TVNet: Temporal Voting Network for Action Localization

Hanyuan Wang[a], Dima Damen[b], Majid Mirmehdi[c] and Toby Perrett[d]
Department of Computer Science, University of Bristol, Bristol, U.K.

Keywords: Action Detection, Action Localization, Action Proposals.

Abstract: We propose a Temporal Voting Network (TVNet) for action localization in untrimmed videos. This incorporates a novel Voting Evidence Module to locate temporal boundaries, more accurately, where temporal contextual evidence is accumulated to predict frame-level probabilities of start and end action boundaries. Our action-independent evidence module is incorporated within a pipeline to calculate confidence scores and action classes. We achieve an average mAP of 34.6% on ActivityNet-1.3, particularly outperforming previous methods with the highest IoU of 0.95. TVNet also achieves mAP of 56.0% when combined with PGCN and 59.1% with MUSES at 0.5 IoU on THUMOS14 and outperforms prior work at all thresholds. Our code is available at https://github.com/hanielwang/TVNet.

1 INTRODUCTION

While humans are capable of identifying event boundaries in long videos (Zacks et al., 2001), in a remarkably consistent fashion, current approaches fall short of optimal performance (Zhao et al., 2017; Chao et al., 2018; Lin et al., 2018; Zeng et al., 2019; Lin et al., 2019; Xu et al., 2020; Lin et al., 2020; Su et al., 2021) despite being trained on large and varied datasets (Jiang et al., 2014; Caba Heilbron et al., 2015), primarily due to varying action durations and densities (Lin et al., 2018; Liu et al., 2019; Su et al., 2021).

Current state-of-the-art action-recognition methods attach equal importance to each frame when determining where action boundaries occur (Lin et al., 2018; Lin et al., 2019; Liu et al., 2019; Su et al., 2021). Intuitively, a frame near the start of an action should be better suited to predicting the start point of the action than a frame in the middle of the action, and similarly for end points. One would expect this to result in more accurate boundary locations. In this paper, we incorporate this intuition, while still utilising all frames in the untrimmed video, such that each and every frame can contribute evidence when predicting boundary locations. Distinct from previous approaches, we weight this evidence depending on its distance to the boundary. We propose the Temporal Voting Network (TVNet), which models relative boundary locations via contextual evidence voting. Specifically, this contains a proposed Voting Evidence Module to locate temporal boundaries based on accumulating temporal contextual evidence.

Our key contributions can be summarized as follows: (1) We introduce a novel voting network, TVNet, for accurate temporal action localization. (2) We evaluate our method on two popular action localization benchmarks: ActivityNet-1.3 and THUMOS14. Our method achieves significant improvements, especially at high IoU thresholds, demonstrating more precise action boundaries. (3) We perform a detailed ablation, finding that both temporal context voting and attention learning are crucial to TVNet's performance.

2 RELATED WORK

Methods for action recognition assume trimmed videos as input, which they directly classify (Simonyan and Zisserman, 2014; Tran et al., 2015; Carreira and Zisserman, 2017; Feichtenhofer et al., 2019). Instead, temporal action localization works aim to locate actions in untrimmed videos as well as classify them. Most works, like ours, investigate proposal generation *and* proposal evaluation (Zhao et al., 2017; Chao et al., 2018; Lin et al., 2018; Lin et al.,

[a] https://orcid.org/0000-0002-9349-9597
[b] https://orcid.org/0000-0001-8804-6238
[c] https://orcid.org/0000-0002-6478-1403
[d] https://orcid.org/0000-0002-1676-3729

Wang, H., Damen, D., Mirmehdi, M. and Perrett, T.
TVNet: Temporal Voting Network for Action Localization.
DOI: 10.5220/0010868900003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP*, pages 550-558
ISBN: 978-989-758-555-5; ISSN: 2184-4321

2019; Liu et al., 2019; Long et al., 2019; Xu et al., 2020; Bai et al., 2020; Chen et al., 2020; Su et al., 2021), but some just focus on proposal evaluation, such as (Zeng et al., 2019; Liu et al., 2021).

Notable studies include MGG (Liu et al., 2019), which uses frame-level action probabilities to generate segment proposals. BSN (Lin et al., 2018) and BMN (Lin et al., 2019) generate proposals based on boundary probabilities, using frame-level positive and negative labels as supervision. However, their probabilities are calculated independently and each temporal location is considered as an isolated instance, leading to sensitivity to noise. Therefore, several works focus on exploiting rich contexts. G-TAD (Xu et al., 2020) proposes a model based on a graph convolutional network to incorporate multi-level semantic context into video features. GTAN (Long et al., 2019) involves contextual information with the feature using Gaussian kernels. The recently introduced BSN++ (Su et al., 2021) uses a complementary boundary generator to extract both local and global context. However, these works ignore that different contextual frames have different degrees of importance, which may lead to insufficient exploitation of context, and generate imprecise boundaries, especially in complex scenes.

In contrast, a context-aware loss function for action spotting in sports videos is proposed in (Cioppa et al., 2020), which treats frames according to their different temporal distances to the ground-truth moments. This method is designed to detect an event has occurred, and does not perform as well as specialist action localization methods when adapted to predict precise starting and ending boundaries. Our work is inspired by the notion of voting to incorporate contextual information, which was used for the task of moment localization in action completion (Heidarivincheh et al., 2018; Heidarivincheh et al., 2019). However, these works aim to recognise a single moment for retrieval, rather than start and end times of actions. We thus offer the first attempt to incorporate voting into action localization. We detail our method next.

3 METHOD

Our proposed method, TVNet, takes a feature sequence as input. It produces a set of candidate proposals using a Voting Evidence Module (VEM), where each frame in the sequence can contribute to boundary localization through voting, whether itself is a boundary frame or not. We then calculate confidence scores and classify these candidate proposals with an action classifier. An overview of TVNet is illustrated in Figure 1.

Section 3.1 offers a formal problem formulation. Section 3.2 introduces our main contribution, the Voting Evidence Module. Finally, Section 3.3 discusses how the VEM is used within the full proposal generation process.

3.1 Problem Definition

Given an untrimmed video V with length L, the feature sequence is denoted as $F = \{f_t\}_{t=1}^{T}$ with length T, extracted at rate $\frac{L}{T}$. Annotations of action instances in video V can be denoted as $\Psi = \{(s,\ e,\ a)\}_{k=1}^{K}$, where K is the total number of ground truth action instances, and s, e and a are starting time, ending time and label of action instance, respectively. The goal of the temporal action localization task is to predict a set of possible action instances $\hat{\Phi} = \{(\hat{s},\ \hat{e},\ \hat{c},\ \hat{a})\}_{m=1}^{M}$. Here, $\hat{s}$ and $\hat{e}$ are the starting and ending boundaries of the predictions, $\hat{c}$ is a confidence score for the proposal, $\hat{a}$ is the predicted class, and M is the number of predicted action instances. The annotation set Ψ is used to calculate the loss for training. The predicted instance set $\hat{\Phi}$ is expected to cover Ψ with high overlap and recall, so M is likely to be larger than K.

3.2 Voting Evidence Module (VEM)

Our premise is that each frame in the sequence offers information that would assist in locating the start and end of nearby frames. For each frame, we aim to predict its relative signed distance (in frames) to the start of an action of interest. Assume a frame is 5 frames past the start of the action. We denote this relative distance '-5', indicating the start of the action is 5 frames ago. In contrast, when a frame precedes the start of the action by say 2 frames, we denote '+2'. Similarly, we can predict the relative signed distance from a frame to the end of an action. To better understand our objective, we consider two cases. First, if a frame is within an action, the optimal model would be able to predict the relative start frame and the end frame of the ongoing action, from this frame. The second case is when the frame is part of the background. The optimal model can still predict the elapsed time since the last action concluded as well as the remaining time until the next action starts.

Evidently, a single frame's predictions can be noisy, so we utilise two techniques to manage the noise. The first is to only make these predictions within local temporal neighbourhoods, where evidence can be more reliable, and the second is to accumulate evidence from all frames in the sequence. We

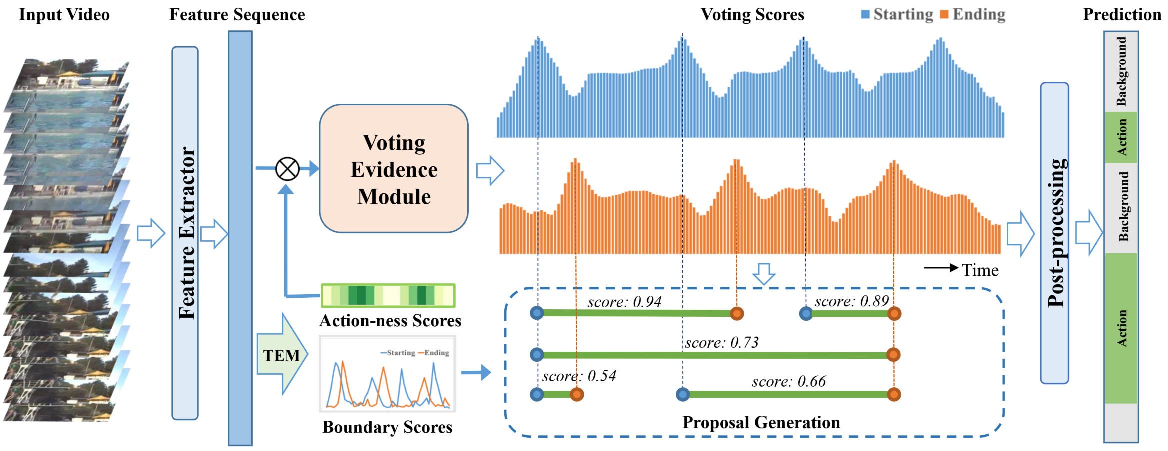

Figure 1: Overview of our proposed TVNet. Given an untrimmed video, frame-level features are extracted. Our main contribution, the Voting Evidence Module, takes in this feature sequence and outputs sequences of starting and ending voting scores. Local maxima in these voting scores are combined to form action proposals, which are then scored and classified.

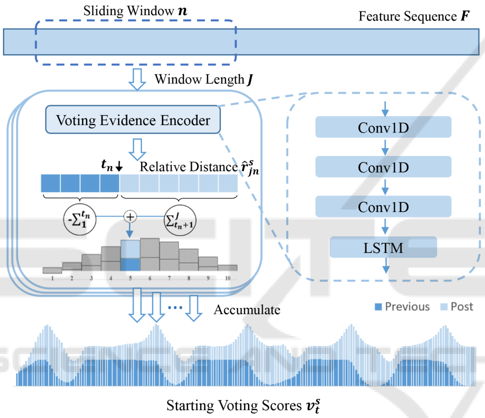

Figure 2: Illustration of our proposed Voting Evidence Module. We accumulate evidence from all frames to calculate boundary scores for starting.

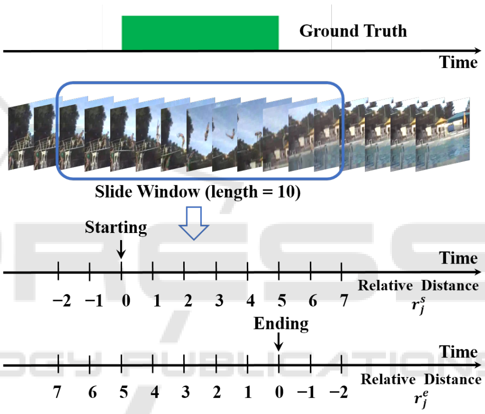
Ground Truth
Time
Slide Window (length = 10)
Starting
Time
−2
−1
0
1
2
3
4
5
6
7
Relative Distance r_j^s
Ending
Time
7
6
5
4
3
2
1
0
−1
−2
Relative Distance r_j^e

Figure 3: Supervision for relative distances of starting and ending action boundaries within a sliding window, from ground truth.

describe these next.

Voting Evidence Encoder. Given the feature sequence F, we use a sliding window of length J. Accordingly, we only use neighbourhood of size J in support of temporal boundaries. This is passed to one-dimensional temporal convolutional layers, in order to attend to local context, as well as an LSTM of past input for sequence prediction. The network construction is illustrated in Figure 2. The output prediction at each frame would be $\hat{R} = \left\{(\hat{r}_j^s, \hat{r}_j^e)\right\}_{j=1}^{J}$, and it denotes the relative distance $\hat{r}$ to the *closest* start/end to frame j.

We supervise the training of the VEM from the ground-truth. For each window, we use the relative distance between the current temporal location and the ground-truth boundaries as training labels. As shown in Figure 3, we have a ground-truth relative distance set $R = \left\{(r_j^s, r_j^e)\right\}_{j=1}^{J}$. The values of r_j^s, r_j^e reflect the distance between location j and the closest start location s^* as well as the closest end e^*. Therefore, the ground-truth relative distance is defined as $r_j^s = j - s^*$ for starting, and $r_j^e = e^* - j$ for ending.

We normalize the relative distance value to -1 to 1. We train the VEM for starting and ending separately, using the following MSE losses, shown here for a single window:

$$L^s = \frac{1}{J}\sum_{j=1}^{J}(\hat{r}_j^s - r_j^s)^2 \quad \text{and} \quad L^e = \frac{1}{J}\sum_{j=1}^{J}(\hat{r}_j^e - r_j^e)^2. \tag{1}$$

Voting Accumulation. We accumulate the predicted relative distance votes from all frames in the sequence as

$$
\begin{aligned}
v_t^s = & \sum_{n=1}^{N}\left(\sum_{j=1}^{t_n}(-\hat{r}_{jn}^s)+\sum_{j=t_n+1}^{J}\hat{r}_{jn}^s\right), \\
v_t^e = & \sum_{n=1}^{N}\left(\sum_{j=1}^{t_n}\hat{r}_{jn}^e+\sum_{j=t_n+1}^{J}(-\hat{r}_{jn}^e)\right),
\end{aligned}
\tag{2}
$$

where v_t^s, v_t^e are the voting scores for our predictions, N is the number of windows sliding over location t for which we are voting, r_{jn}^s is the j^{th} relative distance in the n^{th} window, and similarly for r_{jn}^e, with t_n being the corresponding location of frame t in window n. The higher the voting score at location t, the more likely t is to be a boundary location. The sequences of voting scores for starting and ending are denoted as $V^s = \{v_t^s\}_{t=1}^T$ and $V^e = \{v_t^e\}_{t=1}^T$.

From V^s, V^e, we can generate action proposals, explained next in Section 3.3.

3.3 Proposal Generation and Post-processing

Now that we have introduced our main contribution, the Voting Evidence Module, we explain how it is used within our proposal generation pipeline to produce $\hat{\Phi}$. This requires combining our predicted start/end boundaries to form proposals as well as score and classify these proposals. We describe this next.

Proposal Generation. Given the voting scores V^s and V^e, we consider all start/end times above a predefined threshold, ξ, and then consider local maxima as candidate start/end times. We form proposals from every valid start and end combination, i.e. start occurs before end and below a maximum action duration τ. We assign these proposals confidence scores and action classes as follows.

Proposal Confidence Scores. We begin by processing the feature sequence F with a learned Temporal Evaluation Module (TEM), as used in (Lin et al., 2018; Lin et al., 2019). We make slight modifications to the architecture, as shown in Figure 4, by using two 1D-convolutional branches, which performed better experimentally. TEM outputs naive boundary starting (B^s) and ending (B^e) scores, as well as actionness scores (B^a). These scores use only local information, attaching the same importance to each frame. We use the actionness score B^a as background suppression, using element-wise multiplication with the feature sequence. This gives a filtered feature sequence which is the input to VEM. Additionally, we calculate confidence in the proposal directly from its feature sequence $p(\hat{s},\hat{e})$, using the Proposal Evaluation Module from (Lin et al., 2019).

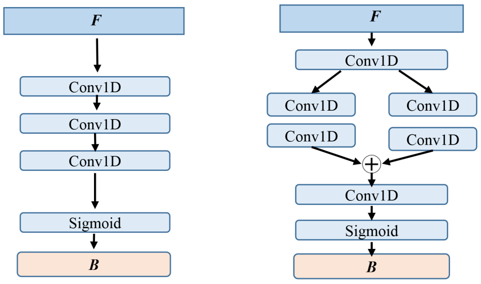

Figure 4: Temporal Evaluation Module from (Lin et al., 2018; Lin et al., 2019) (left) and our improved version (right). They take in a feature sequence F and output sequences of naive starting (B^s) and ending (B^e) boundary scores and actionness scores (B^a), denoted as B.

We fuse V^s, V^e, B^s, B^e and $p(\hat{s},\hat{e})$ to calculate confidence scores which are used to rank proposals. The confidence for a single proposal is:

$$
\hat{c} = (v_{\hat{s}}^s + \alpha b_{\hat{s}}^s)(v_{\hat{e}}^e + \alpha b_{\hat{e}}^e)p(\hat{s},\hat{e}) \tag{3}
$$

where α is a fusion weight, $b_{\hat{s}}^s \in B^s$ is the starting boundary score for $\hat{s}$, and $b_{\hat{e}}^e \in B^e$ is the ending boundary score for $\hat{e}$ from TEM.

Redundant Proposal Suppression. Like other action localization works (Lin et al., 2018; Lin et al., 2019; Su et al., 2021), we use Soft-NMS (Bodla et al., 2017) to suppress redundant proposals using the scores $\hat{c}$.

Proposal-to-Proposal Relations. We also explore relations between proposals as proposed by PGCN (Zeng et al., 2019), through constructing an action proposal graph. We evaluate with/without PGCN, for direct comparison to published works.

Classification. We classify each candidate proposal to obtain the class label $\hat{a}$ and obtain the final prediction set $\hat{\Phi} = \{(\hat{s},\ \hat{e},\ \hat{c},\ \hat{a})\}_{m=1}^{M}$. Note that action classification is performed after proposal generation, making our proposal generation action-independent.

4 EXPERIMENTS

4.1 Experimental Setup

Datasets. We conduct experiments on two temporal action localization datasets: ActivityNet1.3 (Caba Heilbron et al., 2015) and THUMOS14 (Jiang et al., 2014) as in (Lin et al., 2018; Lin et al., 2019; Zeng et al., 2019). ActivityNet-1.3 consists of 19,994 videos with 200 classes. THUMOS14 contains 413 untrimmed videos with 20 classes for the action localization task.

Table 1: Action localization results on ActivityNet-1.3 and THUMOS14. **Bold** for best model and underline for second best.

Method	Publication	ActivityNet-1.3 (mAP@IoU)				THUMOS14 (mAP@IoU)				
		0.5	0.75	0.95	Average	0.3	0.4	0.5	0.6	0.7
SSN (Zhao et al., 2017)	ICCV2017	43.26	28.70	5.63	28.28	51.9	41.0	29.8	-	-
TAL-Net (Chao et al., 2018)	CVPR 2018	38.23	18.30	1.30	20.22	53.2	48.5	42.8	33.8	20.8
BSN (Lin et al., 2018)	ECCV 2018	46.45	29.96	8.02	30.03	53.5	45.0	36.9	28.4	20.0
BMN (Lin et al., 2019)	ICCV 2019	50.07	34.78	8.29	33.85	56.0	47.4	38.8	29.7	20.5
MGG (Liu et al., 2019)	CVPR 2019	-	-	-	-	53.9	46.8	37.4	29.5	21.3
GTAN (Long et al., 2019)	CVPR 2019	**52.61**	34.14	8.91	34.31	57.8	47.2	38.8	-	-
G-TAD (Xu et al., 2020)	CVPR 2020	50.36	34.60	9.02	34.09	54.5	47.6	40.2	30.8	23.4
BC-GNN (Bai et al., 2020)	ECCV 2020	50.56	34.75	9.37	34.26	57.1	49.1	40.4	31.2	23.1
BSN++ (Su et al., 2021)	AAAI 2021	51.27	**35.70**	8.33	**34.88**	59.9	49.5	41.3	31.9	22.8
TVNet	-	51.35	34.96	**10.12**	34.60	**64.7**	**58.0**	**49.3**	**38.2**	**26.4**

Comparative Analysis. We compare our work to all seminal efforts that evaluate on these two standard benchmarks (Zhao et al., 2017; Chao et al., 2018; Lin et al., 2018; Lin et al., 2019; Liu et al., 2019; Long et al., 2019; Zeng et al., 2019; Xu et al., 2020; Bai et al., 2020; Chen et al., 2020; Su et al., 2021). As in previous efforts (Zeng et al., 2019; Chen et al., 2020; Xu et al., 2020), we perform an additional test when combining our work with the additional power of proposal-to-proposal relations from PGCN (Zeng et al., 2019) and the temporal aggregation from MUSES (Liu et al., 2021).

Evaluation Metrics. We use mean Average Precision (mAP) to evaluate the performance of action localization. To compare to other works, we report the same IoU thresholds. On ActivityNet-1.3 these are $\{0.5, 0.75, 0.95\}$, and on THUMOS14 they are $\{0.3, .., 0.7\}$, as well as the average mAP of the IoU thresholds from 0.5 to 0.95 at step size of 0.05.

Implementation Details. For feature extraction, we adopt the two-stream structure (Simonyan and Zisserman, 2014) as the visual encoder following previous works (Lin et al., 2018; Paul et al., 2018; Lin et al., 2019; Su et al., 2021). We parse the videos every 16 frames to extract features as in (Lin et al., 2018; Paul et al., 2018; Lin et al., 2019). To unify the various video lengths as input, following previous work (Lin et al., 2018; Paul et al., 2018), we sample to obtain a fixed input length, which is $T = 100$ for ActivityNet-1.3 and $T = 750$ for THUMOS14. The sliding window length is set to $J = 15+5$ for ActivityNet-1.3 and $J = 10+5$ for THUMOS14. We first train the TEM, then the PEM following the process in (Lin et al., 2018). We then train the Voting Evidence Module using the Adam optimizer (Kingma and Ba, 2014) with a learning rate 0.001 for the first 10 epochs and 0.0001 for the remaining 5 epochs for THUMOS14, and 0.0001 and 0.00001 for ActivityNet-1.3. The batch size is set to 512 and 256 for ActivityNet-1.3 and THUMOS14.

For proposal generation, we set the threshold $\xi = 0.3$, the maximum action length $\tau = 100$ for ActivityNet-1.3, and $\tau = 70$ for THUMOS14. To ensure a fair comparison for classifying our proposals, we use the same classifier as previous works (Lin et al., 2018; Lin et al., 2019; Xu et al., 2020). We use the top video classification from (Xiong et al., 2016) for ActivityNet-1.3. On THUMOS14, we assign the top-2 video classes predicted by (Wang et al., 2017) to all the proposals in that video. For Soft-NMS, we select the top 200 and 400 proposals for ActivityNet-1.3 and THUMOS14, respectively.

4.2 Results

Main Results. We compare TVNet with the state-of-the-art methods in Table 1. TVNet achieves comparable performance on ActivityNet-1.3, with a mAP of 10.12% at an IoU of 0.95, which outperforms all previous methods and shows that our temporal voting can distinguish the boundaries more precisely. On THUMOS14, we exceed all other methods across the range of IoUs commonly reported, for example reaching 49.3% mAP at 0.5.

Similar to other works, such as (Xu et al., 2020; Chen et al., 2020), in Table 2 we provide results on THUMOS14 for the PGCN (Zeng et al., 2019) proposal evaluation scheme applied to our proposals. We also provide results combining our proposals with the recently-introduced MUSES evaluation scheme (Liu et al., 2021). We report strong performance outperforming all prior works.

Qualitative Results. We also present qualitative results of TVNet on on both datasets. Figure 5 shows challenging examples on long THUMOS14 videos contain dense (top) and sparse (bottom) short actions. Figure 6 shows two examples from ActivityNet-1.3. A success case is shown, where long actions are de-

Table 2: Action localization results on THUMOS14 for methods combined with proposal-to-proposal relations from PGCN (Zeng et al., 2019) and MUSES (Liu et al., 2021). **Bold** for best and underline for second best.

Method	Publication	THUMOS14 (mAP@IoU)				
		0.3	0.4	0.5	0.6	0.7
BSN + PGCN (Zeng et al., 2019)	ICCV 2019	63.6	57.8	49.1	-	-
Uty + PGCN (Chen et al., 2020)	BMVC 2020	66.3	59.8	50.4	37.5	23.5
G-TAD + PGCN(Xu et al., 2020)	CVPR 2020	66.4	60.4	51.6	37.6	22.9
TVNet + PGCN	-	**68.3**	**63.7**	**56.0**	**39.9**	**24.2**
BSN + MUSES (Liu et al., 2021)	CVPR 2021	68.9	64.0	56.9	46.3	31.0
TVNet + MUSES (Liu et al., 2021)	-	**71.1**	**66.4**	**59.1**	**47.8**	**32.1**

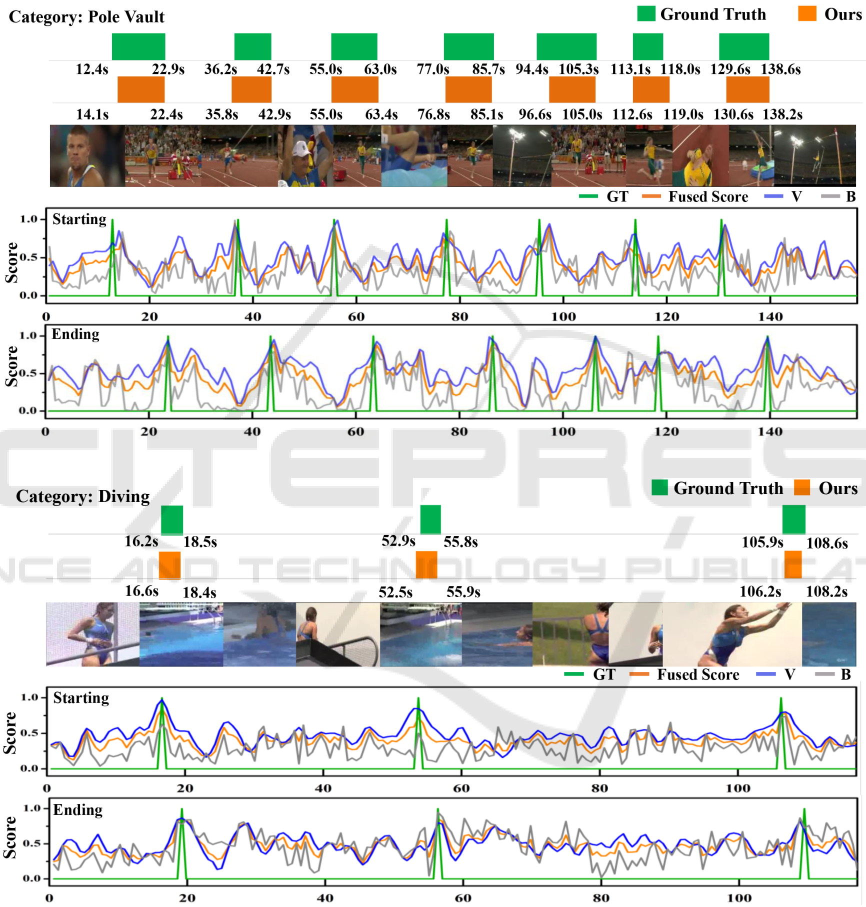

Figure 5: Qualitative results on THUMOS14, where TVNet detects multiple dense (top) and sparse (bottom) actions with accurate boundaries. The green bars indicate ground truth instances and the orange bars indicate TVNet detections. The green, orange, blue and grey lines are ground truth boundaries, weighted boundaries scores, voting scores and boundary scores respectively.

tected with accurate boundaries. A failure case is also shown, where two actions are mistaken as one, but the overall start and end times are accurate. In all examples, note how the orange boundary curves do not look identical to the peaks in the ground truth. This is desired behaviour, as they are designed to be probabilities found by voting, and their maxima are used to form action proposals which are then evaluated for fused confidence scores.

5 ABLATION STUDY

To investigate the behaviour of our model, we conduct several ablation studies on ActivityNet-1.3 and

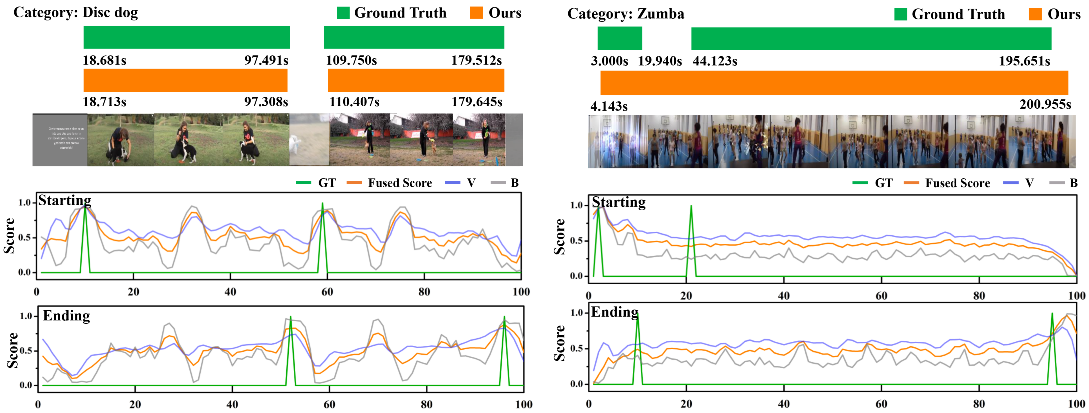

Figure 6: Qualitative TVNet results on ActivityNet. Left: a success case, where the actions are detected with boundaries closely matching the ground truth. Right: a failure case, where the wrong start/end times are matched forming one long action.

Table 3: The effect of actionness scores (B^a) and boundary scores (B^s,B^e) on ActivityNet-1.3. The top row indicates neither, which equates to the model with the TEM removed.

B^a	B^s/B^e	mAP@IoU			
		0.5	0.75	0.95	Average
✗	✗	50.45	34.26	7.97	33.55
✓	✗	51.30	34.89	8.90	34.40
✗	✓	51.14	34.74	9.68	34.35
✓	✓	**51.35**	**34.96**	**10.12**	**34.60**

THUMOS14 (as in (Lin et al., 2019; Su et al., 2021)).

Effectiveness of TEM. We test two contributions of the TEM. These are the actionness score B^a and the boundary scores B^s and B^e. Table 3 shows small improvements when using either, and best results when using both. All parts demonstrate an improvement over using neither, which can be considered as the full model without the TEM.

Voting and boundary scores. We fuse the voting scores and the boundary scores to calculate the final boundaries score for each proposal, which is used for ranking. Table 4 evaluates the importance of these. Just using the voting score outperforms the boundary score, but the combination of the two is best, suggesting they learn complementary information. Figure 7 shows how performance can be improved by combining these with a suitable weighting. We tried different values from 0 to 1 at step size 0.1 for α in Equation 3, the best is $\alpha = 0.6$.

Sliding Window Length. A larger sliding window length in the VEM could lead to more context information being used for voting at the expense of some missed boundary locations, especially for small action instances. Table 6 and Table 6 shows that TVNet results are actually relatively stable with respect to sliding window lengths between 5 and 20, with best results at 15+5 for ActivityNet-1.3 and 5+10 for THUMOS14.

Table 4: The effect of different combinations of boundary score (B), voting scores (V) and proposal generation (G) based on V on ActivityNet-1.3. All results are from our implementation, apart from *, which denotes the original B from (Lin et al., 2019), and can be considered as TVNet without the VEM.

B	G	V	mAP@IoU			
			0.5	0.75	0.95	Average
✓*	✗	✗	50.13	33.18	9.50	33.15
✓	✗	✗	50.64	34.30	8.93	33.84
✓	✓	✗	50.68	34.17	9.73	33.87
✗	✓	✓	51.30	34.89	8.90	34.40
✓	✓	✓	**51.35**	**34.96**	**10.12**	**34.60**

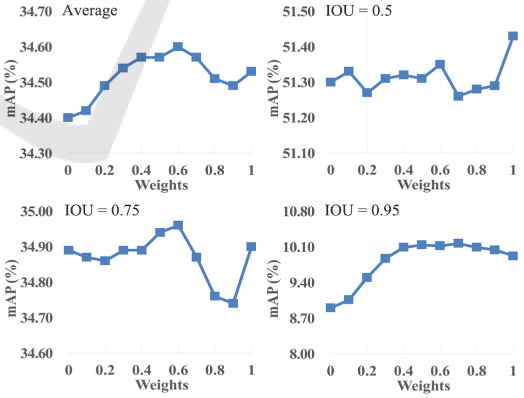

Figure 7: Performance of different weights used to fusion on ActivityNet1.3.

Maximum Action Length. This hyperparameter determines the maximum length of candidate proposals. Table 8 shows results on ActivityNet-1.3, where all video sequences are scaled to have length less than 100, following standard practice (Lin et al., 2018; Lin et al., 2019; Xu et al., 2020; Su et al., 2021). Table

Table 5: The effect of different sliding window lengths *J* on ActivityNet-1.3.

J	mAP@IoU			
	0.5	0.75	0.95	Average
5	50.84	34.57	9.47	34.08
10	50.97	34.61	9.84	34.24
15	**51.11**	**34.63**	10.00	**34.36**
20	50.97	34.57	**10.03**	34.29
15 + 5	**51.35**	**34.96**	10.12	**34.60**
15 + 10	51.27	34.87	9.94	34.52
15 + 20	51.01	34.65	**10.13**	34.42

Table 6: The effect of different sliding window lengths *J* on THUMOS14.

J	mAP@IoU				
	0.3	0.4	0.5	0.6	0.7
5	**64.6**	**57.9**	**48.6**	**38.0**	**25.6**
10	63.9	56.4	47.9	36.2	25.5
15	63.6	56.3	47.3	35.4	24.5
20	61.9	54.8	46.2	34.8	24.0
5 + 10	**64.7**	**58.0**	**49.3**	**38.2**	**26.4**
5 + 15	63.8	56.9	48.5	37.3	25.7
5 + 20	64.0	57.0	47.8	36.3	25.0

Table 7: The effect of different maximum action lengths τ on ActivityNet-1.3.

τ	mAP@IoU			
	0.5	0.75	0.95	Average
60	35.88	9.60	0.76	14.44
70	42.92	14.29	1.02	20.21
80	47.55	27.67	1.70	26.62
90	50.42	33.50	3.12	32.10
100	**51.35**	**34.96**	**10.12**	**34.60**

Table 8: The effect of different maximum action lengths τ on THUMOS14.

τ	mAP@IoU				
	0.3	0.4	0.5	0.6	0.7
40	53.2	44.2	32.9	22.0	13.7
50	57.8	50.4	40.5	29.0	18.8
60	60.7	53.6	42.8	33.5	23.7
70	**64.7**	**58.0**	**49.3**	**38.2**	**26.4**
80	64.3	57.7	48.3	37.1	25.9

Table 9: The effect of different score threshold ξ on ActivityNet-1.3.

ξ	mAP@IoU			
	0.5	0.75	0.95	Average
0.1	51.34	34.94	10.09	**34.60**
0.3	**51.35**	**34.96**	**10.12**	**34.60**
0.5	51.08	34.73	9.49	34.27
0.7	50.72	34.27	9.54	33.96

Table 10: The effect of different score threshold ξ on THUMOS14.

ξ	mAP@IoU				
	0.3	0.4	0.5	0.6	0.7
0.1	**64.8**	57.6	48.4	36.9	25.5
0.3	64.7	**58.0**	**49.3**	**38.2**	**26.4**
0.5	64.5	57.4	48.5	37.0	25.6
0.7	63.7	56.9	47.7	36.6	25.9

Table 11: Ablation studies of the impact of LSTM on ActivityNet-1.3. SRF: Small Receptive Field. SLL: Single Linear Layer.

Layer	mAP@IoU			
	0.5	0.75	0.95	Average
SRF	47.59	31.20	3.32	29.93
SLL	50.29	34.38	9.56	33.97
LSTM	**51.35**	**34.96**	**10.12**	**34.60**

8 shows results on THUMOS14, where the average video sequence is longer and most action instances are short. The best maximum action length on this dataset ($\tau = 70$) is very similar to others ($\tau = 64$ for (Lin et al., 2019; Xu et al., 2020; Su et al., 2021)). We stop at $\tau = 80$ due to GPU memory constraints.

Score Threshold. Table 10 and Table 10 show the effect of varying the threshold ξ, which is used to reject potential start and end points if they have low confidence before forming proposals. We use 0.3 for the main experiments, but results are stable for $0.1 < \xi < 0.5$.

Effectiveness of LSTM. To demonstrate the effect of the LSTM when accumulating evidence, Table 11 shows results when the LSTM is replaced with different architectures. First is a method with a receptive field of 1 (i.e. no temporal accumulation). Second is a single linear layer with the same receptive field as the LSTM. Finally, results from the full LSTM are shown, which performs best across all IoUs. Note that the small receptive field performs very badly on the high IoU.

6 CONCLUSION

This paper introduced a Temporal Voting Network (TVNet) for temporal action localization. TVNet incorporates a novel Voting Evidence Module, which allows each frame to contribute to boundary localization through voting, whether or not it is a boundary itself. On ActivityNet-1.3, TVNet achieves significantly better performance than other state-of-the-art methods at IoU of 0.95, highlighting its ability to accurately locate starting and ending boundaries. On THUMOS14, TVNet outperforms other methods when combining its proposals with the powerful PGCN proposal-to-proposal relations and MUSES. We provide qualitative examples, as well as a detailed ablation, which showcases the benefits of our voting-based approach.

ACKNOWLEDGEMENTS

We use publicly-available datasets. Hanyuan Wang is funded by the China Scholarship Council. Toby Perrett is funded by the SPHERE Next Steps EPSRC Project EP/R005273/1.

REFERENCES

Bai, Y., Wang, Y., Tong, Y., Yang, Y., Liu, Q., and Liu, J. (2020). Boundary content graph neural network for temporal action proposal generation. In *European Conference on Computer Vision*.

Bodla, N., Singh, B., Chellappa, R., and Davis, L. S. (2017). Soft-nms improving object detection with one line of code. In *International Conference on Computer Vision*.

Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J. (2015). ActivityNet: A large-scale video benchmark for human activity understanding. In *Computer Vision and Pattern Recognition*.

Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the Kinetics dataset. In *Computer Vision and Pattern Recognition*.

Chao, Y.-W., Vijayanarasimhan, S., Seybold, B., Ross, D. A., Deng, J., and Sukthankar, R. (2018). Rethinking the faster r-cnn architecture for temporal action localization. In *Computer Vision and Pattern Recognition*.

Chen, Y., Chen, M., Wu, R., Zhu, J., Zhu, Z., Gu, Q., and Robotics, H. (2020). Refinement of boundary regression using uncertainty in temporal action localization. In *British Machine Vision Conference*.

Cioppa, A., Deliège, A., Giancola, S., Ghanem, B., Droogenbroeck, M. V., Gade, R., and Moeslund, T. B. (2020). A context-aware loss function for action spotting in soccer videos. In *Computer Vision and Pattern Recognition*.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). SlowFast networks for video recognition. In *International Conference on Computer Vision*.

Heidarivincheh, F., Mirmehdi, M., and Damen, D. (2018). Action completion: A temporal model for moment detection. In *British Machine Vision Conference*.

Heidarivincheh, F., Mirmehdi, M., and Damen, D. (2019). Weakly-supervised completion moment detection using temporal attention. In *International Conference on Computer Vision Workshop*.

Jiang, Y., Liu, J., Zamir, A. R., Toderici, G., Laptev, I., Shah, M., and Sukthankar, R. (2014). THUMOS challenge: Action recognition with a large number of classes. In *European Conference on Computer Vision Workshop*.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Lin, C., Li, J., Wang, Y., Tai, Y., Luo, D., Cui, Z., Wang, C., Li, J., Huang, F., and Ji, R. (2020). Fast learning of temporal action proposal via dense boundary generator. In *AAAI Conference on Artificial Intelligence*.

Lin, T., Liu, X., Li, X., Ding, E., and Wen, S. (2019). BMN: Boundary-matching network for temporal action proposal generation. In *International Conference on Computer Vision*.

Lin, T., Zhao, X., Su, H., Wang, C., and Yang, M. (2018). BSN: Boundary sensitive network for temporal action proposal generation. In *European Conference on Computer Vision*.

Liu, X., Hu, Y., Bai, S., Ding, F., Bai, X., and Torr, P. H. (2021). Multi-shot temporal event localization: a benchmark. In *Computer Vision and Pattern Recognition (CVPR)*.

Liu, Y., Ma, L., Zhang, Y., Liu, W., and Chang, S.-F. (2019). Multi-granularity generator for temporal action proposal. In *Computer Vision and Pattern Recognition*.

Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., and Mei, T. (2019). Gaussian temporal awareness networks for action localization. In *Computer Vision and Pattern Recognition*.

Paul, S., Roy, S., and Roy-Chowdhury, A. K. (2018). W-TALC: Weakly-supervised temporal activity localization and classification. In *European Conference on Computer Vision*.

Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Neural Information Processing Systems*.

Su, H., Gan, W., Wu, W., Yan, J., and Qiao, Y. (2021). BSN++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. In *AAAI Conference on Artificial Intelligence*.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision*.

Wang, L., Xiong, Y., Lin, D., and Van Gool, L. (2017). UntrimmedNets for weakly supervised action recognition and detection. In *Computer Vision and Pattern Recognition*.

Xiong, Y., Wang, L., Wang, Z., Zhang, B., Song, H., Li, W., Lin, D., Qiao, Y., Gool, L. V., and Tang, X. (2016). CUHK & ETHZ & SIAT submission to activitynet challenge 2016.

Xu, M., Zhao, C., Rojas, D. S., Thabet, A., and Ghanem, B. (2020). G-TAD: Sub-graph localization for temporal action detection. In *Computer Vision and Pattern Recognition*.

Zacks, J., Braver, T., Sheridan, M., Donaldson, D., Snyder, A., Ollinger, J., Buckner, R., and Raichle, M. (2001). Human brain activity time-locked to perceptual event boundaries. *Nature Neuroscience*, 4:651–655.

Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., and Gan, C. (2019). Graph convolutional networks for temporal action localization. In *International Conference on Computer Vision*.

Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., and Lin, D. (2017). Temporal action detection with structured segment networks. In *International Conference on Computer Vision*.