Pehlivan, Selen; Laaksonen, Jorma

Temporal teacher with masked transformers for semi-supervised action proposal generation

Published in:
Machine Vision and Applications

DOI:
10.1007/s00138-024-01521-7

Published: 15/03/2024

Document Version
Publisher's PDF, also known as Version of record

Machine Vision and Applications (2024) 35:36
https://doi.org/10.1007/s00138-024-01521-7

RESEARCH

Temporal teacher with masked transformers for semi-supervised action proposal generation

Selen Pehlivan[1,2] · **Jorma Laaksonen**[2]

Received: 1 June 2023 / Revised: 16 September 2023 / Accepted: 15 February 2024

Abstract
By conditioning on unit-level predictions, anchor-free models for action proposal generation have displayed impressive capabilities, such as having a lightweight architecture. However, task performance depends significantly on the quality of data used in training, and most effective models have relied on human-annotated data. Semi-supervised learning, i.e., jointly training deep neural networks with a labeled dataset as well as an unlabeled dataset, has made significant progress recently. Existing works have either primarily focused on classification tasks, which may require less annotation effort, or considered anchor-based detection models. Inspired by recent advances in semi-supervised methods on anchor-free object detectors, we propose a teacher-student framework for a two-stage action detection pipeline, named Temporal Teacher with Masked Transformers (TTMT), to generate high-quality action proposals based on an anchor-free transformer model. Leveraging consistency learning as one self-training technique, the model jointly trains an anchor-free student model and a gradually progressing teacher counterpart in a mutually beneficial manner. As the core model, we design a Transformer-based anchor-free model to improve effectiveness for temporal evaluation. We integrate bi-directional masks and devise encoder-only Masked Transformers for sequences. Jointly training on boundary locations and various local snippet-based features, our model predicts via the proposed scoring function for generating proposal candidates. Experiments on the THUMOS14 and ActivityNet-1.3 benchmarks demonstrate the effectiveness of our model for temporal proposal generation task.

Keywords Temporal proposal generation · Semi-supervised learning · Anchor-free model · Transformer network

1 Introduction

High-quality temporal action proposals are crucial for a successful two-stage action localization pipeline on long-term video sequences. Deep learning models achieve remarkable performances in the temporal action proposal generation task using either boundary-based [1, 2] or proposal-based [3] approaches within a fully-supervised setting. Complementary characteristics of these two techniques motivate the introduction of joint models with improved performance [3–5]. Despite advances in deep learning architectures for temporal action proposals, the performance usually relies on human-annotated data as it scales up with growing labeled data. However, only relatively limited data is available in the video domain compared to that in image datasets.

Semi-supervised learning (SSL) algorithms aim to learn prediction functions jointly from labeled and unlabeled observations. Inspired by the advances in rapidly developed semi-supervised image classification models [6, 7], recent studies [8–11] show promising results on semi-supervised object-detection with limited labeled data compared to that of fully-supervised versions. To our knowledge, there are few recent approaches adapted and applied to semi-supervised action detection and proposal generation tasks on untrimmed videos. Available action models [12, 13] are designed on top of anchor-based models, where the former [12] investigates the SSL approach using the Boundary Sensitive Network (BSN) model [2] and the latter [13] applies the SSL on the Boundary Matching Network (BMN) model [4]. However, anchor-free models have been receiving more attention in the

Selen Pehlivan: Work done when Selen Pehlivan was at Aalto University.

✉ Selen Pehlivan
selen.pehlivantort@vtt.fi

Jorma Laaksonen
jorma.laaksonen@aalto.fi

[1] VTT Technical Research Centre of Finland, Oulu, Finland

[2] Department of Computer Science, Aalto University, Espoo, Finland

Published online: 15 March 2024

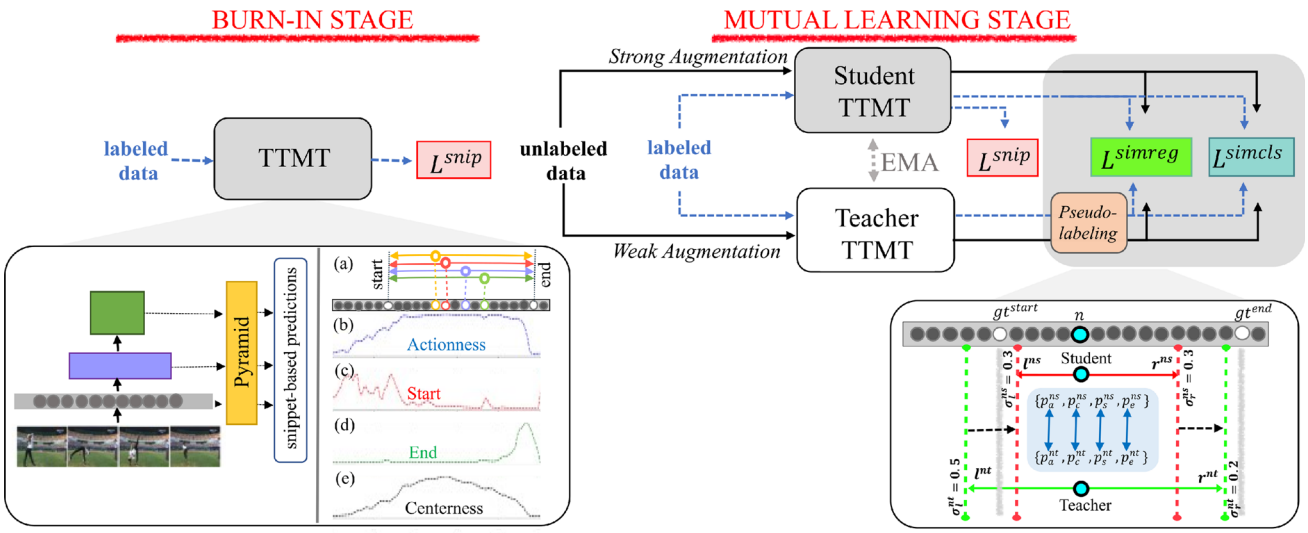


Fig. 1 Overview of Temporal Teacher with Masked Transformers (TTMT). The teacherstudent framework consists of two steps: Build-in Stage and Mutual Learning Stage. The backbone model is based on a multiscale transformer architecture.

fully-supervised setting with a few recent studies proposed for action detection [14, 15] with the promise of achieving competitive accuracy and computational efficiency. Therefore, targeting anchor-free semi-supervised action models looks like a future trend in the field as well.

Particularly, semi-supervised techniques are not well explored on anchor-free models for temporal action proposal generation. Focusing on a two-stage detection pipeline as in [12, 13], we aim to propose an anchor-free model within a semi-supervised training methodology. Following the teacher-student framework [16] as a semi-supervised technique, we introduce an anchor-free temporal proposal generation model to achieve comparable performance with that of fully-supervised anchor-based [3–5] and semi-supervised anchor-based models [13]. Recently, Unbiased Teacher v2 [11] evaluates a new pseudo-labeling semi-supervised approach on anchor-free object detectors with extensive analysis. Following the observations on object detectors from the study, we investigate the performance of our semi-supervised and anchor-free approach for action proposal generation. An SSL-based anchor-free model [17] has been recently introduced for temporal action detection, but direct comparison is not reasonable as this work is proposed in a one-stage detection pipeline. One-stage pipelines are often powered by a refinement stage, since they directly target the action detection task, e.g. [17]. Instead, our work neither requires an extra refinement stage nor uses an existing one, and it focuses mainly on a two-stage pipeline that places more emphasis on the task of creating a proposal. As one advantage, two-stage pipelines with good proposal candidates can be flexibly integrated for strengthening various downstream tasks in different granularities, e.g. action recognition on a coarse scale or human-object interaction detection on a fine scale. As another advantage, they are less dependent on action categories than one-stage pipelines and can be easily fine-tuned into new action categories, also with greater potential for class-incremental scenarios. In particular, we perform action recognition in the second stage of the pipeline as a downstream task by adding a simple pre-trained action classifier, i.e., UntrimmedNet [18], to rescore extracted proposals for classification as in [3–5]. With a two-stage pipeline, our proposal generation model can be integrated with proposal refinement techniques such as P-GCN [19] for action detection as was previously effective in [20].

Our semi-supervised approach, shown in Fig. 1, is a teacher-student framework that follows the training methodology from Liu et al. [21]. Our model observes a set of labeled videos and a set of unlabeled ones in a two-step training pipeline. The first step of the training mechanism, i.e., the burn-in phase, draws on labeled data during the first iterations of training using our anchor-free backbone model as the Student, while the second step, i.e., the mutual learning phase, makes use of both labeled and unlabeled videos using the teacher-student framework with the competing Teacher and Student built on the same anchor-free backbone for the rest of the training. Our teacher-student framework contains both snippet-based classification and regression objectives on supervised and unsupervised data with pseudo-labeling to support strong integration with the snippet-based anchor-free backbone predictions.

The backbone model, i.e., the Masked Transformer model, is based on the detection of multiple per-snippet-based local clues on video sequences via an encoder-only Transformer architecture designed in our recent study [22]. The traditional Transformer model [23] supports the detection of entities along with their pairwise relationships. Because it

explores local snippet-based features at multiple levels of detail, a multiscale Transformer can be a good video processing technique for anchor-free models. There exist recent models for multiscale image classification and our strategy is to extend one of these models, namely Improved Multiscale Vision Transformers (MViTv2) [24], for the video action proposal generation task. Our model improves the pooling attention [24] by using bi-directional masks to better model temporal ordering [25]. Using the proposed Masked Transformer model, we primarily aim to demonstrate that our anchor-free model can be integrated into a semi-supervised teacher-student framework with performance comparable to that of both fully- and semi-supervised anchor-based models [3–5, 12, 13]. Next, we aim to demonstrate how our teacher-student framework can be applied to temporal sequences by both snippet-based classification and regression through consistency regularization and pseudo-labeling, taking into account the localization uncertainty in boundary estimations. Mean teacher models are mostly examined for classification scenarios. Instead, our semi-supervised model is based on a teacher-student framework with multiple snippet-based classification and regression functions formulated specifically for our snippet-based anchor-free design.

We demonstrate that our end-to-end trainable anchor-free Transformer-based generator network, called Temporal Teacher with Masked Transformers (TTMT), achieves promising performance when compared to the state-of-the-art models on action proposal generation. We validate our model with experiments on the THUMOS14 [26] and ActivityNet-1.3 [27] datasets. Experiments reveal that our anchor-free Transformer-based model is a good candidate for video processing as it performs as well as the proposal-based models. The generated proposals are highly overlapping with the ground truth and have accurate boundary localization. The main contributions of our study are (i) a new teacher-student model with an encoder-only Transformer model for anchor-free temporal action proposal generation, (ii) a Masked Transformer model with a temporal extension of pooling attention unit [24] via bi-directional masks for temporal encoding, and (iii) an improved anchor-free model with uncertainty-aware boundary estimations.

2 Background

The target of our work is on semi-supervised two-stage action localization pipeline on untrimmed video sequences. Although the literature is dense on studies of action detection with robust one-stage and two-stage detection models, we here discuss the recent studies on the two-stage detection models. For a robust two-stage pipeline, high-quality proposal generation means better capturing the ground-truth segments with highly confident foreground action regions and accurate boundaries [2, 4, 28]. Most existing studies focus on fully-supervised models for action localization, while few recent ones aim for semi-supervised models.

2.1 Fully-supervised models

Existing proposal generation models in fully-supervised settings can be categorized as anchor-based and anchor-free approaches. Anchor-based approaches can be categorized as top-down and bottom-up approaches. While the former group relies on the sliding window or the Faster R-CNN [29] strategies to extract proposal-level regions as candidate segments [30], the latter is based on detecting boundary-level features for extracting candidates [2, 28]. Temporal Unit Regression Network (TURN) [31] generates proposals via decomposition into short units and employs regression to adjust boundaries from the sliding windows. Temporal Action Grouping (TAG) [28] connects high-scoring regions by a watershed algorithm. Boundary Sensitive Network (BSN) [2] detects local boundaries and evaluates proposal confidence scores within a region. On the other hand, Complementary Temporal Action Proposal (CTAP) [1] jointly uses sliding windows and grouping-based methods for high-quality proposals. Another approach, Snippet Relatedness-based Generator (SRG) [32], represents long-range dependencies among snippets by a score map.

Both proposal-level and boundary-level features are critical for obtaining high-quality proposals with precise boundaries [1, 5]. Complementary characteristics of these features are the key motivations for many joint models integrating proposal- and boundary-level features, e.g. BMN [4] and MGG [5]. One recent model, Boundary Content Graph Neural Network (BC-GNN) [33], uses a graph neural network for the interactions of boundaries and content of proposals. Another model, Relaxed Transformer Decoder (RTD-Net) [34], proposes a transformer-based architecture for temporal proposal generation inspired by a recent transformer object detection framework DETR [35].

In addition to anchor-based models, more recent studies focus on anchor-free approaches. Anchor-Free Saliency-based Detector (AFSD) [14] proposes a saliency-based refinement module that gathers boundary features, and ActionFormer [15] uses multiscale Transformers. Contrary to these studies aiming for single-stage action detection, we target anchor-free models for proposal generation within two-stage action detectors and we devise an SSL-based model.

2.2 Semi-supervised models

A powerful technique for training models on both labeled and unlabeled data is semi-supervised learning (SSL). A popular class of SSL methods produces artificial labels for

unlabeled data and trains a model to predict the artificial label when unlabeled data is inputted. The majority of the recent SSL methods typically consist of pseudo-labeling and consistency regularization approaches. Pseudo-labeling [36] uses the model itself to obtain predictions for unlabeled data. Besides, consistency regularization [37] leverages the idea of obtaining similar predictions when the models are fed with the perturbed data. Early approaches apply exponential moving average (EMA) of model parameters [16] or self-ensembling [38] when producing artificial labels.

SSL for image classification has been rapidly developed with promising results in recent years. Existing SSL image classification works [7, 39] apply input augmentations and consistency regularization on unlabeled images. Inspired by these works, several semi-supervised object detection works have been proposed to exploit similar ideas to train object detectors in a semi-supervised manner [8, 40]. Despite the significant improvement, there are still two remaining issues: (i) there are few studies on SSL-based proposal generation and action detection models, (ii) prior works are mainly focused on anchor-based models [12, 13]. Both models, [12, 13], adopt the Mean Teacher framework in the semi-supervised temporal action proposal task. We devise an alternative teacher-student framework based on our anchor-free masked Transformer network with a lightweight uncertainty-aware proposal refinement component.

One recent SSL-based study [17] introduces an anchor-free one-stage approach to action detection. The study integrates a two-stream model based on a standard Transformer backbone into a semi-supervised model via pseudo-labeling applied to both classification and mask predictions. Similarly, we offer a teacher-student framework, but unlike [17], our proposed SSL-based framework relies on a new anchor-free masked Transformer network and our framework integrates pseudo-labeling not only for the classification of various snippet-based features but also for boundary regression. Our framework leverages the relative uncertainties between the Teacher and Student to select the boundary-level pseudo-labels [11]. Moreover, a direct comparison is not reasonable since we are proposing a two-stage pipeline contrary to Nag et al. [17], which is a one-stage model.

3 Masked transformer pyramid model

Core models replicated under the proposed teacher-student framework are Transformer-based. The Transformers were first introduced for language modeling on text sequences [23] with its support on learning long-range dependencies via the self-attention mechanism. Following the success in NLP [41], attention mechanisms later became an integral part of many vision tasks, including image recognition, object detection, video understanding, text-image synthesis and visual question answering [42, 43]. In particular, we use a multiscale encoder-only Transformer network introduced in our previous study [22] designed for directional temporal dependency modeling on long-range video snippet sequences.

Based on the Masked Transformer network that reveals the local clues in multiple scales besides interactions among snippets, we aim to extract proposal candidates within a pyramid structure. In this section, we first describe the encoder-only Transformer architecture with a bi-directional multi-head attention unit and then give the details of pyramid architecture.

3.1 Multiscale encoder-only transformers

Our core model is based on a multiscale transformer architecture. For the multiscale purpose, we exploit the pooling attention units devised as the self-attention blocks by Multiscale Vision Transformers (MViTv2) [24]. In MViTv2, the pooling attention has been originally proposed as part of a Vision Transformer model for image classification, object detection and video recognition tasks. In this work, we integrate it to process 1D sequences of snippet embeddings extracted using a pre-trained CNN model and we leverage it for temporal proposal generation task.

Multiscale Transformer architecture comprises the concept of stages. Each stage consists of multiple transformer blocks with specific time resolution and channel dimension. Reducing the sequence length from input to output of the network stages, the architecture gradually expands the channel width via pooling attention units. For an input sequence, $F \in \mathbb{R}^{T \times D}$, a Transformer block packs it into query, key and value matrices, Q, K, V, with a pooling attention unit as

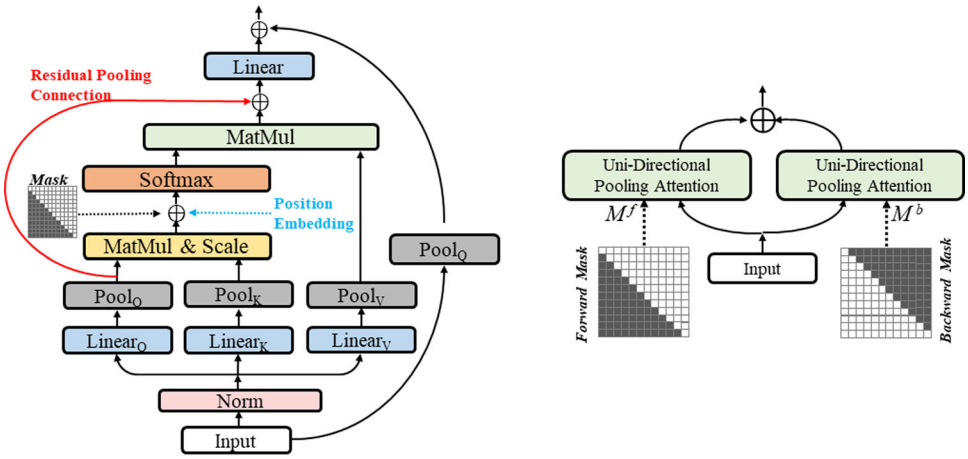
$$Q = P_Q(FW_Q), K = P_K(FW_K), V = P_V(FW_V), \quad (1)$$

where W_Q, W_K and $W_V \in \mathbb{R}^{D \times D}$. The pooling attention unit first projects input F using W_Q, W_K and W_V and then applies pooling operators (P) that are 1×3 convolution layers. The pooling operator can reduce the time resolution, i.e., the sequence length, using a convolutional stride.

Following the pooling operators, the standard (i.e., unmasked) version of the multi-head attention block is applied as

$$Z' = \frac{QK^\top}{\sqrt{D}} + E^r ,$$
$$Attn(Q, K, V) = softmax(Z')V , \quad (2)$$

where E^r is the relative position embedding along temporal axes. Later, we apply the residual pooling connection and add the pooled query tensor to the output sequence, $Z = Attn(Q, K, V) + Q$.



(a) Uni-directional Pooling Attention (b) Bi-directional Pooling Attention

Fig. 2 Bi-directional pooling attention unit proposed as one extension of the attention unit from [24] with integrated directional masks

3.2 Bi-directional multi-head attention

In this work, we integrate a directional mask into the pooling attention unit and introduce a bi-directional version of the attention unit to model temporal ordering in attention output [25]. Given a mask $M \in \mathbb{R}^{T\times T}$, we first apply dot product attention among Q and K with a scaling factor as in Eq. (2) and then add with the mask component as

$$Z'_{ij} = \sum_{d=1}^{D}(Q_{id}K_{dj})/\sqrt{D} + E^r_{ij} + M_{ij}, \tag{3}$$

where i and j are snippet indices. If $M_{ij} = -\infty$, then $Z'_{ij} = -\infty$. This implies the $Attn_{ij}$ turns into zero in Eq. (2), since softmax output results in 0.

For the bi-directional version, we use two masks—one for modeling forward ordering and the other for modeling backward ordering, M^f and M^b, respectively, as

$$M^f_{ij} = \begin{cases} 0 & i < j, \\ -\infty & \text{otherwise} \end{cases}, \tag{4}$$

$$M^b_{ij} = \begin{cases} 0 & i > j, \\ -\infty & \text{otherwise} \end{cases}. \tag{5}$$

We apply forward and backward masks as in Eq. (3) to compute Z'^f and Z'^b outputs respectively, and multiply by V. The final $Attn$ matrix is then merged with a simple addition operation as

$$Attn(Q, K, V) = softmax(Z'^f)V + softmax(Z'^b)V\ . \tag{6}$$

Figure 2 illustrates the details of the pooling attention unit with bi-directional mask extension. Note that the proposed attention model can be generalized to various other mask structures.

3.3 Transformer-based pyramid architecture

In the proposed multiscale transformer architecture, while the bottom stages perform fine-scale evaluation, the higher stages perform coarse-scale evaluation on video sequences. The architecture is converted into a simple pyramid structure with attachments of lateral connections. In this structure, the bottom-up pathway consists of multiple stages each having a various number of blocks. The last block of each stage doubles the channel width D_i while reducing the sequence length T_i by a factor of two using the bi-directional pooling attention unit (see Sect. 3.2). The last block output of a stage corresponds to a level sequence map. The top-down pathway integrates sequence maps iteratively via lateral connections to form a pyramid network [44]. In each iteration, a coarse-scale sequence map is upsampled by a factor of two using the nearest neighbor and added to the previous bottom-up map that is filtered using a 1×1 convolutional layer. The merged map is smoothed using a 1×3 convolutional filter into $P_i \in \mathbb{R}^{D_i\times T_i}$ (we fix the numbers of channels D_i to 1024 in this paper). This process continues until the finest resolution map is constructed.

All levels of the pyramid use shared network heads including classifiers and regressors as in a traditional image pyramid. Our network heads consist of (i) snippet-based prediction branches including {actionness, centerness, start-boundary, end-boundary}, (ii) boundary regression branch, and (iii) localization uncertainty branch. Given a feature map P_i, the head of *actionness* predicts the actionness score, p^n_a, the head of *centerness* measures the centerness score, p^n_c, while the heads of *start-* and *end-boundary* classifiers estimate the scores of being a start and an end position, p^n_s and p^n_e for the snippet n, respectively. The prediction heads

are designed using two linear layers. Besides, there exists a *boundary regression* branch with two linear layers that returns a pair of relative distance estimations, $v^n = (l^n, r^n)$, from a snippet n to start and end boundaries. Finally, our network has a *localization uncertainty* branch to estimate uncertainties [45] for predicted relative distances, $\sigma^n = (\sigma_l^n, \sigma_r^n)$, with a linear layer attached to the first linear layer of the boundary regression branch.

Given a ground-truth segment at an interval $[s^*, e^*]$, the snippets are defined as positive within this interval for the actionness, i.e., snippet n within a ground-truth segment has an actionness value of $p_a^{n*} = 1$. Adapting the centerness formulation for temporal segments from FCOS [46], snippet n at location t within a ground-truth segment has a centerness value on the same interval as $p_c^{n*} = \sqrt{\frac{\min(l^{n*}, r^{n*})}{\max(l^{n*}, r^{n*})}}$ where l^{n*} and r^{n*} are distances of snippet n to start and end boundaries, $l^{n*} = t - s^*$ and $r^{n*} = e^* - t$ (otherwise the centerness value is zero). Corresponding start and end boundary labels are defined as positive within intervals $[s^* - \tau^*, s^* + \tau^*]$ and $[e^* - \tau^*, e^* + \tau^*]$, respectively, with an extra offset $\tau^* = (e^* - s^*)/10$. Following FCOS [46], positive snippets, that lie within a ground-truth segment, are participated in boundary regression and uncertainty prediction using $v^{n*} = (l^{n*}, r^{n*})$.

4 Teacher-student framework

The teacher-student framework is borrowed by many deep neural network models for semi-supervised learning [16] to reduce over-fitting with a large number of learning parameters and to train robust models with more abstract invariances. The framework jointly trains a student and a teacher model in a mutually beneficial way in which the student model learns and updates the teacher model using exponential moving average (EMA) [16]; while the teacher model generates targets to train student model. In this section, we describe the stages in the training process of the proposed teach-student framework; burn-in and mutual learning stages, respectively.

4.1 Burn-in stage

In a teacher-student framework, good initialization is important, since the teacher generates targets to be used by the student for learning. We utilize *Burn-in* training strategy [21] to optimize the student model weights θ using supervised data and supervised loss.

Let $P_i \in \mathbb{R}^{D_i \times T_i}$ be the feature map at layer i of pyramid network with feature dimension D_i and length T_i. Once we have ground truth labels at each location t on the feature map, we train our student model on supervised data with a fixed number of epochs using the following supervised loss

$$\begin{aligned}\mathcal{L}_{sup}^{snip} = \frac{1}{N_s}\Bigg(&\sum_n \ell_a(p_a^n, p_a^{n*}) + \sum_n \ell_c(p_c^n, p_c^{n*}) \\ &+ \sum_n \ell_s(p_s^n, p_s^{n*}) + \sum_n \ell_e(p_e^n, p_e^{n*})\Bigg) \\ &+ \frac{1}{N_{ps}}\Bigg(\sum_n \mathbf{1}^n \ell_{diou}(v^n, v^{n*}) \\ &+ \sum_n \mathbf{1}^n \ell_{unc}(v^n, \sigma^n, v^{n*})\Bigg),\end{aligned} \tag{7}$$

where we predict the actionness score, the centerness score, and the start-end boundary scores and regress the target segment intervals assuming each snippet location as an anchor point. $\mathbf{1}^n$ indicates that the n-th snippet is a positive instance within a ground-truth segment interval, and p_a^n, p_c^n, p_s^n, p_e^n, v^n and σ^n show the prediction outputs of corresponding network heads. N_s and N_{ps} are the numbers of all locations and positive locations in a batch, respectively. ℓ_a is defined as a cross-entropy loss, while ℓ_c, ℓ_s and ℓ_e are binary cross-entropy losses with logits. ℓ_{diou} is a temporal Intersection-over-Union (tIoU) based loss that is computed using predicted boundary distances $v^n = (l^n, r^n)$ and ground-truth boundaries $v^{n*} = (l^{n*}, r^{n*})$, where we adapt the Distance-IoU loss [47] for temporal segments as

$$\begin{aligned}\ell_{diou} &= 1 - tIoU + \frac{d(v, v^*)}{a^2}\,, \\ tIoU &= \frac{\min(l, l^*) + \min(r, r^*)}{\max(l, l^*) + \max(r, r^*)}\,, \\ d(v, v^*) &= |r - l - r^* + l^*|/2\,, \\ a &= \max(l, l^*) + \max(r, r^*)\,,\end{aligned} \tag{8}$$

where $d(\cdot, \cdot)$ is the Euclidean distance between the centers of the predicted and the ground truth segments and a is the length of the shortest enclosing segment covering the two segments.

The localization uncertainty branch is jointly trained with the boundary regression branch using ℓ_{unc} that is the negative power log-likelihood loss (NPLL) [45] as

$$\ell_{unc} = \eta \cdot \left[\Big(\sum_{k \in \{l,r\}} \Big(\frac{(k^* - k)^2}{2{\sigma_k}^2} + \frac{\log \sigma_k^2}{2}\Big)\Big) + 2\log 2\pi\right], \tag{9}$$

where η is either 1 or tIoU score between the predicted and the ground-truth boundaries $v = (l, r)$ and $v^* = (l^*, r^*)$, respectively. $k \in \{l, r\}$ and σ_k is the predicted uncertainty for left or right direction.

After 15 epochs in *Burn-in* stage, we copy the trained weights θ for both the teacher and the student models, $(\theta_t \leftarrow \theta, \theta_s \leftarrow \theta)$.

4.2 Mutual learning stage

In the mutual learning stage, the student and teacher models are jointly training using the EMA [16] strategy. Consistency learning as one self-training technique constrains model outputs to be comparable using transformed unlabeled data with some randomness. Therefore, the technique has been well-adopted for the SSL as reducing dependency on limited labeled data. We apply consistency regularization on both supervised and unsupervised data splits.

On supervised data, while the student model continues to learn using $\mathcal{L}_{sup}^{snip}$, the teacher model generates targets for student models on augmented copies of the data. Alongside $\mathcal{L}_{sup}^{snip}$, a regularization loss is used with two components, $\mathcal{L}_{sup}^{simcls}$ and $\mathcal{L}_{sup}^{simreg}$, respectively, given as

$$\mathcal{L}_{sup}^{simcls} = \frac{1}{N_s}\Big(\sum_n \ell_{con}(p_a^{nt}, p_a^{ns}) + \sum_n \ell_{con}(p_c^{nt}, p_c^{ns}) + \sum_n \ell_{con}(p_s^{nt}, p_s^{ns}) + \sum_n \ell_{con}(p_e^{nt}, p_e^{ns})\Big) , \tag{10}$$

where ℓ_{con} is the mean square error loss to compare the softmax activations over the actionness predictions and the sigmoid activations over the centerness, the start- and end-boundary predictions by the student and teacher models, $p_\cdot^{ns}$ and $p_\cdot^{nt}$, respectively. N_s is the number of augmented snippet copies in the batch. Besides, there exists a regression part with ℓ_{sup}^{simreg} given as

$$\ell_{sup}^{simreg} = \begin{cases} \ell_{diou}(v^{nt}, v^{ns}) & \text{if } \sigma^{nt} + \delta \le \sigma^{ns} \\ 0 & \text{otherwise} \end{cases} ,$$
$$\mathcal{L}_{sup}^{simreg} = \frac{1}{N_s}\sum_n \ell_{sup}^{simreg}(v^{ns}, v^{nt}, \sigma^{ns}, \sigma^{nt}) , \tag{11}$$

where $\delta \ge 0$ is a small margin between the localization uncertainties of teacher and student models and we set it to 0.01. Following Liu et al. [11], we first remove the boundaries where the student model has small localization certainty, e.g. $\sigma^{ns} \le 0.5$. Then, the loss between the boundary predictions of the student and the teacher models are compared using ℓ_{diou} given in Eq. (8) if the teacher certainty is higher than the student certainty value.

When we have unsupervised data as well, we follow a similar methodology with consistency regularization, but each batch contains both supervised and unsupervised data. Consistency regularization is also applied to unsupervised data predictions of teacher and student models using Eqs. (10) and (11). Finally, the objective function is extended as follows

$$\mathcal{L} = \mathcal{L}_{sup}^{snip} + \mathrm{w}^{cls}\mathcal{L}_{sup}^{simcls} + \mathrm{w}^{reg}\mathcal{L}_{sup}^{simreg} + \mathrm{w}^{usup}(\mathrm{w}^{cls}\mathcal{L}_{usup}^{simcls} + \mathrm{w}^{reg}\mathcal{L}_{usup}^{simreg}) , \tag{12}$$

where we have used three weights, w^{cls}, w^{reg} and w^{usup}, respectively.

Our model leverages two kinds of augmentations, *weak* and *strong* augmentations, on supervised data as well as unsupervised data. The student model is trained using *strongly* augmented data while the teacher model is trained using *weakly* augmented one. In all of our experiments, *weak* augmentation is a snippet-dropping strategy with a probability of 5% on input videos of the teacher model, i.e., 5% of the feature channels are dropped. For *strong* augmentation, we apply both (i) the snippet-dropping strategy with a probability of 20% and (ii) the temporal shifting operations on randomly chosen μ of feature channels [13, 48] on input videos of the student model.

5 Proposal inference in multiple scales

During inference, we follow similar inference steps and use the scorig function from our previous study [22]. We generate lists of proposals from feature maps and merge the lists. For a feature map P_i, we first extract the candidate proposal locations and then score these candidates with a scoring function. Later, we prune proposals via non-maximum suppression (NMS) and select top M candidates.

To extract candidate locations, i.e., start and end boundaries, we compute two vectors for each boundary type: g_s and g_e that are the boundary estimates via the boundary regression and uncertainty branches, and g_s' and g_e' that are the boundary estimates via the snippet-based start- and end boundary prediction branches. Given a snippet n at location t on a video test instance, the boundary regression and uncertainty branches return estimate $v^n = (l^n, r^n)$ with uncertainty scores $\sigma^n = (\sigma_l^n, \sigma_r^n)$, respectively. These values are translated into start boundary scores using a probability density function of $\mathcal{N}(s^n, \sigma_l^n)$ within a neighborhood $[s^n - \tau', s^n + \tau']$ where $s^n = t - l^n$ and τ' is a small margin. We similarly generate end boundary scores within a neighborhood $[e^n - \tau', e^n + \tau']$ where $e^n = t + r^n$. Translating start and end scores for all snippets, the final start and end score vectors, g_s and g_e, are built as the maximum of all start and end scores at each location, respectively.

Concurrently, the start-end boundary heads return vectors of predictions with scores p_s^n and p_e^n of a snippet n. We prune these vectors by setting scores to zero at locations that are *not peak* and having scores lower than a threshold value. Then, we obtain two vectors of boundary estimates, g_s' and g_e' for start and end, respectively. A value p_s^n is a *peak* if $p_s^n > p_s^{n-1}$, $p_s^n > p_s^{n+1}$ and $p_s^n > thr$ (similarly for p_e^n).

Lastly, a snippet is in start set, S, if $g_s(n) + g'_s(n) > 0$ and in end set, E, if $g_e(n) + g'_e(n) > 0$, respectively. Having the boundary start locations S and end locations E, we generate $|S| \times |E|$ candidate proposal locations.

Given a proposal candidate m with snippets $\mathcal{X}^m = \{x_1, \ldots, x_c, \ldots, x_n\}$ where x_1, x_c, x_n are the start, center and end points of the proposal candidate, $x_1 \in S$ and $x_n \in E$, we devise a scoring function with three components. sc^m_{action} is the average of actionness scores of all proposal snippets. Next, sc^m_{center} is computed over the centerness scores of the start, center and end snippets, where the score is high for a proposal with low centerness scores on the boundaries and a high centerness score in the middle. Finally, sc^m_{bound} is computed over the start and end scores of the start, center and end snippets, where the score is high for a proposal with low start-end boundary scores in the center and high boundary scores in the edges. Then, we combine the scores with equal weights as $sc^m = sc^m_{action} + sc^m_{center} + sc^m_{bound}$ and the components are given as

$$
\begin{aligned}
sc^m_{action} &= \frac{1}{|\mathcal{X}^m|} \sum_{x \in \mathcal{X}^m} p^x_a, \\
sc^m_{center} &= \sqrt[4]{p^{x_c}_c (1-p^{x_1}_c)(1-p^{x_n}_c)(p^{x_c}_c / \max_{x \in \mathcal{X}^m} p^x_c)}, \\
sc^m_{bound} &= \sqrt[4]{p^{x_1}_s \, p^{x_n}_e (1-p^{x_c}_s)(1-p^{x_c}_e)}.
\end{aligned} \tag{13}
$$

After generating candidate proposals, we prune redundant ones using non-maximum suppression (NMS) or soft-NMS to achieve higher recall rates [2, 49].

6 Experimental evaluation

Our goal is to demonstrate the robustness and performance of our method, i.e., TTMT, on the task of generating accurate action proposals on two benchmark datasets, the THUMOS14 [26] and ActivityNet-1.3 [27]. Detailed ablation comparisons on the THUMOS14 dataset are also presented to analyze the model.

6.1 Datasets

THUMOS14 [26]. The dataset includes 1010 and 1574 videos of 20 action categories in the validation and test splits, respectively. Among these videos, 200 validation videos and 212 test videos have temporal annotations of actions. Following the previous studies [1, 2], we conduct our training on the validation set and performance evaluation on the test set.

ActivityNet-1.3 [27]. The dataset consists of 19, 994 long-term untrimmed video sequences in 200 action categories. The dataset splits into training, validation and testing subsets with 10, 024, 4, 926 and 5, 044 video samples, respectively. Each video sequence contains one or more actions with annotated segment intervals. We train our model on the training set and evaluate on the validation set.

6.2 Visual encodings and training settings

Given an untrimmed video, it is represented as a sequence of T' snippets encoded using pre-trained CNN models, $F' \in \mathbb{R}^{T' \times D'}$. For the THUMOS14, we use feature encoding precomputed by [20] based on TSN pre-trained model on Kinetics [50]. We split each video sequence during inference with overlapped windows of size 128 and stride 64. For the ActivityNet, we used the Slowfast features precomputed by [13]. We scale the feature length to $T' = 128$ for all videos.

Following Ji et al. [12] and Wang et al. [13], we split the training data with available labels into labeled and unlabeled subsets. We have three data settings represented as TTMT@M% where M $\in \{100, 90, 60\}$ and M% of the training data is reserved as labeled data for supervised learning within the temporal teacher pipeline, e.g. TTMT@60% means that our model is trained following the proposed teacher-student framework using 60% of available data as labeled in supervised training and 40% of data as unlabeled in unsupervised training. We obtain predictions of the student and teacher models concurrently. Since we have observed that the teacher model outperforms the student model, we report the teacher results throughout the experiments. The predictions are from the best student and teacher models that are the ones with the lowest validation loss. For both datasets, the learning rate of 10^{-4} is used with a weight decay of 10^{-9}. We use the Adam optimizer during training.

For the THUMOS experiments, we use the weight combination of $\mathrm{w}^{cls} = 6$, $\mathrm{w}^{reg} = 0.005$ in TTMT@100% training setting (where $\mathrm{w}^{usup} = 0$), and $\mathrm{w}^{usup} = 1$ in TTMT@60% and @90% training settings. For the ActivityNet experiments, we use the weight combination of $\mathrm{w}^{cls} = 6$, $\mathrm{w}^{reg} = 0.05$ in TTMT@100% training setting (where $\mathrm{w}^{usup} = 0$), and $\mathrm{w}^{usup} = 1$ in TTMT@60% and @90% training settings. For temporal augmentation, we have experimented with various snippet-dropping percentages and temporal shift parameters. Based on our empirical observation, we report our results for the randomly chosen $\mu = 64$ of feature channels where half of the channels move forward, and the other half of them move backward by a shift amount of 1.

6.3 Proposal generation

Following previous works [1, 2, 4, 5], proposal generation task is evaluated by means of Average Recall (AR) and Area Under Curve (AUC) metrics. AR is evaluated under various tIoU thresholds in the range [0.5, 0.95] for the ActivityNet

Table 1 Evaluation of the model TTMT@100% in various Transformer settings on the THUMOS14 dataset

#Blocks (B)	AR@50	AR@100	AR@200	AR@500	AR@1000
B[5]	45.03	52.96	60.12	67.11	67.58
B[6]	44.90	52.55	59.69	66.81	67.83
B[7]	45.92	53.49	60.39	66.90	67.77
B[8]	45.97	53.12	60.03	66.75	67.53
B[9]	44.97	52.56	59.26	66.85	67.97
B[10]	45.90	52.93	59.57	66.50	67.24
B[11]	45.56	52.76	59.85	66.54	67.35
B[7 + 1]	44.20	52.04	59.15	66.28	66.99
B[7 + 2]	46.16	53.25	60.53	66.55	67.14
B[7 + 3]	45.80	53.20	59.90	66.34	67.19
B[8 + 1]	45.99	53.18	**60.59**	67.13	67.69
B[8 + 2]	**46.47**	**53.94**	60.47	**67.24**	**68.17**
B[8 + 3]	46.41	53.14	59.81	66.53	67.34

#Blocks (B) column specifies the number of blocks in each stage of the transformer model. The channel width value of the first stage is set 1024 and the head value of the first stage is set 8 in transformer models. Bold marks the highest score

and in the range [0.5, 1.0] for the THUMOS14 with a step of 0.05. The AUC is calculated using AR under various Average Number of Proposals (AN) as AR@AN, where AN varies from 0 to 100 for the ActivityNet and from 0 to 1000 for the THUMOS14.

6.3.1 Proposal generation on THUMOS dataset

We first examine the pyramid setting of the core transformer architecture to see its effect on the performance of model TTMT@100% in the proposal generation task. Following an incremental strategy, we experiment with up to three stages of a pyramid with various block numbers and report two-stage results since we have not observed further improvement with more stages.

Table 1 shows the AR@AN performances with AN varying from 50 to 1000 on the test set with NMS pruning (threshold is set to 0.83). Using a single stage Transformer network with a number of blocks B in range $[1, \ldots, 11]$, the initial channel depth of 1024 and the head number of 8, we observe that while the model TTMT@100% with B[8] shows the highest performance of 45.97 AR@50, the model TTMT@100% with B[7] shows better performance at higher AR values. We build the pyramid iteratively and extend the models B[7] and B[8] with a second stage. Adding a second stage to the pyramid, we observe that the model TTMT@100% with B[7+2] over B[7] and the model TTMT@100% with B[8+2] over B[8] gain improvement in AR@50, and TTMT@100% with B[8+2] outperforms all other single-stage models we have tested so far. Evaluations on the THUMOS datasets show that the second stage of resolution can help to improve AR performances, as we expect from a pyramid structure.

We also conduct a set of experiments to search for weight combinations, w^{cls} and w^{reg} given in Eq. (12), over the model TTMT@100% with B[8+2]. The plot in Fig. 3 shows that increasing w^{cls} in consistency regularization has a significant effect in performance, while the variation in w^{reg} has a minor effect.

Comparisons with semi-supervised models Selecting the best transformer settings on TTMT@100%, we examine the performances of the models TTMT@60% and TTMT@90% with B[8] and B[8+2] where models use less supervision due to fewer labeled data than supervised models. Reported in Table 2, we have been outperforming [12, 13], except [13] in AR@1000. The performance improvement at low recall values are more important and we particularly outperform others in AR@50 and AR@100. For instance, we achieve better than [13] by +1.87 @60% and +4.91 @90% in AR@50, respectively. Moreover, model B[8+2] outperforms model B[8] in AR@50 due to its multiscale nature at 90% and 60% settings as well. We have observed slightly lower performance than [13] only in AR@1000 with −1.94 @60% and −1.35 @90%. Both Ji et al. [12] and SSTAP [13] are teacher-student frameworks, but they rest on anchor-based models as the core architecture. While the former is built on the BSN [2] proposal generation model, the latter is built on the BMN [4] model. Using an anchor-free model, we achieve better performance in AR metrics except AR@1000, but improving performance at low AR values is important.

Comparisons with fully-supervised models. Similarly, we examine the performance of our model TTMT@100% with some related fully-supervised approaches. TTMT@100% is trained using all the available labeled data and only the supervised loss components from Eq. (12), i.e., $w^{usup} = 0.0$. Table 3 reports our results, in comparison to other studies. We

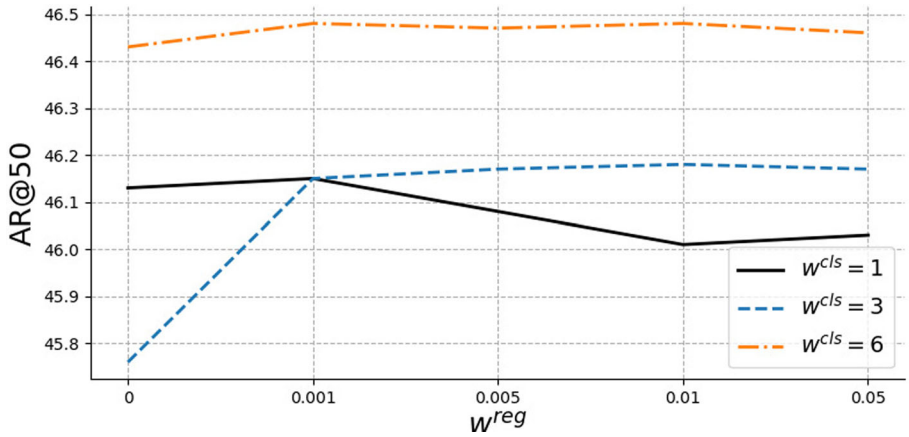


Fig. 3 Evaluation of the model TTMT@100% with B[8+2] in various weight combinations $\{w^{cls}, w^{reg}\}$

Table 2 Comparison of semi-supervised baseline models with our proposal generation models TTMT@60% and TTMT@90% that use less number of labeled data on the THUMOS14

Semi-supervised models	AR@50	AR@100	AR@200	AR@500	AR@1000
Ji et al.@60% [12]	37.42	46.71	53.96	61.01	65.10
SSTAP@60% [13]	39.42	48.02	55.03	–	67.07
SSTAP@90% [13]	40.12	49.22	55.86	–	68.21
TTMT@60% B[8]	40.93	49.14	55.77	64.18	65.13
TTMT@60% B[8 + 2]	41.29	49.03	55.13	62.83	64.51
TTMT@90% B[8]	44.32	52.15	59.23	66.20	66.86
TTMT@90% B[8 + 2]	45.03	52.50	59.09	66.01	66.80

Table 3 Comparison of fully-supervised baseline models with our proposal generation model TTMT on the THUMOS14

Models	AR@50	AR@100	AR@200	AR@500	AR@1000
CTAP [1]	32.49	42.61	51.97	–	–
BSN [2]	37.46	46.06	53.21	60.64	64.52
MGG [5]	39.93	47.75	54.65	61.36	64.06
BMN+SNMS [4]	39.36	47.72	54.70	62.07	65.49
DBG+SNMS [3]	37.32	46.67	54.50	62.21	66.40
BC-GNN+SNMS [33]	40.50	49.60	56.33	62.80	66.57
TCANet [51]	42.05	50.48	57.13	63.61	66.88
RTD-Net [34]	41.52	49.32	56.41	62.91	–
CPN [52]	39.90	49.98	58.22	66.47	**70.20**
SSTAP@100% [13]	41.01	50.12	56.69	–	68.81
TTMT@100% B[8]	45.97	53.12	60.03	66.75	67.53
TTMT@100% B[8 + 2]	**46.47**	**53.94**	**60.47**	**67.24**	68.17
TTMT@90% B[8 + 2]	45.03	52.50	59.09	66.01	66.80
TTMT@60% B[8 + 2]	41.29	49.03	55.13	62.83	64.51

Best scores are in bold

have observed better results than the other fully-supervised methods except in AR@1000. In AR@1000, the performance of model TTMT@100% with B[8+2] is lower than [13] by −0.64 and [52] by −2.03, respectively.

6.3.2 Proposal generation on ActivityNet dataset

On the ActivityNet dataset, we perform the same iterative strategy to check the pyramid settings as we perform for the THUMOS dataset. Examining model TTMT@100% with B in range [1, . . . , 11], the initial channel depth of 1024 and the head number of 8, we observe that B[1] and B[4] show comparable and the best performances. Increasing pyramid stages, we have observed no improvement thus we do not report the performance here. Performance in AR and AUC metrics are reported in Table 4 with comparison to some state-of-the-art studies. Most state-of-the-art results except models of Ji et al. [12] and SSTAP [13] are taken in fully-

Table 4 Comparison with some state-of-the-art fully-supervised and semi-supervised anchor-based models on the ActivityNet

Models	AR@1 (val)	AR@100 (val)	AUC (val)
CTAP [1]	–	73.17	65.72
BSN [2]	32.17	74.16	66.17
MGG [5]	–	74.54	66.43
BMN [4]	–	75.01	67.10
DBG [3]	–	76.65	**68.23**
BC-GNN [33]	–	**76.73**	68.05
TCANet [51]	**34.55**	76.08	68.08
RTD-Net [34]	33.05	73.21	65.78
Ji et al.@60% [12]	–	75.07	66.35
BMN*@60% [4]	–	74.42	66.47
SSTAP@60% [13]	–	75.20	67.23
BMN*@100% [4]	–	75.01	67.10
SSTAP@100% [13]	–	75.54	67.53
TTMT@60% B[4], η=1 in Eq. (9)	33.70	74.80	66.48
TTMT@100% B[1], η=1 in Eq. (9)	33.69	75.13	66.61
TTMT@60% B[4], η=tIoU in Eq. (9)	33.42	74.69	66.63
TTMT@100% B[1], η=tIoU in Eq. (9)	33.66	75.20	66.70

BMN$*$ results are from Wang et al. [13]
Best scores are in bold

supervised setting where there is also no teacher-student framework. Besides, all the models reported in Table 4 including Ji et al. [12] and SSTAP [13] are anchor-based. Our model TTMT@100% results in better performance than CTAP [1], BSN [2], MGG [5] and BMN [4], while both TTMT@100% and TTMT@60% show competitive performance with anchor-based semi-supervised models Ji et al. [12], and SSTAP [13]. Both Ji et al. [12] and SSTAP [13] are teacher-student frameworks, and they rely on anchor-based BSN [2] and BMN [4] architectures, respectively.

We report the ActivityNet results for two settings in which we modify the η in ℓ_{unc} [see Eq. (9)] and set η=tIoU or η=1. The results are similar with minor variations.

Computational analysis With our current implementation, we have analyzed the average network inference time using an Nvidia Tesla P100 graphics card on a sample of 600 videos from the ActivityNet Dataset. Following Lin et al. [2, 4] and Tan et al. [34], we exclude the computation of the backbone feature extractor, since it is pre-computed. As mentioned in Sect. 3.3, the depth of channels, i.e., D_i, has been set to 1024 and the network inference took an average of 0.0018 s for the model TTMT@100 B[1]. In this study, we inherited the MViTv2 implementation. The standard attention unit has quadratic complexity in computing and memory [23]. Some recent works aim to reduce quadratic time complexity to make transformers more efficient with linear time [53, 54]. Although our proposed bi-directional mask strategy for forward and backward computations is slightly lower, doubling the attention units still maintains the same computational complexity.

Number of parameters is another measure of model complexity. Since our architecture includes pooling-layers within the pooling attention unit, the total number of learnable parameters is related to D_i. If D_i is set to 256, the total number of learnable parameters for the model TTMT@100 B[1] is 4.8 M. If D_i is set to 1024, then the total number for the same model is 25.7M. As we stated in Sect. 3.3, we use $D_i = 1024$ in all our experiments on both benchmarks. The BMN [4] model that is integrated into SSTAP [13] includes a 5.7M total number of learnable parameters when the channel depth is set to 256. Additionally, a Transformer-based model RTD-Net contains a total of 32.1M learnable parameters. However, we refrain from making a direct comparison because various architectures rely either on various submodel structures at different network depths or on various hyperparameters, and the tuning of these parameters acts differently on performance in each model.

6.4 Ablation studies on THUMOS dataset

A set of ablation studies are conducted to further investigate the proposed teacher-student transformer network. We examine (i) the impact of teacher-student training over traditional fully-supervised training, (ii) the impact of uni-directional and bi-directional masks on the proposal generation task, (iii) the impact of each component in the scoring function,

Table 5 Evaluation of the core model on the THUMOS14 dataset in comparison with the model integrated into the teacher-student framework

Models	AR@50	AR@100	AR@200	AR@500	AR@1000
Masked transformer@100%B[8]	41.64	48.50	54.63	62.70	62.91
Masked transformer@100%B[8 + 2]	41.09	48.46	54.87	63.04	64.58
TTMT@100% B[8]	45.97	53.12	60.03	66.75	67.53
TTMT@100% B[8 + 2]	**46.47**	**53.94**	**60.47**	**67.24**	**68.17**

Bold marks the highest score
Best scores are in bold

Table 6 Evaluation of different mask integrations within the model TTMT@100% with B[8 + 2] on the THUMOS14 dataset

Mask Structures	AR@50	AR@100	AR@200	AR@500	AR@1000
None (original pooling unit)	45.65	52.62	59.51	66.18	66.97
Bidirectional-GL	46.28	53.17	60.00	66.51	67.22
Bidirectional-G	**46.47**	**53.94**	**60.47**	**67.24**	**68.17**
Bidirectional-L	45.88	53.15	59.94	66.49	67.07
Backward-G	41.89	49.63	56.25	64.76	66.28
Forward-G	44.22	51.64	58.38	65.92	67.33

Best scores are in bold

and (iv) the impact of two pathways for extracting candidate boundaries.

6.4.1 Impact of teacher-student framework

Under the same evaluation settings, we examine the performance of the core encoder-only Masked Transformer model introduced in Sect. 2 without the integration into the teacher-student framework. We have conducted the experiments for B[8] and B[8+2], keeping the setting we have used in model TTMT@100% (i.e., the student model in the build-in stage has the equivalent setting to the core model as well). As reported in Table 5, we obtain significant improvement within the TTMT framework over the core Transformer model in all AR metrics, e.g., the model TTMT@100% with B[8 + 2] improves by 5.38 in AR@50 over B[8 + 2]. In TTMT, we have two competing models where the teacher model (EMA model) is trained smoothly over the student model weights, and we apply pseudo-labeling and consistency regularization. It shows that the integration into the teacher-student framework helps in improving the performance of the core model.

6.4.2 Impact of bi-directional masks

To see the impact of the masking strategy introduced in Sect. 3.2, we examine the performance of the model TTMT@100% with B[8+2] using different mask structures. We explore the TTMT model using: None, Bidirectional-GL, Bidirectional-G, Bidirectional-L, Backward-G and Forward-G mask structures. The None is equivalent to using the original pooling attention unit [24] without any mask. The Bidirectional-G and Bidirectional-L are based on using a bi-directional pooling attention unit with two local (L) and two global (G) masks, respectively, in forward and backward directions. While the masks in Bidirectional-G cover the whole video, the masks in Bidirectional-L cover the maximum of T/2 of the neighborhood of the entities and disable the interactions between rest of the snippets (i.e., M_{ij}^{f} is $-\infty$ as well, if $i < (j - T/2)$ and M_{ij}^{b} is $-\infty$ as well, if $i > (j + T/2)$). We also experiment on the Bidirectional-GL that includes 4 branches in the pooling attention unit with four masks of two local (L) and two global (G) masks in forward and backward directions.

Given in Table 6, we have observed that the Bidirectional-G outperform other cases in all AR metrics, both Bidirectional-G and Bidirectional-L are better than None case, and also G masks result in better performance than L masks. Investigating, the uni-directional versions of the pooling attention unit, results show that both the Bidirectional-G and Bidirectional-L have better performance than the uni-directional Forward-G and Backward-G versions (uni-directional versions contain a global mask in a specific direction). It suggests using bi-direction masks over uni-directional ones for video evaluation when the offline evaluation setting is possible. Moreover, the results verify the benefits of the masked Transformer models for temporal video evaluation.

6.4.3 Impact of scoring function

To see the impact of each component of the scoring function given in Eq. (13), we examine the individual components as well as their combinations. Table 7 presents the AR performances and we see that the combined score has a significant improvement over other combinations. A weighting strategy can also be applied at this level of the inference to improve

Table 7 Evaluation of the proposed scoring function with actionness sc_{action}, centerness sc_{center} and boundary sc_{bound} components with the model TTMT@100% with B[8 + 2] on the THUMOS14 dataset (see Sect. 5)

sc_{action}	sc_{center}	sc_{bound}	AR@50	AR@100	AR@200	AR@500	AR@1000
✓			12.21	21.18	36.45	57.31	61.09
	✓		33.87	44.34	53.96	63.89	64.86
		✓	34.96	46.48	53.49	62.21	63.95
✓	✓		41.16	49.45	57.01	65.92	67.34
✓		✓	44.72	51.57	56.99	65.07	66.43
✓	✓	✓	**46.47**	**53.94**	**60.47**	**67.24**	**68.17**

Best scores are in bold

Table 8 Evaluation of g and g' on the proposed model TTMT@100% (see Sect. 5)

Models		AR@50	AR@100	AR@200	AR@500	AR@1000
B[8]	g_s, g_e	45.97	52.27	58.4	63.75	63.84
B[8]	g'_s, g'_e	**47.78**	**53.87**	56.29	56.29	56.29
B[8]	$g_s + g'_s, g_e + g'_e$	45.97	53.12	**60.03**	**66.75**	**67.53**
B[8 + 2]	g_s, g_e	47.31	53.43	58.95	64.66	64.73
B[8 + 2]	g'_s, g'_e	**48.29**	53.84	56.95	56.95	56.95
B[8 + 2]	$g_s + g'_s, g_e + g'_e$	46.47	**53.94**	**60.47**	**67.24**	**68.17**

Best scores are in bold

Table 9 Comparison of the detection result with mAP@tIoU in various tIoU values [20]

Models	0.7	0.6	0.5	0.4	0.3
TURN+UNet [1]	6.3	14.1	24.5	35.3	46.3
BSN+UNet [2]	20.0	28.4	36.9	45.0	53.5
MGG+UNet [5]	21.3	29.5	37.4	46.8	53.9
BMN+UNet [4]	20.5	29.7	38.8	47.4	56.0
DBG+UNet [3]	21.7	30.2	39.8	49.4	57.8
BC-GNN+UNet [33]	23.1	31.2	40.4	49.1	57.1
G-TAD+UNet [20]	23.4	30.8	40.2	47.6	54.5
Ji et al. [12] @100%+UNet	21.9	32.2	41.7	51.2	57.9
SSTAP@100%+UNet [13]	22.8	32.8	42.3	51.5	58.4
TTMT@100% B[10]+UNet	**24.2**	**34.7**	**45.2**	**52.6**	60.0
TTMT@100% B[8]+UNet	22.6	32.4	42.7	50.8	58.3
TTMT@100% B[8 + 2]+UNet	23.3	33.0	43.5	51.9	**60.2**

Our proposals are trained using the visual encoding and the video-level classification results from the G-TAD
Best scores are in bold

performance, but we have here simply added the computed scores and obtained convincing results.

As can be seen from the combined results, each component of the scoring function contributes effectively to the overall score. This emphasizes that our snippet-based structure requires a well-designed scoring function with powerful components and thus will perform better. The function we introduce here gives good results on our snippet-based prediction structure, removing some parts will cause a decrease in performance. Moreover, a better designed scoring function can further improve performance, while a poorly designed one can degrade it.

6.4.4 Impact of branches on boundaries

As we discuss in Sect. 5, we compute the boundaries of candidate proposals via two pathways, g and g', respectively. We conduct a set of experiments to see the impact of each pathway on the boundary predictions and report the inference performances in Table 8. We observe that boundary estimation via g' has benefits over g in AR@50 and AR@100 while g results in better inference in higher AR metrics.

6.5 Temporal action localization

To examine the performance for action detection, Mean Average Precision (mAP) is calculated with tIoU threshold values in the range [0.3, 0.7] with a step of 0.1 for the THUMOS14 dataset. Following other two-stage detection pipelines [2, 4], we first create the top 200 proposals using our model on THUMOS14, and we then use UntrimmedNet (UNet) model [18] to get video-level classification results and keep the top-2 class for each video. Finally, we compute a detection score for each proposal using the proposal score by our TTMT network and the classification score by UNet. In particular, the final scores for the top-2 action categories of each proposal are calculated by a simple multiplication of the proposal scores (see Eq.(13)) and UNet scores. Comparative results are shown in Table 9. Using the same classifier, i.e., UNet, we can observe that our TTMT@100% model outperforms many state-of-the-art anchor-based architectures (e.g. [2, 5, 13]) in high tIoU settings.

7 Conclusion

In this paper, we incorporate a new anchor-free proposal generation model into a teacher-student framework for a semi-supervised two-stage detection pipeline. We apply the pseudo-labeling techniques for classification and regression to improve generated proposals and integrate relative teacher-student uncertainties for selecting effective pseudo-labels in the proposed anchor-free model. We further provide a detailed evaluation of the Masked Transformer network within the teacher-student framework. The proposed Transformer-based model is designed for modeling temporal ordering with a lighter structure compared to anchor-based alternatives and the architecture can be extended with many local predictors by just simply integrating them into the pyramid network branch.

We show how our transformer-based anchor-free SSL method can achieve comparable performance with the state-of-the-art anchor-based methods, besides many architectural benefits. We find that our model benefits from uncertainty estimations and that a good scoring function for merging local estimates is necessary for a good performance.

Acknowledgements This work has been funded by the Academy of Finland Project Numbers 329268 and 345791. We also acknowledge the computational resources provided by the Aalto University's Aalto Science IT project, CSC–IT Center for Science and the LUMI Supercomputer.

Author Contributions S.P. worked on the conceptualization, methodology and visualizations, conducted the experiments, and wrote the main manuscript text. J. L. contributed in manuscript drafting and text editing. All authors reviewed the manuscript.

Funding Open Access funding provided by Aalto University.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Gao, J., Chen, K., Nevatia, R.: Ctap: complementary temporal action proposal generation. In: ECCV, pp. 68–83 (2018)
2. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: BSN: boundary sensitive network for temporal action proposal generation. In: ECCV, pp. 3–19 (2018)
3. Lin, C., Li, J., Wang, Y., Tai, Y., Luo, D., Cui, Z., Wang, C., Li, J., Huang, F., Ji, R.: Fast learning of temporal action proposal via dense boundary generator. In: AAAI, pp. 11499–11506 (2020)
4. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: BMN: boundary-matching network for temporal action proposal generation. In: ICCV, pp. 3889–3898 (2019)
5. Liu, Y., Ma, L., Zhang, Y., Liu, W., Chang, S.-F.: Multi-granularity generator for temporal action proposal. In: CVPR, pp. 3604–3613 (2019)
6. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: a holistic approach to semi-supervised learning. In: NIPS (2019)
7. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.-L.: Fixmatch: simplifying semi-supervised learning with consistency and confidence. In: NIPS (2020)
8. Jeong, J., Lee, S., Kim, J., Kwak, N.: Consistency-based semi-supervised learning for object detection. In: NIPS, vol. 32 (2019)
9. Sohn, K., Zhang, Z., Li, C.-L., Zhang, H., Lee, C.-Y., Pfister, T.: A simple semi-supervised learning framework for object detection. arXiv preprint arXiv:2005.04757 (2020)
10. Zhou, Q., Yu, C., Wang, Z., Qian, Q., Li, H.: Instant-teaching: an end-to-end semi-supervised object detection framework. In: CVPR, pp. 4081–4090 (2021)
11. Liu, Y.-C., Ma, C.-Y., Kira, Z.: Unbiased teacher v2: semi-supervised object detection for anchor-free and anchor-based detectors. In: CVPR, pp. 9819–9828 (2022)
12. Ji, J., Cao, K., Niebles, J.C.: Learning temporal action proposals with fewer labels. In: ICCV, pp. 7073–7082 (2019)
13. Wang, X., Zhang, S., Qing, Z., Shao, Y., Gao, C., Sang, N.: Self-supervised learning for semi-supervised temporal action proposal. In: CVPR, pp. 1905–1914 (2021)
14. Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Learning salient boundary feature for anchor-free temporal action localization. In: CVPR, pp. 3320–3329 (2021)
15. Zhang, C.-L., Wu, J., Li, Y.: Actionformer: localizing moments of actions with transformers. In: ECCV (2022)
16. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. In: NIPS, vol. 30 (2017)
17. Nag, S., Zhu, X., Song, Y.-Z., Xiang, T.: Semi-supervised temporal action detection with proposal-free masking. In: ECCV, pp. 663–680 (2022)
18. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: CVPR (2017)
19. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: ICCV (2019)
20. Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-tad: Sub-graph localization for temporal action detection. In: CVPR (2020)
21. Liu, Y.-C., Ma, C.-Y., He, Z., Kuo, C.-W., Chen, K., Zhang, P., Wu, B., Kira, Z., Vajda, P.: Unbiased teacher for semi-supervised object detection. arXiv preprint arXiv:2102.09480 (2021)
22. Pehlivan, S., Laaksonen, J.: Anchor-free action proposal network with uncertainty estimation. In: ICME (2023)

23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS, vol. 30 (2017)
24. Li, Y., Wu, C.-Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., Feichtenhofer, C.: Mvitv2: improved multiscale vision transformers for classification and detection. In: CVPR, pp. 4804–4814 (2022)
25. Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., Zhang, C.: Disan: directional self-attention network for RNN/CNN-free language understanding. In: AAAI (2018)
26. Jiang, Y., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: action recognition with a large number of classes (2014)
27. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: a large-scale video benchmark for human activity understanding. In: CVPR, pp. 961–970 (2015)
28. Zhao, Y., et al.: Temporal action detection with structured segment networks. In: ICCV (2017)
29. Girshick, R.: Fast r-CNN. In: ICCV (2015)
30. Xu, H., Das, A., Saenko, K.: R-c3d: region convolutional 3d network for temporal activity detection. In: ICCV (2017)
31. Gao, J., Yang, Z., Chen, K., Sun, C., Nevatia, R.: Turn tap: temporal unit regression network for temporal action proposals. In: ICCV (2017)
32. Eun, H., Lee, S., Moon, J., Park, J., Jung, C., Kim, C.: Srg: snippet relatedness-based temporal action proposal generator. IEEE Trans. Circuits Syst., Video Technol. 11: 4232–4244 (2019)
33. Bai, Y., Wang, Y., Tong, Y., Yang, Y., Liu, Q., Liu, J.: Boundary content graph neural network for temporal action proposal generation. In: ECCV (2020)
34. Tan, J., Tang, J., Wang, L., Wu, G.: Relaxed transformer decoders for direct action proposal generation. In: ICCV (2021)
35. Carion, N., Massa, F., Synnaeve, G., et al.: End-to-end object detection with transformers. In: ECCV (2020)
36. Lee, D.-H., et al.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning. ICML (2013)
37. Bachman, P., Alsharif, O., Precup, D.: Learning with pseudo-ensembles. In: NIPS (2014)
38. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
39. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: NIPS (2016)
40. Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X., Liu, Z.: End-to-end semi-supervised object detection with soft teacher. In: ICCV (2021)
41. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
42. Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., Zhang, S.-H., Martin, R.R., Cheng, M.-M., Hu, S.-M.: Attention mechanisms in computer vision: a survey. Comput. Vis. Media **8**(3), 331–368 (2022)
43. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: a survey. ACM Comput. Surv. (CSUR) **54**(10s), 1–41 (2022)
44. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
45. Lee, Y., Hwang, J.-w., et al.: Localization uncertainty estimation for anchor-free object detection. arXiv preprint arXiv:2006.15607 (2020)
46. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: ICCV, pp. 9627–9636 (2019)
47. Zheng, Z., Wang, P., Liu, W., Li, J., et al.: Distance-iou loss: faster and better learning for bounding box regression. In: AAAI (2020)
48. Lin, J., Gan, C., Han, S.: Tsm: temporal shift module for efficient video understanding. In: ICCV, pp. 7083–7093 (2019)
49. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: ICCV (2017)
50. Kay, W., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
51. Qing, Z., Su, H., Gan, W., Wang, D., Wu, W., Wang, X., Qiao, Y., Yan, J., Gao, C., Sang, N.: Temporal context aggregation network for temporal action proposal refinement. In: CVPR (2021)
52. Hsieh, H.-Y., Chen, D.-J., Liu, T.-L.: Contextual proposal network for action localization. In: WACV (2022)
53. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: self-attention with linear complexity. arXiv preprint arXiv:2006.04768 (2020)
54. Shen, Z., Zhang, M., Zhao, H., Yi, S., Li, H.: Efficient attention: attention with linear complexities. In: WACV (2021)

Selen Pehlivan received her PhD in the field of computer vision from the Computer Engineering Department at Bilkent University in 2012. She is currently working as a research scientist at VTT Technical Research Centre of Finland. Her research primarily focuses on human activity understanding and multimodal representations. She has been a visiting scholar at the University of Illinois at Urbana-Champaign (2009-2011), worked as a postdoctoral research associate in the Center for Research in Computer Vision at the University of Central Florida (2013), and also served as a faculty member in the Computer Engineering Department at TED University, Turkey (2014-2019). Prior to joining VTT, she worked as a researcher at Aalto University School of Science, Finland (2019-2023).

Jorma Laaksonen received his Dr. of Science in Technology degree in 1997 from Helsinki University of Technology, Finland, and is presently a senior university lecturer at the Department of Computer Science of the Aalto University School of Science. He is an author of 40 journal and 180 conference papers on pattern recognition, statistical classification, machine learning, and neural networks. His research interests are in content-based multimodal information analysis and retrieval and computer vision. Dr. Laaksonen is a Former Associate Editor of Pattern Recognition Letters, IEEE senior member, and a founding member of the SOM and LVQ Programming Teams and the PicSOM Development Group.