Self-Supervised Learning for Semi-Supervised Temporal Action Proposal

Xiang Wang[1] Shiwei Zhang[2] Zhiwu Qing[1] Yuanjie Shao[1]* Changxin Gao[1] Nong Sang[1]

[1]Key Laboratory of Image Processing and Intelligent Control,
School of Artificial Intelligence and Automation,
Huazhong University of Science and Technology, China
[2]DAMO Academy, Alibaba Group, China

{wxiang,qzw,shaoyuanjie,cgao,nsang}@hust.edu.cn, zhangjin.zsw@alibaba-inc.com

Abstract

*Self-supervised learning presents a remarkable performance to utilize unlabeled data for various video tasks. In this paper, we focus on applying the power of self-supervised methods to improve semi-supervised action proposal generation. Particularly, we design an effective **S**elf-supervised **S**emi-supervised **T**emporal **A**ction **P**roposal (SSTAP) framework. The SSTAP contains two crucial branches, i.e., temporal-aware semi-supervised branch and relation-aware self-supervised branch. The semi-supervised branch improves the proposal model by introducing two temporal perturbations, i.e., temporal feature shift and temporal feature flip, in the mean teacher framework. The self-supervised branch defines two pretext tasks, including masked feature reconstruction and clip-order prediction, to learn the relation of temporal clues. By this means, SSTAP can better explore unlabeled videos, and improve the discriminative abilities of learned action features. We extensively evaluate the proposed SSTAP on THUMOS14 and ActivityNet v1.3 datasets. The experimental results demonstrate that SSTAP significantly outperforms state-of-the-art semi-supervised methods and even matches fully-supervised methods. Code is available at* https://github.com/wangxiang1230/SSTAP.

1. Introduction

Temporal action proposal aims to localize action instances in untrimmed videos by predicting both action-ness probabilities and temporal boundaries. Recently, various approaches [16, 29, 27] for the task have been proposed and achieve significant progress with the quick development of spatio-temporal feature learning [40, 42, 8, 14]. Almost all the methods rely on dense temporal annotations for the training videos. However, the annotating task is tedious and requires large amounts of human labor. Thus these methods may have limited abilities to meet practical demands.

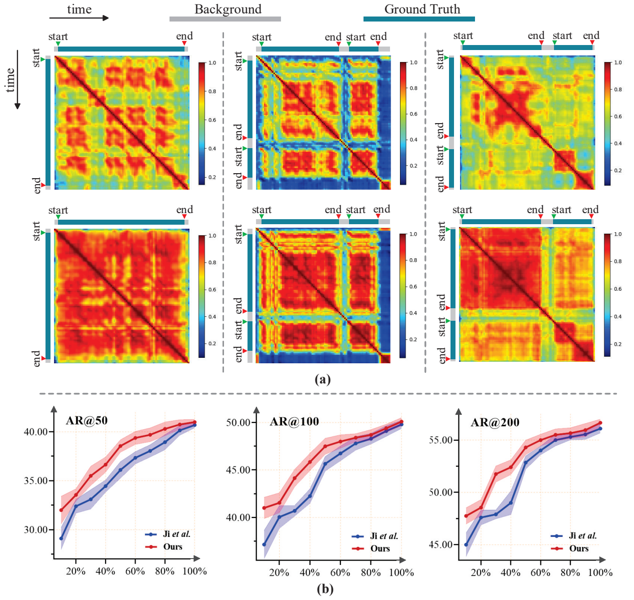

Figure 1: **(a)** Feature similarity matrix visualization. We use cosine similarity to measure the degree of similarity between arbitrary two snippet-level feature vectors within the same video. Note that, snippet-level features of the action as similar as possible while separating actions from backgrounds. Compared to Ji *et al.* [20] (top), better representations of the features can be learned by adding our relation-aware self-supervised branch (bottom). **(b)** Our SSTAP consistently exceeds the state-of-the-art semi-supervised method (Ji *et al.* [20]) in terms of Average Recall when trained with different percentages of labels on the THUMOS14 dataset.

To alleviate the dependence of labeled videos, Ji *et al.* [20] first apply the semi-supervised method, *i.e.*, Mean Teacher [41], to temporal action proposal. In this method, Ji *et al.* only use a small portion of labeled videos and

[*]Corresponding Author

reach high performances. Due to perturbation is an essential component of semi-supervised methods, the method proposes two sequential perturbations, *i.e.*, time warping and time masking, to improve robustness and generalization. However, the perturbations ignore the temporal interactions, which is critical to learn robust action representations. Another line of paradigm to utilize unlabeled videos is about self-supervised methods. These methods explore undergoing video structure by predefining pretext tasks, *e.g.*, learning temporal order [26, 50], pace prediction [44], and learning playback rate [53]. They have reached an impressive performance in several video-related tasks [43, 26, 12], and thus self-supervised learning is proved to be a promising methodology. However, the methodology has never been explored to generate temporal action proposals. We believe that it can contribute to improving the performance by fully utilizing unlabeled videos.

Based on the above observations, we propose to apply self-supervised methods to improve the semi-supervised temporal action proposal by designing the SSTAP framework. The proposed SSTAP contains two main branches, *i.e.*, temporal-aware semi-supervised branch and relation-aware self-supervised branch. The temporal-aware semi-supervised branch targets to improve the method in [20] by designing two simple but effective perturbations, *i.e.*, temporal feature shift and temporal feature flip. The first perturbation bidirectionally moves some randomly selected channels of feature maps, which is inspired by [28], and the second perturbation flips the total features, both of them along the temporal dimension. By this means, the proposal model can be more robust and generalized. In the relation-aware self-supervised branch, we define two pretext tasks, *i.e.*, masked feature reconstruction and clip-order prediction. The pretext tasks respectively reconstruct the randomly masked features and predict the correct order of the randomly shuffled clip features. Therefore, SSTAP can better explore the unlabeled videos and learns discriminative features. In Figure 1(a), the covariance-like similarity matrixes show that the self-supervised branch can help to decrease intra-class distance and increase inter-class distance simultaneously. Hence SSTAP can improve proposal performance (Figure 1(b)). We evaluate the proposed SSTAP on the challenging THUMOS14 [21] and ActivityNet v1.3 [7] datasets and achieve a remarkable improvement on both datasets.

In summary, our main contributions are as follows:

- To the best of our knowledge, we are the first to incorporate self-supervised learning in semi-supervised temporal action proposal by designing a unified SSTAP framework;
- We have designed two simple but effective types of temporal sequential perturbations and defined two types of self-supervised pretext tasks for SSTAP;
- We extensively test the proposed SSTAP on two public datasets and achieve state-of-the-art performance.

2. Related Work

Fully-Supervised Temporal Action Proposal. There are two mainstream approaches: anchor-based methods and boundary-based methods. Anchor-based methods generate proposals by designing multi-scale anchors or sliding windows. The works in [39, 13] adopt the C3D network [42] as the binary classifier for sliding window proposal evaluation. The works in [6, 5, 4] use LSTM networks to evaluate the pre-defined anchors. [17, 18, 51, 47, 9] propose to apply temporal regression to adjust the action boundaries. [16] proposes to use the complementarity of multi-scale anchors and sliding windows to improve performance. Instead, boundary-based methods evaluate each temporal location in the video. TAG [49] generates proposals by a temporal watershed algorithm to group continuous high-score regions. BSN [31] generates proposals via locally locating temporal boundaries and globally evaluating confidence scores. MGG [32] combines anchor-based methods and boundary-based methods to generate proposals. The works in [52, 1] propose to use graph convolutional networks [22] to model temporal relationships in the input video. BMN [29] proposes a boundary-matching mechanism for the confidence evaluation of densely distributed proposals in an end-to-end pipeline. BMN has become the champion method on ActivityNet Challenge 2019 [7] and the mainstream solution on ActivityNet Challenge 2020 [7]. In this work, we focus on evaluating our SSTAP with the BMN due to its superior performance.

Semi-Supervised Learning. Semi-supervised learning describes a class of algorithms that seek to learn from both unlabeled and labeled data, typically assumed to be sampled from the same or similar distributions. Approaches differ on what information to gain from the structure of the unlabeled data. In the image classification task, there are two important approaches for semi-supervised learning: pseudo-labeling and consistency regularization. Pseudo-label [25] imputes approximate classes on unlabeled data by making predictions from a model trained only on labeled data. Consistency regularization methods measure the discrepancy between predictions made perturbed data points. Approaches of this kind include Π-Model [23], Temporal ensembling [23], Mean Teacher [41], and Virtual Adversarial Training [34]. In the semi-supervised temporal action proposal task, [20] adopts the Mean Teacher framework and proposes two perturbations.

Self-Supervised Learning. Self-supervised learning is a general learning framework that relies on surrogate tasks that can be formulated using only unlabeled data. For im-

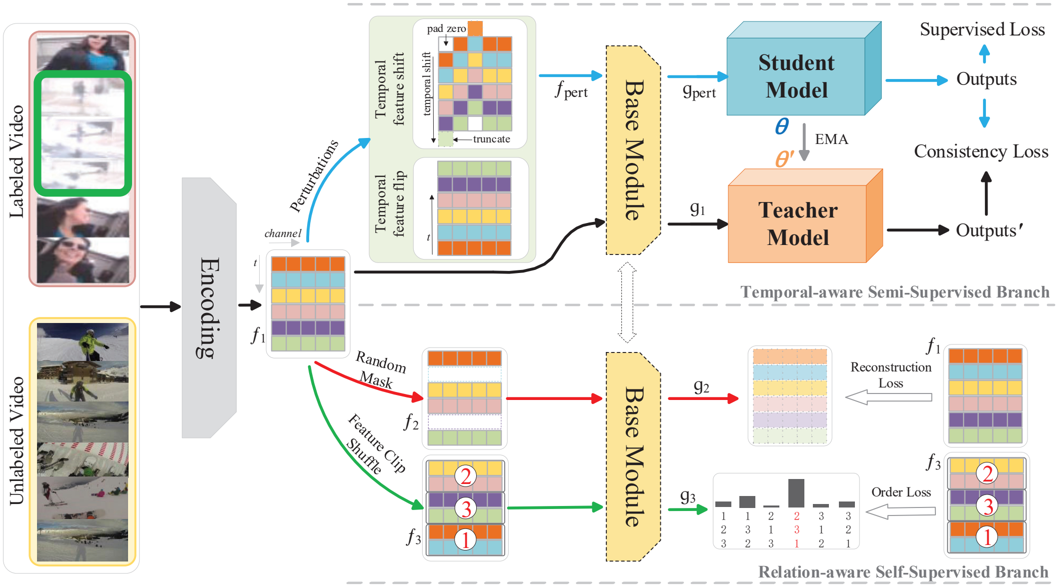

Figure 2: Overview of our SSTAP. We first encode a sampled untrimmed input video into a feature sequence f_1. In the temporal-aware semi-supervised branch (top right), there are two sequential perturbation operations: temporal feature shift and temporal feature flip. And the Base Module takes the perturbed sequences f_{pert} and the unobstructed f_1 as inputs. Next, the student model and the teacher model of the same network structure generate outputs. In the relation-aware self-supervised branch (bottom right), there are two self-supervised pretext tasks: masked feature reconstruction and clip-order prediction. In the end, a unified multi-task framework is exploited for optimization. Color-coded arrows denote the associations between the features in the framework and the respective modules.

age data, there exist self-supervised tasks such as predicting relative positions of image patches [11], jigsaw puzzles [35], image inpainting [36] and image color channel prediction [24]. Since the particular property of the video is temporal information, recent works also attempt to leverage the temporal relations among frames, such as order verification [33, 15], order prediction of frames [26, 50], and perceive multiple temporal resolutions [53].

3. SSTAP

Following the previous work [20], we build our SSTAP on top of a state-of-the-art fully-supervised proposal generation network, Boundary-Matching Network (BMN) [29]. Note that, compared with the multi-stage BSN [31] framework employed by [20], the BMN with end-to-end training can eliminate the mutual influence between multiple stages. At the same time, we also have conducted a fair comparison with [20] using the same BMN as our SSTAP. We extend the Mean Teacher [41] framework with two types of sequential perturbations, *i.e.,* temporal feature shift and temporal feature flip in the temporal-aware semi-supervised branch. And in the relation-aware self-supervised branch, two types of self-supervised auxiliary tasks, *i.e.,* masked feature reconstruction and clip-order prediction, are utilized to assist in training the proposal model. Figure 2 shows an overview of our method.

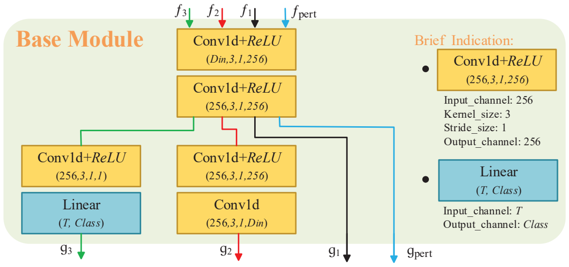

Figure 3: The details of our Base Module. Our Base Module is an extension of the "Base Module" in BMN [29]

3.1. Problem Description

Given an untrimmed video sequence $S = \{s_n\}_{n=1}^{l_s}$ with its length as l_s, our method aims at detecting action instances $\varphi_p = \{\xi_n = [t_{s,n}, t_{e,n}]\}_{n=1}^{M_s}$ with a relatively small amount of training labels, where M_s is the total number of action instances, and $[t_{s,n}, t_{e,n}]$ denotes the starting and ending points of an action instance ξ_n, respectively. Note that, classes of these action instances are not considered in the semi-supervised temporal action proposal task.

3.2. Feature Encoding

Following recent proposal generation methods [31, 29, 20, 52], we construct SSTAP framework upon visual feature sequence extracted from the raw video. Given an untrimmed video sequence $S = \{s_n\}_{n=1}^{l_s}$ with length

l_s, we first divide it into non-overlapping short snippets that contain σ frames each. Then the two-stream network [46] is adopted to extract a visual feature sequence $\phi = \{\phi_{t_n}\}_{n=1}^{T} \in \mathbb{R}^{T\times C}$, where C is the dimension of feature and $T = l_s/\sigma$.

3.3. Temporal-aware Semi-Supervised Branch

In this section, we present our temporal-aware semi-supervised branch in SSTAP. We first provide a brief description of the proposal generation network and mean teacher framework. Afterward, we introduce two types of sequential perturbations proposed by us, *i.e.,* temporal feature shift and temporal feature flip.

Proposal Generation Network. To validate our semi-supervised framework and better illustrate our approach, we build our method on top of the Boundary-Matching Network (BMN) [29] , an effective and end-to-end proposal generation method.

The same feature encoding is performed as the first step. The BMN comprises three modules: “Base Module”, “Temporal Evaluation Module” (TEM), “Proposal Evaluation Module” (PEM). “Base Module” handles the input feature sequence ϕ and outputs feature sequence ϕ' shared by the following TEM and PEM. TEM evaluates the starting and ending probabilities of each location in the video to generate boundary probability sequences. PEM contains a Boundary-Matching layer (like the ROI Pooling in Faster-RCNN [37]) to transfer the feature sequence ϕ' to a boundary-matching feature map and contains a series of 3D and 2D convolutional layers to generate boundary-matching confidence maps. The three modules are trained in a unified framework. Therefore, given an untrimmed video, BMN can simultaneously generate (1) boundary probability sequences to construct proposals and (2) boundary-matching confidence maps to evaluate the confidences of all proposals densely. Please refer to [29] for more details of BMN.

Mean Teacher Framework. In the Mean Teacher framework, there are two models: a student proposal model f_θ and a teacher proposal model $f_{\theta'}$. The student proposal model learns as in fully-supervised learning, with its weights θ optimized by the supervised losses applied on labeled videos. The teacher proposal model has the identical neural network architecture as the student, while its weights θ' are updated with an exponential moving average (EMA) of the weights from a sequence of student models of different training iterations:

$$\theta'_\tau = \alpha\theta'_{\tau-1} + (1-\alpha)\,\theta_\tau, \tag{1}$$

where τ denotes the training iteration, and α is a smoothing coefficient, which is always set to 0.999.

Sequential Perturbations. In the temporal-aware semi-supervised branch, the Mean Teacher framework is adopted on BMN to form our semi-supervised learning framework. Meanwhile, in the literature, stochastic perturbations have been found crucial for learning robust models by many semi-supervised learning works [23, 34, 41, 2, 20, 48]. And a typical way of perturbation is adding noise to feature maps. The work in [41] adds gaussian noise to intermediate feature maps of both student and teacher models. Ji *et al*. [20] add two perturbations to the input sequence. However, those perturbations ignore the temporal interactions, which is critical to temporal action proposal task. In our work, we further explore what other specific perturbations are necessary for sequential learning and propose two essential sequential perturbations: temporal feature shift and temporal feature flip.

The **temporal feature shift** perturbation is bi-directional moving some randomly selected channels on the feature map of input video along the temporal dimension (Figure 2). Therefore, temporal feature shift can significantly increase the diversity of the input features. Note that, this perturbation is inspired by [28]. The differences between [28] and temporal feature shift include: (1) that [28] chooses fixed channels (select the first $1/4$ of the feature channels, with half moving forward and the other half moving backward). While we randomly choose μ of feature channels (μ is a hyper-parameter, $\mu/2$ of feature channels move forward, and the other $\mu/2$ of channels move backward). Hence ours will add more feature diversity. And in the experiments, we observe that the method [28] can lead to a sharp decline in performance since the perturbed training features are completely misaligned compared to the testing features without perturbations. (2) the purpose of [28] is to achieve the effect of 3D convolution (*i.e.,* to capture the spatio-temporal interactive information between adjacent time points) by inserting this 2D disturbance in residual blocks for action recognition task. Our temporal feature shift serves as a way of data augmentation, providing more data for training.

Besides temporal feature shift, we propose **temporal feature flip** as another source of sequential perturbation. Since sequential video features with different perturbations may have different numbers of proposals with various locations and sizes, it is challenging to match the given video features. Therefore, the horizontally flipped video features are adopted so that one-to-one correspondence between the proposals in the original and the flipped video features can be easily aligned (Figure 2). During the training, the student models at each iteration are encouraged to generate the symmetric outputs with the teacher models.

During the training process, each mini-batch includes both labeled and unlabeled data, and we also adopt a dropout strategy to prevent overfitting. The labeled samples are trained using supervised loss. However, without ground truth labels, the supervised loss is undefined upon unlabeled videos. Consistency regularization in mean teacher framework utilizes unlabeled data based on the assumption that the model should output similar predictions when fed perturbed versions of the same input. In our temporal-aware semi-supervised branch, the consistency loss is applied to both the labeled and unlabeled data. Note that, we add consistency loss (L2-loss) to both boundary probability sequences and boundary-matching confidence maps output by BMN. Therefore, in the temporal-aware semi-supervised branch, the total loss formula is:

$$L_{semi} = L_{supervised} + \lambda_1 L_{pert\_shift} + \lambda_2 L_{pert\_flip}, \quad (2)$$

where weight terms λ_1 and λ_2 are set to 1 and 0.1 separately, L_{pert_shift} and L_{pert_flip} are consistency losses for temporal feature shift perturbation and temporal feature flip perturbation separately.

3.4. Relation-aware Self-Supervised Branch

Inspired by recent progress in self-supervised learning in video analysis [33, 15, 26, 50, 53], we hypothesize that the semi-supervised temporal action proposal method could dramatically benefit from self-supervised learning techniques. And based on this insight, in the relation-aware self-supervised branch, we propose two auxiliary tasks. The two auxiliary tasks, *i.e.,* masked feature reconstruction and clip-order prediction, can assist the network in learning temporal relations and discriminative representations.

Masked feature reconstruction. As shown in Figure 2, the key idea of this self-supervised auxiliary task is to generate the feature f_2 by randomly masking the video feature f_1 at some time points along the time dimension. The Base Module then utilizes f_2 to reconstruct f_1. The details of the Base Module are shown in Figure 3. Masked feature reconstruction produces self-supervised signals from the original feature f_1, which can learn discriminative representations in a simple-yet-effective way.

In the masked feature reconstruction auxiliary task, the Base Module will be driven to perceive and aggregate information from the context to predict the dropped snippets. In this way, the learned temporal semantic relations and discriminative features are conducive to semi-supervised temporal action proposal naturally. We use ω to represent the degree of the random mask, and we measure the effect of ω later in Section 4.3.

Clip-order prediction. This auxiliary task needs to predict clip feature sequences' correct order in a randomly scrambled feature map. Specifically, the reordering of three randomly shuffled feature sequences is shown in Figure 2. Actually, the clip-order prediction is formulated as a classification task. The input is a tuple of clip feature sequences, and the output is a probability distribution over different orders. In the experiment, we empirically designed a reordering of two randomly shuffled feature sequences. The module used for clip-order prediction is shown in Figure 3.

Clip-order prediction can leverage the chronological order of feature f_1 to learn discriminative temporal representations. And clip-order prediction is at the clip sequence level, which can reduce the uncertainty of orders and is more appropriate to learn video feature representations.

3.5. Overall Loss

The total training loss is composed of the losses from section 3.3 and section 3.4, as follows:

$$L_{total} = L_{semi} + \lambda_3 L_{aux\_recons} + \lambda_4 L_{aux\_order}, \quad (3)$$

where loss functions L_{aux_recons} and L_{aux_order} are designed for masked feature reconstruction and clip-order prediction mentioned above separately. Among them, L_{aux_recons} is L2-loss and L_{aux_order} is typical cross-entropy loss for both labeled and unlabeled data. Eventually, the final loss function L_{total} is composed of the L_{semi} in the temporal-aware semi-supervised branch and the losses in the relation-aware self-supervised branch. Hyper-parameters λ_3 and λ_4 are set to 0.0001 and 0.001 separately. To jointly learn the semi-supervised pattern and the self-supervised pattern, a unified multi-task framework is exploited for optimization in an end-to-end manner.

4. Experiments

4.1. Dataset and Setup

THUMOS14. This dataset has 1010 validation videos and 1574 testing videos with 20 classes. There are 200 validation videos and 213 testing videos labeled with temporal annotations for the action proposal or detection task. We train our model on the validation set and evaluate on the test set. To make a fair comparison with the previous works [29, 20], we employ the same two-stream features [46].

ActivityNet v1.3. This dataset is a large-scale dataset containing 19994 videos with 200 activity classes for action recognition, temporal action proposal generation and detection. The quantity ratio of training, validation, and testing sets satisfy 2:1:1. Two-stream features are employed to make a fair comparison with the previous works [29, 20]. Meanwhile, in order to show that our method is feature-agnostic, we also adopt I3D features [8] pre-trained on Kinetics [8] and without fine-tuned on ActivityNet v1.3.

We follow the same pre-processing and post-processing steps as the BMN [29], including parameters adopted in Soft-NMS [3] and network structure parameters for a fair comparison.

4.2. Temporal Action Propsal Generation

The proposal generation task's goal is to generate high-quality proposals to cover action instances with high recall and high temporal overlap. To evaluate proposal quality, Average Recall (AR) under multiple IoU thresholds

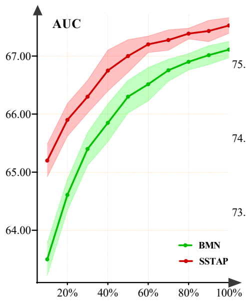

Figure 4: Varying the percentages of labels for training on ActivityNet v1.3, we compare the AUC (left) and AR@100 (right) of the proposals generated by our semi-supervised method and the fully-supervised BMN counterpart.

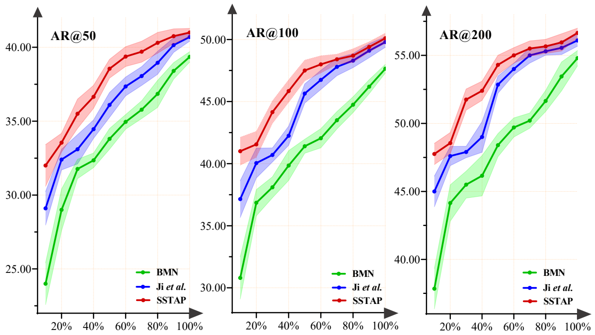

Figure 5: We compare AR@50 (left), AR@100 (middle), and AR@200 (right) of the proposals generated by and the fully-supervised BMN [29], semi-supervised Ji *et al.* [20] and our SSTAP when trained with different percentages of labels on the THUMOS14 dataset. Note that, Ji *et al.* [20] did not publish the source code. For fair comparisons, the results of Ji *et al.* [20] are carefully reproduced based on BMN [29] by us.

are calculated. Following conventions, IoU thresholds $[0.5:0.05:0.95]$ and $[0.5:0.05:1.0]$ are used for ActivityNet v1.3 and THUMOS14 respectively. We calculate AR under different Average Number of proposals (AN) as AR@AN and calculate the Area under the AR vs. AN curve (AUC) as metrics on ActivityNet v1.3, where AN is varied from 0 to 100.

Comparsions with fully-supervised methods. Like [20], we compare the temporal action proposal results under two training setups: (1) Our semi-supervised framework, where $x\%$ of training videos are labeled with temporal boundaries and $(100-x)\%$ of training videos are unlabeled; (2) Fully-supervised methods, where the same amount of labeled videos are employed for training while no other data are used. With this comparison, we can see how our semi-supervised framework performs against the fully-supervised counterpart under different training ratios.

# Method	AR@100	AUC
TCN	-	59.58
Prop-SSAD	73.01	64.40
CTAP	73.17	65.72
BSN	74.16	66.17
MGG	74.54	66.43
SSTAP@60%	**75.20**	**67.23**

(a) # Versus supervised methods.

# Method	AR@100	AUC
BMN@60%(I3D)	74.47	66.52
SSTAP@60%(I3D)	**75.00**	**67.04**
BMN@60%	74.42	66.47
SSTAP@60%	75.20	67.23
BMN@90%	74.99	67.02
SSTAP@90%	75.46	67.48
BMN@100%	75.01	67.10
SSTAP@100%	**75.54**	**67.53**

(b) # Versus BMN.

Table 1: Comparisons between our SSTAP and fully-supervised temporal action proposal generation methods on the validation set of ActivityNet v1.3 dataset in terms of AR@AN and AUC.

Feature	Method	@50	@100	@200	@1000
2-Stream	TAG [49]	18.55	29.00	39.61	-
Flow	TURN [17]	21.86	31.89	43.02	64.17
2-Stream	CTAP [16]	32.49	42.61	51.97	-
2-Stream	BSN [31]	37.46	46.06	53.21	64.52
2-Stream	MGG [32]	39.93	47.75	54.65	64.06
2-Stream	DBG [27]	37.32	46.67	54.50	66.40
2-Stream	BC-GNN [1]	40.50	49.60	56.33	66.57
2-Stream	BMN@60%	34.88	42.11	49.76	61.15
2-Stream	BMN@90%	38.45	46.31	53.36	65.29
2-Stream	BMN@100%	39.36	47.72	54.70	65.49
2-Stream	**SSTAP@60%**	**39.42**	**48.02**	**55.03**	**67.07**
2-Stream	**SSTAP@90%**	**40.12**	**49.22**	**55.86**	**68.21**
2-Stream	**SSTAP@100%**	**41.01**	**50.12**	**56.69**	**68.81**

Table 2: Comparisons between our method and fully-supervised proposal generation methods on THUMOS14 in terms of AR@AN.

We further compare our SSTAP with fully-supervised methods on the validation set of ActivityNet v1.3. Table 1 lists a set of proposal generation methods, including TCN [10], Prop-SSAD [30], CTAP [16], BSN [31], MGG [32], and BMN [29]. Specifically, with only 60% of the videos labeled, our SSTAP surpasses the fully-supervised BMN trained with all labels (100%) and other fully-supervised methods (Figure 4 and Table 1). Meanwhile, Table 1 also shows that the performance of our SSTAP can be further improved when more labels are available (*i.e.*, 90% and 100%). Our approach also performs well with I3D feature inputs, which proves that our SSTAP is feature-agnostic. Similarly, Table 2 and Figure 5 show the proposal generation performance comparisons on the testing set of THUMOS14.

Comparisons with semi-supervised baselines. Table 3 compares semi-supervised proposal generation methods on the testing set of the THUMOS14 dataset. To ensure a fair comparison, we adopt the same video feature and post-processing steps. Table 3 shows that our method using two-stream video features outperforms other semi-supervised methods significantly when the proposal number is set within $[50, 100, 200, 500, 1000]$. Especially, Figure 5 demonstrates that our SSTAP outperforms the strong semi-supervised method in Ji *et al.* [20] consistently under the different ratios of $labeled/(labeled + unlabeled)$ training

Method	Label	@50	@100	@200	@500	@1000
Vanilla BMN	10%	23.71	31.11	37.98	46.35	52.25
Mean Teacher [41]	10%	27.95	36.27	43.42	51.68	57.28
Pseudo-label [25]	10%	26.89	35.48	42.11	50.89	55.56
Ji *et al.* [20]	10%	29.10	37.43	45.07	52.67	57.96
SSTAP	**10%**	**32.33**	**40.92**	**48.27**	**54.99**	**59.38**
Vanilla BMN	60%	34.88	42.11	49.76	56.76	61.15
Mean Teacher [41]	60%	36.77	45.23	52.26	59.50	64.04
Pseudo-label [25]	60%	36.46	45.43	53.08	59.94	63.93
Ji *et al.* [20]	60%	37.42	46.71	53.96	61.01	65.10
SSTAP	**60%**	**39.42**	**48.02**	**55.03**	**62.64**	**67.07**

Table 3: Comparisons between semi-supervised baselines trained with 10% and 60% of the labels. For fair comparisons, semi-supervised baselines are all based on BMN. We report AR at various AN on THUMOS14.

Method	Label	@50	@100	@200	@500	@1000
Vanilla BMN	10%	23.71	31.11	37.98	46.35	52.25
SSTAP - F	10%	32.07	40.52	47.88	54.59	58.77
SSTAP - F - R	10%	30.82	39.24	46.85	54.31	58.71
SSTAP - F - R - C	10%	30.23	38.75	46.12	53.96	58.16
SSTAP - R - C	10%	30.80	38.96	46.31	54.28	58.23
SSTAP - S - R - C	10%	29.21	37.57	45.10	52.92	57.99
SSTAP (ALL)	**10%**	**32.33**	**40.92**	**48.27**	**54.99**	**59.38**
Vanilla BMN	60%	34.88	42.11	49.76	56.76	61.15
SSTAP - F	60%	39.26	48.00	54.95	62.07	66.65
SSTAP - F - R	60%	38.52	47.24	54.69	61.89	66.72
SSTAP - F - R - C	60%	38.04	46.71	54.35	62.17	66.51
SSTAP - R - C	60%	38.57	46.89	54.48	62.35	66.83
SSTAP - S - R - C	60%	37.44	46.86	54.07	61.23	65.21
SSTAP (ALL)	**60%**	**39.42**	**48.02**	**55.03**	**62.64**	**67.07**

Table 4: Ablation study of the effectiveness of components in our SSTAP on THUMOS14. Abbreviations: F for temporal feature flip, R for masked feature reconstruction, C for clip-order prediction, and S for temporal feature shift.

Method	Label	@50	@100	@200	@500	@1000
Vanilla BMN	10%	23.71	31.11	37.98	46.35	52.25
BMN + C	10%	26.47	34.77	41.95	49.24	54.57
BMN + R	10%	27.45	34.89	41.51	48.71	53.75
BMN + C + R	10%	28.45	36.13	42.60	49.34	54.99
Vanilla BMN	60%	34.88	42.11	49.76	56.76	61.15
BMN + C	60%	36.75	45.76	53.05	61.78	65.84
BMN + R	60%	37.14	45.83	53.17	61.20	65.75
BMN + C + R	60%	37.82	47.00	54.02	61.53	65.93

Table 5: Ablation study of the effectiveness of self-supervised branch. Abbreviations: R for masked feature reconstruction, and C for clip-order prediction.

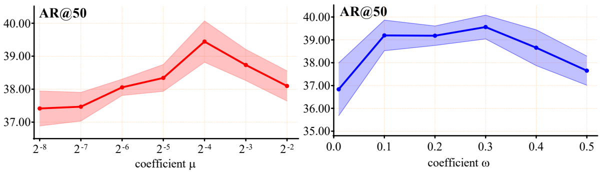

Figure 6: Ablation comparisons. The effects of temporal feature shift perturbation and masked feature reconstruction auxiliary task under different hyper-parameter choices on the THUMOS14 dataset.

videos. Unless otherwise stated, the results of Ji *et al.* [20] are all based on BMN.

4.3. Ablation Study

In this section, we present ablation studies of several components of our algorithm. We use different values of hyper-parameters that give the best result for each architectural change. The THUMOS14 dataset is employed in all studies performed in this section.

Complementarity between components. We further conduct detailed ablation studies to evaluate different components of the proposed framework, including temporal feature shift (S), temporal feature flip (F), clip-order prediction (C), and masked feature reconstruction (R). Ablation studies include the following:

Vanilla BMN : All of the above four components are discarded.

SSTAP - F : Only temporal feature flip perturbation is discarded.

SSTAP - F - R : The temporal feature flip perturbation and masked feature reconstruction auxiliary task are discarded.

SSTAP - F - R - C : The temporal feature flip perturbation and the two self-supervised auxiliary tasks in the relation-aware self-supervised branch are discarded.

SSTAP - R - C : The two self-supervised auxiliary tasks in the relation-aware self-supervised branch are discarded.

SSTAP - S - R - C : The temporal feature shift perturbation and the two self-supervised auxiliary tasks in the relation-aware self-supervised branch are discarded.

Table 4 demonstrates that the four components are complementary in terms of improving performance. In particular, when combined with the four components (*i.e.*, *SSTAP (ALL)*), the best performance is achieved. And the results for *SSTAP - F - R - C* and *SSTAP - S - R - C* show that our single perturbation also performs very well.

Effectiveness of self-supervised branch. As illustrated in Table 5, we compare the results of applying clip-order prediction (C) and masked feature reconstruction (R) directly to the original BMN. That shows the effectiveness of the two self-supervised auxiliary tasks for performance improvement. Note that, both labeled and unlabeled data are used for training the auxiliary tasks.

Selection of hyper-parameters. Figure 6 illustrates the comparison of the selection of hyper-parameters. It can be observed that the adjustment of parameters has a certain effect on the performance of AR@50 on THUMOS14, meanwhile, $\mu = 2^{-4}$ and $\omega = 0.3$ appear to be the optimal operating points.

4.4. Action Detection with Our Proposals

To further examine the quality of the proposals generated by SSTAP, we put the proposals in a temporal action detection framework. The evaluation metric of temporal action detection is mAP, which calculates the Average Precision under multiple IoU thresholds for each action cate-

Method	Reference	0.5	0.75	0.95	Average
SCC [19]	CVPR'17	40.00	17.90	4.70	21.70
CDC [38]	CVPR'17	45.30	26.00	0.20	23.80
R-C3D [51]	ICCV'17	26.80	-	-	-
BSN [31]+ [55]	ECCV'18	46.45	29.96	8.02	30.03
TAL-Net [9]	CVPR'18	38.23	18.30	1.30	20.22
P-GCN [54]	ICCV'19	48.26	33.16	3.27	31.11
G-TAD [52]+ [55]	CVPR'20	50.36	34.60	9.02	34.09
BC-GNN [1]+ [55]	ECCV'20	50.56	34.75	**9.37**	34.26
BMN@60%+ [55]	ICCV'19	49.50	33.68	8.15	33.17
BMN@90%+ [55]	ICCV'19	49.94	33.73	8.23	33.74
BMN@100%+ [55]	ICCV'19	50.07	34.78	8.29	33.85
Ji *et al.* @60%+ [55]	ICCV'19	49.82	34.53	7.01	33.52
Ji *et al.* @90%+ [55]	ICCV'19	50.24	34.97	7.35	34.13
Ji *et al.* @100%+ [55]	ICCV'19	50.55	35.01	7.58	34.23
SSTAP@60%+ [55]	-	50.14	34.92	7.43	34.01
SSTAP@90%+ [55]	-	50.64	35.12	7.80	34.35
SSTAP@100%+ [55]	-	**50.72**	**35.28**	7.87	**34.48**

Table 6: Action detection results on the validation set of ActivityNet v1.3, where our proposals are combined with video-level classification results generated by [55].

Method	Reference	0.7	0.6	0.5	0.4	0.3
SST [5]+UNet	CVPR'17	4.7	10.9	20.0	31.5	41.2
TURN [17]+UNet	ICCV'17	6.3	14.1	24.5	35.3	46.3
BSN [31]+UNet	ECCV'18	20.0	28.4	36.9	45.0	53.5
MGG [32]+UNet	CVPR'19	21.3	29.5	37.4	46.8	53.9
DBG [27]+UNet	AAAI'20	21.7	30.2	39.8	49.4	57.8
G-TAD [52]+UNet	CVPR'20	**23.4**	30.8	40.2	47.6	54.5
BC-GNN [1]+UNet	ECCV'20	23.1	31.2	40.4	49.1	57.1
BMN@60%+UNet	ICCV'19	17.0	25.5	34.0	44.7	53.4
BMN@90%+UNet	ICCV'19	19.7	28.9	38.2	46.8	55.5
BMN@100%+UNet	ICCV'19	20.5	29.7	38.8	47.4	56.0
Ji *et al.* @60%+UNet	ICCV'19	19.2	28.1	37.1	47.2	55.4
Ji *et al.* @90%+UNet	ICCV'19	21.5	31.6	41.2	50.6	57.2
Ji *et al.* @100%+UNet	ICCV'19	21.9	32.2	41.7	51.2	57.9
SSTAP@60%+UNet	-	20.7	30.5	39.4	48.8	56.5
SSTAP@90%+UNet	-	22.1	32.3	41.9	51.2	57.8
SSTAP@100%+UNet	-	22.8	**32.8**	**42.3**	**51.5**	**58.4**

Table 7: Action detection results on the testing set of THUMOS14 in terms of mAP@tIoU. We compare with "proposal + classification" methods, where classification results are generated by UntrimmedNet [45].

Method	Label	0.7	0.6	0.5	0.4	0.3
Vanilla G-TAD	10%	6.8	12.6	20.4	28.5	37.3
Ji *et al.* [20]+G-TAD	10%	9.5	17.4	25.8	34.4	43.4
SSTAP+G-TAD	**10%**	**11.1**	**18.4**	**27.6**	**35.9**	**45.5**
Vanilla G-TAD	60%	16.5	25.3	35.4	44.8	50.9
Ji *et al.* [20]+G-TAD	60%	20.1	29.4	39.6	47.5	53.8
SSTAP+G-TAD	**60%**	**21.8**	**31.1**	**41.4**	**50.2**	**56.3**
Vanilla G-TAD	100%	**23.4**	30.8	40.2	47.6	54.5
Ji *et al.* [20]+G-TAD	100%	21.3	31.3	41.2	49.6	55.3
SSTAP+G-TAD	**100%**	22.6	**32.4**	**42.7**	**51.3**	**57.0**

Table 8: Generalizing our SSTAP to G-TAD [52] in terms of mAP@tIoU on THUMOS14. The comparison experiments all use the same two-stream feature [46] as in G-TAD [52].

gory. On ActivityNet v1.3, the IoU thresholds for mAP are set to $\{0.5, 0.75, 0.95\}$, and the IoU thresholds for average mAP are set to $[0.5 : 0.05 : 0.95]$. On THUMOS14, the IoU thresholds for mAP are set to $\{0.3, 0.4, 0.5, 0.6, 0.7\}$.

We adopt the two-stage "detection by classifying proposals" temporal action detection framework to combine our proposals with action classifiers. For fair comparisons, following [31, 29, 52, 1], on ActivityNet v1.3, we adopt top-1 video-level classification results generated by method [55] and use confidence scores of BMN proposals for detection results retrieving. On THUMOS14, following BMN [29], we also use both top-2 video-level classification results generated by UntrimmedNet [45]. And the same classifiers are also used for other proposal generation methods, including SST [5], TURN [17], BSN [31], MGG [32], DBG [27], G-TAD [52], and BC-GNN [1].

Table 6 illustrates the performance comparisons, which are evaluated on the testing set of THUMOS14. With only 60% of the videos labeled, our SSTAP achieves better performance than fully-supervised BMN trained with all labels in metrics of average mAP. Especially, with 100% of the videos labeled, our SSTAP outperforms the fully-supervised proposal methods, namely BMN [29], G-TAD [52], BC-GNN [1], and Ji *et al.* [20]. Similar results on THUMOS14 are shown in Table 7, thus demonstrating the effectiveness of our proposed SSTAP.

4.5. Generalization Experiments

To prove the SSTAP method is valid for other network architectures and frameworks, we introduce SSTAP to G-TAD [52]. G-TAD proposes to use graph convolutional networks [22] to model temporal relationships between each time point in the input video. As illustrated in Table 8, introducing SSTAP to G-TAD also improves performance. In particular, our SSTAP outperforms the strong semi-supervised baseline [20] by a large margin.

5. Conclusion

In this paper, we incorporate self-supervised learning in the semi-supervised temporal action proposal task and propose a unified SSTAP framework. Specially, we have designed two simple but effective types of temporal sequential perturbations and defined two types of self-supervised pretext tasks for SSTAP. We show empirically that SSTAP consistently outperforms the state-of-the-art semi-supervised methods and even matches the fully-supervised methods. Furthermore, we indicate that our SSTAP is agnostic to specific proposal methods and can be effectively applied to other temporal action proposal approaches.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grant (61871435, 61901184), and the Fundamental Research Funds for the Central Universities no.2019kfyXKJC024.

References

[1] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. *ECCV*, 2020. 2, 6, 8

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NIPS*, pages 5049–5059, 2019. 4

[3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, pages 5561–5569, 2017. 5

[4] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. 2019. 2

[5] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *CVPR*, pages 2911–2920, 2017. 2, 8

[6] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *CVPR*, pages 1914–1923, 2016. 2

[7] 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. 2

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 1, 5

[9] 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*, pages 1130–1139, 2018. 2, 8

[10] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. In *ICCV*, pages 5793–5802, 2017. 6

[11] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 3

[12] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *CVPR*, pages 1801–1810, 2019. 2

[13] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *ECCV*, pages 768–784. Springer, 2016. 2

[14] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *ICCV*, pages 6201–6210, 2019. 1

[15] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, pages 3636–3645, 2017. 3, 5

[16] Jiyang Gao, Kan Chen, and Ram Nevatia. Ctap: Complementary temporal action proposal generation. In *ECCV*, pages 68–83, 2018. 1, 2, 6

[17] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, pages 3628–3636, 2017. 2, 6, 8

[18] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. *arXiv preprint arXiv:1705.01180*, 2017. 2

[19] Fabian Caba Heilbron, Wayner Barrios, Victor Escorcia, and Bernard Ghanem. Scc: Semantic context cascade for efficient action detection. In *CVPR*, pages 3175–3184. IEEE, 2017. 8

[20] Jingwei Ji, Kaidi Cao, and Juan Carlos Niebles. Learning temporal action proposals with fewer labels. In *ICCV*, pages 7073–7082, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[21] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014. 2

[22] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2, 8

[23] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 2, 4

[24] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, pages 6874–6883, 2017. 3

[25] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICMLW*, volume 3, 2013. 2, 7

[26] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, pages 667–676, 2017. 2, 3, 5

[27] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *AAAI*, pages 11499–11506, 2020. 1, 6, 8

[28] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, pages 7083–7093, 2019. 2, 4

[29] 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, 4, 5, 6, 8

[30] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *ACMMM*, pages 988–996, 2017. 6

[31] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, pages 3–19, 2018. 2, 3, 6, 8

[32] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *CVPR*, pages 3604–3613, 2019. 2, 6, 8

[33] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order

verification. In *ECCV*, pages 527–544. Springer, 2016. 3, 5

[34] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 2, 4

[35] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016. 3

[36] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 3

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 4

[38] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, pages 5734–5743, 2017. 8

[39] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, pages 1049–1058, 2016. 2

[40] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *NeurIPS*, 1, 2014. 1

[41] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, pages 1195–1204, 2017. 1, 2, 3, 4, 7

[42] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 1, 2

[43] Michael Tschannen, Josip Djolonga, Marvin Ritter, Aravindh Mahendran, Neil Houlsby, Sylvain Gelly, and Mario Lucic. Self-supervised learning of video-induced visual invariances. In *CVPR*, pages 13806–13815, 2020. 2

[44] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. *arXiv preprint arXiv:2008.05861*, 2020. 2

[45] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, pages 4325–4334, 2017. 8

[46] 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. 4, 5, 8

[47] Xiang Wang, Changxin Gao, Shiwei Zhang, and Nong Sang. Multi-level temporal pyramid network for action detection. In *PRCV*, pages 41–54. Springer, 2020. 2

[48] Xiao Wang, Daisuke Kihara, Jiebo Luo, and Guo-Jun Qi. Enaet: Self-trained ensemble autoencoding transformations for semi-supervised learning. *arXiv preprint arXiv:1911.09265*, 2019. 4

[49] Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaoou Tang. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716*, 2017. 2, 6

[50] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, pages 10334–10343, 2019. 2, 3, 5

[51] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, pages 5783–5792, 2017. 2, 8

[52] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, pages 10156–10165, 2020. 2, 3, 8

[53] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *CVPR*, pages 6548–6557, 2020. 2, 3, 5

[54] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, pages 7094–7103, 2019. 8

[55] Y Zhao, B Zhang, Z Wu, S Yang, L Zhou, S Yan, L Wang, Y Xiong, D Lin, Y Qiao, et al. Cuhk & ethz & siat submission to activitynet challenge 2017. *arXiv preprint arXiv:1710.08011*, 8, 2017. 8