Post-Processing Temporal Action Detection

Sauradip Nag[1,2] Xiatian Zhu[1,3] Yi-Zhe Song[1,2] Tao Xiang[1,2]

[1] CVSSP, University of Surrey, UK [2] iFlyTek-Surrey Joint Research Center on Artificial Intelligence, UK

[3] Surrey Institute for People-Centred Artificial Intelligence, UK

{s.nag,xiatian.zhu,y.song,t.xiang}@surrey.ac.uk

Abstract

Existing Temporal Action Detection (TAD) methods typically take a pre-processing step in converting an input varying-length video into a fixed-length snippet representation sequence, before temporal boundary estimation and action classification. This pre-processing step would temporally downsample *the video, reducing the inference resolution and hampering the detection performance in the original temporal resolution. In essence, this is due to a temporal quantization error introduced during resolution downsampling and recovery. This could negatively impact the TAD performance, but is largely ignored by existing methods. To address this problem, in this work we introduce a novel* model-agnostic *post-processing method without model redesign and retraining. Specifically, we model the start and end points of action instances with a Gaussian distribution for enabling temporal boundary inference at a* sub-snippet *level. We further introduce an efficient Taylor-expansion based approximation, dubbed as* Gaussian Approximated Post-processing *(***GAP***). Extensive experiments demonstrate that our GAP can consistently improve a wide variety of pre-trained off-the-shelf TAD models on the challenging ActivityNet (+0.2%∼0.7% in average mAP) and THUMOS (+0.2%∼0.5% in average mAP) benchmarks. Such performance gains are already significant and highly comparable to those achieved by novel model designs. Also, GAP can be integrated with model training for further performance gain. Importantly, GAP enables lower temporal resolutions for more efficient inference, facilitating low-resource application. The code is available at* https://github.com/sauradip/GAP

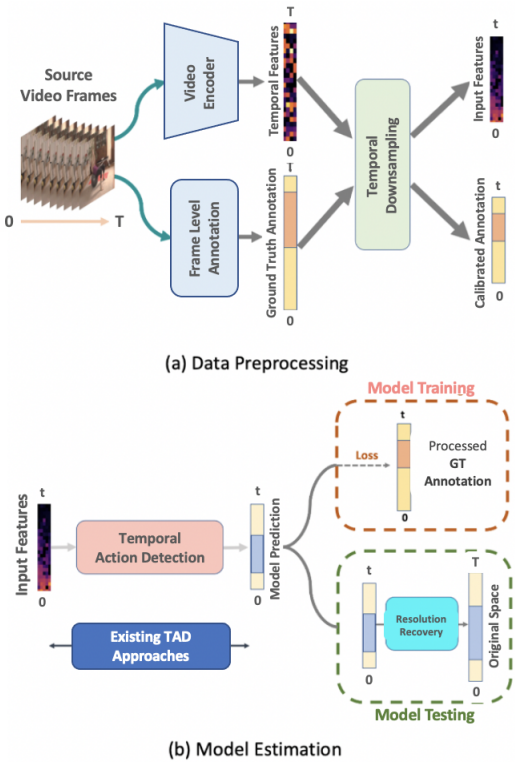

Figure 1. A typical pipeline for temporal action detection. (a) For efficiency and model design ease, temporal resolution reduction is often applied during pre-processing. This causes model inference at lower (coarse) temporal resolutions. (b) After bringing the prediction results back to the original temporal resolution during inference, quantization error will be introduced inevitably.

1. Introduction

The objective of Temporal action detection (TAD) is to identify both the temporal interval (*i.e.*, start and end points) and the class label of all action instances in an untrimmed video [3, 7]. Given a test video, existing TAD methods typically generate a set of action instance candidates via proposal generation based on regressing predefined anchor boxes [4, 6, 13, 23] or directly predicting the start and end times of proposals [2, 9, 10, 15, 25–27] and global segmentation masking [14]. To facilitate deep model design and improve computational efficiency, most TAD methods would pre-process a varying-length video into a fixed-length snippet sequence by first extracting frame-level visual features

Figure 2. Conventional *snippet-level* TAD inference along with our proposed *sub-snippet-level* post-processing.

with a frozen video encoders and subsequently sampling a smaller number of feature points (*i.e.*, snippet) evenly (see Fig. 1(a)). As a result, a TAD model performs the inference at *lower temporal resolutions*. This introduces a ***temporal quantization error*** that could hamper the model performance. For instance, when decreasing video temporal resolution from 400 to 25, the performance of BMN [9] degrades significantly from 34.0% to 28.1% in mAP on ActivityNet. Despite the obvious connection between the error and performance degradation, this problem is largely ignored by existing methods.

In this work, we investigate the *temporal quantization error* problem from a post-processing perspective. Specifically, we introduce a model-agnostic post-processing approach for improving the detection performance of existing off-the-shelf TAD models without model retraining. To maximize the applicability, we consider the TAD inference as a black-box process. Concretely, taking the predictions by any model, we formulate the start and end points of action instances with a Gaussian distribution in a *continuous* snippet temporal resolution. We account for the distribution information of temporal boundaries via Taylor-expansion based approximation. This enables TAD inference at *sub-snippet* precision (Fig. 2), creating the possibility of alleviating the temporal quantization error. We name our method as *Gaussian Approximated Post-processing* (**GAP**).

We summarize the ***contributions*** as follows. (I) We identify the previously neglected harming effect of temporal resolution reduction during the pre-processing step in temporal action detection. (II) For the first time, we investigate the resulting temporal quantization error problem from a model generic post-processing perspective. This is realized by modeling the action boundaries with a Gaussian distribution along with an efficient Taylor-expansion based approximation. (III) Extensive experiments show that a wide range of TAD models [2, 9, 10, 15, 25–27] can be seamlessly benefited from our proposed GAP method without algorithmic modification and model retraining, achieving the best single model accuracy on THUMOS and ActivityNet. Despite this simplicity, the performance improvement obtained from GAP can match those achieved by designing novel models [5]. At the cost of model retraining, our GAP can be integrated with existing TAD models for achieving further gain. Further, our GAP favorably enables lower temporal resolutions for higher inference efficiency with little performance degradation. Crucially, GAP can be applied generally in a variety of learning settings (*e.g.*, supervised, semi-supervised, zero-shot, few-shot).

2. Related Works

Temporal action detection Inspired by object detection in static images [17], R-C3D [23] uses anchor boxes by following the design of proposal generation and classification. With a similar model design, TURN [6] aggregates local features to represent snippet-level features for temporal boundary regression and classification. SSN [32] decomposes an action instance into three stages (starting, course, and ending) and employs structured temporal pyramid pooling to generate proposals. BSN [10] predicts the start, end and actionness at each temporal location and generates proposals with high start and end probabilities. The actionness was further improved in BMN [9] via additionally generating a boundary-matching confidence map for improved proposal generation. GTAN [13] improves the proposal feature pooling procedure with a learnable Gaussian kernel for weighted averaging. G-TAD [27] learns semantic and temporal context via graph convolutional networks for more accurate proposal generation. BSN++ [20] further extends BMN with a complementary boundary generator to capture rich context. CSA [19] enriches the proposal temporal context via attention transfer. VSGN [31] improves short-action localization using a cross-scale multi-level pyramidal architecture. Recently, Actionformer [29] and React [18] proposed a purely DETR based design for temporal action localization at multiple scales. Mostly, existing TAD models suffer from temporal quantization error as the actions are detected in the reduced temporal space. We present a model-agnostic post-processing strategy for generally tackling this problem without model redesign and retraining at a negligible cost.

Temporal boundary refinement methods can designed particularly for improving proposal localization. but still at the snippet level [8, 11, 16, 21, 28]. However, they still perform at the snippet level, and not solve the temporal quantization error problem as we focus on here. Specifically, PGCN [28] modeled the intra-action proposals using graph convolution networks to refine the boundaries. PBRNet [11] refined the anchor proposals using a two-stage refinement architecture with a complicated loss de-

sign. Recent focus has been shifted to anchor-free proposal refinement [8, 16, 21] where coarse action proposals are refined using local and global features to obtain fine-grained action proposals. However, the refinement modules are very design specific and cannot be easily adapted to any existing approaches. AFSD [8] used a pyramidal network to generate coarse action proposals and then refined them with boundary pooling based contrastive learning. Very recently, [14] developed a lightweight transformer based proposal-free model with boundary refinement. Often, large model size and complicated model/loss design are involved in each of these previous methods. In contrast, we take a completely different perspective (model-agnostic post-processing) and solve uniquely the temporal quantization error problem. Crucially, our method can be seamlessly integrated into prior temporal boundary refinement techniques *without* complex model redesign.

This work is inspired by [30] tackling human pose estimation in images, a totally different problem compared to more complex TAD we study here. Technically, we make non-trivial contributions by investigating both post-processing and model integration. Also, the temporal boundaries come with start /end pair form, rather than individual human joint keypoints. In the literature, human pose estimation and TAD are two independent research fields with sparse connections. However, at high level they could share generic challenges such as prediction post-processing as we study here. Importantly, post-processing is significantly understudied yet critical to TAD, as we reveal for the first time.

3. Method

We denote an untrimmed video as $X = \{x_n\}_{n=1}^{l_v}$ including a total of l_v frames. Ground-truth annotation of a training video X_i has M_i action instances $\Psi_i = \{(\psi_j, \xi_j, y_j)\}_{j=1}^{M_i}$ where ψ_j/ξ_j denote the start/end time, and y_j is the action category. During both training and inference, any video V is typically *pre-processed* into a unified representation format by first applying a pre-trained, frozen video encoder (*e.g.*, TSN [22]) and then sampling equidistant temporal points for a fixed number of (*e.g.*, 100) times. Each sampled point is called *a snippet* representing a short sequence of consecutive video frames. Obviously, this pre-processing is a ***temporal downsampling*** procedure, resulting in TAD at low temporal resolution as:

$$\mathbb{P} = \phi(F_v) = \{s_i, e_i\}_{i=1}^{N_p}, \tag{1}$$

where s_i and e_i are the start and end time of i-th predicted action instance, N_p specifies the number of action predictions per video, and F_v denotes the downsampled snippet feature. To generate the final temporal boundaries, the action predictions $\mathbb{P}$ need to be ***temporally upsampled*** linearly back to the original temporal resolution.

This temporal downsampling and upsampling process introduces temporal quantization errors negative to model performance. To address this problem, we propose a model-agnostic *Gaussian Approximated Post-processing* (GAP) method as detailed below.

3.1. Temporal Boundary Calibration

GAP aims to calibrate the start and end points of a given action boundary prediction. Our key idea is to explore the per-snippet score distribution structure of the predicted proposals P to infer the underlying maximum activation for both the start and end points. Specifically, we assume the predicted score distribution follows a univariate Gaussian distribution. This is conceptually similar with existing TAD methods [9, 27] using the overlap ratio over anchors against the annotated action intervals to create the ground-truth learning objective. Given a predicted boundary point at a discrete snippet temporal location $x \in [1, 2, \cdots, T]$ with T the total number of snippets per video, we formulate the temporal boundary distribution as:

$$P(x; \mu) = \frac{\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}{\sigma\sqrt{2\pi}}, \tag{2}$$

where μ refers to the underlying boundary point at sub-snippet resolution and σ refers to the standard deviation.

In order to reduce the approximation difficulty, we use logarithm to transform the original exponential form P to a quadratic form G to facilitate inference while keeping the original maximum activation location as:

$$G(x; \mu) = ln(P) = -\frac{1}{2}ln(2\pi) - ln(\sigma) - \frac{(x-\mu)^2}{2\sigma^2} \tag{3}$$

Our objective is to reason the value μ which refers to the underlying boundary point at sub-snippet resolution.

As an extreme point in a curve, it is known that the first derivative at the location μ meets the condition:

$$\mathcal{D}^{'}(x)\big|_{x=\mu} = (\tfrac{\partial G}{\partial x})\big|_{x=\mu} = -\tfrac{(x-\mu)}{\sigma^2}\Big|_{x=\mu} = 0 \tag{4}$$

To explore this condition, we adopt the Taylor's theorem. Formally, we approximate the activation $G(\mu)$ by a Taylor series up to the quadratic term, evaluated at *the maximal activation* x of the predicted snippet distribution as

$$G(\mu) = G(x) + \mathcal{D}^{'}(x)(\mu - x) + \frac{1}{2}(\mu - x)^2\mathcal{D}^{''}(x) \tag{5}$$

where $\mathcal{D}^{''}$ is the second derivative (*i.e.*, Hessian) of G evaluated at x, formally defined as:

$$\mathcal{D}^{''}(x) = \frac{\partial \mathcal{D}^{'}(x)}{\partial x} = -\frac{1}{\sigma^2} \tag{6}$$

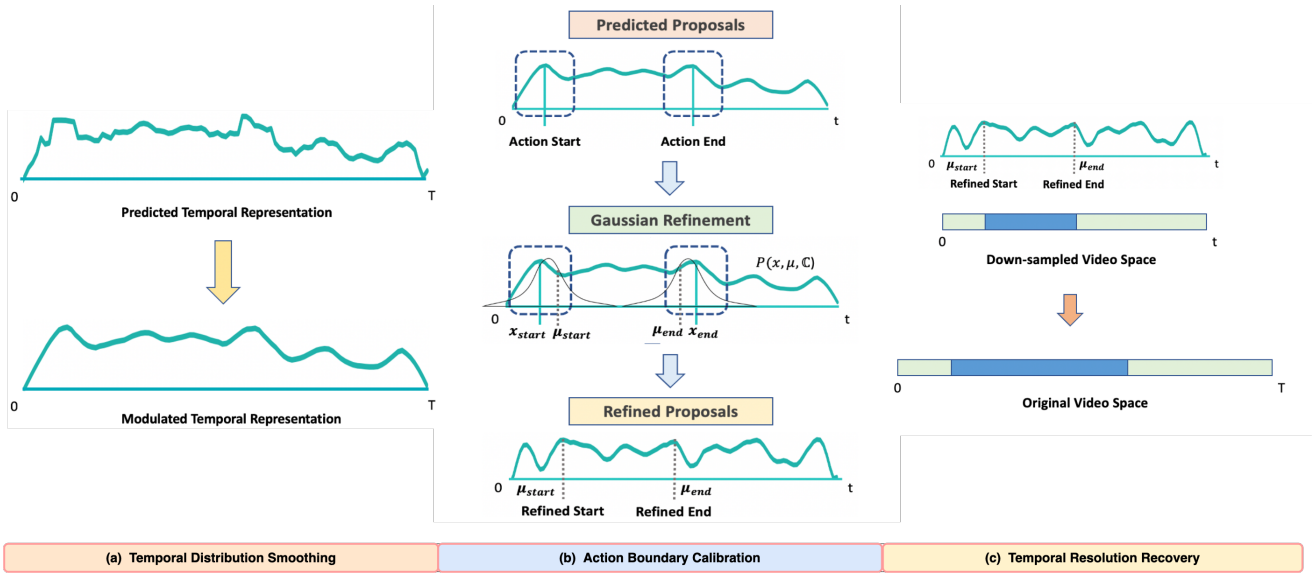

Figure 3. **Overview of the proposed *Gaussian Approximated Post-processing* (GAP) method.** Given a test video, an existing TAD model generates 1-D temporal score distribution of candidate foreground action instances. With our GAP, (a) we first regulate the temporal distribution (Eq (11)) by smoothing the score curve, followed by (b) detecting the boundary (*i.e.*, start/end points) and distributional refinement using a Gaussian kernel to obtain more accurate prediction at the sub-pixel precision. (c) We finally recover the original temporal resolution by multiplying the video-duration with the refined proposals.

The intuition is that, x is typically close to the underlying unseen optimal prediction so that the approximation could be more accurate. Combining Eq. (4), Eq. (5), Eq. (6) together, we obtain the refined prediction as:

$$\mu = x - \left((D^{''}(x)\right)^{-1} D^{'}(x), \tag{7}$$

where $D^{''}(x)$ and $D^{'}(x)$ can be estimated efficiently from the given score distribution. Finally, we use μ to predict the start and end points in the original video space.

Discussion Our GAP is efficient computationally as it only needs to compute the first and second derivative of predicted boundary points. Existing TAD approaches can be readily benefited without model redesign and retraining.

3.1.1 Temporal distribution smoothing

Often, the temporal boundary predicted by a TAD model does not follow good Gaussian shape. As shown in Fig. 4, the temporal prediction usually comes with multiple peaks. To avoid potential negative effect, we first smooth the temporal distribution h using a Gaussian kernel K with the same variation as: $h' = K * h$ where $*$ denotes the convolution operation. To keep the original magnitude, we further scale h' linearly as:

$$h' = \frac{h' - min(h')}{max(h') - min(h')} * max(h) \tag{8}$$

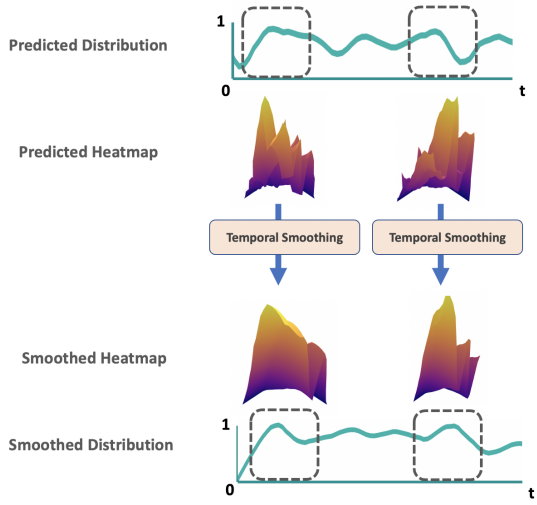

Figure 4. **Illustration of temporal distribution smoothing** operation along the conflicting action boundary snippets.

where $max()$ and $min()$ return the maximum and minimum value. We validate that this step is useful (Table 3), with the resulting visual effect demonstrated in Fig 4.

3.1.2 Summary

Our GAP can be generally integrated with existing boundary regression based TAD models without model redesign

and retraining (Fig 5(a)). At test time, we take as input the predicted snippet prediction predicted by any model such as BMN, and output more accurate start and end points per prediction in the original video space. The pipeline of using GAP is summarized in Fig. 3. Totally three steps are involved: (a) Temporal distribution smoothing (Eq. (11)); (b) Action boundary calibration by Taylor expansion at sub-snippet precision (Eq. (2)-(7)); (c) Temporal resolution recovery linearly to the original video length.

3.2. Integration with Existing Model Training

When model retraining is allowed, our GAP can also be integrated with existing TAD training without altering design nor adding learnable parameters (refer to Fig 5(b)). The only change is to applying GAP on the intermediate coarser predictions by prior methods (*e.g.*, AFSD [8] and RTDNet [21]). While retraining a model with predicted outputs could bring good margin, our post processing mode is more generally useful with little extra cost.

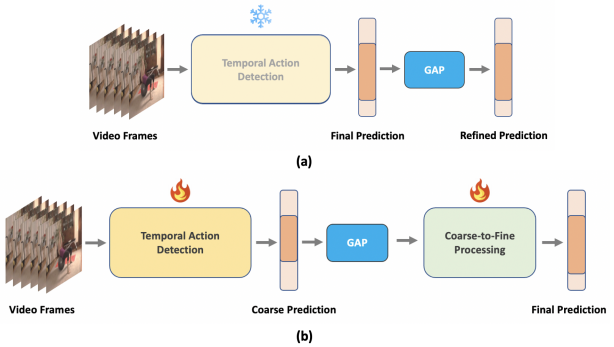

Figure 5. **Integrating GAP** during (a) post-processing the existing TAD predictions in inference, or (b) model training when applied on intermediate coarse predictions.

3.2.1 Ground-truth calibration

As model inference, the ground-truth for training is also affected by temporal resolution reduction. Specifically, during pre-processing by evenly sampling temporal points from the whole raw video length, the ground-truth start/end snippet locations need to be transformed accordingly. Formally, we denote the ground-truth of a video as $g = \Psi_g = \{(\hat{s}_j, \hat{e}_j, y_j)\}_{j=1}^{M_i}$ including the start and end annotations of each action instance. The temporal resolution reduction is defined as:

$$g' = (s', e') = \frac{g}{\lambda} = (\frac{\hat{s}}{\lambda}, \frac{\hat{e}}{\lambda}), \tag{9}$$

where λ is the downsampling parameter conditioned on the temporal sampling ratio and video duration. Conventionally, in the downsampling step, we often quantise g':

$$g'' = (s'', e'') = quantize(g') = quantize(\frac{\hat{s}}{\lambda}, \frac{\hat{e}}{\lambda}), \tag{10}$$

where $quantize()$ specifies a quantization function (*e.g.*, including floor, ceil and round). Noted that, g'' is a scalar term which represents an individual start/end point. Next, the start/end snippet distribution centred at the quantized location g'' can be synthesized via:

$$P(x; g'') = \frac{1}{\sigma\sqrt{2\pi}} exp\left(\frac{-(x-g'')^2}{2\sigma^2}\right), \tag{11}$$

where x denotes a point in the temporal distribution and σ denotes a fixed spatial variance. This is applied separately on both the ground-truth start and end points. Nonetheless, such start/end snippet distributions generated are clearly inaccurate due to the quantization error, as illustrated in Fig 6. This may cause sub-optimal supervision signals and degraded performance. To address this issue, we instead place the start/end centre at the *original non-quantized location* g' as it represents the accurate ground-truth location. Afterwards, we still apply Eq (11) with g'' replacing by g'. We will evaluate the effect of ground-truth calibration (Table 6).

Figure 6. **Illustration of quantization error** in the standard ground-truth (GT) generation: Obtaining the start/end points with floor based snippet quantization. As a result, an error (indicated by red marker) is introduced. Other quantization (*e.g.*, ceiling, rounding) share the same problem.

4. Experiments

Datasets We conduct extensive experiments on two major TAD benchmarks. (1) ActivityNet-v1.3 [3] provides 19,994 videos from 200 action classes. We adopt the standard setting to split all the videos into training, validation and testing subsets in a ratio of 2:1:1. (2) THUMOS14 [7] offers 200 validation videos and 213 testing videos from 20 action categories with labeled temporal boundary and class label.

Table 1. Evaluating the generic benefits of our GAP method on improving state-of-the-art TAD models on the ActivityNetv1.3 and THUMOS14 datasets. Empty results are due to the unavailability of open-source code.

Category	Method	ActivityNet mAP				THUMOS14 mAP			
		0.5	0.75	0.95	Avg	0.3	0.5	0.7	Avg
Anchor-based	MUSES [12]	50.0	34.9	6.5	34.0	68.9	56.9	31.0	53.4
	MUSES [12] + **GAP**	**50.3**	**35.5**	**6.9**	**34.3**	**69.3**	**57.8**	**31.9**	**53.8**
	PBRNet [11]	53.9	34.9	8.9	35.0	58.5	51.3	29.5	-
	PBRNet [11] + **GAP**	**54.4**	**35.4**	**9.2**	**35.2**	**59.2**	**51.9**	**30.0**	-
Anchor-Free	BMN [9]	50.1	34.8	8.3	33.9	56.0	38.8	20.5	38.5
	BMN [9] + **GAP**	**50.5**	**35.2**	**8.6**	**34.3**	**56.6**	**39.4**	**21.0**	**38.9**
	GTAD [24]	50.4	34.6	9.0	34.1	54.5	40.2	23.4	39.3
	GTAD [24] + **GAP**	**50.8**	**34.9**	**9.2**	**34.4**	**55.0**	**40.5**	**23.8**	**39.6**
	DCAN [5]	51.8	35.9	9.4	35.4	68.2	54.1	32.6	-
	DCAN [5] + **GAP**	**52.4**	**36.4**	**9.6**	**35.8**	**68.6**	**54.6**	**33.0**	-
	RTDNet [21]	47.2	30.7	8.6	30.8	68.3	51.9	23.7	-
	RTDNet [21] + **GAP**	**47.7**	**31.1**	**8.8**	**31.2**	**68.8**	**52.3**	**24.2**	-
	AFSD [8]	52.4	35.3	6.5	34.4	67.3	55.5	31.1	52.0
	AFSD [8] + **GAP**	**53.0**	**35.9**	**7.1**	**34.8**	**68.0**	**56.1**	**31.5**	**52.5**
	ActionFormer [8]	53.5	36.2	8.2	35.6	82.1	71.0	43.9	66.8
	ActionFormer [8] + **GAP**	**53.9**	**36.4**	**8.5**	**36.0**	**82.3**	**71.4**	**44.2**	**66.9**
	React [8]	-	-	-	-	69.2	57.1	35.6	55.0
	React [8] + **GAP**	**-**	**-**	**-**	**-**	**69.5**	**57.3**	**35.7**	**55.2**
Proposal-Free	TAGS [14]	56.3	36.8	9.6	36.5	68.6	57.0	31.8	52.8
	TAGS [14] + **GAP**	**56.7**	**37.2**	**9.8**	**36.7**	**69.1**	**57.4**	**32.0**	**53.0**

Implementation details We have adopted all the original training and inference details of existing TAD methods. For re-training AFSD [8] and RTDNet [21], we have used the reported hyperparameters in the respective papers. All the training has been performed on an Intel i9-7920X CPU with two Nvidia RTX 2080 Ti GPU. We used the same feature encoders as the original papers. During inference, all the full-resolution proposals are passed into SoftNMS for final output similar to [9].

4.1. Improving State-of-the-Art Methods

We evaluate the effect of our GAP on top TAD performers across all the anchor-based, anchor-free and proposal-free methods (MUSES [12], BMN [9], AFSD [8] and TAGS [14]) on ActivityNet and THUMOS dataset.

Results on ActivityNet From Table 1 we make the following observations: **(1)** The performance for anchor-based approaches [11, 12] is improved by at max 0.3% in avg mAP and by a constant gain of 0.3% to 0.5% in mAP@IOU 0.5. In particular, GAP can further improve over previous offset-based boundary refinement like PBRNet [11]. **(2)** When applying GAP on anchor-free approaches, the performance gain is in the range from 0.2% to 0.4%. Noticeably, AFSD [8] is benefited by an impressive improvement of 0.6% in mAP@IOU0.5 and 0.4% in avg mAP. This gain is already similar to that ($\sim$ 0.4%) of AFSD's complex learnable boundary refinement component. GAP is also effective for multi-scale DETR based approaches like ActionFormer [29] with similar margins achieved on ActivityNet. This gain is consistent with those for anchor-free based models. **(3)** With very different masking based architecture design in TAGS [14], GAP can still consistently yield an improvement of 0.2% in avg mAP. This further validates the model-agnostic advantage of our method.

Results on THUMOS14 Overall, similar conclusions can be drawn on THUMOS. All the models with our proposed GAP post-processing achieve the best results, often by a margin of 0.2$\sim$0.5% in avg mAP. There is a noticeable difference that the improvement by GAP is more significant than on ActivityNet, indicating the more severe quantization error on THUMOS due to longer videos.

Discussion We note that while TAD performance is saturating and a very challenging metric (average mAP over IoU thresholds from 0.5 to 0.95 for ActivityNet and from 0.3 to 0.7 for THUMOS) is applied, GAP can still push the performance at the comparable magnitude as recent state-of-the-art model innovation [5]. This is encouraging and meaningful, except for neglectable cost added and no model retraining. Additional results on other TAD settings is provided in the `Supplementary`.

4.2. Ablation Studies

(i) Input temporal resolution We examined the impact of snippet temporal resolution/size, considering that it is an

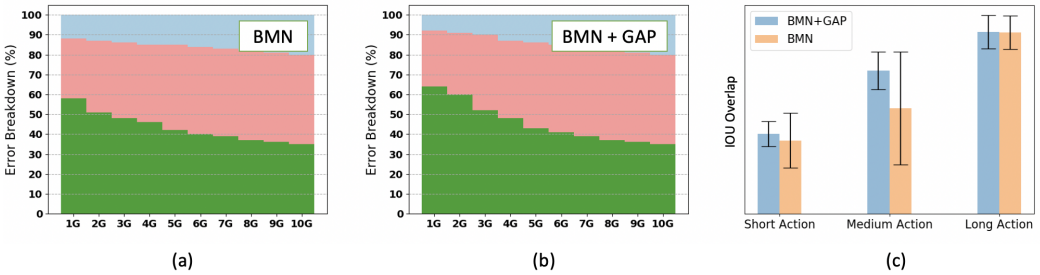

Figure 7. (a) False positive profile of BMN. (b) BMN with GAP on ActivityNet. (c) Proposal overlap analysis on various video lengths. We use top up-to $10G$ predictions per video, where G is the number of ground-truth action instances.

Table 2. Effect of temporal size on the ActivityNet using BMN [9] model.

Method	Temporal Resolution	mAP			
		0.5	0.75	0.95	Avg
BMN [9]	25	44.7	27.9	7.0	28.1
BMN+**GAP**	25	**45.5**	**28.4**	**7.3**	**28.5**
BMN	100	50.1	34.8	8.3	33.9
BMN+**GAP**	100	**50.5**	**35.2**	**8.6**	**34.3**
BMN	400	50.9	34.9	8.1	34.0
BMN+**GAP**	400	**51.1**	**35.0**	**8.2**	**34.1**

Table 3. Effect of temporal smoothing on ActivityNet

Method	Smoothing	mAP	
		0.5	Avg
BMN [9]	-	50.1	33.9
BMN +**GAP**	✗	50.3	34.0
BMN +**GAP**	✓	**50.5**	**34.3**

important efficiency factor. We used BMN [9] as the baseline TAD model in the standard training and testing setting. From Table 2 we have a couple of observations: (a) When reducing the input temporal resolution, as expected the model performance consistently degrades whilst the inference cost drops. (b) With the support of GAP, the model performance loss can be effectively mitigated, especially at very small input resolution. This facilitates the deployment of TAD models on low-resource devices as desired in emerging embedded AI.

(ii) Effect of temporal smoothing We evaluated the effect of temporal smoothing. From the results in Table 3, it can be observed that this step is useful and necessary otherwise the original prediction scores are less compatible with our GAP.

(iii) Error sensitivity analysis We compare our GAP (with BMN backbone) with original BMN [9] (anchor-free) via false positive analysis [1]. We sort the predictions by the scores and take the top-scoring predictions per video. Two major errors of TAD are considered: (1) *Localization error*, which is defined as when a proposal/mask is predicted as foreground, has a minimum tIoU of 0.1 but does not meet the tIoU threshold. (2) *Background error*, which happens when a proposal/mask is predicted as foreground but its tIoU with ground truth instance is smaller than 0.1. In this test, we use ActivityNet. We observe in Fig. 7(a,b) that GAP has the most true positive samples at every amount of predictions. The proportion of localization error with GAP is also notably smaller, which is the most critical metric for improving average mAP [1]. Also based on various video lengths [1], we estimated the standard deviation of all the proposal overlap with GT for both BMN and BMN with GAP variant. From Fig 7(c), it is interesting to note that our GAP indeed improves the overlap in challenging short and medium length videos and also BMN has a significant standard deviation in shorter-action instances. This explains the gain of GAP refinement over existing BMN which is caused due to the quantization error.

Table 4. Speed analysis of existing TAD method w/ our GAP on a NVIDIA RTX 2080 Ti GPU

Method	Inference Time	Speed
AFSD [8]	0.29 sec	1931 FPS
AFSD + **GAP**	0.31 sec	1792 FPS

(iv) Complexity We tested the inference efficiency impact by our method in AFSD at input size of 100 snippets for ActivityNet on a machine with one i9-7920X CPU and one RTX 2080 GTX GPU. From Table 4 it can be observed that the running speed is reduced from 1931 FPS to 1792 FPS in the low-efficient python environment, *i.e.*, a drop of 7.2%. There is a minor affordable increase from post-processing. Other programming language (*e.g.*, C/C++) based software can further reduce the overhead addition.

(v) Ground-truth calibration We tested the effect of our ground-truth calibration. We considered both cases with and without GAP post-processing. We observed from Table 5 that our ground-truth calibration brings positive performance margin consistently. In particular, it contributes

Table 5. Effect of ground-truth calibration on ActivityNet. *TAD model*: BMN [9] w/ GAP.

Ground-truth	Post-processing	mAP	
		0.5	Avg
W/O Calibration	W/O GAP	50.1	33.9
W/ Calibration	W/O GAP	50.2	34.0
W/O Calibration	W/ GAP	**50.5**	34.2
W/ Calibration	W/ GAP	**50.5**	**34.3**

Table 6. Effect of ground-truth quantization on ActivityNet. *TAD model*: BMN [9] w/ GAP.

Quantization Type	mAP			
	0.5	0.75	0.95	Avg
Ceiling	50.2	34.9	8.2	34.0
Rounding	**50.6**	35.1	8.4	34.2
Floor	50.5	**35.2**	**8.6**	**34.3**

Table 7. Results of integrating GAP in training and inference on ActivityNet.

Method	GAP		FLOPS	mAP	
	Train	Test		0.5	Avg
RTDNet [21]	✗	✗	85.7	47.2	30.8
	✓	✗		47.8	31.4
	✓	✓		**47.9**	**31.5**
AFSD [8]	✗	✗	157.1	52.4	34.4
	✓	✗		53.2	35.0
	✓	✓		**53.4**	**35.1**

consistently a gain of around 0.1% in avg mAP in both cases particularly for the stricter IOU metrics. This is reasonable since such fine-grained tuning matters most to more demanding metrics.

(vi) Quantization function We evaluated the quantization function in ground-truth calibration (Eq (10)). Common quantization options include `floor`, `ceiling` and `rounding`. From Table 6 we observed that `rounding` and `floor` are similarly effective, whilst `ceiling` gives the worst performance with a drop of 0.3% in avg mAP.

4.3. Integrating GAP with model training

Other than post-processing, our GAP can also be integrated into the training of existing TAD models. We experimented AFSD [8] and RTDNet [21] by applying GAP to their intermediate coarse start/end points during training. Table 7 shows that GAP can bring in more significant gains of 0.7%∼1.0% in IOU@0.5 mAP without adding extra parameters nor loss design complexity. This is also clearly reflected in the feature visualization as shown in Fig 8, where the previously ambiguous boundaries between action foreground and background can be well separated. This suggests more promising benefit of our GAP when model re-training is allowed. We also observed additional gain when integrating GAP during both training and post-processing, indicating flexible usage of our proposed GAP in existing TAD models.

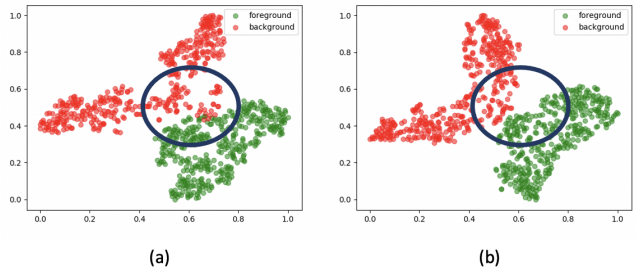

Figure 8. **T-SNE visualization** of the feature representation of a random ActivityNet val video (a) without and (b) with GAP assisted training. As seen from the encircled region that the original TAD model suffers from the ambiguous boundaries between action foreground and background. This can be well resolved once GAP is integrated during training.

5. Limitations

Although GAP enjoys the flexibility of being a plug-and-play module it comes with a few limitations. While being model agnostic and simple, it does not have high gain when the temporal resolution is large (Table 2), *e.g.*, greater than 400 snippets. This is because, at high temporal resolutions there is no much quantization error due to more duration per instance, and post-processing is hence less needed. Nonetheless, our GAP still gives a gain of 0.1% in avg mAP, which is a meaningful boost considering that the metric is very strict and the model performance is saturating. The snippet duration issue can only be solved if the snippet sampling procedure is automated based on the quantized error, which will be a good research direction for future research.

6. Conclusion

For the first time we systematically investigated largely ignored yet significant problem of *temporal quantization error* for temporal action detection in untrimmed videos. We not only revealed the genuine significance of this problem, but also presented a novel *Gaussian Aware Post-processing* (GAP) for more accurate model inference. Serving as a ready-to-use plug-in, existing state-of-the-art TAD models can be seamlessly benefited without any algorithmic adaptation at a neglectable cost. We validated the performance benefits of GAP over a wide variety of contemporary models on two challenging datasets. When model re-training is allowed, more significant performance gain can be achieved without complex model redesign and change.

References

[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *ECCV*, pages 256–272, 2018. 7

[2] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *CVPR*, 2017. 1, 2

[3] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 1, 5

[4] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, 2018. 1

[5] Guo Chen, Yin-Dong Zheng, Limin Wang, and Tong Lu. Dcan: improving temporal action detection via dual context aggregation. In *AAAI*, volume 36, pages 248–257, 2022. 2, 6

[6] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, 2017. 1, 2

[7] 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*, 155:1–23, 2017. 1, 5

[8] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *CVPR*, 2021. 2, 3, 5, 6, 7, 8

[9] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, pages 3889–3898, 2019. 1, 2, 3, 6, 7, 8

[10] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018. 1, 2

[11] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *AAAI*, volume 34, pages 11612–11619, 2020. 2, 6

[12] Xiaolong Liu, Yao Hu, Song Bai, Fei Ding, Xiang Bai, and Philip HS Torr. Multi-shot temporal event localization: a benchmark. In *CVPR*, pages 12596–12606, 2021. 6

[13] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *CVPR*, 2019. 1, 2

[14] Sauradip Nag, Xiatian Zhu, Yi-zhe Song, and Tao Xiang. Proposal-free temporal action detection via global segmentation mask learning. In *ECCV*, 2022. 1, 3, 6

[15] Sauradip Nag, Xiatian Zhu, and Tao Xiang. Few-shot temporal action localization with query adaptive transformer. *BMVC*, 2021. 1, 2

[16] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. In *CVPR*, pages 485–494, 2021. 2, 3

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, 2016. 2

[18] Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. React: Temporal action detection with relational queries. In *ECCV*, 2022. 2

[19] Deepak Sridhar, Niamul Quader, Srikanth Muralidharan, Yaoxin Li, Peng Dai, and Juwei Lu. Class semantics-based attention for action detection. In *ICCV*, pages 13739–13748, 2021. 2

[20] Haisheng Su, Weihao Gan, Wei Wu, Yu Qiao, and Junjie Yan. Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. *arXiv preprint arXiv:2009.07641*, 2020. 2

[21] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *ICCV*, 2021. 2, 3, 5, 6, 8

[22] 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*, pages 20–36. Springer, 2016. 3

[23] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017. 1, 2

[24] Mengmeng Xu, Juan-Manuel Pérez-Rúa, Victor Escorcia, Brais Martinez, Xiatian Zhu, Li Zhang, Bernard Ghanem, and Tao Xiang. Boundary-sensitive pre-training for temporal localization in videos. *arXiv*, 2020. 6

[25] Mengmeng Xu, Juan-Manuel Pérez-Rúa, Victor Escorcia, Brais Martinez, Xiatian Zhu, Li Zhang, Bernard Ghanem, and Tao Xiang. Boundary-sensitive pre-training for temporal localization in videos. In *ICCV*, pages 7220–7230, 2021. 1, 2

[26] Mengmeng Xu, Juan-Manuel Perez-Rua, Xiatian Zhu, Bernard Ghanem, and Brais Martinez. Low-fidelity end-to-end video encoder pre-training for temporal action localization. In *NeurIPS*, 2021. 1, 2

[27] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020. 1, 2, 3

[28] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, 2019. 2

[29] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *ECCV*, pages 492–510. Springer, 2022. 2, 6

[30] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *CVPR*, pages 7093–7102, 2020. 3

[31] Chen Zhao, Ali K Thabet, and Bernard Ghanem. Video self-stitching graph network for temporal action localization. In *ICCV*, pages 13658–13667, 2021. 2

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