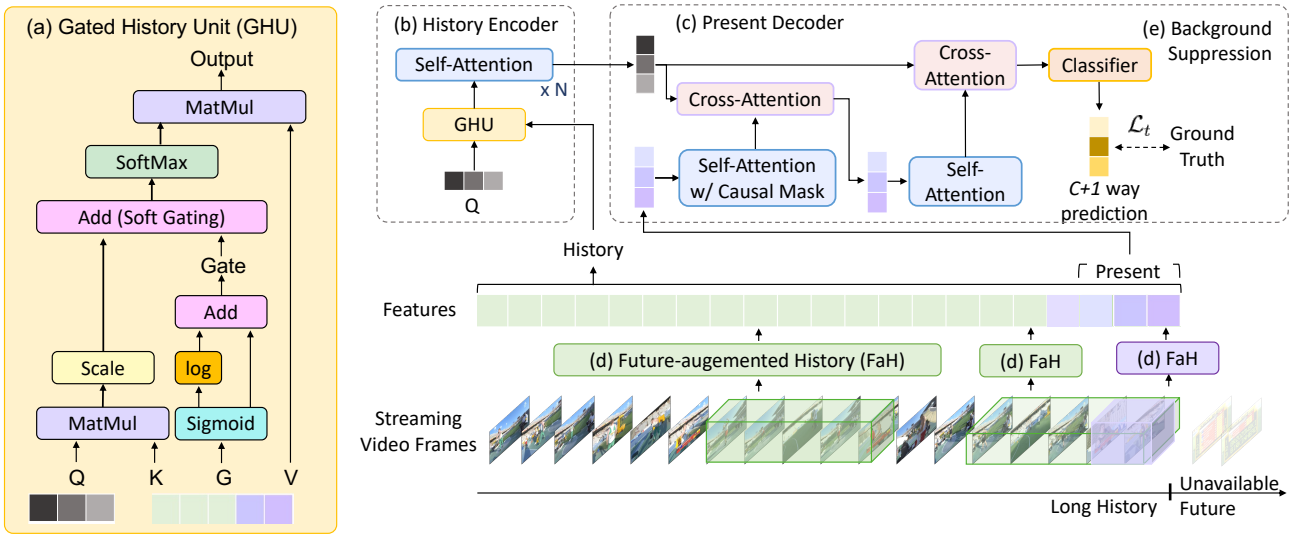

Figure 2. **Model Overview.** GateHUB comprises a novel Gated History Unit (GHU) (a) as part of History Encoder (b) to explicitly enhance or suppress history frames, *i.e.* streaming video frames observed so far, as per how informative they are to current frame prediction. GHU encodes them by cross-attending with a latent encoding (Q). GateHUB uses Future-augmented History features (FaH) (d) to encode each history frame using t_f subsequently observed future frames. The Present Decoder (c) correlates with history by cross-attending the encoded history with the present, *i.e.*, a small set of most recent frames, to make current frame prediction. We subject the prediction to a background suppression loss (e) to reduce false positives by effectively separating action frames from closely resembling background frames.

[44] applies transformer to predict the activity boundary for TAP. However, unlike TAL and TAP which are both offline tasks having access to the entire video, online action detection does not have access to the future and requires causal understanding from history to present. We follow the existing transformer-based streaming tasks [8, 20, 49] and apply a causal mask to address online action detection.

Long Sequence Modeling. To model long input sequences, recent work [13] proposes to reduce complexity by factorizing [41] or subsampling the inputs [7]. Another group of work focuses on modifying the internal dense self-attention module to boost the efficiency [2, 45]. More recently, Perceiver [24] and PerceiverIO [23] propose to cross-attend long-range inputs to a small fixed-sized latent encoding, adding further flexibility in terms of input and reducing the computational complexity. However, unlike our GHU, PerceiverIO lacks an explicit mechanism to enhance/suppress history frames making it sub-optimal for online action detection. Our method uses LSTM-inspired gating to calibrate cross-attention to enhance/suppress history frames per their informative-ness while employing transformers to learn from long history sequences effectively.

3. Methodology

Given a streaming video sequence $\mathbf{h} = [h_t]_{t=-T+1}^{0}$, our task is to identify *if* and *what* action $y_0 \in \{0, 1, ..., C\}$ occurs at the current frame h_0. We have a total of C action classes and label 0 for background frames with no action. Since future frames $h_1, h_2, ...,$ are NOT accessible, the model makes the $C+1$-way prediction for the current frame based on the recent T frames, $[h_t]_{t=-T+1}^{0}$, observed up until the current frame. While T may be large in an untrimmed video stream, as shown in the top row of Fig. 1, all frames observed in past history $[h_t]_{t=-T+1}^{-1}$ may not be equally informative to the prediction for the current frame.

3.1. Gated History Unit based History Encoder

To make the $C+1$-way prediction accurately for current frame h_0 based on T history frames, $\mathbf{h} = [h_t]_{t=-T+1}^{0}$, we employ transformers to first encode the video sequence history and then associate the current frame with the encoding for prediction. Inspired by the recently introduced PerceiverIO [23], our method consists of a History Encoder (Fig. 2b) that uses cross-attention to project the variable length history to a fixed-length learned latent encoding. Using cross-attention is more efficient than using self-attention because its computational complexity is quadratic w.r.t. latent encoding size instead of the video sequence length which is typically orders of magnitude larger. This is crucial to developing a model for the online setting. However, as shown in Fig. 1, vanilla cross-attention, as used in PerceiverIO and LSTR [49], fails to learn attention weights for history frames that correlate with how informative each history frame is for h_0 prediction. We therefore introduce a novel Gated History Unit (GHU) (Fig. 2a) that has a position-guided gated cross-attention mechanism which learns a set of gating scores G to calibrate the attention weights to effectively enhance or suppress history frames based on how informative they are to current frame prediction.

Specifically, given $\mathbf{h} = [h_t]_{t=-T+1}^{0}$ as the streaming sequence of T history frames ending at current frame h_0, we encode $\mathbf{h}$ with a feature extraction backbone, u, followed by a linear encoding layer $\mathbf{E}$. We then subject the output to a learnable position encoding, $\mathbf{E_{pos}}$, relative to the current frame, h_0, to give $\mathbf{z^h} = u(\mathbf{h})\mathbf{E} + \mathbf{E_{pos}}$ where $u(\mathbf{h}) \in \mathbb{R}^{T\times M}$, $\mathbf{E} \in \mathbb{R}^{M\times D}$, $\mathbf{z^h} \in \mathbb{R}^{T\times D}$ and $\mathbf{E_{pos}} \in \mathbb{R}^{T\times D}$. M and D denote the dimensions of extracted features and post-linear encoding features, respectively. We also define a learnable latent query encoding, $\mathbf{q} \in \mathbb{R}^{L\times D}$, that we cross-attend with $\mathbf{h}$. Following the standard multi-headed cross-attention setup [23, 24], let N_{heads} be the number of heads in GHU such that $Q_i = \mathbf{q}\mathbf{W}_i^q$, $K_i = \mathbf{z^h}\mathbf{W}_i^k$, $V_i = \mathbf{z^h}\mathbf{W}_i^v$ be the queries, keys and values, respectively, for each head $i \in \{1, \ldots, N_{heads}\}$ (Fig. 2a) where projection matrices $\mathbf{W}_i^q, \mathbf{W}_i^k \in \mathbb{R}^{D\times d_k}$ and $\mathbf{W}_i^v \in \mathbb{R}^{D\times d_v}$. We assign $d_k = d_v = D/N_{heads}$ in our set up [42]. Next, we obtain the position-guided gating scores, G, for $\mathbf{h}$ as,

$$\mathbf{z^g} = \sigma(\mathbf{z^h}\mathbf{W}^g) \tag{1}$$

$$G = \log(\mathbf{z^g}) + \mathbf{z^g} \tag{2}$$

where $\mathbf{W}^g \in \mathbb{R}^{D\times 1}$ is the matrix projecting each history frame to a scalar. $\mathbf{z^g} \in \mathbb{R}^{T\times 1}$ is a sequence of scalars for the history frames $\mathbf{h}$ after applying sigmoid σ. $G \in \mathbb{R}^{T\times 1}$ is the gating score sequence for history frames in GHU. By using $\mathbf{z^h}$ which already contains the position encoding, the gates are guided by the relative position of the history frame to the current frame h_0. As also shown in Fig. 2a, we now compute the gated cross-attention for each head, GHU_i, as,

$$GHU_i = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}} + G\right) V_i \tag{3}$$

and multi-headed gated cross-attention defined as,

$$\text{MultiHeadGHU}(Q, K, V, G) = \text{Concat}([GHU_i]_{i=0}^{N_{heads}})\mathbf{W^o} \tag{4}$$

where $\mathbf{W}^o \in \mathbb{R}^{D\times D}$ re-projects the attention output to D dimension. It is possible to define G separately for each head but in our method, we find sharing G across all heads to perform better (Sec. 4.4). From Eqn. 1 and 2, we can observe that each scalar in $\mathbf{z^g}$ lies in $[0, 1]$ due to sigmoid which implies that each gating score in G lies in $[-\inf, 1]$. This enables the softmax function in Eqn. 3 to calibrate the attention weight for each history frame by a factor in $[0, e]$ such that a factor in $[0, 1)$ suppresses a given history frame and a factor in $(1, e]$ enhances a given history frame. This provides an explicit ability to GHU to learn to calibrate the attention weight of a history frame based on how informative the history frame is for prediction of h_0. Unlike previous methods with relative position bias [11, 31], G is input-dependent and learns based on the history frame and its position w.r.t. current frame. This enables GHU to assess how informative each history frame is based on its feature representation and relative position from the current frame h_0.

We feed the output of GHU to a series of N self-attention layers to obtain the final history encoding (Fig. 2b).

3.2. Hindsight is 2020: Future-augmented History

Existing methods [14, 19, 46, 48, 49] extract features for each frame by feed-forwarding the frame and optionally, a small set of past consecutive frames through pretrained networks like TSN [43] and I3D [6]. It is worth noting that although for current frame prediction its future is not available, for the history frames their *future* is accessible and this *hindsight* can potentially improve the encoding of history for current frame prediction. Existing methods do not have a mechanism to leverage this. This inspires us to propose a novel feature extraction scheme, Future-augmented History (FaH), where we aggregate observed future information into the features of a history frame to make it aware of its so far observable future. Fig. 2d illustrates the FaH feature extraction process. For a history frame h_t and a feature extraction backbone u, when t_f *future history frames* for h_t can be observed, FaH extracts features for h_t using a set of frames $[h_i]_{i=t}^{t+t_f}$ (*i.e.* history frame itself and its subsequently observed t_f future frames). Otherwise, FaH extracts features for h_t using a set of frames $[h_i]_{i=t-t_{ps}}^{t}$ (*i.e.* history frame itself and its past t_{ps} frames),

$$u(h_t) = \begin{cases} u([h_i]_{i=t-t_{ps}}^{t}) & \text{if } t > -t_f \\ u([h_i]_{i=t}^{t+t_f}) & \text{if } t <= -t_f \end{cases} \tag{5}$$

At each new time step with one more new frame getting observed, FaH will feed-forward through u twice to extract features for (1) the new frame using $[h_i]_{i=-t_{ps}}^{0}$ frames and (2) h_{-t_f} that is now eligible to aggregate future information using $[h_i]_{i=-t_f}^{0}$ frames (as shown in Fig. 2d purple and green cuboid respectively). Thus, FaH has the same time complexity as existing feature extraction methods. FaH does not trivially incorporate all available subsequently observed frames. Instead, it encodes only from a set of future frames that are the most relevant to a history frame (as we empirically explain later in Section 4.4).

3.3. Present Decoder

In order to correlate the present with history to make current frame prediction, we sample a subset of t_{pr} most recent history frames $[h_t]_{t=-t_{pr}-1}^{0}$ to model the present (*i.e.* the most immediate context) for h_0 using the Present Decoder (Fig. 2c). After extracting the features via FaH, we apply a learnable position encoding, $\mathbf{E_{pos}^{pr}}$, to each of the t_{pr} frame features and subject them to a multi-headed self-attention with a causal mask. The causal mask limits the influence of only the preceding frames on a given frame. We then cross-attend the output from self-attention with the history encoding from the History Encoder. Inspired by Perceiver [24], we repeat this process twice and the self-attention does not need a causal mask the second time.

Finally, we feed the output corresponding to each of t_{pr} frames to the classifier layer for prediction.

3.4. Background Suppression Objective

Existing online action detection methods [14, 19, 46, 48, 49] apply standard cross entropy loss for $C+1$-way multi-label per-frame prediction. Standard cross entropy loss does not consider that the "no action" background class does not belong to any specific action distribution and is semantically different from the C action classes. This is because background frames can be anything from completely blank at the beginning of a video to closely resemble action frames without actually being action frames (*e.g.*, aiming before making a billiards shot). The latter is a common cause for false positives in online action detection. In addition to the complex distribution of background frames, untrimmed videos suffer from a sharp data imbalance where background frames significantly outnumber action frames.

To tackle these challenges, we design a novel background suppression objective that applies separate emphasis on low-confident action and background predictions during training to increase the margin between action and background frames (Fig. 2e). Inspired by focal loss [29], our objective function, $\mathcal{L}_t$ for frame h_t is defined as,

$$\mathcal{L}_t = \begin{cases} -y_t^0(1-p_t^0)^{\gamma_b}\log(p_t^0) & \text{if } y_t^0 = 1 \\ -\Sigma_{i=1}^{C} y_t^i(1-p_t^i)^{\gamma_a}\log(p_t^i) & \text{otherwise} \end{cases} \quad (6)$$

where $\gamma_a, \gamma_b > 0$ enables low-confident samples to contribute more to the overall loss forcing the model to put more emphasis on correctly predicting these samples. Unlike original focal loss [29], our background suppression objective specializes for online action detection by applying separate γ to action classes and background. This separation is necessary to distinguish the action classes that have a more constrained distribution from the background class whose distribution is more complex and unconstrained. Our objective is the first attempt in online action detection to put separate emphasis on low-confident hard action and background predictions.

3.5. Flow-free Online Action Detection

Existing methods [14, 46, 48] for online action detection use optical flow in addition to RGB to capture fine-grained motion among frames. Computing optical flow takes much more time than feature extraction or model inference, and can be unrealistic for time-critical applications such as autonomous driving. This motivates us to develop an optical flow-free version of GateHUB that is able to achieve higher or close accuracy compared to existing methods without time-consuming optical flow estimation. To capture motion without optical flow using only RGB frames, we leverage multiple temporal resolutions using a spatio-temporal backbone such as TimeSformer [3]. We extract two feature vectors for a frame h_t by encoding a frame sequence sampled at a higher frame rate spanning a smaller time duration and another frame sequence sampled at a lower frame rate spanning a larger time duration. Similar to the setup using RGB and optical flow features, we concatenate the two feature vectors before feeding them to GateHUB.

4. Experiments

4.1. Datasets

Following existing online action detection work [14, 17, 46, 48, 49], we evaluate GateHUB on three common benchmark datasets – THUMOS'14, TVSeries, and HDD.

THUMOS'14 [22] consists of over 20 hours of sports video and is annotated with 20 actions. We follow prior work [46, 48] and train on the validation set (200 untrimmed videos) and evaluate on the test set (213 untrimmed videos).

TVSeries [12] includes 27 episodes of 6 popular TV shows with a total duration of 16 hours. It is annotated with 30 real-world everyday actions, *e.g.* open door, run, drink.

HDD (Honda Research Institute Driving Dataset) [37] includes 137 driving videos with a total duration of 104 hours. Following prior work [46], we use the vehicle sensor as input signal and divide data into 100 sessions for training and 37 sessions for testing.

4.2. Implementation Details

For TVSeries and THUMOS'14, following [14, 17, 46, 48, 49], we resample the videos at 24 FPS (frames per second) and then extract frames at 4 FPS for training and evaluation. The sizes of *history* and *present* are set to 1024 and 8 most recently observed frames, respectively, spanning durations of 256s and 2s correspondingly at 4 FPS. For HDD, following OadTR [46], we extract the sensor data at 3 FPS for training and evaluation. The sizes of *history* and *present* are 48 and 6 most recently observed frames respectively, spanning durations of 16s and 2s correspondingly at 3 FPS.

Feature Extraction. Following [46, 49], we use mmaction2 [10]-based two-stream TSN [43] pretrained on Kinetics-400 [6] to extract frame-level RGB and optical flow features for THUMOS'14 and TVSeries. We concatenate the RGB and optical flow features along channel dimension before feeding to the linear encoding layer in GateHUB. For HDD, we directly feed the sensor data as input to GateHUB. To fully leverage the proposed FaH, the feature extraction backbone needs to support multi-frame input. Since TSN only supports single-frame input, we explore spatio-temporal TimeSformer [3] (pretrained on Kinetics-600 using 96×4 frame sampling) that supports multiple-frame input. We set the time duration for past t_{ps} and future t_f frames under FaH to be 1s and 2s respectively. We use TimeSformer to extract RGB features and use TSN-based optical flow features as TimeSformer only supports RGB.

We also demonstrate FaH using RGB features from I3D [6] with results in the supplementary. For our flow-free version, we replace optical flow features with features obtained from an additional multi-frame input of RGB frames uniformly sampled from a duration of 2s. Please refer to supplementary for additional details.

Training. We train GateHUB for 10 epochs using Adam optimizer [26], weight decay of $5e^{-5}$, batch size of 50, OneCycleLR learning rate schedule of PyTorch [34] with pct_start of 0.25, $D = 1024$, latent encoding size $L = 16$, number of self-attention layers in History Decoder $N = 2$, $N_{heads} = 16$ for each attention layer and $\gamma_a = 0.6, \gamma_b = 0.2$ for background suppression.

Evaluation Metrics We follow the protocol of per-frame mean average precision (mAP) for THUMOS and HDD and calibrated average precision (mcAP) [12] for TVSeries.

4.3. Comparison with State-of-the-Art

Method	Feature Backbone		THUMOS14
	RGB	Optical Flow	mAP (%)
FATS [25]	TSN	TSN	59.0
IDN [14]			60.3
TRN [48]			62.1
PKD [52]			64.5
OadTR [46]			65.2
WOAD [19]			67.1
LSTR [49]			69.5
GateHUB (Ours)			**70.7**
TRN [48]	TimeSformer	TSN	68.5
OadTR [46]			65.5
LSTR [49]			69.6
GateHUB (Ours)			**72.5**

Table 1. Online action detection results on THUMOS'14 comparing GateHUB with SoTA methods on mAP (%) when the RGB-based features are extracted from either TSN or TimeSformer. Optical flow-based features are extracted from TSN in all settings.

Table 1 compares GateHUB with existing state-of-the-art (SoTA) online action detection methods on THUMOS'14 for two different setups, one using RGB features from TSN [43] and the other using RGB features from TimeSformer [3]. Both setups use optical flow features from TSN. WOAD [19] uses RGB features from I3D (equivalent to TSN). For TSN RGB features, all mAP in Table 1 are as reported in the references. For TimeSformer RGB features, we use the official code for TRN, OadTR and LSTR for fair comparison. From the table, we can observe that GateHUB outperforms all existing methods by at least 1.2% when using RGB features from TSN. Moreover, GateHUB outperforms existing methods by a larger margin of at least 2.9% using RGB features from TimeSformer. GateHUB is also the first approach to surpass 70% on THUMOS'14 benchmark. This validates that GateHUB, comprising GHU, Background Suppression and FaH to holistically leverage the long history more informatively, outperforms all SoTA on THUMOS'14.

We further compare GateHUB with SoTA on TVSeries and HDD in Table 2a and 2b, respectively. Following protocol, we use RGB and optical flow features from TSN for TVSeries and sensor data for HDD. All results from SoTA are as reported in the references. We can observe that GateHUB outperforms all SoTA on both TVSeries and HDD. The large improvement on HDD using sensor data validates that GateHUB is also effective on data modalities other than RGB or optical flow.

Method	mcAP (%)
FATS [25]	84.6
IDN [14]	86.1
TRN [48]	86.2
PKD [52]	86.4
OadTR [46]	87.2
LSTR [49]	89.1
GateHUB (Ours)	**89.6**

(a)

Method	mAP (%)
CNN [12]	22.7
LSTM [36]	23.8
RED [17]	27.4
TRN [48]	29.2
OadTR [46]	29.8
GateHUB (Ours)	**32.1**

(b)

Table 2. Online action detection results comparing GateHUB with state-of-the-art methods on (a) TVSeries using RGB + Optical Flow data as input on mcAP metric and (b) HDD using sensor data as input on mAP metric.

4.4. GateHUB: Ablation Study

In this section, we conduct an ablation study to highlight the impacts of the novel components of GateHUB. Unless stated otherwise, all experiments are on THUMOS'14 using RGB and optical flow features from TSN.

Impact of Gated History Unit (GHU). We conduct an experiment where we test different variants of our Gated History Unit (GHU) by removing one or more of its design elements. Table 3a summarizes the results of this experiment. In the table, 'w/o GHU' refers to replacing GHU with vanilla cross-attention from Perceiver IO [23] and LSTR [49], *i.e.*, CrossAttention$(Q, K, V) =$ SoftMax$(QK^\intercal/\sqrt{d})$. In 'w/ GHU enhance only', we remove $\log(\mathbf{z^g})$ from Eqn. 2 that suppresses history frames, *i.e.* $G = \mathbf{z^g}$. Conversely, in 'w/ GHU suppress only', we remove $\mathbf{z^g}$ from Eqn. 2 that enhances history frames, *i.e.* $G = \log(\mathbf{z^g})$. In 'w/ GHU w/o position guidance', we operate on frame features before subjecting them to learned position encoding, *i.e.* $G = \log(\mathbf{z^{\tilde{g}}}) + \mathbf{z^{\tilde{g}}}$ where $\mathbf{z^{\tilde{g}}} = q(\mathbf{h})\mathbf{E}$. We also compare with 'w/ GHU per head' where G is learned separately for each cross-attention head.

Table 3a shows that our implementation of GHU significantly outperforms all other variants of GHU and cross-attention. We can observe that 'w/o GHU' performs 1.1% worse than 'w/ GHU'. This is because, without explicit gating, vanilla cross-attention fails to learn attention weights for history frames that correlate with how informative history frames are to current frame prediction (also depicted in Figure 1). Moreover, the lower performances of 'w/ GHU suppress only' and 'w/ GHU enhance only' validate that we need to both enhance the informative history frames and

Method	mAP (%)
w/ GHU (Ours)	**70.7**
w/o GHU	69.6
w/ GHU suppress only	70.5
w/ GHU enhance only	70.5
w/ GHU w/o position-guidance	70.3
w/ GHU per head	68.0

(a)

Method	mAP (%)
Ours $\gamma_a > \gamma_b$	**70.7**
Ours $\gamma_a < \gamma_b$	70.2
w/ cross-entropy	69.9
w/ standard focal loss	70.2

(b)

Method	Future Duration	mAP (%)
w/o FaH	-	71.5
w/ FaH	0.5	71.1
	1s	72.0
	2s	**72.5**
	4s	71.4

(c)

Table 3. Ablation study comparing different variants of (a) Gated History Unit (GHU), (b) background suppression objective and (c) Future-augmented History (FAH). Ablation in (a) and (b) is conducted with RGB features from TSN and in (c) are conducted with RGB features from TimeSformer. Optical flow features from TSN are used in all settings.

suppress the uninformative ones to achieve the best performance. Without the ability to both enhance and suppress, the model may encode uninformative history frames into the latent encoding or inadequately emphasize the informative ones, leading to worse performance. The performance is also lower when using history frame features without position encoding ('w/ GHU w/o position guidance'). This is because without position guidance, the model cannot assess the relative position of a particular history frame w.r.t. the current frame which is an important factor in deciding how informative a history frame is to current frame prediction. We also find having separate G per head ('w/ GHU per head) performs much worse than sharing G across heads due to overfitting from N_{heads} times more parameters.

Impact of Background Suppression. We compare our background suppression objective with standard cross-entropy loss (*i.e.*, $\gamma_a = \gamma_b = 0$) and standard focal loss(*i.e.*, $\gamma_a = \gamma_b \neq 0$) [29] as shown in Table 3b. First, compared to our background suppression objective, both standard cross-entropy and focal loss achieve lower accuracy. This validates that it is important to put separate emphasis on the low-confident action and background predictions to effectively differentiate action frames and closely resembling background frames. Furthermore, we find that across different combinations of γ_a and γ_b, choosing a pair where $\gamma_a > \gamma_b$ leads to higher accuracy. Specifically, we find $\gamma_a = 0.05$ and $\gamma_b = 0.025$ to give the highest accuracy. This can be attributed to the high data imbalance. Action frames are much lower in number than background frames and therefore require a stronger emphasis than background.

Impact of Future-augmented History (FaH). Table 3c shows the ablation on FaH. Since the TSN backbone is not compatible with multi-frame input, we conduct this study using RGB features from TimeSformer. The table shows that with 2s of future information incorporated into history features, we achieve the best accuracy which is 1% higher than without future-augmented history ('w/o FaH'). The accuracy is also improved with 1s of future information incorporated into history features. We further observe that the accuracy drops when future duration is much longer *e.g.* 4s or much shorter *e.g.* 0.5s. This shows that making a history frame aware of its future enables it to be more informative for current frame prediction. At the same time, future duration up to a certain extent (in our case, 2s) can encode meaningful future into history frames. Much beyond that, the future changes enough to be of little use for a given history frame, while much shorter future duration may also add noise rather than information. We wish to emphasize that all future duration are bound by the frames observed so far and do not extend into inaccessible future frames.

GateHUB Present Decoder. Table 4a shows the ablation study on our Present Decoder by altering different aspects of the design. Unlike the original PerceiverIO [23], where the output queries are independent, we model the present (equivalent of output queries in our method) via a causal self-attention and cross-attend it with history encoding multiple times (inspired by Perceiver [24]). We can observe in Table 4a that treating present frames independently ('*i.e.* w/o self-attention') and having only one cross-attention ('*i.e.* w/ cross-attention only at first layer') both reduce the accuracy considerably. Unlike LSTR [49] that uses a FIFO queue with disjoint long-term and short-term memory, in our design, the sequences of history and present frames fully overlap. Table 4a shows that having disjoint history and present frames (*i.e.*, 'w/ disjoint history and present') leads to a 1.3% lower performance, further validating our design of Present Decoder and GateHUB overall.

4.5. GateHUB Efficiency

For online action detection setting, model efficiency is an important metric. We compare GateHUB with existing methods w.r.t. parameter count, GFLOPs, and inference speed in terms of FPS as shown in Table 4b. We first observe that GateHUB achieves the highest accuracy with the least number of model parameters compared to all existing methods. We also note that while methods like OadTR [46] and TRN [48] are more efficient in terms of GFLOPs, their accuracy is much lower. GateHUB achieve a more favorable accuracy-efficiency trade-off with fewer GFLOPs than the existing best method LSTR [49] while obtaining a higher accuracy. All aforementioned methods require optical flow computation which is time-consuming, therefore the inference speed of these methods is governed by the optical flow computation speed of 8.1 FPS. Meanwhile, our flow-free model obviates optical flow computation by using RGB features from TimeSformer at two different frame

Method	mAP (%)
Ours	**70.7**
w/o self-attention	67.7
w/ cross-attention only at layer 1	68.6
w/ disjoint history and present	69.4

(a)

Method	Model		Inference Speed (FPS)					mAP(%)
	Parameter Count	GFLOPs	Optical Flow Computation	RGB Feature Extraction	Flow Feature Extraction	Model	Overall	
TRN [50]	402.9M	1.46	8.1	70.5	14.6	123.3	8.1	62.1
OadTR [46]	75.8M	2.54	8.1	70.5	14.6	110.0	8.1	65.2
LSTR [49](Flow-free)	54.2M	6.33	-	22.7	-	99.2	22.7	63.5
LSTR [49]	58.0M	7.53	8.1	70.5	14.6	91.6	8.1	69.5
Ours (Flow-free)	41.8M	3.47	-	22.7	-	83.3	**22.7**	66.5
Ours	45.2M	6.98	8.1	70.5	14.6	71.2	8.1	**70.7**

(b)

Table 4. (a) Ablation study for Present Decoder. (b) Efficiency comparison of GateHUB using RGB and optical flow features and our optical flow-free version with existing methods. GateHUB using RGB and optical flow has the least parameter count compared to existing methods, and higher accuracy and lower GFLOPs than the existing best method. Moreover, our flow-free version attains higher or close accuracy compared to existing methods that require RGB and optical flow features at 2.8× faster inference speed.

rates, and attains higher or close accuracy compared to existing work at 2.8× faster inference speed. When compared with flow-free LSTR, GateHUB achieves 3% higher mAP, thus providing a significantly better speed-accuracy tradeoff than the existing best method.

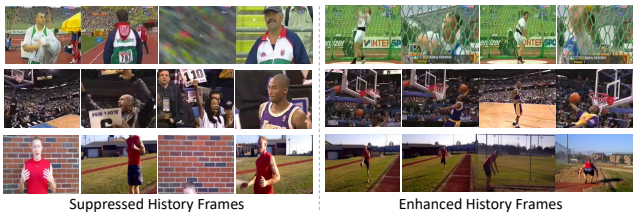

Figure 3. Examples of the most suppressed and most enhanced history frames as per the gating score learned by GHU. Frames in the same row belong to the same video.

4.6. Qualitative Evaluation

Gated History Unit (GHU). We qualitatively assess the effect of GHU by visualizing examples of the most suppressed and most enhanced history frames in a streaming video when ordered as per the gating scores G learned by GHU in Eqn. 2. Fig. 3 shows examples from three videos where frames in the same row belong to the same video. From the figure, we can observe that GHU learns to suppress frames that exhibit no discernible action from the C action classes. The suppressed frames either have people arbitrarily moving or are uninformative background frames (*e.g.* crowd cheering) that convey no useful information to predict action for the current frame. On the other hand, GHU learns to maximize emphasis on history frames with action from the C classes and on background frames that provide meaningful context to determine the current frame action (*e.g.* long jump athlete running toward the pit).

Current Frame Prediction. We visualize GateHUB's current frame prediction in Fig. 4. The confidence in the range $[0, 1]$ on y-axis denotes the probability of predicting the correct action (*i.e. High Jump* in Fig. 4). We can observe that GateHUB with GHU (red) is effective in reducing false positives for background frames that closely resemble action frames compared to without GHU (orange). Please refer to supplementary material with visualizations highlighting more online action detection scenarios.

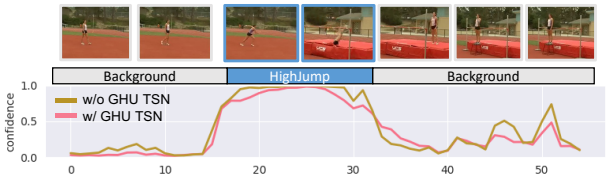

Figure 4. Visualization of GateHUB's online prediction. The curves indicate the predicted confidence of the ground-truth class (*High Jump*) using TSN backbone with and without GHU.

5. Conclusion and Future Work

We present GateHUB for online action detection in untrimmed streaming videos. It consists of novel designs including Gated History Unit (GHU), Future-augmented History (FaH), and a background suppression loss to more informatively leverage history and reduce false positives for current frame prediction. GateHUB achieves higher accuracy than all existing methods for online action detection, and is more efficient than the existing best method. Moreover, its optical flow-free variant is 2.8× faster than previous methods that require both RGB and optical flow while obtaining higher or close accuracy.

While GateHUB outperforms all existing methods, there is ample room for improvement. Although GateHUB can leverage long history, the length is still finite and may not be adequate when actions occur infrequently over long duration. It would be worthwhile to investigate ways to leverage history sequences of any length. Another challenge is slow motion action which is uncommon and can have considerably different temporal distribution, making it difficult to predict as accurately as common actions.

Acknowledgements: At Rochester Institute of Technology, Junwen Chen and Yu Kong are supported by NSF SaTC award 1949694, and the Army Research Office under grant number W911NF-21-1-0236. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv:2103.15691*, 2021. 2

[2] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 3

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *ICML*, 2021. 2, 5, 6

[4] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. SST: Single-stream temporal action proposals. In *CVPR*, 2017. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *CVPR*, 2017. 2, 4, 5, 6

[7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 3

[8] Xie Chen, Yu Wu, Zhenghao Wang, Shujie Liu, and Jinyu Li. Developing real-time streaming transformer transducer for speech recognition on large-scale dataset. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021. 3

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP*, 2014. 1

[10] MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2, 2020. 5

[11] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. *ACL*, 2019. 4

[12] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *ECCV*, 2016. 1, 2, 5, 6

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3

[14] Hyunjun Eun, Jinyoung Moon, Jongyoul Park, Chanho Jung, and Changick Kim. Learning to discriminate information for online action detection. In *CVPR*, 2020. 1, 2, 4, 5, 6

[15] Hyunjun Eun, Jinyoung Moon, Jongyoul Park, Chanho Jung, and Changick Kim. Temporal filtering networks for online action detection. *Pattern Recognition*, 2021. 1

[16] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *ICCV*, 2021. 2

[17] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. RED: Reinforced encoder-decoder networks for action anticipation. In *BMVC*, 2017. 2, 5, 6

[18] Jiyang Gao, Zhenheng Yang, Chen Sun, Kan Chen, and Ram Nevatia. TURN TAP: Temporal unit regression network for temporal action proposals. In *ICCV*, 2017. 2

[19] Mingfei Gao, Yingbo Zhou, Ran Xu, Richard Socher, and Caiming Xiong. WOAD: Weakly supervised online action detection in untrimmed videos. In *CVPR*, 2021. 1, 2, 4, 5, 6

[20] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. *ICCV*, 2021. 3

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1, 2

[22] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *CVIU*, 2017. 2, 5

[23] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. 1, 3, 4, 6, 7

[24] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. *arXiv:2103.03206*, 2021. 3, 4, 7

[25] Young Hwi Kim, Seonghyeon Nam, and Seon Joo Kim. Temporally smooth online action detection using cycle-consistent future anticipation. *Pattern Recognition*, 2021. 6

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[27] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-matching network for temporal action proposal generation. In *ICCV*, 2019. 2

[28] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018. 2

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 5, 7

[30] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *CVPR*, 2019. 2

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, 2021. 4

[32] Megha Nawhal and Greg Mori. Activity graph transformer for temporal action localization. *arXiv:2101.08540*, 2021. 2

[33] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv:2102.00719*, 2021. 2

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[35] Sanqing Qu, Guang Chen, Dan Xu, Jinhu Dong, Fan Lu, and Alois Knoll. LAP-Net: Adaptive features sampling via learning action progression for online action detection. *arXiv:2011.07915*, 2020. 1, 2

[36] Vasili Ramanishka, Yi-Ting Chen, Teruhisa Misu, and Kate Saenko. Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *CVPR*, pages 7699–7707, 2018. 2, 6

[37] Vasili Ramanishka, Yi-Ting Chen, Teruhisa Misu, and Kate Saenko. Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *CVPR*, 2018. 5

[38] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 2017. 2

[39] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016. 2

[40] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. *arXiv:2102.01894*, 2021. 2

[41] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 3

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 1, 4

[43] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 4, 5, 6

[44] Lining Wang, Haosen Yang, Wenhao Wu, Hongxun Yao, and Hujie Huang. Temporal action proposal generation with transformers. *arXiv preprint arXiv:2105.12043*, 2021. 3

[45] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv:2006.04768*, 2020. 3

[46] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. Oadtr: Online action detection with transformers. *ICCV*, 2021. 1, 2, 4, 5, 6, 7, 8

[47] Huijuan Xu, Abir Das, and Kate Saenko. R-C3D: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017. 2

[48] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S Davis, and David J Crandall. Temporal recurrent networks for online action detection. In *ICCV*, 2019. 1, 2, 4, 5, 6, 7

[49] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. *NeurIPS*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[50] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. *arXiv:2104.06399*, 2021. 8

[51] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Temporal query networks for fine-grained video understanding. In *CVPR*, pages 4486–4496, 2021. 2

[52] Peisen Zhao, Jiajie Wang, Lingxi Xie, Ya Zhang, Yanfeng Wang, and Qi Tian. Privileged knowledge distillation for online action detection. *arXiv:2011.09158*, 2020. 1, 2, 6

[53] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017. 2

[54] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. Enriching local and global contexts for temporal action localization. In *ICCV*, pages 13516–13525, 2021. 2