Semi-supervised Learning for Multi-label Video Action Detection

Hongcheng Zhang
absolutezh@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Xu Zhao*
zhaoxu@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Dongqi Wang
wangdq0124@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

ABSTRACT

Semi-supervised multi-label video action detection aims to locate all the persons and recognize their multiple action labels by leveraging both labeled and unlabeled videos. Compared to the single-label scenario, semi-supervised learning in multi-label video action detection is more challenging due to two significant issues: generation of multiple pseudo labels and class-imbalanced data distribution. In this paper, we propose an effective semi-supervised learning method to tackle these challenges. Firstly, to make full use of the informative unlabeled data for better training, we design an effective multiple pseudo labeling strategy by setting dynamic learnable threshold for each class. Secondly, to handle the long-tailed distribution for each class, we propose the unlabeled class balancing strategy. We select training samples according to the multiple pseudo labels generated during the training iteration, instead of the usual data re-sampling that requires label information before training. Then the balanced re-weighting is leveraged to mitigate the class imbalance caused by multi-label co-occurrence. Extensive experiments conducted on two challenging benchmarks, AVA and UCF101-24, demonstrate the effectiveness of our proposed designs. By using the unlabeled data effectively, our method achieves the state-of-the-art performance in video action detection on both AVA and UCF101-24 datasets. Besides, it can still achieve competitive performance compared with fully-supervised methods when using limited annotations on AVA dataset.

CCS CONCEPTS

• **Computing methodologies** → *Activity recognition and understanding*.

KEYWORDS

Semi-supervised learning, video understanding, action detection, multi-label classification

ACM Reference Format:
Hongcheng Zhang, Xu Zhao, and Dongqi Wang. 2022. Semi-supervised Learning for Multi-label Video Action Detection. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), Oct. 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3503161.3547980

(a) Left: bend/bow, take a photo, hold (an object); Right: bend/bow, watch

(b) Left: stand, read, listen to; Right: stand, talk to

Figure 1: Action instances with multiple action labels

1 INTRODUCTION

Video action detection aims to detect the bounding boxes for all the persons in an input video and estimate their action labels. The training process of fully-supervised action detection methods [12, 18, 20, 29, 35, 39] relies on a large amount of manually annotated training data and thus the performance of the model depends on the quality and quantity of annotations. However, it is time-consuming and cost-intensive to obtain large datasets for video action detection like AVA [14] and AVA-Kinetics [24], as spatio-temporal annotation is required on each clip of the whole video.

Hence, it is noteworthy that the numerous unlabeled videos containing human actions in real life can promote the training of action detection models. Compared with fully-supervised methods that relies entirely on labeled data, the semi-supervised method can take full advantage of both labeled and unlabeled videos and boost the performance of action detectors. Furthermore, the fully-supervised action detection approaches will suffer performance degradation when the amount of labeled data is limited, while semi-supervised action detection can still achieve competitive performance by leveraging the unlabeled data effectively.

In action detection task, as shown in Fig.1, the action instance (actor) in a short clip from labeled dataset (such as AVA) or unlabeled real-world videos rarely belongs to a single action category but is usually associated with multiple labels. Therefore, unlike the previous work [21] that performs semi-supervised action detection on videos which merely contain single action label, we focus on semi-supervised learning for multi-label video action detection, which is more challenging and has greater practical prospects. However, it is inappropriate to directly apply the usual semi-supervised learning methods like [33, 46] to the multi-label action detection task. Because semi-supervised video action detection in the multi-label scenario has two major challenges: one is the generation of multiple pseudo labels, the other is the class-imbalanced data distribution in

*indicates corresponding author.

MM '22, October 10–14, 2022, Lisboa, Portugal.

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3547980

both the labeled and unlabeled videos. These two problems are usually coupled together, which bring difficulties to semi-supervised multi-label video action detection.

For the first problem, semi-supervised multi-label action detection requires the generation of multiple pseudo labels for the unlabeled data. However, simply pursuing high pseudo-label quality by setting fixed high thresholds for all classes like methods [2, 3, 33, 41, 42, 46] in semi-supervised image classification will be detrimental to the performance. Because the minority classes tend to have lower prediction scores due to the insufficient feature learning caused by the class imbalance issue, while the prediction scores of majority classes are higher. If a high threshold is set for all classes, the generated pseudo labels will be inclined to the majority, which will exacerbate the problem of class imbalance. Inspired by Flexmatch [46], we propose the multiple pseudo labeling method, which generates multiple pseudo labels by setting dynamic thresholds according to the class distributions. This method increases the number of positive samples for minority classes and thus can leverage more informative unlabeled data for the training.

For the second problem, the class-imbalanced data distribution is common in action videos. For example, actions such as *stand* and *walk* appear more frequently in both the dataset and real-world videos, while actions such as *fall down* and *hit* appear with lower frequency, as shown in Fig.3. Both labeled and unlabeled videos suffer from the class imbalance issue, which makes the action classifier bias towards the majority categories during the training. And the skewed pseudo labels generated from the classifier will be detrimental to semi-supervised learning process. In addition, the impact of multi-label should be considered when performing the class-balanced strategy. It is inappropriate to directly adopt the class-balanced data re-sampling method used by [15, 19, 40, 47], because it requires category information in advance, which is not available for unlabeled videos. And the pseudo labels for unlabeled data should be generated before the training iteration, which will lead to low quality of pseudo labels and bring additional cost. To address the imbalance issue, we propose a class-balanced method for unlabeled data, named as unlabeled class balancing.

We demonstrate the effectiveness of our method with extensive experiments on AVA [14] and UCF101-24 [34] datasets. Our semi-supervised learning method outperforms the fully-supervised methods by using the labeled and unlabeled data effectively. Besides, with limited annotations, our method can achieve competitive performance when compared with fully-supervised methods with 100% annotations. Our contributions are summarized as follows.

- For multiple pseudo label generation, we propose the multiple pseudo labeling strategy by setting dynamic threshold for each class, which can leverage more informative unlabeled data for better training.
- To tackle the class imbalance issue, we propose unlabeled class balancing. It first samples the unlabeled data during the training iteration instead of the usual data re-sampling, and then perform balanced re-weighting to mitigate the class imbalance caused by multi-label co-occurrence.
- Experiments conducted on AVA and UCF101-24 datasets demonstrate the effectiveness of our designs. Our method outperforms the fully-supervised action detection methods and can still achieve competitive performance with limited annotations. And to the best of our knowledge, this is the first attempt on semi-supervised learning in the task of multi-label video action detection.

2 RELATED WORK

2.1 Video Action Detection

Video action detection [8, 12, 18, 20, 25, 29, 35, 38, 39] has evolved rapidly in recent years, due to the development of convolutional neural networks and available high-quality datasets. Datasets (such as AVA and AVA-Kinetics) are annotated with atomic actions for all action instances in the video. The action instances are annotated both spatially and temporally and are usually associated with multiple action labels. Typical action detection methods [12, 29, 35, 38, 39] extended object detectors on 3D-CNN features to handle videos. The bounding boxes of action instances were first predicted by person detector and then labeled with some action classes. Recently, several approaches [26, 28, 29, 35, 38, 39] focused on leveraging context information and modeling the relations between actors to improve recognizing human action. In this work, following [12, 29, 35, 38, 39], we leverage a person detector to generate the bounding boxes, and utilize the SlowFast [12] as our action detection network.

2.2 Class-imbalanced learning

For the multi-label action detection task, the class-imbalanced distribution of data is common in both labeled and unlabeled videos, which poses an great challenge to this task. Most existing approaches for class-imbalanced learning can be divided into two categories: re-sampling [4, 5, 10, 31] and re-weighting [6, 10, 17, 37]. In the category of re-sampling, under-sampling the majority classes [4, 5, 31] and over-sampling the minority classes [4, 10] are two general strategies. For re-weighting, some researchers set the weight to be inversely proportional to the class frequency [6, 10, 17, 37]. As for the class-imbalanced learning on multi-label classification, [40] first performs re-sampling on the dataset, and then uses re-weighting to handle the imbalance caused by multi-label co-occurrence.

2.3 Semi-Supervised learning

Semi-supervised learning (SSL) has attracted increasing attention in recent years due to its superiority in utilizing both labeled and unlabeled data. There are two powerful techniques for semi-supervised learning, consistency regularization [1, 22, 36] and pseudo labeling [23, 41, 42]. FixMatch [33] achieved competitive performance by combining these techniques with weak and strong data augmentations and using cross-entropy loss for consistency regularization. Flexmatch [46] pointed out that, it is not optimal to generate pseudo labels from unlabeled data by setting fixed threshold, which limits the performance of FixMatch and other pseudo-label methods. For semi-supervised video action detection, [21] trains the network with consistency-based regularization. However, this method is only applicable to those simple videos which merely contain single action label, and is not suitable for handling the scene with multiple actions in videos. Thus, in this work, we aim at tackling this problem. To the best of our knowledge, this is the first attempt to perform semi-supervised learning on action detection task in the multi-label scenario.

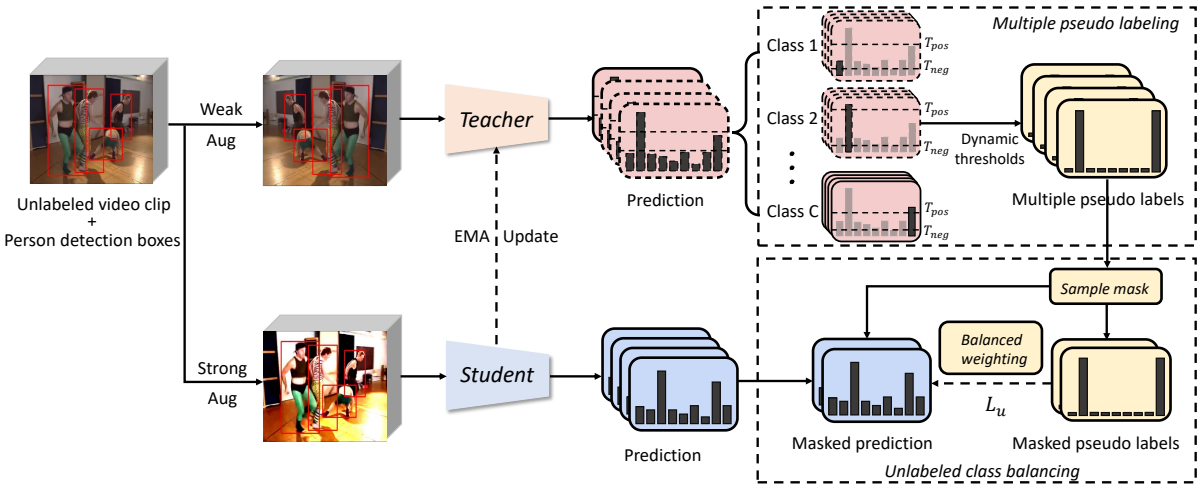


Figure 2: An overview of our proposed method. It contains a student network and a teacher network, where the teacher is momentum-updated with the student. Given an unlabeled video clip, we first use the teacher model to make a prediction and generate the multiple pseudo labels by setting dynamic thresholds for each class. Then to handle the class imbalance issue, we perform unlabeled class balancing. With sample mask generated from multiple pseudo labels, we select action instances from student's prediction to compute L_u via a balanced re-weighting *BCE* loss.

3 METHOD

3.1 Overall Pipeline

Spatio-temporal action detection aims to locate all persons in the input video clip and predict their action labels. Following the typical action detection framework [29, 35, 38, 39], we leverage an off-the-shelf person detector to obtain N_{act} detection boxes on the key frame (center frame) from the input clip. Then, based on the detected boxes and the input clip of T frames $x \in R^{T\times 3\times H\times W}$, the action model produces spatio-temporal features and outputs the final action predictions $pred \in R^{N_{act}\times C}$, where C, H, W are class numbers, height and width respectively. We denote the number of training samples containing class j from both labeled and unlabeled data as n_l^j, n_u^j, respectively. Without loss of generality, we assume that the classes are sorted in descending order, i.e., $n^1 \geq n^2 \geq ... \geq n^C$ for both labeled and unlabeled data.

Given a labeled video clip dataset $X_l = \{(x_l^i, y_l^i)\}_{i=1}^{N_l}$ and unlabeled video clip dataset $X_u = \{x_u^i\}_{i=1}^{N_u}$, where y_l^i denotes the action labels for the video clip x_l^i, we want to train the action detection model by utilizing both the labeled and unlabeled data. Our method follows the typical semi-supervised framework [2, 3, 33, 46] with a teacher model $f_t(\cdot)$ and a student model $f_s(\cdot)$ with the same architecture. Receiving the unlabeled input clip with weak augmentation, the teacher model makes the prediction $pred_t^u = f_t(Aug_w(x_u))$. While the student model makes the predictions from both the labeled input clip with weak augmentation and the unlabeled input clip with strong augmentation, as shown in Eq.(1).

$$\begin{aligned} pred_s^l &= f_s(Aug_w(x_l)) \\ pred_s^u &= f_s(Aug_s(x_u)) \end{aligned} \tag{1}$$

Then the student model is trained by the *BCE* (binary cross entropy) loss applied on both the labeled clips and the unlabeled clips with pseudo action labels. And the weights of teacher model is an exponential moving average (EMA) of the student model's weights. However, different from the semi-supervised methods in image classification [2, 3, 33, 46], semi-supervised action detection mainly faces two problems in multi-label scenario, generation of multiple pseudo labels and class-imbalanced data distribution.

The first is the generation of pseudo labels. Unlike the semi-supervised learning methods [21, 33, 46] in single-label scenario, we need to generate multiple pseudo labels from the teacher model's prediction for multi-label action detection. Most semi-supervised methods [2, 3, 21, 33] determine the pseudo label by setting a fixed threshold for all categories, which we believe is not optimal. Because this setting takes neither the learning status of each class nor the class-imbalanced data distribution into consideration. Therefore, we propose a pseudo-label generation method based on the multi-label scenario, as described in Sec 3.2. It generates multiple pseudo labels by setting learnable thresholds which can be dynamically adjusted for each class during the training iteration.

Secondly, the detection performance is also hindered by the class-imbalanced distribution, which exists in both labeled and unlabeled datasets, as shown in Fig.3. The class imbalance issue will make the model biased towards the majority classes and limit the performance of the minority classes. This phenomenon becomes severe when we perform semi-supervised learning on action detection, because the biased teacher model pre-trained from imbalanced data will produce skewed action predictions on the unlabeled input clips. And the pseudo labels generated from the teacher model will suffer a more imbalanced distribution, thus undermining the performance of semi-supervised learning. However, the class balanced methods [15, 40, 45] will encounter difficulties when performing data re-sampling with unlabeled data. Since data re-sampling requires the class information for all unlabeled data before training,

an additional inference process is required to generate the corresponding pseudo labels for the unlabeled data. This process will lead to inaccurate pseudo labels and additional cost.

To tackle these problems, we propose an unlabeled class balancing strategy, which avoids to generate labels for all unlabeled data in advance. It firstly generates an instance level mask obeying *Bernoulli* distribution to perform instance level sampling during the training iteration instead of data re-sampling. And then we perform balanced weighting to alleviate the imbalance effect caused by multi-label co-occurrence. The overview of our method is shown in Fig.2, and we will elaborate the details in the following sections.

3.2 Multiple pseudo labeling

Many methods [2, 3, 33, 41] determine the category of pseudo labels by simply taking the argmax of softmax probabilities. However, in the scenario of multi-label action detection, one action instance is usually associated with several action categories, which makes the generation of pseudo labels challenging. Our target is to generate multiple pseudo labels $y_u \in \{0,1\}^{N_{act} \times C}$ by the sigmoid prediction $pred_t \in R^{N_{act} \times C}$ from the teacher model classification head for each class. An intuitive design for generating multiple pseudo label is to set a threshold to select the highly confident predictions as positives and those below the threshold as negatives. However, this design is not conducive to the model's feature learning, because some positive predictions below the threshold may be mistakenly regarded as negatives. Therefore, we contend that thresholds should be set for both positives and negatives, as shown in Eq.(2). If the prediction score from the i-th action instance and j-th class is greater than T^j_{pos}, it will be regarded as positive, where $i = 1, 2, ..., N_{act}$ and $j = 1, 2, ..., C$. Otherwise it will be regarded as negative if the score is less than T^j_{neg}. We believe the rest are unreliable for pseudo labeling, and we ignore them in the training.

$$y_u^{i,j} = \begin{cases} 1, & pred_t^{i,j} > T^j_{pos} \\ 0, & pred_t^{i,j} < T^j_{neg} \\ ignore & otherwise \end{cases} \tag{2}$$

We implement this process by setting weights for the *BCE* (binary cross entropy) loss, we set the weight of unreliable predictions to 0, and the weight of positives and negatives to 1, as shown below.

$$w_{pos}^{i,j} = w_{neg}^{i,j} = \begin{cases} 1, & pred_t^{i,j} > T^j_{pos} \text{ or } pred_t^{i,j} < T^j_{neg} \\ 0, & otherwise \end{cases} \tag{3}$$

And the weighted *BCE* loss is shown as Eq.(4).

$$l_{cls}^{i,j} = -(w_{pos}^{i,j} y_u^{i,j} \log(pred_s^{i,j}) + w_{neg}^{i,j}(1 - y_u^{i,j}) \log(1 - pred_s^{i,j})) \tag{4}$$

Based on this setting, we further investigate reasonable threshold designs for multiple pseudo-label generation. We propose three methods as follows.

3.2.1 Fixed threshold. We set fixed thresholds for each class, so the positive and negative thresholds are denoted as $T^j_{pos} = \tau_{pos}$ and $T^j_{neg} = \tau_{neg}$, $j = 1, 2, ..., C$, respectively.

3.2.2 Class-related threshold. We further consider the category information in the threshold setting for the class-imbalanced data distribution. Intuitively, the majority categories have a high classification accuracy while the actions belonging to minority categories are difficult to recognize, due to the insufficient feature learning. Setting a fixed high threshold for all classes is not optimal, since it will produce fewer positives of minority categories and thus exacerbate the imbalance issue. Therefore, we set the positive threshold for each class according to the class imbalance ratio $\eta^j_u = n^j_u / n^1_u$, which ranges between 0 to 1. And we use a non-linear mapping function $M(\cdot)$ on the imbalance ratio. $M(\eta^j_u)$ also ranges between 0 to 1 and then will be leveraged to scale the positive threshold τ_{pos}, as shown in Eq.(5). And for the negatives, we set $T^j_{neg} = 0.1$ for all categories.

$$T^j_{pos} = M(\eta^j_u) \cdot \tau_{pos} \tag{5}$$

3.2.3 Learnable threshold. To dynamically adjust the threshold during the training process, we set T^j_{pos} as a learnable parameter. We set the weight for *BCE* loss as a function related to T^j_{pos}, so that T^j_{pos} can be updated in each training iteration, and the T^j_{pos} is initialized with the value calculated by Eq.(5). We design a sigmoid function for the weight, as shown in Eq.(18).

$$w_{pos}^{i,j} = \frac{1}{1 + \exp(-a(pred_t^{i,j} - T^j_{pos}))} \tag{6}$$

Besides, a regularization term is added to the classification loss to prevent the value of T^j_{pos} from being too large, as shown in Eq.(19). The details of the regularization term are listed in the supplementary material.

$$l_r^{i,j} = -\log T^j_{pos} + 2T^j_{pos} \tag{7}$$

3.3 Unlabeled class balancing

Multi-label action detection on both labeled and unlabeled data suffers from severe class imbalance issue as shown in Fig.3, which makes the prediction of the model biased towards the majority classes. The common solution is to adopt a class balancing strategy in multi-label scenario, such as [15, 40]. They firstly perform class balanced data re-sampling and then use re-balanced weighting to alleviate the class imbalance caused by multi-label co-occurrence. However, it is inappropriate to perform data re-sampling with unlabeled data as mentioned in Sec.3.1.

We propose the unlabeled class balancing strategy for semi-supervised multi-label action detection. Different from the usual data re-sampling strategy, we first sample action instances dynamically according to their pseudo labels in each training iteration on the unlabeled data. And then we alleviate the imbalance caused by multi-label co-occurrence via re-weighting. To perform balanced sampling for each class, we expect to sample the same number of action instances for each category, which is denoted as $\overline{n_u} = \frac{1}{C} \sum_{j=1}^{C} n^j_u$. Then, we select action instances assigned as class j with probability p^j during each training iteration by using mask, where the mask for the i-th action instance $m^i \in \{0, 1\}$ follows a *Bernoulli* distribution with probability p^j, as shown in Eq.(8).

$$m^i \sim Bernoulli(1, p^j), i \in \{1, ..., N_{act}\} \tag{8}$$

To balance the class distribution of action instances, we design the function of probability p^j as shown in Eq.(9), we under-sample the majority classes and sample as many action instances in the

minority classes as possible. And it is worth noting that, the sample probability of some minority classes is set to 1, since over-sampling is not used for the unlabeled data.

$$p^j = \begin{cases} (\frac{\overline{n}_u}{n_u^j})^\beta, & n_u^j > \overline{n}_u, \beta > 0 \\ 1, & otherwise \end{cases} \tag{9}$$

To perform the sampling method mentioned above, we need to assign action instance to a specific class according to their multiple pseudo labels generated in Sec.3.2. The class with the minimum η_u^j among the positive multiple pseudo labels will be treated as the class of the action instance, which is formulated as Eq.(10). In the multi-label scenario, one action instance is usually associated with majority and minority classes simultaneously, and this design will assign the action instance to the minority class. Thus we can sample the action instance with higher probability, which increases the number of minority classes in the training process.

$$k = \arg\min_{j \in J}(\eta_u^j), J = \{j | y_u^j = 1, j = 1, ..., C\} \tag{10}$$

However, this strategy will inevitably affect the sample numbers of the other classes due to multi-label co-occurrence when classifying one action instance into a specific category. Therefore, the class imbalance issue is not completely eliminated after sampling, as show in Fig.3, and a re-weighting procedure is needed. Based on the unlabeled sampling strategy mentioned above, the number of instances containing class j to be sampled during training is $\sum_{k=1}^{C} n_u^{j,k} p^k$, where $n_u^{j,k}$ denotes the number of action instances containing class j but assigned as class k. According to Eq.(10), $n_u^{j,k} = 0$ when $j < k$ and $n_u^{j,j} = n_u^j - \sum_{k=j+1}^{C} n_u^{j \cap k}$, where $n_u^{j \cap k}$ denotes the number of action instances containing both class j and k. Then the numbers of sampled action instances containing class j can be formulated as Eq.(11).

$$s_u^j = n_u^j p^j + \sum_{k=j+1}^{C} n_u^{j \cap k}(p^k - p^j) \tag{11}$$

$$g_u^j = \frac{s_u^j}{\overline{n_u}} = \frac{p^j + \sum_{k=j+1}^{C} \frac{\eta_u^{j \cap k}}{\eta_u^j}(p^k - p^j)}{\frac{1}{C}\sum_{r=1}^{C} \frac{\eta_u^r}{\eta_u^j}} \tag{12}$$

The gap coefficient between the actual instance-level sampling numbers and our class-balanced expectation can be denote as Eq.(12). The balanced weight is formulated as Eq.(13), which is used for calculating unsupervised loss as shown in Sec.3.4. And if we set hyper parameters $\beta = 1$ and $\gamma = 1$, the whole pipeline based on unlabeled balancing strategy could be approximately considered as class-balanced.

$$w_g^j = (g_u^j)^{-\gamma} \tag{13}$$

It is worth noting that obtaining the actual class distribution η_u for the unlabeled data requires an inference process, which will incur additional cost and the generated distribution may be inaccurate. We assume that pseudo labels generated from unlabeled data and ground truth labels from labeled data have similar distributions, as shown in Fig.3. Based on this assumption, we use η_l instead of η_u, and the effect of this substitution can be ignored, which is demonstrated in supplementary material. Then the gap coefficient can be calculated by analyzing the labeled data distribution efficiently before the training process as shown in Algorithm 1, and a similar substitution setting will also apply to T_{pos} in Sec.3.2.2.

Algorithm 1 Pipeline of our method

Input: Labeled video action clips X_l and unlabeled video action clips X_u.

// Pre-process for unlabeled class balancing
for $j = 1$ to C **do**
  Calculate gap coefficient g_u^j on Eq.(11)-(12)
  Generate the class balanced weight w_g^j via Eq.(13)
// Training
while not reach the maximum iteration **do**
  Get predictions: $pred_s^l$,$pred_t^u$,$pred_s^u$ via Eq.(1)
  // Multiple pseudo labeling
  Set thresholds T_{pos}, T_{neg}
  Get multiple pseudo labels y_u
  Calculate weights w_{pos} on Eq.(18)
  // Unlabeled class balancing
  Get sample mask m on Eq.(8)
  Select action instances from $pred_s^u$ with mask m
  Re-weight the *BCE* loss according to w_g^j
  Compute the loss L_u,L_s and L_{total} via Eq.(14)-(17)
return Model's parameter

3.4 Loss function

Following [2, 3, 33, 46], our total loss consists of supervised and unsupervised losses. The unsupervised loss for the i-th action instance and j-th class is a combination of the weighted *BCE* loss and the regularization term in Sec.3.2, as shown in Eq.(14). And the λ_1 represents the weight for the regularization term.

$$l_u^{i,j} = l_{cls}^{i,j} + \lambda_1 l_r^{i,j} \tag{14}$$

Based on the unlabeled class balancing strategy, we mask the action instances and re-weight the unsupervised loss in class-level. The formulation of unsupervised loss is shown as bellow.

$$L_u = \frac{1}{N_{act}C}\sum_{i=1}^{N_{act}} m_i \sum_{j=1}^{C} w_g^j l_u^{i,j} \tag{15}$$

For the supervised loss, we also perform the unlabeled class balancing strategy in Sec.3.3 on the labeled data by using ground truth labels instead of the pseudo labels. The supervised loss can be formulated as in Eq.(16), where *BCE* loss is leveraged to calculate classification loss for the student prediction on labeled action clips.

$$L_s = \frac{1}{N_{act}C}\sum_{i=1}^{N_{act}} m_i \sum_{j=1}^{C} w_g^j BCE(pred_l^{i,j}, y_l^{i,j}) \tag{16}$$

Then the final training objective can be formalized as combination of both supervised and unsupervised loss, which is formulated as:

$$L_{total} = L_s + \lambda L_u \tag{17}$$

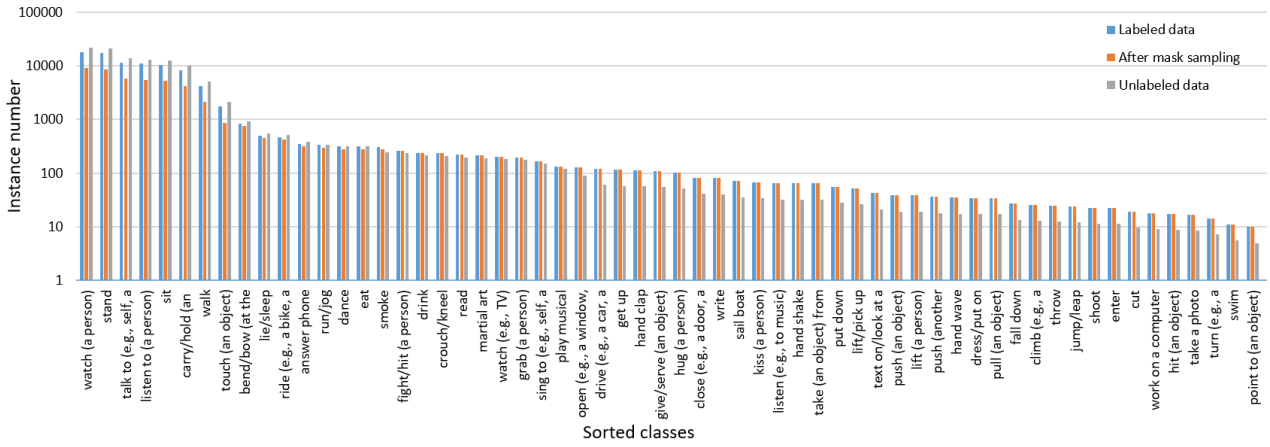


Figure 3: The class distribution of action instances, which is generated from 10k video clips randomly sampled from both the labeled and unlabeled dataset. The class imbalance issue exists in both labeled and unlabeled videos. The pseudo labels generated from unlabeled data exhibit a more imbalanced distribution. The mask sampling in Sec.3.3 can alleviate the imbalance issue, but cannot eliminate it due to the effect of multi-label co-occurrence.

where L_s and L_u represent supervised loss and unsupervised loss applied on labeled and unlabeled action clips respectively. And the overall pipeline of our semi-supervised learning approach is shown in Algorithm 1.

4 EXPERIMENTS

4.1 Datasets

4.1.1 AVA. AVA [14] is a labeled dataset of spatio-temporally localizing atomic visual actions, containing 430 15-minute videos. For AVA, the action instances in videos are annotated with bounding boxes and their corresponding multiple action labels. And the annotations are provided on key frames which are sparsely sampled at 1 FPS. We use both v2.1 and v2.2 of AVA in comparison to the state-of-the-art methods and v2.2 for ablation study. Following the official guidelines, we evaluate 60 action classes with frame-level mAP as the metric, and the IoU threshold is 0.5.

4.1.2 UCF101-24. UCF101-24 is a labeled dataset for spatio-temporal video action detection, which contains spatio-temporal annotations on 3,207 videos for 24 action classes. Following the common settings of previous methods [18, 20, 21, 29, 44], we perform experiments on the first split of UCF101-24 and use frame-mAP@0.5 as the metric for evaluation.

4.1.3 Unlabeled dataset. We notice that in the AVA training set, the ground truth action labels and bounding boxes are provided every 1 second on the key frames. It means that only the key frames are annotated with action labels, the rest frames remain unlabeled. We collect the unlabeled frames without action labels and spatio-temporal annotations from AVA and treat them as key frames. Centering on these unlabeled key frames, we can obtain unlabeled video clips that constitute our unlabeled dataset. The number of video clips in the unlabeled dataset is 184k. And the ratio compared to the entire labeled AVA dataset is approximately 1:1.

4.2 Implementation details

4.2.1 Person detector. For the labeled video clips on AVA, we leverage the predicted human detection boxes from [38] for the person detection on key frames, following the routine setting [29, 35, 39]. The person detection model is Faster R-CNN [30] with a ResNeXt-101-FPN backbone pre-trained on ImageNet [11], COCO [27] dataset, and finetuned on the AVA dataset. And for the UCF101-24 dataset, we adopt the person detector designed in [20], which is pretrained on COCO dataset and finetuned on UCF101-24 dataset. For the unlabeled video clips, we leverage the above person detectors to get the detection bounding boxes of the action instances.

4.2.2 Network Structure. For semi-supervised learning on AVA dataset, we use SlowFast[12] and ACAR [29] as our baseline model to verify the effectiveness of our method. Following [29], we use SlowFast R50 8 × 8 instantiation and increase the spatial resolution of res5 by 2×. For the ACAR, we instantiate the ACAR R50 8 × 8 without actor feature bank. The backbone for both SlowFast and ACAR is pre-trained on the Kinetics-400 [7] dataset. For UCF101-24 dataset, we use SlowFast as our baseline model and set the temporal sampling for the slow pathway to 8 × 4 and the input for the fast pathway is 32 consecutive frames, following [29].

4.2.3 Training and Inference. We first train the baseline model on the labeled AVA dataset, and the training setups follow the original papers. Then for semi-supervised training , we train the network for 35k iterations employing the SGD with an initial learning rate of 0.0064, momentum 0.9 and weight decay 10^{-7} respectively. Also, we use the onecycle [32] learning procedure to schedule the learning rate. We perform an exponential moving average with the momentum of 0.999. We set the batchsize for labeled data as 16, and the ratio of unlabeled data to labeled data is set as 1:1. We use horizontal flip for the weak augmentation. And for the strong augmentation performed on video clips, we extend the RandAugment [9] in the temporal dimension in our experiments. We set the unsupervised weight $\lambda = 0.3$ and regularization weight $\lambda_1 = 1.0$ for calculating

the total loss in Eq.(14)-(17). We perform multiple pseudo labeling by using learnable class-related threshold mentioned in Sec.3.2, and set the hyper parameter $a = 20$ and $\tau_{pos} = 0.9$. We leverage the unlabeled class balancing strategy in Sec.3.3 on both the labeled and unlabeled training data, setting the $\beta = 0.9$ and $\gamma = 0.05$. And we use η_l instead of η_u for calculating the balanced weights before training iteration. For inference on AVA, we scale the shorter side of input frames to 256 pixels and use detected person boxes with scores greater than 0.85 for final action classification following the common setting in [12, 29, 35, 38, 39]. The training and inference details for UCF101-24 dataset are listed in supplementary material.

4.3 Comparison with Existing methods

To demonstrate the effectiveness of our design, we compare the proposed method with other representative semi-supervised methods, both training with 100% labeled data and unlabeled data. Comprehensive experiments of all methods have been conducted on AVA and UCF101-24 datasets, and the implemetation details for Pseudo-label [23], Noisy student [42] and Fixmatch [33] can be found in the supplementary material. As shown in Table 1, our method outperforms all other existing methods by a considerable improvement, lifting the best mAP@0.5 from 25.4% to 26.4% and from 81.9% to 82.8% on AVA v2.2 and UCF101-24, respectively. This results substantiate the superiority of our proposed method.

We also conduct experiments on AVA v2.2 to validate the effectiveness of our methods when training with limited labeled data, and the results are shown in Table 2. Our method outperforms the fully-supervised baseline (Our implementation of SlowFast R50) and other semi-supervised methods in the case of different labeled data ratios. With 50% labeled data, our method can still outperform the fully-supervised baseline trained with the whole labeled data.

Table 1: Comparison to other representative semi-supervised methods on AVA and UCF101-24 datasets, measured by frame-mAP@0.5.

Method	AVA v2.1	AVA v2.2	UCF101-24
Baseline	24.8	25.2	81.6
Pseudo-label [23]	23.4	23.9	78.8
Noisy student [42]	24.5	24.7	81.7
Fixmatch [33]	24.9	25.4	81.9
Ours	**26.1**	**26.4**	**82.8**

Table 2: Comparison to the semi-supervised methods on AVA v2.2 when training with different ratio of labeled data.

Method	Ratio of labeled data				
	5%	10%	25%	50%	100%
Baseline	19.0	21.2	23.3	24.9	25.2
Pseudo-label [23]	18.7	20.9	22.1	23.6	23.9
Noisy student [42]	19.1	21.1	22.9	24.3	24.7
Fixmatch [33]	19.3	20.9	23.2	24.8	25.4
Ours	**19.6**	**21.9**	**23.8**	**25.9**	**26.4**

To compare our method with the existing fully-supervised methods, we use 100% labeled and unlabeled data to train our framework. Note that for the sake of fairness, we compare our method with fully-supervised methods pre-trained on Kinetics-400 without long-term feature/memory bank [29, 35, 38]. As shown in Table 3 and Table 4, respectively, our method achieves the state-of-the-art performance in video action detection on both AVA and UCF101-24 dataset. Our method improves the performance over different fully-supervised baselines and validates the benefits of leveraging unlabeled data.

Besides, Fig. 4 shows the mAP increment of our method with respect to the fully-supervised baseline on all categories. We can see that our semi-supervised method gives more performance boosts on the minority classes, which is consistent with our insight to balance the long-tailed data distribution.

Table 3: Comparison to the sota fully-supervised methods on AVA dataset. FB means the long-term feature bank.

model	backbone	AVA	val mAP
VAT [13]	I3D	v2.1	25.0
C-RCNN w/o FB [39]	Res50	v2.1	25.3
SlowFast [12]	Res50	v2.1	24.2
WOO [8]	SlowFast R50	v2.1	25.2
Ours + SlowFast	SlowFast R50	v2.1	26.1
SlowFast [12]	Res50	v2.2	24.9
WOO [8]	SlowFast R50	v2.2	25.4
ACAR w/o FB [29]	SlowFast R50	v2.2	27.8
Ours + SlowFast	SlowFast R50	v2.2	26.4
Ours + ACAR w/o FB	SlowFast R50	v2.2	**28.2**

Table 4: Comparison to the fully-supervised methods on UCF101-24 dataset.

Method	Inputs	val mAP
T-CNN [16]	RGB	67.3
STEP [44]	RGB+FLOW	75.0
S3D-G [43]	RGB+FLOW	78.8
YOWO [20]	RGB	80.4
MOC [25]	RGB+FLOW	73.1
Baseline (SlowFast R50)	RGB	81.6
Ours	RGB	**82.8**

4.4 Ablation Study

4.4.1 Effectiveness of components. To verify the impact of different components in our semi-supervised learning method, experiments are performed as shown in Table 5, where MPL stands for *Multiple pseudo labeling* and UCB for *Unlabeled class balancing*. For the semi-supervised learning without MPL and UCB, we set fixed thresholds $T_{pos} = 0.5$ and $T_{neg} = 0.1$ for generating multiple pseudo labels. The results in Table 5 illustrate the effectiveness of the two modules and also show that semi-supervised learning method will get a worse performance (24.9%) than the fully-supervised baseline (25.2%) without MPL and UCB. Furthermore, comparing the training results on 5% and 100% labeled data, we can find different effects of MPL and UCB on the performance improvement. Unlike the case of 5% data, the improvement of UCB is more than that of MPL in

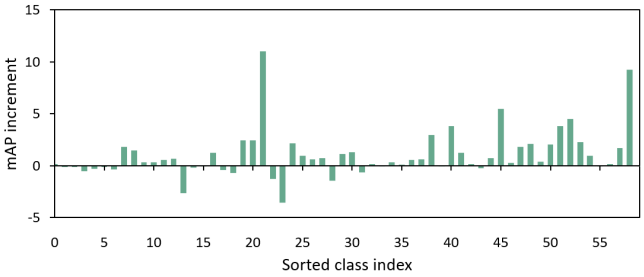


Figure 4: Gains of mAP for each class on the AVA dataset with respect to the baseline.

the case of 100% labeled data. Because the model is well-trained on 100% labeled data and overfits towards the majority classes, leading to serious class imbalance issue, which can be effectively alleviated by the UCB. This validates that the imbalanced class distribution of training data is a major factor limiting the performance of semi-supervised learning on multi-label video action detection.

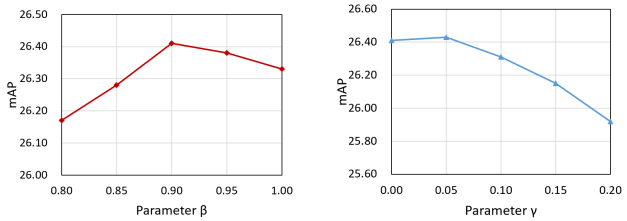
Table 5: Effectiveness of components on AVA dataset. MPL: Multiple pseudo labeling. UCB: Unlabeled class balancing.

MPL	UCB	Ratio of labeled data	
		5%	100%
		19.02	24.93
✓		19.21	25.71
	✓	19.19	26.22
✓	✓	**19.56**	**26.43**

4.4.2 Effectiveness of multiple pseudo labeling. For the ablation study on multiple pseudo labeling, we perform experiments on AVA dataset with both 5% and 100% labeled data. For the fixed threshold strategy, we fix the positive threshold $T_{pos} = 0.5$ and negative threshold $T_{neg} = 0.1$. And for the class-related threshold strategy, we set parameter $\tau_{pos} = 0.9$ and the negative threshold $T_{neg} = 0.1$. For the mapping function in Sec.3.2, we set a concave function $M(x) = x^{0.1}$ to map the imbalance ratio to get the positive threshold for each class. We initialize the thresholds in the learnable threshold strategy with the thresholds calculated by the class-related method and we set the parameter $a = 20$ and $\lambda_1 = 1$ in the experiments. The ablation study details for mapping function and learnable strategy are listed in supplementary material. We use baseline with UCB and compare the performance of different multiple pseudo labeling strategies, and the results are shown in Table 6. The results show that class-related threshold strategy outperforms the method that set fixed positive threshold to generate multiple pseudo labels for all categories, and the advantage of class-related threshold is more obvious with limited training annotations. This is because setting a fixed threshold for all classes will generate more positive pseudo labels for the majority classes, which makes the imbalance issue severe and weakens the feature learning for the minority classes. Furthermore, compared with the class-related threshold strategy, the learnable threshold strategy improves the performance by setting the learnable thresholds, which can be dynamically adjusted during the training iteration.

Table 6: Ablation study on threshold strategy in multiple pseudo labeling.

Method	Ratio of labeled data	
	5%	100%
Fixed threshold	19.19	26.22
Class-related threshold	19.44	26.37
Learnable threshold	**19.56**	**26.43**



(a) Parameter β (b) Parameter γ

Figure 5: Ablation study for hyper parameters in unlabeled class balancing.

4.4.3 Effectiveness of unlabeled class balancing. We conducted experiments with 100% labeled data to determine hyper parameters in unlabeled class balancing. We first explore the sampling rate parameter β in Eq.(9). We use baseline with MPL for training and set the parameter γ to 0, which is equivalent to not performing subsequent re-weighting operation. According to the results shown in Fig.5(a), we set the β as 0.9 and perform the experiments for parameter γ. The results are shown in Fig.5(b). The experimental results show that setting a lower sample rate is not always better for the majority classes. Because the mask sampling will omit a number of valuable instances, and thus weakens the model's feature learning capacity on the majority classes. In addition, the results in Fig.5(b) show that over weighting the minority classes also reduces the performance of the model.

5 CONCLUSION

In this paper, we propose a novel semi-supervised learning method for multi-label video action detection. This is the first work to perform semi-supervised learning for video action detection in multi-label scenario. We point out the two major challenges in this task: generation of multiple pseudo labels and class-imbalanced data distribution. First, we design an effective multiple pseudo-label generation method by setting dynamic learnable thresholds according to the class distributions. Besides, to alleviate the imbalance between different classes, we propose the unlabeled class balancing, which selects training samples dynamically according to their pseudo labels during the training phase and then performs re-weighting to balance the effect of multi-label co-occurrence. Extensive experiments conducted on AVA and UCF101-24 datasets demonstrate the effectiveness of our proposed method.

ACKNOWLEDGMENTS

This work has been funded in part by the NSFC grants 62176156 and the Science and Technology Commission of Shanghai Municipality under Grant 20DZ2220400.

REFERENCES

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. 2014. Learning with pseudo-ensembles. *Advances in neural information processing systems* 27 (2014).

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. 2019. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785* (2019).

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems* 32 (2019).

[4] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks* 106 (2018), 249–259.

[5] Jonathon Byrd and Zachary Lipton. 2019. What is the effect of importance weighting in deep learning?. In *International Conference on Machine Learning*. PMLR, 872–881.

[6] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems* 32 (2019).

[7] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[8] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. 2021. Watch only once: An end-to-end video action detection framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8178–8187.

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 702–703.

[10] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9268–9277.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6202–6211.

[13] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. 2019. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 244–253.

[14] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. 2018. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6047–6056.

[15] Hao Guo and Song Wang. 2021. Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15089–15098.

[16] Rui Hou, Chen Chen, and Mubarak Shah. 2017. Tube convolutional neural network (t-cnn) for action detection in videos. In *Proceedings of the IEEE international conference on computer vision*. 5822–5831.

[17] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5375–5384.

[18] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. 2017. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 4405–4413.

[19] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. 2019. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217* (2019).

[20] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. 2019. You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. *arXiv preprint arXiv:1911.06644* (2019).

[21] Akash Kumar and Yogesh Singh Rawat. 2022. End-to-End Semi-Supervised Learning for Video Action Detection. *arXiv preprint arXiv:2203.04251* (2022).

[22] Samuli Laine and Timo Aila. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242* (2016).

[23] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, Vol. 3. 896.

[24] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. 2020. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214* (2020).

[25] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. 2020. Actions as moving points. In *European Conference on Computer Vision*. Springer, 68–84.

[26] Yuxi Li, Boshen Zhang, Jian Li, Yabiao Wang, Weiyao Lin, Chengjie Wang, Jilin Li, and Feiyue Huang. 2021. LSTC: Boosting Atomic Action Detection with Long-Short-Term Context. In *Proceedings of the 29th ACM International Conference on Multimedia*. 2158–2166.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[28] Jingcheng Ni, Jie Qin, and Di Huang. 2021. Identity-aware Graph Memory Network for Action Detection. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3437–3445.

[29] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. 2021. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 464–474.

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497* (2015).

[31] Li Shen, Zhouchen Lin, and Qingming Huang. 2016. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*. Springer, 467–482.

[32] Leslie N Smith and Nicholay Topin. 2019. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, Vol. 11006. International Society for Optics and Photonics, 1100612.

[33] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems* 33 (2020), 596–608.

[34] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[35] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. 2020. Asynchronous interaction aggregation for action detection. In *European Conference on Computer Vision*. Springer, 71–87.

[36] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems* 30 (2017).

[37] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. *Advances in Neural Information Processing Systems* 30 (2017).

[38] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. 2019. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 284–293.

[39] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu. 2020. Context-aware rcnn: A baseline for action detection in videos. In *European Conference on Computer Vision*. Springer, 440–456.

[40] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. 2020. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *European Conference on Computer Vision*. Springer, 162–178.

[41] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems* 33 (2020), 6256–6268.

[42] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10687–10698.

[43] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*. 305–321.

[44] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. 2019. Step: Spatio-temporal progressive learning for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 264–272.

[45] Yuzhe Yang and Zhi Xu. 2020. Rethinking the value of labels for improving class-imbalanced learning. *Advances in Neural Information Processing Systems* 33 (2020), 19290–19301.

[46] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems* 34 (2021).

[47] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. 2020. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9719–9728.

A ABLATION STUDY ON MULTIPLE PSEUDO LABELING

A.1 Class-related threshold

Inspired by [46], We explore three different mapping functions for getting class-related threshold in Table 7: (1) concave: $M(x) = x^{0.1}$, (2) linear: $M(x) = x$, and (3) convex: $M(x) = x/(2-x)$. We set the parameter $\tau_{pos} = 0.8$ for the experiment. And we see that the concave function shows the best performance and the linear function shows the worst.

Table 7: Ablation study on mapping function.

Mapping function	mAP
Concave	26.30
Linear	25.73
Convex	25.89

We also conduct experiments on the positive threshold in class-related threshold strategy, as shown in Fig.6. The optimal choice of positive threshold τ_{pos} is around 0.9, either increasing or decreasing this value will lead to a performance decay. Beside, we also try to set class-related thresholds for T_{neg}, but the experimental results are worse than setting $T_{neg} = 0.1$ for all classes.

A.2 Learnable threshold

We design a sigmoid function for the positive weight, where $i = 1, 2, ..., N_{act}$ and $j = 1, 2, ..., C$, as shown in Eq.(18).

$$w_{pos}^{i,j} = \frac{1}{1 + \exp(-a(pred_t^{i,j} - T_{pos}^j))} \tag{18}$$

The function curves are shown in Fig.7, where T_{pos}^j is set to 0.5. The parameter a determines the shape of the function, and it will be steeper when setting a larger a. For example, when $a = 20$, $T_{pos}^j = 0.5$, for the j-th class of the i-th action instance, the positive weight $w_{pos}^{i,j}$ will grow from around 0 to 1 with $pred_t^{i,j}$ growing from around 0.3 to 0.7. According to the experimental results in Fig.8(a), we set $a = 20$.

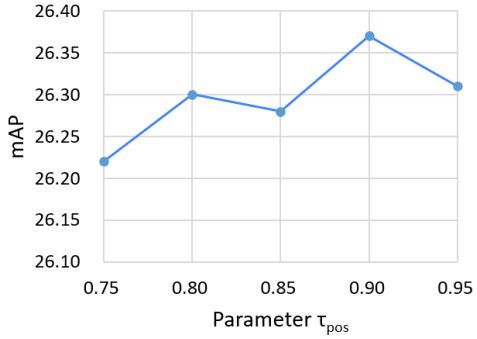


Figure 6: Ablation study on parameter τ_{pos}.

The regularization term is used for preventing the value of T_{pos}^j from being too large, as shown in Eq.(19). The regularization term will achieve the minimum value when $T_{pos}^j = 0.5$. And it will become larger when T_{pos}^j approaches 0 or 1, thus making the value of T_{pos}^j reasonable.

$$l_r^{i,j} = -\log T_{pos}^j + 2T_{pos}^j \tag{19}$$

To explore the weight λ_1 of regularization term, we conduct experiment as shown in Fig.8(b). We can see that it is optimal when $\lambda_1 = 1$. When a larger λ_1 is set, regularization term will tend to make $T_{pos}^j = 0.5$, approaching the fixed threshold strategy. And when setting a smaller λ_1, T_{pos}^j tends to a larger value during the training, which is not conducive to the feature learning of the model.

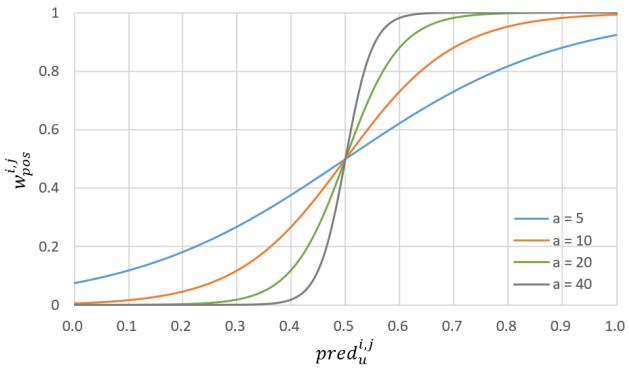


Figure 7: Curve of $w_{pos}^{i,j}$ when setting different a, $T_{pos}^j = 0.5$.

B ABLATION STUDY ON UNLABELED CLASS BALANCING

B.1 Data Imbalanced ratio

We assume that pseudo labels generated from unlabeled data and ground truth labels from labeled data have similar distributions. Based on this assumption, we use imbalanced ratio of labeled data η_l instead of η_u, and the effect of this substitution can be ignored, which is demonstrated in Table 8.

Table 8: Experimental results using labeled and unlabeled imbalanced ratio.

Method	mAP
Using unlabeled distribution η_u	26.37
Using labeled distribution η_l	26.43

B.2 Pseudo-label Based Class Balancing

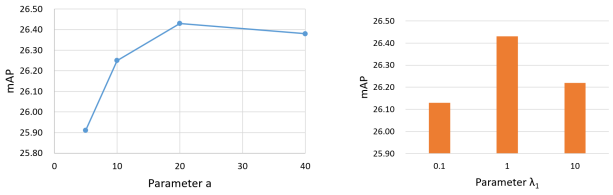
The labeled class balanced methods will encounter difficulties when performing data re-sampling with unlabeled data. Since data re-sampling requires the class information for all unlabeled data before training, an additional inference process is required to generate the corresponding pseudo labels for the unlabeled data. We call this method "pseudo-label based class balancing". And the experimental results on AVA v2.2 are shown in Table 9

C EXPERIMENT ON THE WEIGHT OF UNSUPERVISED LOSS

We perform experiments on the weight for the unsupervised loss, as shown in Fig.9. And we set $\lambda = 0.3$ according to the results.

Table 9: Experimental results using Pseudo-label Based Class Balancing and Unlabeled Class Balancing.

Method	mAP
Pseudo-label Based Class Balancing	26.02
Unlabeled Class Balancing	26.43



(a) Parameter a (b) Parameter λ_1

Figure 8: Ablation study for hyper parameters in learnable threshold.

D TRAINING AND INFERENCE DETAILS ON UCF101-24

On the UCF101-24 dataset, we firstly train the baseline model following the training setups in [29]. Then for semi-supervised training, we train the whole framework end-to-end for 5.4k iterations with a base learning rate of 0.0008 and use the onecycle learning procedure to schedule the learning rate. We use all boxes generated by the person detector for inference. Other hyperparameter settings for semi-supervised learning are similar to experiments on AVA.

E IMPLEMENTATION DETAILS OF OTHER SEMI-SUPERVISED METHODS

We adopt several representative semi-supervised methods for comparison with our method. We implement Pseudo-label [23], Noisy Student [42] and Fixmatch [33] on the multi-label video action detection task. The implementation details are as follows.

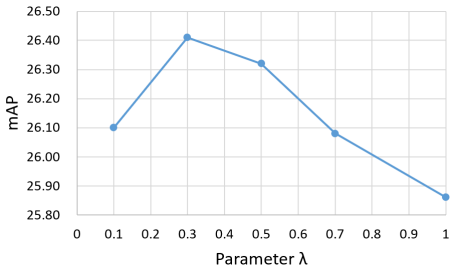


Figure 9: Ablation study on the weight λ of unsupervised loss.

E.1 Pseudo-label

Based on the pipeline of our method, we perform weak augmentation on both the labeled and unlabeled input clip, and we do not update the teacher model during the training, following [23]. We train the model without the UCB (unlabeled class balancing) and leverage fixed threshold strategy for MPL (multiple pseudo labeling).

E.2 Noisy student

For Noisy student, we leverage the same augmentation function as our semi-supervised method. And following [42], we preform the iterative training twice. The other settings are the same as the Pseudo-label method in Sec.E.1.

E.3 Fixmatch

For Fixmatch, we update the teacher model's parameter in the same way as our method. When training the network, we do not use UCB and the iterative training strategy, and set a fixed threshold for MPL