UTS UNIVERSITY
OF TECHNOLOGY
SYDNEY

Learning from Imperfect Supervision in Visual Pattern Classification and Localization

by Fan Ma

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Yi Yang

University of Technology Sydney

Faculty of Engineering and Information Technology

August 2022

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Fan Ma declare that this thesis, is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy*, in the *Faculty of Engineering and Information Technology* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature: Signature removed prior to publication.

Date: 24th August 2022

Abstract

Machine learning algorithms have achieved tremendous success on various computer vision tasks in past decades. Large-scale well-annotated data, such as ImageNet and ActivityNet, are necessary for learning a valuable model. However, high-quality training samples are often insufficient in practice, and it is labor-intensive and time-consuming to produce intense supervision for different learning tasks. Designing algorithms with imperfect training data thus becomes significant in the current data explosion era.

In this dissertation, imperfect supervision is categorized into three classes: 1) Limited supervision where only a small portion of training samples are annotated; 2) Noisy supervision where some labels of training samples are corrupted; 3) Weak supervision where the labels of training data are imprecise to provide expected outputs. Several models are developed to learn from the supervision of different data types. A self-paced co-training algorithm is proposed to improve the model performance when limited training samples are available. I have also proved that our algorithm can achieve a better model with diverse classifiers. Moreover, a self-reweighting mechanism based on online learned class centroids is introduced to prevent the model from deteriorating by noisy supervision. Experiments are conducted on several image recognition datasets demonstrating the superiority of our designed algorithms under both limited and noisy supervision. Furthermore, two practical applications of temporal localization are studied when weak supervision is available. The first task is the temporal action localization, where only a single frame is annotated for each action instance. The goal is to produce precise temporal boundaries for action instances. An efficient frame expanding algorithm has been introduced to improve the temporal action localization performance. The other task uses query language to temporally localize moments in videos where only language-video pairs are available in the training data. The connections between the video clips and concepts in query sentences are formed by decoupling the core concepts in the query sentence.

This thesis demonstrates that our well-designed algorithms yield excellent results when only imperfect data are available in various vision tasks, ranging from image classification, object detection, and temporal localization in videos.

Dissertation directed by Professor Yi Yang

Australian Artificial Intelligence Institute, School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney

Acknowledgments

First and foremost, I want to acknowledge the help and guidance of my principal supervisor, Prof. Yi Yang. Yi is a brilliant person not only in research fields but also in social communication. In the past four years, I have learned a lot from him. He encourages and helps me to build industry connections to stay at the forefront of several computer vision tasks. Moreover, he always points out my problem quickly and instructs me to make life-long plans. "Think before doing" is the most vital spirit I have learned from him. His calm and rational advice has been essential to my survival of several years as a graduate student and to my development as a researcher.

I would also like to thank my co-supervisor Prof. Deyu Meng, who was also my master tutor. I got to know Yi because of his recommendation. He always encourages me to stay focused on the research and to do some exciting work. His passion and optimism for research and life also inspire me to enjoy the process of doing academic research. Many thanks to my co-authors, Linchao Zhu, Zheng Shou, Xin Yu, Zhicheng Yan, Yu Wu. They have provided many insightful ideas and contributed a lot to the paper writing. I also want to thank my colleagues, Xiaohan Wang, Ruijie Quan, and Zhun Zhong, who are also my roommates and friends. We have had a wonderful time in the past four years.

I also want to thank my group members and colleagues, Xiaojun Chang, Jiaxu Miao, Hehe Fan, Pingbo Pan, Yanbin Liu, Zongxin Yang, Qingji Guan, Yawei Luo, Qianyu Feng, Yunqiu Xu, Guang Li, Youjiang Xu, Xuanmeng Zhang, Yuhang Ding, Guangrui Li, Yang He, Ping Liu, Yutian Lin, Peike Li, Xuanyi Dong, Tianqi Tang, Bingwen Hu, Minfeng Zhu, Fengda Zhu, Hu Zhang, Zhedong Zheng, Liang Zheng, Xiaolin Zhang, Aming Wu, and many others. I was very fortunate to collaborate with or discuss with them. These discussions inspired and motivated many of my research works.

Lastly, I would like to thank my family and my girlfriend Yue Yang for their selfless support and love.

Fan Ma

Sydney, Australia, 2022

List of Publications

Conference Papers:

1. **Fan Ma**, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, Zheng Shou. "SF-Net: Single-Frame Supervision for Temporal Action Localization", European Conference on Computer Vision (ECCV 2020)

2. **Fan Ma**, Zheng Shou, Linchao Zhu, Haoqi Fan, Yilei Xu, Yi Yang, Zhicheng Yan. "Unified Transformer Tracker for Object Tracking", IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2022)

Journal Papers :

3. **Fan Ma**, Deyu Meng, Xuanyi Dong, Yi Yang, "Self-Paced Multi-view Co-training", Journal of Machine Learning Research (JMLR)

4. **Fan Ma**, Yu Wu, Xin Yu, Yi Yang, "Learning With Noisy Labels via Self-Reweighting From Class Centroids", IEEE Transactions on Neural Networks and Learning Systems (TNNLS)

5. **Fan Ma**, Linchao Zhu, Yi Yang, "Weakly Supervised Moment Localization with Decoupled Consistent Concept Prediction", International Journal of Computer Vision (IJCV)

Table of Contents

List of Figures

LIST OF TABLES

Chapter 1

Introduction

1.1 Motivation

Deep neural networks have achieved remarkable successes in various practical applications, including image classification, detection, video understanding, and natural language processing (NLP). Large amounts of training data are critical for obtaining valuable models. The emergence of ImageNet (Deng et al., 2009) has reached a milestone for the image classification community. Afterwards, several significant benchmarks have been set for almost all machine learning problems, such as COCO (Lin et al., 2014) in object detection and segmentation tasks, Kinetics (Kay et al., 2017) in video understanding, and GLUE (Wang et al., 2018) in NLP. These large scale training data provide substantial supervision for training models. However, annotating training samples in different tasks requires enormous human resources and takes an extended time. The labels of training samples could also be incorrect if annotators mistakenly assign the wrong labels to some training samples. Besides, almost unlimited data emerges on the internet every day, so it is impossible to annotate all of them for model learning. Learning with imperfect supervision is thus essential for obtaining valuable machine learning models. In this dissertation, we study three types of data whose annotations are insufficient to provide full supervision for the learning process:

- Only a few training samples are annotated. In this case, the models learn from limited labeled training samples and amounts of unlabeled samples, which are more easily collected. For the image classification, only a few samples of each class are labeled while most samples are unlabeled.

- Training samples are annotated with incorrect labels. Annotators may assign the false labels to the training samples when they are tired during annotation or they fail to identify the labels of samples. For instance, a dog image could be assigned with the cat category by accident.

- Training samples are related but not directly aligned with the expected outputs. The specific task in this set usually requires fine-grained outputs while the training data only have coarse annotations. For instance, we may only have frame annotations without timestamps while the task is to temporally localize events in videos.

Are there ways to train the model with imperfect supervision? We argue that better models can be obtained by diving into various data supervision.

a) When only a few training samples are annotated, the large number of unsupervised samples can be used to boost model performance. It is widely studied in semi-supervised learning (SSL) which aims to learn from both labeled and unlabeled data. Building connections between labeled samples and unlabeled ones is significant in SSL. The supervised knowledge delivered by labeled data and potential data structure underlying unlabeled ones is mined in various researches. Co-training (Blum and Mitchell, 1998), which trains different classifiers and exchanges labels of unlabeled instances in an iterative way, is one of the most classical and well-known SSL approaches. In recent years, co-training has been attracting much attention attributed to both of its wide applications (Nigam and Ghani, 2000; Wan, 2009; Kumar and Iii, 2011; Zhu et al., 2012; Do et al., 2016) and rational theoretical supports (Blum and Mitchell, 1998; Balcan et al., 2004; Balcan and Blum, 2010; Wang and Zhou, 2007, 2010, 2013, 2017). However, current co-training style methods lack optimization objectives that can measure the performance and explain the intrinsic iterative mechanism for the learning with limited training samples. This thesis introduces the self-paced co-training (SPaCo), which contains a specified objective function where the optimization process complies with the learning procedure of conventional co-training. Moreover, the rationality of our proposed algorithm is guaranteed. Experimental evaluations on image classification, image retrieval, and object detection demonstrate the generalization capacity and effectiveness of our algorithm.

b) Noisy labels are commonly encountered in practical machine learning tasks since datasets collected from search engines (Liang et al., 2016; Zhuang et al., 2017) or annotated by crowdsourcing systems (Bi et al., 2014) usually contain false supervision. Besides, many erroneous labels are from human annotations since annotators may mistakenly assign false labels to given samples (Deng et al., 2009; Ma et al., 2020b). Noisy labels in general handicap the performance of machine learning models from two aspects. On the one hand, the increasing number of falsely annotated samples results in the insufficient sampling of effective samples during training. On the other hand, training on noisy data could deteriorate the model performance as models, especially the deep neural networks, are easily overfitted on falsely annotated samples. Identifying these corrupted training samples and preventing the model from overfitting on these samples are thus critical to obtaining robust models (Raykar et al., 2010; Ren et al., 2018).

A standard solution is to assign dynamic weights to samples. Impacts of noisy labeled data will be potentially reduced in training when weighted with smaller weights. (Bi et al., 2014). For instance, assigning zero weights to wrongly annotated samples prevents training from fallacious supervision signals. Recent methods produce sample weights according to the losses of training samples (Shu et al., 2019; Ren et al., 2018), where the sample is deemed as an incorrectly annotated sample when the corresponding loss is big. However, a model often assigns large weights to noise samples when the model fits the noisy samples well. This thesis proposes a novel reweighting method, namely self-reweighting from class centroids (SRCC), to ameliorate the weight assignment for noisy data. Concretely, the centroid of each class is first obtained based on sample features. The similarities between samples and class centroids are then calculated to produce sample weights. Our SRCC thus alleviates false supervision signals and improves the reliability of sample weights. Theoretical analysis has also been given to demonstrate the significance of our proposed method.

c) Sometimes annotated labels are not the same as the outputs we expected from the model. Less information is contained in the annotated labels compared to the desired prediction. In such cases, relations between inputs and expected predictions can not be directly built. For instance, the model is required to produce video-level predictions such as temporal localization of action instances while only action frame annotations are available. Relations between training annotations and desired outputs should be built to enable learning on these training samples. In this thesis, we study two practical applications of such cases.

We first study the single frame supervision for temporal action localization (TAL). It is time-consuming to annotate the temporal boundary of action instances in a video. The annotators often watch the video several times to annotate a single action instance. In this thesis, we make the first attempt to temporally localization action instances with only frame labels. The annotators only watch the video once to record the action class and timestamp when they notice each action. This significantly reduces annotation resources compared to full supervision. To make full use of single-frame supervision, we make several inventions in our single frame network (SF-Net) for enhancing the localization capability via identifying background frames and action frames. A novel background and foreground mining strategy is introduced to expand the training pool through pseudo labelling action frames. Our model has archived impressive performance compared to models trained with full supervision.

We further investigate the weakly-supervised moment localization with natural language. This is similar but a more complicated task compared to the TAL. The temporal boundary of the target video clip is also produced by the model but is given based on query language sentences. When the precise temporal boundary supervision signals are absent, localizing the visual representations is challenging with video sentence pairs. In this thesis, we divide this problem into the localization of atomic objects and actions in the query sentence. We propose the decoupled consistent concept prediction (DCCP) for weakly-supervised video localization where a novel pairing module is developed to match word features with video clip features for localizing the atomic concepts. Extensive experiments on three benchmarks demonstrate the superiority of our proposed design.

In this thesis, we focus on handling the three types of imperfect data. We demonstrate that models could achieve impressive results even without high-quality human-annotated data. Therefore, we believe that machine learning models could be learned from data of arbitrary types.

1.2 Research Contribution

The contributions of this thesis are summarized as follows:

- We make an effort to teach models to learn with imperfect training samples, which forms three types of problems, including semi-supervised learning (SSL), noisy learning, and weakly-supervised learning. We analyze each problem thoughtfully and propose practical algorithms for different tasks.

- We introduce a unified self-paced co-training (SPaCo) framework, which provides an optimization objective and shares a similar iterative process with the conventional co-training algorithms for SSL. We also analyze the rationality of the proposed SPamCo. The superiority of the proposed algorithms is comprehensively substantiated on multiple tasks.

- We propose a simple yet effective self-reweighting from class centroids method (SRCC) to address training with erroneous labels. We produce a robust sample weight for each sample based on its feature similarity to the class centroids for mitigating the negative effect of false annotated samples. Our work also makes the first attempt to exploit mixed data with noisy labels to enhance the learning process. Extensive experimental results on the various datasets confirm that our method achieves promising classification performance.

- We make the first attempt to use single frame supervision for the challenging problem of localizing temporal boundaries of actions. We present that the single frame annotation reduces annotation time significantly than annotating precise decision boundaries. A single frame network (SF-Net) is introduced to explore the temporal boundaries of actions based on action frames. Extensive experiments are conducted on three benchmarks, showing that the performances on both segment localization and single-frame localization tasks are impressive with our SF-Net.

- We study a complicated weakly-supervised learning task, retrieving moments via natural language without temporal boundary annotations. A novel decoupled consistent concept prediction (DCCP) framework is introduced to facilitate visual and language representation learning. Our framework decouples this task into several sub-tasks, localizing critical concepts in the query sentence. We develop a pairing module for each critical concept to match the video clip that contains the concept. Our framework achieves state-of-the-art performance on three standard moment retrieval datasets, verifying the effectiveness of our proposed framework.

1.3 Thesis Organization

- *Chapter 2*: This chapter reviews the literature on semi-supervised, noisy, and weakly-supervised learning. Specifically, the theoretical developments and applications for SSL in recent years are present. We investigate the common algorithms in the noisy learning problem for the noisy training. We have also studied two practical problems, including temporal action localization and localizing moments with natural language, when annotations do not contain fine-grained information for supervision.

- *Chapter 3*: In this chapter, we present a self-paced co-training (SPaCo) algorithm for SSL when only a small portion of annotated training simples and a large amount of unlabeled samples are available. The training objective and optimization process are well presented to reveal the learning process. We also theoretically showed that the success of our proposed algorithm is guaranteed with weak conditions. An exciting conclusion, "a better model can be obtained if we have several models with diversity", is drawn in this section. Experiments on toy data, image classification, image retrieval, and object detection demonstrate the superiority of our present algorithm.

- *Chapter 4*: This chapter focuses on training with noisy data where two types of noisy training data, symmetric and asymmetric noise, are investigated. To mitigate the impact of false annotated samples, we introduce a novel self-reweighting algorithm from class centroids (SRCC). We also provide theoretical analysis to demonstrate the significance of our SRCC. Experimental results on several benchmark datasets also show that our proposed method outperforms other state-of-the-art algorithms.

- *Chapter 5*: This chapter investigates a practical application to temporal localize action instances in videos. We first show that annotating time could be largely reduced when only the single frame rather than the precise temporal boundary of the action instance is annotated. A useful single frame network (SF-Net) is then introduced in this chapter to build action instances with single frame annotations. Experiments are also conducted on several benchmarks to validate the merit of our SF-Net.

- *Chapter 6*: In this chapter, we solve a more challenging problem, localizing moments with natural language under the weakly-supervised annotation. This task not only requires the feature processing of visual and language signals but also needs a well-designed learning task to build connections between visual and language signals. Without temporal annotations indicating the start and end timestamps of events, we decouple this task into atomic concept localization by extracting main components, such as objects and actions, in videos and languages. Extensive experiments demonstrate that our proposed method achieves impressive results even without precise temporal boundaries of action instances.

- *Chapter 7*: This chapter summarizes the thesis and instructs potential fields to be pursued in the future.

CHAPTER 2

Literature Review

In this chapter, we provide readers with an overview of learning from imperfect data. We begin with semi-supervised learning when limited labeled samples are avilable (Section 2.1). The development of SSL algorithms, especially the co-training style methods, are reviewed. We then present the developments of methods on noisy learning when false annotated labels are contained in training data (Section 2.2). Literature on weakly supervised learning is discussed in Section 2.3 and two practical problems are reviewed.

2.1 Semi-supervised Learning

2.1.1 Co-training

As unlabeled data can be easily obtained with minimal human labor, Semi-supervised learning has attracted lots of attention for boosting model performance with limited training data. Blum and Mitchell (1998) proposed co-training algorithm which trains different classifiers with training samples and expands the training pool of one classifier based on pseudo predictions of unlabeled data from the other classifier. Multiple methods were then developed to improve the model performance. A standard solution is to alter the process of adding unlabeled instances. Nigam and Ghani (2000) proposed Co-EM to add all unlabeled samples at each iteration for training. All unlabeled samples are used in the Co-EM and the labels of unlabeled samples are dynamic changed. Several methods managed to improve the quality of unlabeled samples. Zhang and Zhou (2011) formed a graph to enhance the confidence of selected unlabeled samples. Xu et al. (2016) used predictions of different classifiers to construct a pseudo-label vector for achieving a robust prediction. Zhou (2019) tempted to correct pseudo labels by logical reasoning. Another solution is to constrain the predictions of different classifiers on unlabeled data. Sindhwani et al. (2005b) proposed co-regularized least squares to minimize the difference between predictions of different classifiers. Sindhwani and Rosenberg (2008) introduced Co-MR to exploit Reproducing Kernel Hilbert Spaces defined over the labeled and unlabeled data. Yu et al. (2011) proposed Bayesian co-training with the Bayesian undirected graphical model. Ye et al. (2015) enforced an affixed rank constraint on unlabeled predictions in the optimization function. However, these methods are designed for the specific task, especially the classification task, and they are not easily adapted to more practical tasks, such as object detection.

Moreover, the rationality of co-training is theoretical analyzed in several researches. Blum and Mitchell (1998) proved that the model is learnable in the PAC model when the features used in different classifiers are independent given the class. Abney (2002) relaxed this assumption by providing a weaker view-independence condition. Balcan et al. (2004) introduced the ϵ-expansion assumption, which further relaxed the condition for guaranteeing the success of the co-training algorithm. Wang and Zhou (2010) analyzed the co-training in the view of label propagation. Wang and Zhou (2013) introduced diversity theory that differences between models could be used. Nevertheless, most analyses contain subjective assumptions, such as independence between models or idealistic assurance on pseudo labels. These assumptions are not intuitive and are hard to be verified in practical scenarios.

2.1.2 Self-paced Learning

Bengio et al. (2009) introduced a learning paradigm called *curriculum learning* where training samples from easy to complex are ranked to learn a model. Kumar et al. (2010) treated the curriculum design as a regularization term in the self-paced learning (SPL) objective. A weight is assigned to each sample to indicate the training importance during model iteration. SPL has recently attracted increased attention in various applications. Jiang et al. (2015) introduced a self-paced curriculum learning regime via a loss of prior knowledge. Meng et al. (2017a) showed that the optimization process of SPL is equivalent to a robust loss minimization problem, which can be optimized with the majorization-minimization algorithm. Shu et al. (2019) presented that weights of training samples can be obtained from a meta-network rather than calculating weights from losses. However, all training samples are annotated and used in these methods, failing to build relations between labeled and unlabeled data. Recently, SPL has been also sued in several practical scenarios. Ghasedi et al. (2019) applied the SPL to the generative adversarial clustering and achieves impressive performance gain. Ge et al. (2020) recently used self-paced contrastive learning to narrow the gaps between source and target domains. Yu et al. (2022) adopted SPL to the specific anomaly detection problem and achieves significant improvement.

2.2 Learning from Noisy Labels

When trained with noisy data, many efforts have been taken in recent years to improve the model robustness in terms of theoretical analyses, loss objectives, and sample reweighting. Manwani and Sastry (2013) proved that the risk minimization of 0-1 loss function obtains noise-tolerance properties. Liu and Tao (2015) proposed class-probability estimators with order statistics of predictions on training samples. Zhang and Sabuncu (2018) introduced a generalized cross-entropy loss (GCE), which used mean absolute error and cross-entropy loss together for noisy samples. Wang et al. (2019) introduced a reverse cross-entropy (RCE) to facilitate robust learning, and Lyu and Tsang (2020) presented that curriculum loss can be used to select samples adaptively during training. Although the robust loss enhances the model capacity with noisy data, these methods only work on certain types of noisy data, especially the symmetric noisy data where the noise samples in one class are uniformly sampled for other categories.

Sample reweighting methods have achieved appealing performance for noisy learning in recent years (Shu et al., 2019; Ren et al., 2018; Yu et al., 2019). The core step is to assign sample weights to training samples for the loss calculation. Meng et al. (2017b) found that a monotonically decreasing weighting function is equivalent to optimizing a robust loss function. Inspired by meta-learning (Finn et al., 2017), Shu et al. (2019) proposed a complicated sample reweighting schemes with meta-learning scheme on the validation data. Although significant performance gain is obtained, the training procedure is extremely long on small datasets since the second derivatives are required in meta-learning methods. Zhou et al. (2021) introduced the sparse regularization to reduce the learning impact from noisy samples. Bai et al. (2021) investigated when to stop the training process before the overfitting on the noise samples. Li et al. (2022) integrated contrastive learning with sample selection to improve the model generalization, and Liang et al. (2022) considered the few-shot learning when noise samples are contained. However, these reweighting methods produce sample weights according to individual sample losses, neglecting that individual sample losses would be inaccurate if the model overfits the training data.

2.3 Weakly Supervised Learning

There are some cases where the annotation data could not provide precise supervision for given tasks. The labeled information is also changed in various applications, such as object detection, image segmentation, and temporal localization. The critical issue among these tasks is to build the relationship between the limited annotation data and the required prediction. In this dissertation, we investigate two challenging practical applications, temporal action localization, and moment retrieval via natural language.

2.3.1 Temporal Action Localization

Temporal action localization is to predict the start and end timestamp of action instances. Given temporal boundary annotations, the background and action instances can be distinguished. Zhao et al. (2017b) adopted a structural temporal pyramid pooling to measure the completeness of action instances. Yuan et al. (2017) introduced temporal evolution by splitting an action instance into three parts. Long et al. (2019) proposed a Gaussian kernel to optimize temporal scale of action proposals dynamically. However, these methods use fully temporal annotations, which are not easily obtained in practice. Weakly-supervised temporal action localization is then studied where only action classes in each video are contained. The temporal action score sequence is then used to provide action proposals (Wang et al., 2017; Nguyen et al., 2018; Liu et al., 2019; Narayan et al., 2019). Wang et al. (2017) introduced UntrimmedNet to reason about the temporal duration of action instances. Shou et al. (2018) investigated foreground and background scores in the temporal dimension for identifying action instances. Narayan et al. (2019) designed three loss functions to learn action features for temporally localizing action instances. Liu et al. (2019) introduced a multi-branch network to model the completeness of actions. However, these methods are hard to localize action boundaries accurately as no temporal information can be used.

2.3.2 Localizing Moments with Natural Language

This task is to temporally localize the video clip that contains the moment described by the query sentence. Anne Hendricks et al. (2017) introduced a moment context network for moments retrieval and released the DiDeMo dataset for this task. Ning et al. (2018) used Long Short-Term Memory (LSTM) for language encoding and adopted an attention scheme for retrieving the moment. Chen et al. (2018) introduced a temporal groundnet to exploit interactions between frames and words for final localization. Gao et al. (2017) released a new dataset Charades-STA where the start and end timestamp of the moment should be predicted for a query sentence. Ge et al. (2019) managed to learn activities from both video and language modalities via activity concepts based localizer. Xu et al. (2019) proposed a multi-task loss to build sentence-video connections. Nevertheless, all these methods used temporal annotations, which are not available in most scenarios, to build the semantic relations between languages and videos.

CHAPTER 3

Semi-supervised Learning with Self-paced Co-training

3.1 Introduction

Semi-supervised learning (SSL) aims to learn about labeled and unlabeled data by considering supervised knowledge from limited training samples and potential unlabeled data structures. Co-training (Blum and Mitchell, 1998) is one of the most classical and well-known SSL approaches that train classifiers and exchanges predictions on unlabeled samples in an iterative way. In recent years, co-training has been attracting attention and is widely used in various applications (Nigam and Ghani, 2000; Wan, 2009; Zhu et al., 2012; Do et al., 2016). The rationality of the co-training algorithm is also investigated. Blum and Mitchell (1998) proved its correctness under the assumption that samples from different views are independent given the class label. Later, Balcan et al. (2004) relaxed the conditions that the co-training algorithm would succeed when the classifier makes confident predictions on unlabeled samples in each view. However, these analyses contain strong assumptions that the pseudo labels of unlabeled samples selected in each iteration are of high confidence extent. Such assumptions are too subjective to be satisfied, especially in the early stage of the co-training algorithm. The learned classifiers might not be able to precisely pseudo-annotate unlabeled samples with an expected accuracy. This would degenerate the performance of co-training since the wrongly pseudo-labeled samples involved in training have no chance to be rectified in the latter training process.

Another issue in co-training methods is the absence of an optimization objective to measure the performance and explain the intrinsic iterative mechanism. The performance measure is generally one of the necessary elements for a machine learning method. Some studies introduced objective functions based on the assumption that predictions of different classifiers or views on samples should be consistent (Sindhwani et al., 2005b; Li et al., 2012). These co-regularization approaches encode relations of predictions from different views into a co-regularization term and turn multi-view SSL into a new convex optimization problem. However, the new objective function is often hard to optimize and the learning process is totally different from the co-training process. Thus, it is critical to provide a model with an explicit objective whose optimization process is consistent with the co-training implementation. Moreover, most existing co-training regimes are mainly implemented in two-view cases. When more views or classifiers are available, these methods are not easy to be extended. A reasonable performance measure or an objective function is necessary to inspire sound learning on training classifiers in general multi-view scenarios.

To address these issues, we propose self-paced co-training (SPaCo) in this chapter. The method differs from the previous co-training regimes mainly in two aspects: Firstly, it utilizes a “draw with replacement” manner. An unlabeled sample added to the training pool can be removed if classifiers in later training rounds identify it as a low-confidence annotated one. The pseudo label of an unlabeled sample can also be rectified based on the prediction from classifiers in later training rounds. Secondly, the rationality of

SPaCo is also analyzed in this chapter. We present that our method can obtain a better classifier when classifiers have diverse predictions on unlabeled samples. Moreover, it is substantiated that the new method can attain an evident better performance beyond current state-of-the-art co-training methods in various applications, demonstrating the superiority of the proposed SPaCo algorithm. In summary, this work makes the following contributions:

- A self-paced co-training (SPaCo) framework is presented, which can be easily applied to multiple SSL tasks. Specifically, two regularization forms are introduced, including hard and soft co-regularization terms. The SPaCo with the hard co-regularization term, follows the sample selection of conventional co-training algorithms. The soft co-regularization term would impose continuous weights on samples for cross-view sample training.
- Instead of using fixed predictions on unlabeled samples and selecting examples only based on individual predictions, our model draws the unlabeled samples with replacement and considers predictions from all views to select pseudo examples.
- The effectiveness of the proposed SPaCo is analyzed based on the ϵ-expansion theory (Balcan et al., 2004). The superiority of the proposed algorithms is comprehensively substantiated in various practical tasks.

3.2 Self-paced Co-training

3.2.1 Learning Objective

The general SPL framework introduces a weight for each training instance to decide its learning order. We attach the weight to an unlabeled instance, the attached weight can then determine the status of this sample being selected for training. Considering we have N_l labeled and N_u unlabeled training samples with M classifiers or views, we can present the SPaCo optimization problem as follows:

$$\min_{\Theta,\mathbf{V},\tilde{Y}} E = \sum_{j=1}^{M} \Big(\sum_{i=1}^{N_l} \ell_i^{(j)} + \sum_{i=N_l+1}^{N_l+N_u} \Big(v_i^{(j)} \ell_i^{(j)} + f(v_i^{(j)}, \lambda^{(j)}) \Big) \Big) + \mathcal{R}(\mathbf{V}) + \mathcal{R}(\Theta), \tag{3.1}$$

where

$$\ell_i^{(j)} = \begin{cases} \ell\big(y_i, g(\mathbf{x}_i^{(j)}; \theta^{(j)})\big), i = 1, \cdots, N_l, \\ \ell\big(\tilde{y}_i, g(\mathbf{x}_i^{(j)}; \theta^{(j)})\big), i = N_l + 1, \cdots, N_l + N_u, \end{cases}$$

where y_i and $\tilde{y}_i$ represents the annotated and predicted label of i^{th} sample. ℓ denotes the loss function and v_i^j represents the i^{th} sample weight in the j^{th} classifier. We store all predicted labels and samples weights in $\tilde{Y}$ and $\mathbf{V} \in \mathbf{R}^{N_u \times M}$, respectively. $\Theta = \{\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(M)}\}$ are the classifier model parameters. $\mathcal{R}(\Theta)$ is the regularization term on model parameters. We employ the commonly used L_2 regularization to penalize the weights in the present paper. $\mathcal{R}(\mathbf{V})$ is the specific co-regularizer imposed on the sample weights of unlabeled data.

Note that only unlabeled samples are attached with weights as the labeled ones have been annotated. When and some labels are corrupted, we can also attach weights to the labeled ones to make the model robust to the noise. In this chapter, we assume that all the labeled examples are clean. As class distribution is significant for model learning, we specify a different age parameter for each class to add unlabeled samples. The corresponding regularization term can be written as follows:

$$f(v_i^{(j)}, \lambda^{(j)}) = -\lambda_c^{(j)} v_i^{(j)},$$

where c is the pseudo label of sample $x_i^{(j)}$, and K is the total number of classes. The $\lambda^{(j)} = \{\lambda_c^{(j)} | c = 1, \ldots, K\}$ is the age parameter that controls how many unlabeled examples could be selected for training in each iteration. When λ_j is small, only most confident examples with small losses will be considered. As λ_j grows, more unlabeled examples will be gradually put into the training. As there are M views, we have to set the MK

values of the age parameters in different views, which would be hard to be tuned during training. Instead of directly setting λ_c^j, we simply define the number of selected examples where the λ_c^j can be calculated by ranking the loss values of all unlabeled samples.

The last term $\mathcal{R}(\mathbf{V})$ is to encode the intrinsic correlation among weights of different views and compensate each other by combining knowledge from all views. Without this term, the above equation will degenerate into the traditional self-training semi-supervised problem in each view since all views can be calculated separately with no influence to and from other views. We thus call $\mathcal{R}(\mathbf{V})$ as the co-regularization term since it plays a critical role in our algorithm for multi-view training. We formulate two types of co-regularization terms, including hard and soft regularization terms, and explain how these terms correlate different views.

3.2.2 Hard Co-regularization Term

For co-training style algorithms, the unlabeled samples with high prediction probability of one class in one view would be added into the training pool of the other views. In our SPaCo framework, the weight of an unlabeled example in one view would be 1 if the classifier of this view predicts its corresponding sample with high confidence. To force the algorithm into selecting this sample to others views, we ought to encourage its weight in other views also being 1. The co-regularization term for implementing this can thus be written as follows:

$$\mathcal{R}_h(\mathbf{V}) = -\gamma \sum_{p<q} (\mathbf{v}^{(p)})^T \mathbf{v}^{(q)}, \tag{3.2}$$

where $p, q \in \{1, \dots, M\}$, and $\mathbf{v}^{(p)} = \mathbf{V}_{*p}$ contains all weights of unlabeled samples in the p^{th} view. γ is the co-regularization parameter that controls how strongly the regularization is penalized.

The inner product form of the co-regularization term encodes the relationship of "sample easiness degree" between two views and encourages unlabeled samples of both views to be selected simultaneously. This co-regularization term also follows the basic strategy of co-training that most confident pseudo-labeled samples selected from one view can be used by the other views. Suppose we are minimizing Eq. (3.1) using the regularization term in Eq. (3.2) with all other parameters fixed except the weight vectors of j^{th} view, by calculating the derivative of Eq. (3.1) with respect to $v_i^{(j)}$, we have

$$\frac{\partial E}{\partial v_i^{(j)}} = \ell_i^{(j)} - \lambda_c^{(j)} - \gamma \sum_{q \neq j} v_i^{(q)}. \tag{3.3}$$

Then we can get the closed-form updating equation for $v_i^{(j)}$ as follows:

$$v_i^{(j)*} = \begin{cases} 1, \ell_i^{(j)} < \lambda_c^{(j)} + \gamma \sum_{q \neq j} v_i^{(q)}, \\ 0, otherwise. \end{cases} \tag{3.4}$$

From Eq. (3.4), we can observe that an sample with loss less than $\lambda_c^{(j)} + \gamma \sum_{q \neq j} v_i^{(q)}$ would be selected into training in the next iteration. This indicates that the confident samples of one view (with relatively smaller loss value $\ell_i^{(j)}$ in the classifier of j^{th} view) and samples selected by other views (with $v_i^{(q)} = 1$, meaning that the sample has been taken as confident ones and selected in previous training process), are prone to be selected than those with $v_i^{(q)} = 0$. Note that for an unlabeled sample, its weight can be only 0 or 1 and is related to all other views. Thus we call the regularization term in Eq. (3.2) the hard co-regularization term.

The parameter γ controls the association degree between different views. If γ is set sufficiently large with the quantity of added unlabeled samples fixed, all samples selected from other views will be chosen by the classifier of the current view. It is then equivalent to conventional co-training style algorithms in which the classifier of one view first picks samples and then puts them all into the training pool of other views. However, if predictions from one view are not reliable, we can set a small γ to combine predictions from all views to improve the robustness of predicted results on unlabeled samples.

3.2.3 Soft Co-regularization Term

The correlation information of sample confidence from different views is finely encoded in the SPaCo model by introducing the inner-product-form co-regularizer term. The proposed model can select unsupervised samples in one iteration and replace them with other samples. It makes the model choose the confident pseudo-labeled samples for the next training iteration. However, the weights on the unlabeled samples can only be 0 or 1, meaning that they can only be roughly selected or removed. Compared to the hard-term learning manner, the soft one should be more expected since it tends to more faithfully and comprehensively reflect the correlation information among different views. To this aim, we further design the following soft co-regularization term:

$$\mathcal{R}_s(\mathbf{V}) = \gamma \sum_{p<q} (\mathbf{v}^{(p)} - \mathbf{v}^{(q)})^T (\mathbf{v}^{(p)} - \mathbf{v}^{(q)}). \tag{3.5}$$

As compared to the hard co-regularization term, the meaning of this regularizer should be more evident: it is the square of the difference between weight vectors from any two views and tends to enforce similar importance weights, as well as selected pseudo-labeled samples for further training, among different views. This form is similar to the form in co-regularization style algorithms, while instead of forcing the same predictions from different views, we require that the confidence level of an unlabeled sample should be similar in disparate views. As the confidence level of a sample is intrinsically related to its prediction, the proposed co-regularization term implicitly correlates predictions of all views. In addition, pseudo-labeled samples with high confidence would also be trained to boost further model performance, which can be easily observed from the following solution forms. By taking the derivative with $v_i^{(j)}$, we can get:

$$\frac{\partial E}{\partial v_i^{(j)}} = \ell_i^{(j)} - \lambda_c^{(j)} + \gamma\big((M-1)v_i^{(j)} - \sum_{q \neq j} v_i^{(q)}\big). \tag{3.6}$$

Then we can obtain the closed-form updating equation for $v_i^{(j)}$ as follows:

$$v_i^{(j)*} = \begin{cases} 0, \ell_i^{(j)} \geq \lambda_c^{(j)} + \gamma \sum\limits_{p \neq j} v_i^{(p)}, \\ 1, \ell_i^{(j)} \leq \lambda_c^{(j)} + \gamma \sum\limits_{p \neq j} (v_i^{(p)} - 1), \\ \frac{1}{M-1} (\sum\limits_{p \neq j} v_i^{(p)} + \frac{\lambda_c^{(j)} - \ell_i^{(j)}}{\gamma}), otherwise. \end{cases} \tag{3.7}$$

It can be seen that for each $x_i^{(j)}$, $v_i^{(j)}$ is also calculated as 0 when the $\ell_i^{(j)}$ is larger than the sum of $\lambda_c^{(j)} + \gamma \sum_{q \neq j} v_i^{(q)}$, similar as the 0-weight case in hard SPaCo model. Otherwise, as $\ell_i^{(j)}$ linearly decreases to $\lambda_c^{(j)} + \gamma \sum_{p \neq j} (v_i^{(p)} - 1)$, $v_i^{(j)}$ would linearly increase to 1. This means the sample weight is possible to be arbitrary value in [0,1]. We thus call the term in Eq. (3.5) as the soft co-regularization term. Only for those pseudo-labeled samples with sufficient confidence, $v_i^{(j)}$ will be 1, i.e., the sample will be used in the next training process. There are two possible types of such confident samples: the sample with large $v_i^{(p)}$ for all other views, and that with relatively smaller prediction loss value $\ell_i^{(j)}$ in the current view. Both correspond to the confident samples complying with our intuition.

The parameter γ is similar to that in the hard SPaCo model. A relatively larger γ would make most of the weights of unlabeled samples tend to be one, and a smaller one would make these weights 0. The difference is that it leads to a soft weight updating scheme in soft SPaCo cases and thus tends to get a more accurate evaluation of samples' importance weights.

3.3 Optimization

In the previous section, we propose the SPaCo model with hard and soft co-regularization terms, respectively. The alternative optimization strategy (AOS) can then be employed to solve both models. In this section, we first introduce the traditional optimization strategy in which each view is updated in a serial way. Then to speed up the learning process, we introduce the parallel amelioration of our algorithm.

3.3.1 Alternative Optimization Strategy

Initiation: The first step is to initialize the parameters in the proposed model. The weight matrix $\mathbf{V} \in \mathcal{R}^{N_u \times M}$ is initiated as a zero matrix. Classifiers in all views are first trained based on labeled set, and predictions are made on the unlabeled set. Labels of all unlabeled samples are set based on the average predictions from classifiers in all views. Age parameter $\lambda_c^{(j)}$ in each view is initialized with a small value to allow the most confident unlabeled samples of each class in all views to be selected. The strategy of tuning $\lambda_c^{(j)}$ will be discussed in Section 3.5.2. The $\mathbf{V}$ is then updated based on the rule in Eq. (3.4) or Eq. (3.7) for picking confident unlabeled samples for each view.

Update $\mathbf{v}^{(j)}$: For the current j^{th} view, the weight vector $\mathbf{v}^{(j)}$ is updated for preparing training samples. By taking derivatives with each $v_i^{(j)}$, we can easily get the selected pseudo-labeled into the training process (i.e., obtain their weights). As discussed before, the solution for updating $v_i^{(j)}$ given hard and soft co-regularization terms are presented in Eq. (3.4) and Eq. (3.7), respectively.

Update $\theta^{(j)}$: The training pool in the current view now contains labeled and newly selected pseudo-labeled samples. The problem of updating parameters $\theta^{(j)}$ now becomes the following sub-optimization problem:

$$\min_{\theta^{(j)}} \sum_{i=1}^{N_l} \ell_i^{(j)} + \sum_{i=N_l+1}^{N_l+N_u} v_i^{(j)} \ell_i^{(j)} + \mathcal{R}(\theta^{(j)}), \tag{3.8}$$

This is a standard objective function for supervised learning and can be easily solved by off-the-shelf toolkits. For sample, if a neural network is adopted and the cross-entropy loss is used for image classification tasks, the parameter $\theta^{(j)}$ is simply optimized using the SGD algorithm. Our proposed method has no limitation on the base classifiers which makes it applicable for general applications.

Update $\tilde{Y}$: The newly learned classifier is expected to perform gradually better since more confident data are expected to be used for training. It is then reasonable to make use of the updated predictions on the unlabeled set to update their pseudo-labels. It can be easily done by solving the following minimization sub-problem:

$$\min_{\tilde{y}_i} \sum_{j=1}^{M} v_i^{(j)} \ell(\tilde{y}_i, g(\mathbf{x}_i^{(j)}; \theta^{(j)})). \tag{3.9}$$

It is easy to prove that the global optimum of the above problem can be obtained by setting the pseudo-label $\tilde{y}_i$ as the weighted average predictions directly. Note that some of the wrongly pseudo-labeled samples can be rectified in this manner.

Augment λ and Update $\mathbf{v}^{(j)}$: Once pseudo-labels of unlabeled data are refreshed, $\lambda = \{\lambda_c^{(j)} | c \in [K], j \in [M]\}$ is enlarged to allow more samples with lager loss values, i.e., the unlabeled samples with lower confidences, into the training pool in the next iteration. Specifically, we increase the number of selected unlabeled samples at each iteration in the same way employed by co-training algorithms. Suppose that we increase the number of unlabeled samples by 5 for each class in the current iteration. We first calculate losses of all unlabeled examples by Eq. (3.4) and Eq. (3.7), and then sort the losses for each class in the ascending order. We then set λ_c^j as the value of the top 6^{th} loss for the c^{th} class under hard and soft regularization term settings, respectively.

We then update $\mathbf{v}^{(j)}$ to pick the specific number of unlabeled samples for the next iteration. There are chances that samples selected for previous training (i.e., weight equals 1 in the previous iteration) may not be selected (i.e., the weight is updated as 0) if their loss values increase to a large evident value. Our algorithm possesses the capability of "draw with replacement" instead of "draw without replacement" manner in current co-training approaches.

The iteration will be terminated when all unlabeled samples have been involved in

Algorithm 1 Serial SPaCo

```
Input: Labeled and unlabeled samples, co-regularization parameter $\gamma$, and iteration rounds T.
Output: $\Theta = \{\theta^{(j)} | j = 1, \ldots, M\}$.
Initialize weight matrix $\mathbf{V}$, age parameter $\lambda$, and training round $t = 1$.
Update $\Theta$
Update $\mathbf{V}$
while $t < T$ || no available data  do
    for $vid \leftarrow 1$ to M do
        Update $\mathbf{v}^{(vid)}$: prepare training pool for current view
        Update $\theta^{(vid)}$: learn a new classifier based on added samples
        Update $\tilde{Y}$: renew predictions on all unlabeled samples
        Augment $\lambda$: allow more samples being picked
        Update $\mathbf{v}^{(vid)}$: select confident samples for other views
    end for
end while
Return $\Theta$
```

training, or the preset most significant iteration number is reached. Algorithm 1 presents the entire optimization procedure. It is easy to see that the training steps of Algorithm 1 are very similar to the standard co-training method proposed in Blum and Mitchell (1998). Specifically, it also iteratively trains classifiers on different views by exchanging labels of unlabeled samples. This shows that the proposed algorithm is closely related to other co-training approaches. Yet beyond others, the proposed algorithm complies with an optimization implementation on an underlying self-paced learning model. This model makes the co-training process capable of being quickly executed in multi-view scenarios (more than three views) under sound objective guidance and provides some novel, insightful understandings of the intrinsic effectiveness mechanism under the co-training approach.

3.3.2 Parallel Training

The problem with the above training strategy lies in its training speed. Since the parameters of all views need to be updated one by one serially, the training time will

Algorithm 2 Parallel SPaCo

```
Input: Labeled and unlabeled samples, co-regularization parameter $\gamma$, and iteration rounds T.
Output: $\Theta = \{\theta^{(j)} | j = 1, \dots, M\}$.
Initialize weight matrix $\mathbf{V}$, age parameter $\lambda$, and training round $t = 1$.
Update $\Theta$
Update $\mathbf{V}$:
while $t < T$ || no available data do
    Update $\mathbf{V}$: prepare training data for all views
    Update $\Theta$ : train classifiers for all views in a distributed way
    Update $\tilde{Y}$: renew predictions on all unlabeled samples
    Augment $\lambda$: allow more samples being picked
end while
Return $\Theta$
```

increase, especially in the cases that many views are available for the problem or multi-modal information is expected to be employed. The training time becomes critical when deep neural networks are adopted for each view. The parallel training manner should be not only necessary but also a must. Therefore, we develop a parallel learning strategy for the proposed SPaCo model, as summarized in Algorithm 2.

We can also reduce the costs of communication between different views. The updating rule for importance weight vectors is thus simplified in each view based on weights of all views learned from the previous iteration. If a hard co-regularization term is adopted, $v_i^{(j)}$ is determined by its loss and weights from all other views, and the solution in Eq. (3.4) is modified as follows:

$$v_{i,t}^{(j)*} = \begin{cases} 1, \; \ell_i^{(j)} < \lambda_c^{(j)} + \gamma \bar{v}_{i,t-1}, \\ 0, \; otherwise. \end{cases} \tag{3.10}$$

where t denotes the current training round, and $\bar{v}_{i,t-1} = \frac{1}{M}\sum_j v_{i,t-1}^{(j)}$ is the average weight for x_i in the previous $(t-1)^{th}$ training round. Similarly, given the soft co-regularization term, we can rewrite the updating rule for $v_i^{(j)}$ as below:

$$
v_{i,t}^{(j)*} = \begin{cases} 0, \; \ell_i^{(j)} \geq \lambda_c^{(j)} + \gamma \bar{v}_{i,t-1} \\ 1, \; \ell_i^{(j)} \leq \lambda_c^{(j)} + \gamma(\bar{v}_{i,t-1} - 1) \\ \bar{v}_{i,t-1} + \dfrac{\lambda_c^{(j)} - \ell_i^{(j)}}{\gamma}, \; otherwise. \end{cases} \tag{3.11}
$$

The updating rule, which defines the sample weight in each view, is now correlated with the average sample weight in all views. The classifier of each view can be thus optimized in a distributed way. The training of classifiers in all views can be deployed on several threads or machines, and the bottleneck of training time in one iteration depends on the classifier with the longest training time among all views. Parallel learning can also be quickly executed in distributed machines when multiple deep neural networks are employed. It is useful if we employ multi-classifiers for each view to improve further the probability of selecting correct pseudo-labeled samples.

3.4 Rationality Exploration

Similar to the theoretical support for traditional co-training methods, we prove that the proposed SPaCo algorithm is also a PAC learning algorithm (Valiant, 1984) under certain assumptions. Since traditional investigations mainly focus on the rationality of data with only two views, it is then critical to guarantee the effectiveness of the learning algorithm when applied to the case with more available views. To make this feasible, we define a more general version of ϵ-expansion condition as used in Balcan et al. (2004) and prove its effectiveness when being applied to multi-view data.

Let D be the distribution over a sample space $X = X^1 \times \cdots \times X^M$, and X^+ and X^- denote the positive and negative regions of X, respectively (for simplicity we assume we are doing binary classification). Let D^+ and D^- denote the marginal distributions of D over X^+ and X^-, respectively. Following the definition in Balcan et al. (2004), we denote $\mathbf{S} = \{\mathbf{S}^{(j)} | i = 1, \ldots, M\}$ as confident sets in each view ($\mathbf{S}^j \subseteq X^{j^+}$), and then $Pr(|\bigvee_{j\in[M]} \mathbf{S}^{(j)}|) = Pr(\mathbf{S}^{(1)} \vee \cdots \vee \mathbf{S}^{(M)})$ denotes the probability mass on sample for which we are confident about at least one view. The multi-view ϵ-expansion condition is defined as follows:

Definition 3.1. D^+ is ϵ-expanding if the following inequality holds:

$$Pr\Big(\Big|\bigoplus_{j\in[M]} \mathbf{S}^{(j)}\Big|\Big) \geq \epsilon \, \min\Big(Pr\Big(\Big|\overset{\geq 2}{\bigvee_{j\in[M]}} \mathbf{S}^{(j)}\Big|\Big), Pr\Big(\Big|\bigwedge_{j\in[M]} \bar{\mathbf{S}}^{(j)}\Big|\Big)\Big),$$

where $Pr(|\bigoplus_{j\in[M]} \mathbf{S}^{(j)}|)$ denotes the probability mass on samples for which we are confident about only one view, $Pr(|\overset{\geq 2}{\bigvee_{j\in[M]}} \mathbf{S}^{(j)}|)$ denotes the probability mass on samples being confident at least two views, and $Pr(|\bigwedge_{j\in[M]} \bar{\mathbf{S}}^{(j)}|)$ denotes the probability of samples which none of views are confident about. $Pr(|\bigvee_{j\in[M]} \mathbf{S}^{(j)}|) = Pr(|\overset{\geq 2}{\bigvee_{j\in[M]}} \mathbf{S}^{(j)}|) + Pr(|\bigoplus_{j\in[M]} \mathbf{S}^{(j)}|)$ and $Pr(|\bigvee_{j\in[M]} \mathbf{S}^{(j)}|) + Pr(|\bigwedge_{j\in[M]} \bar{\mathbf{S}}^{(j)}|) = 1$.

Definition 3.1 is a more general version compared to the definition in Balcan et al. (2004). Based on this multi-view ϵ-expansion condition, we have the following two lemmas.

Lemma 3.1. *Suppose* $Pr(|\overset{\geq 2}{\bigvee_{j\in[M]}} \mathbf{S}^{(j)}|) \leq Pr(|\bigwedge_{j\in[M]} \bar{\boldsymbol{S}}^{(j)}|)$ *and* $Pr(\boldsymbol{T}^{(j)}||\bigvee_{j\in[M]} \mathbf{S}^{(j)}|) \geq 1 - \epsilon^{(j)}$ *for every* $\epsilon^{(j)} \leq \frac{\epsilon}{8}$, *and then* $Pr(|\overset{\geq 2}{\bigvee_{j\in[M]}} \mathbf{T}^{(j)}|) \geq (1+\frac{\epsilon}{2}) Pr(|\overset{\geq 2}{\bigvee_{j\in[M]}} \mathbf{S}^{(j)}|)$ *where* $\mathbf{T}^{(j)} = \mathbf{S}^{(j)}_{t+1}$ *denotes the updated confident region of* i^{th} *view.*

Proof:

$$\begin{aligned}
Pr(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{T}^{(j)}|) &\geq Pr_{p\neq q}\Big(\mathbf{T}^{(p)}\wedge\mathbf{T}^{(q)}\Big)\\
&\geq Pr_{p\neq q}\Big(\mathbf{T}^{(p)}\wedge\mathbf{T}^{(q)}\Big||\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big)Pr\Big(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big)\\
&\geq (1-\epsilon_p-\epsilon_q)Pr\Big(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big)\\
&\geq (1-\frac{\epsilon}{4})(1+\epsilon)Pr\Big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\Big)\\
&\geq (1+\frac{\epsilon}{2})Pr\Big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\Big).
\end{aligned}$$

Lemma 3.2. *Suppose* $Pr(|\bigvee_{j\in[M]}^{\geq 2}\boldsymbol{S}^{(j)}|) > Pr(|\bigwedge_{j\in[M]}\bar{\mathbf{S}}_i|)$ *and let* $\alpha = 1-Pr(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|)$, *if* $Pr\big(\boldsymbol{T}^{(j)}\big||\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\big) > 1-\alpha\epsilon^{(j)}$ *for every* $\epsilon^{(j)} < \frac{\epsilon}{8}$, *and then* $Pr(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{T}^{(j)}|) \geq (1+\frac{\alpha\epsilon}{8})Pr(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|)$.

Proof:

$$\begin{aligned}
\alpha &= Pr\Big(|\bigwedge_{j\in[M]}\bar{\mathbf{S}}_i|\Big)+Pr\Big(|\bigoplus_{j\in[M]}\mathbf{S}^{(j)}|\Big)\\
&\geq (1+\epsilon)Pr\Big(|\bigwedge_{j\in[M]}\bar{\mathbf{S}}_i|\Big)\\
&\geq (1+\epsilon)(1-Pr\Big(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big)).
\end{aligned}$$

We get $Pr\Big(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big) \geq 1-\frac{\alpha}{1+\epsilon}$.

$$\begin{aligned}
Pr(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{T}^{(j)}|) &\geq Pr_{p\neq q}\Big(\mathbf{T}^{(p)}\wedge\mathbf{T}^{(q)}\Big||\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big)Pr\Big(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big)\\
&\geq (1-\frac{\alpha\epsilon}{4})(1-\frac{\alpha}{1+\epsilon})\\
&\geq (1-\alpha)(1+\frac{\alpha\epsilon}{8})\\
&\geq (1+\frac{\alpha\epsilon}{8})Pr\Big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\Big).
\end{aligned}$$

Based on above lemmas, we have:

Theorem 3.1. *Let ϵ_{fin} and δ_{fin} be the desired accuracy and confidence parameters. Suppose that the multi-view ϵ-expanding condition is satisfied in each training round, and our algorithm trains classifier in each view with accuracy and confidence parameters set to $\frac{\epsilon\cdot\epsilon_{fin}}{8}$ and $\frac{\delta_{fin}}{K}$, respectively. After running the SPaCo for $K = O(\frac{1}{\epsilon} log \frac{1}{\epsilon_{fin}\cdot\rho_{init}})$ rounds, we can then achieve the error rate as follows:*

$$Pr\big(E_{(\mathbf{x},y)\sim D}(\ell(y, g(\mathbf{x}_{,}\Theta)) < \epsilon_{fin}\big) \geq 1 - \delta_{fin}$$

Proof: For $i \geq 1$, assume that $S_i^{(j)} \subseteq X^{(j)+}$ is the confident set in each view after step $i-1$ of self-paced co-training. Define $p_i = Pr(|\bigvee_{j\in[M]}^{\geq 2} \mathbf{S}_i^{(j)}|)$, $q_i = Pr(|\bigwedge_{j\in[M]} \bar{\mathbf{S}}_i|)$, and $\alpha_i = 1 - p_i$, with all probabilities with respect to D^+. We try to bound $Pr(|\bigvee_{j\in[M]} \mathbf{S}_n^{(j)}|)$ after K rounds of iteration. After each training round, we get that with probability $1 - \frac{\delta_{fin}}{K}$, we have:

$$Pr\big(\mathbf{S}_{i+1}^{(j)}\big|| \bigvee_{j\in[M]} \mathbf{S}_i^{(j)}|\big) \geq 1 - \frac{\epsilon_{fin}\cdot\epsilon}{8}.$$

After first iteration, with probability $1 - \frac{\delta_{fin}}{N}$, we can get:

$$p_1 = Pr\big(|\bigvee_{j\in[M]}^{\geq 2} \mathbf{S}_1^{(j)}|\big) \geq (1 - \frac{\epsilon}{4}) Pr(|\bigvee_{j\in[M]} \mathbf{S}_0^{(j)}|) \geq (1 - \frac{\epsilon}{4})\rho_{init}.$$

Now we consider that for $i \geq 1$, If $p_i \leq q_i$, we can obtain that with probability $1 - \frac{\delta_{fin}}{K}$, we have $Pr\big(|\bigvee_{j\in[M]}^{\geq 2} \mathbf{S}_{i+1}^{(j)}|\big) \geq (1 + \frac{\epsilon}{2}) Pr\big(|\bigvee_{j\in[M]}^{\geq 2} \mathbf{S}_i^{(j)}|\big)$ using Lemma 1. Similarly, if $p_i > q_i$, with probability $1 - \frac{\delta_{fin}}{K}$, we have $Pr\big(|\bigwedge_{j\in[M]}^{\geq 2} \mathbf{S}_{i+1}^{(j)}|\big) \geq (1 + \frac{\alpha_i \epsilon}{8}) Pr\big(|\bigwedge_{j\in[M]}^{\geq 2} \mathbf{S}_i^{(j)}|\big)$ using Lemma 2. And with probability at least $1 - \delta_{fin}$, learning algorithm $A^{(j)}$ of each view will success after K training rounds.

From above observations, we have $p_{i+1} = (1 + \frac{\epsilon}{16})^i (1 - \frac{\epsilon}{4})\rho_{init}$ as long as $p_i \leq \frac{1}{2}$. Then the required training rounds for $p_{K_1} > \frac{1}{2}$ can be calculated by solving the following inequality:

$$(1+\frac{\epsilon}{16})^{K_1}(1-\frac{\epsilon}{4})\rho_{init} > \frac{1}{2}.$$

We then have $K_1 > \frac{\log\frac{2}{4-\epsilon}+\log\rho_{init}}{\log(1+\frac{\epsilon}{16})}$. Since $\frac{\log\frac{2}{4-\epsilon}+\log\rho_{init}}{\log(1+\frac{\epsilon}{16})} < \frac{32}{\epsilon}log\frac{1}{\rho_{init}}$, we have $p_{K_1} > \frac{1}{2}$ after iterations $K_1 = O(\frac{1}{\epsilon}log\frac{1}{\rho_{init}})$. At this point, we compare the relationship between α_i and α_{i+1}. From Lemma 2, we can get:

$$\begin{aligned} 1-\alpha_{i+1} &\geq (1+\frac{\alpha_i\epsilon}{8})(1-\alpha_i) \\ 1+\frac{\alpha_i\epsilon}{8}-\frac{\epsilon}{8} &\geq \frac{\alpha_{i+1}}{\alpha_i} \\ 1-\frac{\epsilon}{16} &\geq \frac{\alpha_{i+1}}{\alpha_i}. \end{aligned}$$

Given $p_{K_1} > \frac{1}{2}$, after K_2 iterations, we have $\frac{\alpha_{K_1+K_2}}{\alpha_{K_1}} \leq (1-\frac{\epsilon}{16})^{K_2}$. Due to $\alpha_{K_1} = 1-p_{K_1} < \frac{1}{2}$, we can then make $\alpha_{K_1+K_2} \leq \epsilon_{fin}$ by calculating the required training rounds through solving the following inequality:

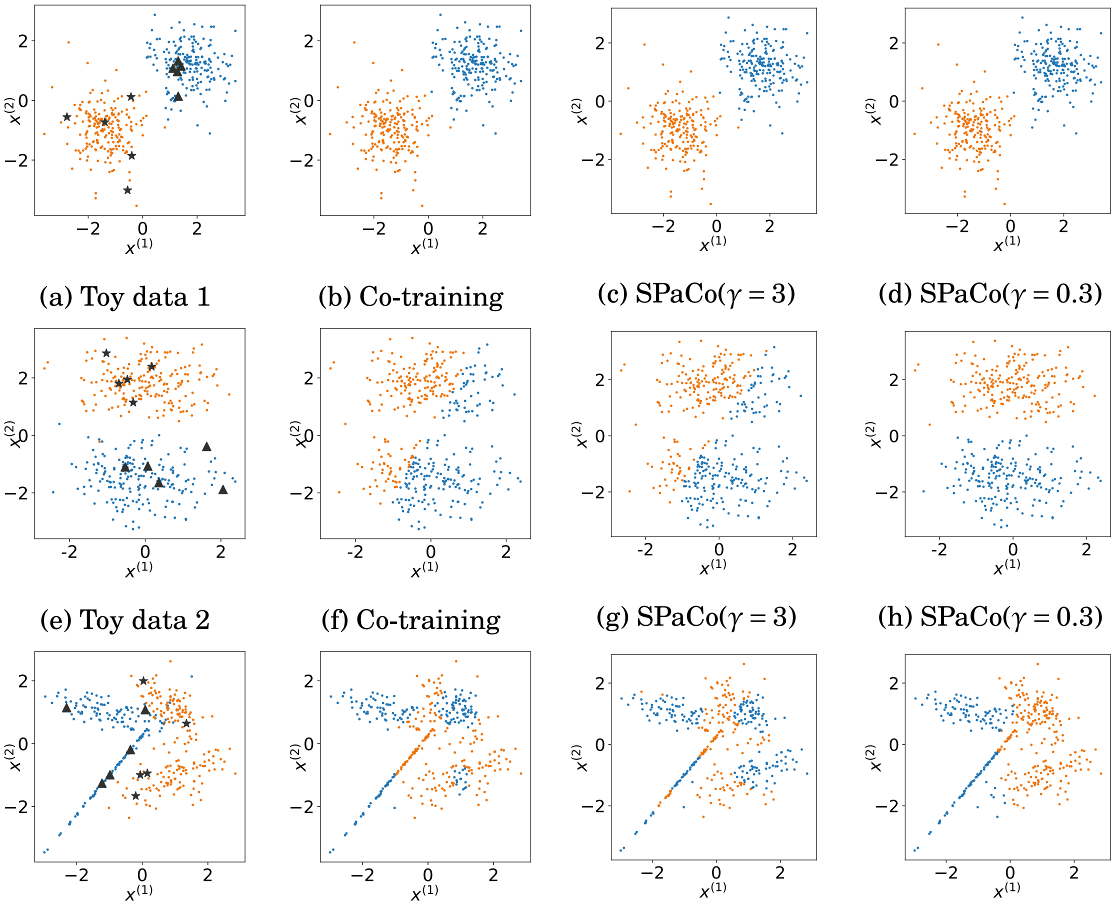
$$\frac{1}{2}(1-\frac{\epsilon}{16})^{K_2} \leq \epsilon_{fin}.$$

By solving the above equation, we obtain that after iterations $K_2 = O(\frac{1}{\epsilon}log\frac{1}{\epsilon_{fin}})$, we have $p_{K_1+K_2} < 1-\epsilon_{fin}$. Therefore, after a total of $O(\frac{1}{\epsilon}log\frac{1}{\epsilon_{fin}\cdot\rho_{init}})$ rounds, we can have a predictor of desired accuracy with the desired confidence.

As a result, the rationality of our proposed algorithm can also be supported in terms of traditional PAC theory. The convergence of the model with both serial and parallel training algorithms can be guaranteed as the definition 3.1 is fulfilled. To the best of our knowledge, this is the first time that the expansion theory is analyzed for general multi-view semi-supervised learning.

3.5 Experiments

To validate the performance of the proposed method, we conduct experiments on five tasks. First, we compare our proposed SPaCo with classical co-training on 3 toy sam-



(a) Toy data 1 (b) Co-training (c) SPaCo($\gamma = 3$) (d) SPaCo($\gamma = 0.3$)

(e) Toy data 2 (f) Co-training (g) SPaCo($\gamma = 3$) (h) SPaCo($\gamma = 0.3$)

(i) Toy data 3 (j) Co-training (k) SPaCo($\gamma = 3$) (l) SPaCo($\gamma = 0.3$)

Figure 3.1: Toy problems for co-training. The first column is toy data generated by different Gaussian distributions, (a) and (e) are two-Gaussian data in which each distribution corresponds to one class and (i) is four-Gaussian data in which two distributions correspond to one class. The canonical co-training results on toy data are shown in the second column. Last two columns are results of SPaCo with different γ. The blue and yellow dots denote the samples from two classes, and black triangles and stars are labeled points.

ples. The progress of how each view selects pseudo-labeled examples in a "draw with replacement" manner is also visualized. We also conduct experiments on multi-view text classification, person re-identification, image recognition and object detection tasks.

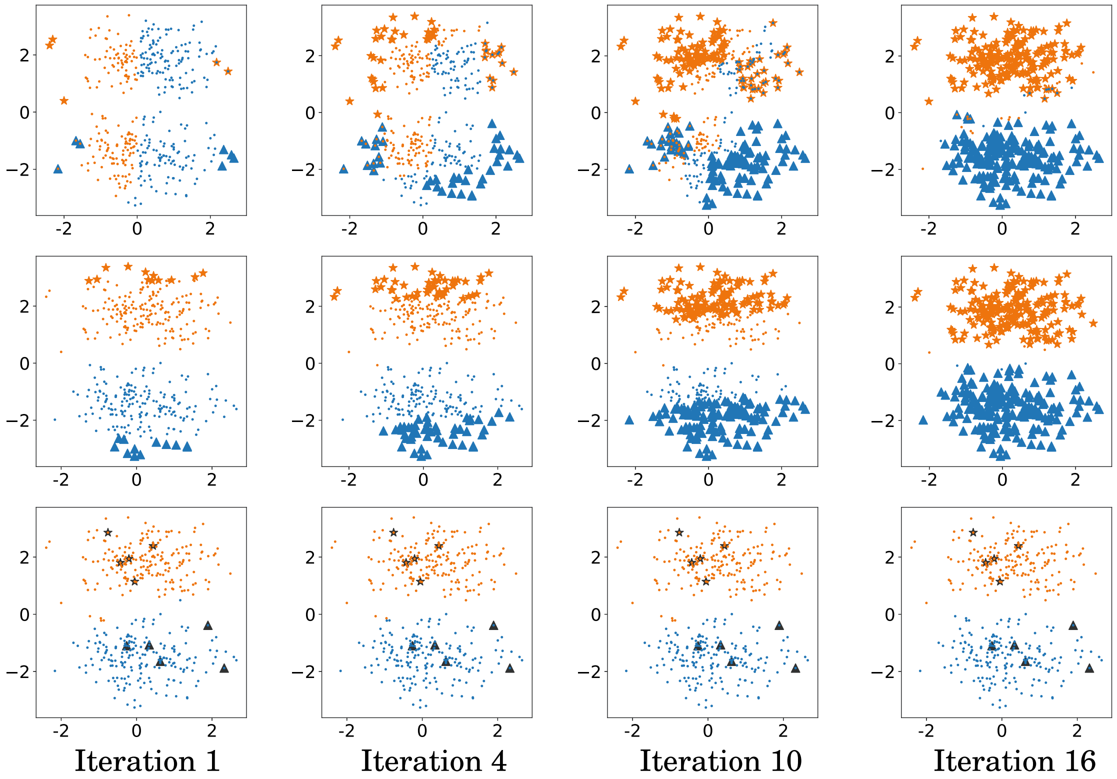


Figure 3.2: Visualized illustrations over the selected unlabeled examples during iterations of our method. Yellow and blue dots denote the predictions on unlabeled examples, respectively. Yellow stars are the selected pseudo-labeled examples of the first class, and blue triangles denote the pseudo-labeled examples of the second class. The first row presents the view using features along the vertical axis, and the second row represents the view using features along the horizontal axis. The third row is the fused predictions from both the first and the second views. Black triangles and stars denote the labeled points.

3.5.1 Toy Data

First, we display three 2D toy classification tasks to visualize the co-training results in Figure 3.1. For each of these 2D problems, we assume that one view only contains one single feature. The traditional co-training algorithm iteratively trains the classifier of each view and adds the most confident unlabeled samples into the training pool of the other view. In SPaCo, we use the hard co-regularization term with $\gamma = 3$ and 0.3, respectively. All samples are generated using scikit-learn Python module (Pedregosa et al., 2011).

The first example is a two-Gaussian case where the two view features of a sample are its two coordinates $x^{(1)}$ and $x^{(2)}$, respectively. Obviously, each view is used to train the classifier for finely separating all samples. SVM with linear kernel function is employed as a base classifier in this case with hinge-loss as its loss function. The canonical co-training handles this problem very well since every view is sufficient to train a classifier, and both views are conditionally independent. Our SPaCo algorithm can also solve this case with γ set to different values.

For the second toy data depicted, only one view feature $x^{(2)}$ can be used to get the correct classifier while $x^{(1)}$ is irrelevant to the classification task. In this case, the traditional co-training fails to separate two clusters since wrong pseudo-labeled samples are selected in the earlier training stage by using the $x^{(1)}$ feature. The SPaCo with a large γ, approximately degenerated into the traditional co-training algorithms, also encounters such issue, while with a relatively small γ, this phenomenon can be relieved since both predictions are considered when adding pseudo-labeled samples, and wrongly labeled ones would be removed in the latter training process even when they are wrongly picked into training pool in the earlier iterations attributed to the "draw with replacement" property of our method. We visualize the process of how the classifier in each view selects unlabeled examples with $\gamma = 0.3$ in Figure 3.2. Predictions from four iterations are presented in the figure. We can obtain the view using the feature $x^{(1)}$ which fails to give right predictions and select some wrongly pseudo-labeled examples during iteration 4 to 10. However, these wrongly labeled examples are rectified in the 16 round. Moreover, some examples selected in early iterations are removed in later training data. This validates that although the first view is bad for generating a good classifier, we can relieve its influence by setting the γ to a small value. By allowing more unlabeled examples into the training, the boundary of each class is also updated and these correct pseudo-labeled examples contribute to the improvement of the classification ability.

Table 3.1: Reuters multilingual dataset summarization. #dim is the dimension of corresponding language, #docs, #c, #l, #u, and #t are the numbers of documents, categories, labeled samples, unlabeled samples and test samples, respectively.

Language	#docs	(%)	#dim	c	#l	#u	#t
English	18,758	16.78	21,531	6	84	2,916	18,674
French	26,648	23.45	24,893	6	84	2,916	26,564
German	12,342	26.80	11,547	6	84	2,916	12,258
Italian	29,953	21.51	34,279	6	84	2,916	29,869
Spanish	24,039	11.46	15,506	6	84	2,916	23,855

Table 3.2: Results for Reuters with different semi-supervised learning algorithms. Mean accuracy with deviation for all competing methods are presented.

	SVM	TSVM	Co-LapSVM	Co-Label	SPaCo(hard)	SPaCo(soft)
Accuracy	66.79±1.11	69.34±1.22	69.34±0.82	72.45±1.12	73.28±1.23	**73.83**±**0.99**

Both of the above cases are linearly separable ones. The third experiment is a more intricate one in which the classification boundary is nonlinear. As shown in the figure, each class of the data is related to a two-Gaussian distribution. We also change the linear kernel function with radial basis function for producing a nonlinear decision surface. The traditional co-training and SPaCo with a large γ both fail to get the right classifier. However, the SPaCo with a smaller γ can learn a good decision boundary in this case, showing its capability in recovering the nonlinear structure under an appropriate γ.

These toy experiments indicate that our SPaCo method with a relatively large γ possesses similar characteristics to the traditional co-training algorithm, and SPaCo with a proper small γ performs better than, at least as well as the traditional co-training model. Therefore, SPaCo model can be viewed as a more adaptive co-training model for various multi-view data structures.

3.5.2 Classification on Multi-view Features

We also evaluate our SPaCo model for multi-view semi-supervised learning on the Reuters multilingual dataset in Amini et al. (2009). This dataset contains newswire

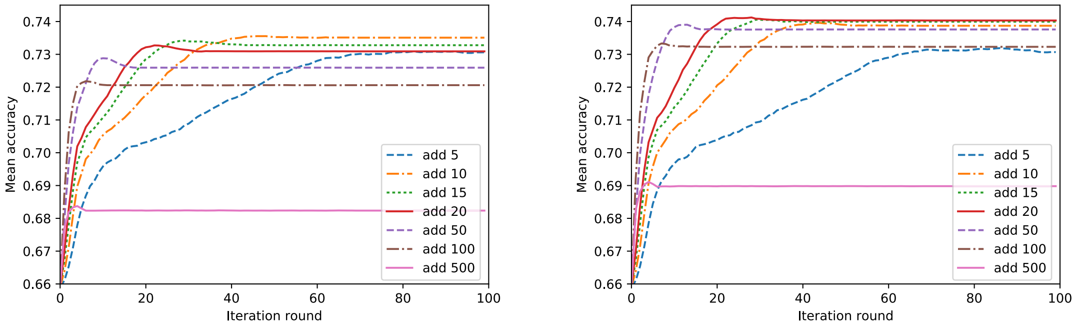


Figure 3.3: Convergence tendency of accuracy for SPaCo with hard and soft regularization terms under different λ updating strategy, and λ is adjusted by the number of samples added in each iteration. The left figure is the trend of the mean accuracy on the test set over iteration rounds for SPaCo with hard regularization term, and the right figure is the result for SPaCo with soft regularization term.

articles written in 5 languages, including *English, French, German, Italian* and *Spanish*, so there are 5 views in total. Each language is categorized into six classes: *C15 (Performance), CCAT (Corporate/Industrial), E21 (Government Finance), ECAT (Economics), GCAT (Government Social), M11 (Equity Markets)*. All documents in the dataset are represented as a bag of words, using a TFIDF-based weighting scheme. Moreover, each document in one language is translated into other four languages using the statistical machine translation system PORTAGE.

To compare with other multi-view semi-supervised algorithms, we follow the experiment setting as described in Xu et al. (2016). For each language class, 14 and 486 documents are selected as labeled and unlabeled training samples, respectively. Thus a total number of 84 and 2916 documents are used as the labeled and unlabeled data for each language. The rest of all the samples are used as test data. Detailed information of this dataset is listed in Table 3.1. SVM with a linear kernel is employed as a base classifier for each view, and a one-versus-all strategy is employed for the multi-class task. The corresponding loss function in our model is thus the sum of k hinge loss function values. All experiments are repeated for five times with random data partitions.

We first analyze the converge rate for SPaCo with hard and soft regularization terms under different λ tuning strategies. Since λ is hard to be tuned for choosing unlabeled samples in each iteration, we specify the value for λ by controlling the number of unlabeled samples after every update. The mean accuracy on the test set with two settings is displayed in Figure 3.3. We employ seven λ tuning strategies by setting the increment of selected unlabeled samples as 5, 10, 15, 20, 50, 100 and 500 for each class in every iteration, and γ is set as 0.3 in this experiment. Results of 100 iterations under these settings are presented for better comparison.

From Figure 3.3, it can be seen that our SPaCo algorithm with both hard and regularization terms under all λ settings converges and improves the performance of initialized model which are only trained on the labeled set. The increment of the selected unlabeled sample is in direct ratio with the converging rate of the proposed model but may degenerate its performance. Adding more unlabeled data with pseudo-labels into the training pool in one iteration would also introduce more noise data which may degenerate the model performance. Moreover, SPaCo with soft regularization term is less sensitive to the increment of selected unlabeled samples. We also compare our proposed method with other competing semi-supervised learning methods. For single view semi-supervised learning algorithms, features from all views are combined for training in SVM and TSVM (Collobert et al., 2006). Two multi-view learning methods, including CoLapSVM (Sindhwani et al., 2005a) and Co-label (Xu et al., 2016), are also compared in this experiment. The Co-LapSVM is a typical co-regularization style algorithm which introduces a prediction consistency regularization term of multi-views. The Co-Label method uses predictions from all views in every iteration with different strategies and forms a pseudo-label vector for obtaining robust predictions. For our SPaCo method, both hard and soft regularization terms are employed with $\gamma = 0.3$ and the increment quantified in each iteration is set to 15. The means and the standard deviations of accuracy of all five languages for different methods on Reuters dataset are presented in

Table 3.3: Performance comparison of all competing methods on CIFAR-10 with different labeled examples (2000 and 4000). The mean error rates (%) and standard deviation are presented. The best performance is marked in bold.

Method	CIFAR-10 (2000 examples)	CIFAR-10 (4000 examples)
LadderNetwork (Rasmus et al., 2015)	–	20.40±0.47
ImprovedGAN (Salimans et al., 2016)	19.61±2.09	18.63±2.32
TripleGAN (Chongxuan et al., 2017)	–	16.99±1.62
GoodBadGAN (Dai et al., 2017)	–	14.14±0.30
Temporal Ensembling (Laine and Aila, 2016)	15.64±0.39	12.16±0.24
Mean Teacher (Tarvainen and Valpola, 2017)	15.73±0.31	12.31±0.28
SNTG (Luo et al., 2018)	13.64±0.32	9.89±0.34
ICT (Verma et al., 2019)	**9.26±0.09**	7.66±0.17
SPaCo(Phard)	12.23±0.43	7.28±0.28
SPaCo(Psoft)	11.97±0.49	**7.05±0.24**

Table 3.2.

From the table, we observe that our SPaCo method with both hard and soft regularization terms perform better than other methods. And SPaCo with soft regularization term achieves relatively higher mean accuracy with lower deviation than that with hard one. This demonstrates that it should be beneficial to select confident unlabeled samples during training with the soft regularization term.

3.5.3 Image Classification

To compare with more latest methods using deep learning models, we conduct experiments on the image recognition task. The CIFAR-10 dataset is employed. The dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. In this experiment, 2000 and 4000 training samples are randomly selected as supervised data, respectively, and other rest training ones are taken as unsupervised samples. The same 10000 test images are used for evaluation in both cases.

We also report results of several recent methods for comparisons. The Ladder network (Rasmus et al., 2015) constrained the predictions of unlabeled examples with different perturbations. Some recent works (Salimans et al., 2016; Chongxuan et al., 2017; Dai et al., 2017) used generative adversarial networks (GAN) to generate samples for additional training. The temporal ensembling method (Laine and Aila, 2016) maintained an exponentially moving average (EMA) of predictions over epochs. Instead of averaging predictions every epoch, the mean teacher algorithm (Tarvainen and Valpola, 2017) updated the targets more frequently by average model parameters. Later Luo et al. (2018) proposed the smooth neighbors on the teacher graph (SNTG) based on previous methods, which considered the connections between data points to induce smoothness on the data manifold. Verma et al. (2019) introduced a co-regularization style algorithm, called ICT, which encourages the prediction at an interpolation of unlabeled points to be consistent with the interpolation of the predictions at those points. We report the error rate of these algorithms on the CIFAR-10 dataset in Table 3.3 for comprehensive performance comparisons.

We evaluate our model on CIFAR-10 dataset with two models employed as two views: the Wide Resnet (Zagoruyko and Komodakis, 2016) and ShakeDrop (Yamada et al., 2018). We set γ to 0.3, and iteration steps to 5 and 4 for the experiment with 2000 and 4000 labeled samples, respectively. The model is trained for 300 epochs in all iterations. The learning rate is 0.1 in the beginning and is reduced 10 times after training of 100 epochs. In each iteration, the number of selected unlabeled examples increase by the number of training examples in the last iteration. We employ the random erasing technique in Zhong et al. (2020) in the data augmentation to increase the diversity of samples from different views.

Table 3.3 summarizes the error rates obtained by all competing methods. It can be observed that our method with both hard and soft regularization terms outperform other algorithms with only 4000 labeled examples. The SPaCo model with soft co-regularization

Table 3.4: Results of SPaCo on the test data of CIFAR-10 during model iterations with 2000 examples labeled. We use ER to denote the error rate (%) and DR to denote the difference rate between predictions from two views. Shake and WRN represent the network names of two views. We report the cross entropy (CE) loss between the predictions from two views in the last column.

Steps	ER (Shake)	ER (WRN)	ER (Fuse)	DR	CE
Iteration 0	32.77	29.62	28.43	27.43	1.10
Iteration 1	24.52	25.40	23.20	19.15	0.81
Iteration 2	22.26	22.15	20.96	14.92	0.64
Iteration 3	19.05	19.14	18.33	11.39	0.48
Iteration 4	15.62	16.90	15.50	9.15	0.38
Iteration 5	12.38	13.04	12.21	6.37	0.32

term achieves 7.05% error rate, lower than that of the ICT method by 8%. Thus, it shows that the SPaCo method also works well with deep learning models on the standard image recognition task.

We further present the error rate of each model in different iterations on the test set in Table 3.4. The algorithm is performed once with the hard co-regularization term for this experiment. As more unlabeled examples are selected for updating classifiers, the error rate on the test data decreases. We also report the diversity degree among different models using difference rate (DR) and cross entropy (CE) loss. From Table 3.4, we can see that the different models indeed introduce diverse predictions. The diversity between classifiers help different views exchange information on unlabeled examples, and the model can thus add confident pseudo-labeled examples into the training to improve the model performance.

3.5.4 Person Re-identification

The person re-identification task is usually viewed as an image retrieval problem, aiming to match pedestrians from the gallery (Zheng et al., 2016). Specifically, given a person-of-interest (query), the person re-identification method aims to determine whether cameras have observed the person.

Table 3.5: Mean average precision (MAP) comparison of all competing methods on Market-1501 dataset with two views. The first line is the supervised learning result using only labeled data. Self iterative training and co-training results are presented in the second and third lines, respectively. The “Rep" denotes that the co-training algorithm is trained with the replacement strategy. The last two lines show the results of our proposed SPaCo model with hard and soft regularization terms.

	Resnet50 & Densenet121			Resnet101 & Densenet121		
	View1	View2	Final	View1	View2	Final
Base	40.5±1.57	38.5±1.20	47.7±0.78	44.5±1.06	38.5±1.20	49.8±0.85
SelfTrain	59.2±0.70	61.7±1.14	67.7±0.72	62.7±0.50	61.7±1.14	69.3±0.42
Cotrain	59.3±0.50	61.9±0.80	67.0±0.33	62.5±0.15	62.2±0.65	68.5±0.29
Cotrain(Rep)	60.1±0.72	62.5±0.77	67.7±0.42	63.1±0.64	63.2±0.52	69.3±0.39
SPaCo(hard)	61.4±0.44	63.8±0.39	68.9±0.37	63.7±0.43	64.4±0.61	70.3±0.30
SPaCo(soft)	**61.7±0.21**	**64.7±0.66**	**69.5±0.33**	**64.6±0.90**	**64.8±0.31**	**70.9±0.35**

Experiments are conducted on Market-1501 dataset for this task. This dataset contains 32,668 detected bounding boxes with persons of 1,501 identities (Zheng et al., 2015). Images of each identity are captured by six cameras at most, and two at least. According to the dataset setting, the training set contains 12936 cropped images of 751 identities and testing set contains 19,732 cropped images of 750 identities. They are directly detected by Deformable Part Model (DPM) instead of hand-drawn bounding boxes, which is closer to the realistic setting. Each identity may have multiple images under each camera. We use the provided fixed train and test sets under the single-query and multi-query evaluation settings.

In this experiment, 20% samples of training data are chosen as the labeled set, and the rest of the data are treated as unlabeled. Since images for different classes are unbalanced, we randomly select 20% labeled samples for each class to ensure that the training set contains images of every class. The experiment is repeated for ten times, and the average performance in test data is reported as the final result.

Three state-of-art deep network structures, including ResNet-50, ResNet-101 (He et al., 2016a) and DenseNet-121 (Huang et al., 2017), are used to get 3-view features for the Market-1501 dataset. All these models are pre-trained with ImageNet datasets,

Table 3.6: Rank 1 accuracy of all competing methods on Market-1501 dataset with two views. The first line is the supervised learning result using only labeled data. Self iterative training and co-training results are presented in the second and third lines, respectively. The "Rep" denotes that the co-training algorithm is trained with the replacement strategy. The last two lines show the results of our proposed SPaCo model with hard and soft regularization terms.

	Resnet50 & Densenet121			Resnet101 & Densenet121		
	View1	View2	Final	View1	View2	Final
Base	63.4±2.06	61.9±1.65	70.1±0.99	66.7±1.04	61.9±1.65	71.8±0.65
SelfTrain	79.5±0.77	81.7±0.59	85.1±0.43	81.5±0.59	82.2±0.45	86.0±0.32
Cotrain	79.5±0.41	81.7±0.45	84.6±0.32	81.4±0.37	81.8±0.56	85.6±0.42
Cotrain(Rep)	79.9±0.50	82.3±0.37	85.1±0.40	81.7±0.40	82.7±0.38	86.0±0.43
SPaCo(hard)	80.6±0.56	83.2±0.57	85.7±0.45	82.5±0.54	83.5±0.24	86.6±0.31
SPaCo(soft)	**81.0±0.57**	**83.8±0.58**	**86.3±0.27**	**82.6±0.87**	**83.6±0.37**	**86.9±0.35**

and input images are resized to 256 × 128 for ResNet50 and Resnet-101, 224 × 224 for DenseNet-101, respectively. In the training phase, images are randomly flipped and cropped for data augmentation. The cross-entropy loss function is used in this experiment, and thus the re-identification task can be well handled using the SPaCo algorithm.

For two-view experiments, two combinations, ResNet-50 with DenseNet-121 and ResNet-101 with DenseNet-121, respectively, are adopted. The base algorithm uses only labeled data in this experiment. Self-train algorithm iteratively trains each classifier and adds unlabeled samples in its view while the co-training algorithm exchanges their selected unlabeled data for training. We also trained the co-training algorithm with the "draw with replacement" strategy to make a fair comparison. Specifically, instead of fixing the pseudo-labeled examples in the training pool, the selected ones are re-selected from all unlabeled ones in each iteration. For the SPaCo method, hard and soft regularization terms are implemented with the same γ set as 0.3. The number of added unlabeled samples is proportional to the number of labeled samples. We set this proportion to 0.5 in algorithms for fair comparison. The maximum iteration round is set as 5 so that all unlabeled samples get their chance to be selected. Both mean average precision (MAP) and rank-1 accuracy measures are employed for performance evaluation. All trials were

Table 3.7: MAP and rank-1 accuracy of all competing methods on Market-1501 dataset with triple-views. The first line is the supervised learning result using only labeled data. SelfTrain result is presented in the second line. Phard and Psoft indicate that parallel training strategy is employed compared to serial training strategy. Fhard and Fsoft denote that the model does not update labels of unlabeled examples during iterations. Last six lines show the results of SPaCo method with hard and soft regularization term under different training strategies.

	Mean average precision				Rank-1 accuracy			
	Res50	Den121	Res101	Final	Res50	Den121	Res101	Final
Base	40.5±1.57	38.5±1.20	44.5±1.06	52.3±0.73	63.4±2.06	61.9±1.65	66.7±1.04	73.8±0.69
Selftrain	59.2±0.70	61.7±1.14	62.7±0.50	70.8±0.37	79.5±0.77	81.7±0.59	81.5±0.59	86.7±0.41
SPaCo(hard)	61.2±0.61	63.8±0.48	63.7±0.47	71.3±0.32	80.6±0.78	83.2±0.64	82.3±0.62	87.0±0.60
SPaCo(Fhard)	54.7±0.83	56.6±0.59	56.8±0.43	64.8±0.52	75.5±0.65	78.2±0.34	77.2±0.59	83.1±0.38
SPaCo(Phard)	61.4±0.74	63.9±0.81	63.7±0.72	71.2±0.52	80.6±0.56	83.0±0.81	82.2±0.70	86.8±0.57
SPaCo(soft)	61.6±0.75	**64.5±0.72**	**64.3±0.43**	**71.8±0.45**	81.1±0.88	**83.6±0.56**	**82.8±0.59**	**87.4±0.47**
SPaCo(Fsoft)	57.3±1.02	59.9±0.82	59.4±0.61	67.3±0.63	77.7±0.50	80.6±0.47	79.1±0.55	84.6±0.50
SPaCo(Psoft)	**61.7±0.61**	64.4±0.75	**64.3±0.38**	71.7±0.44	**81.3±0.45**	**83.6±0.62**	82.7±0.43	87.3±0.27

repeated ten times. The means and standard deviations are shown in Tables 3.5 and 3.6, in terms of both measures. Compared with the traditional co-training, the co-training with the "draw with replacement" strategy performs better. However, it is still inferior to the SPaCo method. This can be explained by the selected pseudo-labeled examples being more confident in our method. These samples are selected based on all views rather than on the predictions from a single view.

The triple view experiment combines all three networks as 3-view features for learning. The traditional co-training is not included since it can only deal with two views. All other settings, including initialized parameters and training strategy, are similar with two-view experiments. The results of all competing methods are compared in Table 3.7. The model with the serial algorithm takes MT (M indicates the number of views and T indicates the iterations) training rounds, while the model with the parallel algorithm only take T rounds as classifiers in all views are paralleled trained. In this experiment, every training round takes four hours and it would be three times faster when the model trained with the parallel algorithm.

From Tables 3.5, 3.6 and 3.7, it is seen that MAP and rank-1 accuracy of all methods are improved compared to the baseline, in which only labeled samples are involved in training. Although multi-view features are generated by employing multi-models, the integrated results are better than results using only any single model. We also fix the labels (as predicted in the first iteration) and only learn sample weights during iterations. It means that Eq. (3.9) is removed when optimizing the whole model. We can obtain that the model without updating predictions on unlabeled examples achieves much lower performance. It indicates that updating labels of unlabeled examples is necessary for generating better predictions. This can be easily explained because the pseudo-labels roughly annotated on the unsupervised samples inevitably contain many false ones, and naturally degenerate the classification capacity. While by allowing the pseudo-labels capable of being ameliorated during the training process, the wrongly annotated labels can thus to be possibly rectified. The soft regularisation term model performs better than the model with hard regularization term. It shows that soft sample weights make the model relatively more robust to unexpected noises. Besides, SPaCo in three-view combination with serial or parallel training strategy performs better than that in two-view settings. This can be explained by the fact that different network structures learn their own representations, which build up a better representation for original images from multiple aspects. However, the performance of traditional co-training performs worse compared with the self-train algorithm. Our proposed method performs better than both co-training and self-train algorithms with hard or soft regularization terms. The mechanism can explain that the proposed model considers adding pseudo-labeled samples from predictions of all views. Some wrongly labeled samples involved into training in an early stage can be removed or rectified in the later iterations. Note that the best rank 1 accuracy and MAP results under every single and combined view are achieved by our SPaCo model with soft regularization term. This indicates that some unlabeled samples may harm the model performance while the SPaCo algorithm with soft model finely relieves this negative effect.

Table 3.8: Performance comparison in average precision (AP) of all competing methods on the PASCAL VOC 2007 test set. The five compared methods make use of full image-level labels for training. Our method (the last four rows) requires only approximately four strong annotated images per class. Results on each class are shown in one column. We use Fast RCNN with VGG16 and RFCN with ResNet50 and ResNet101 as our base detectors to get 3-view features.

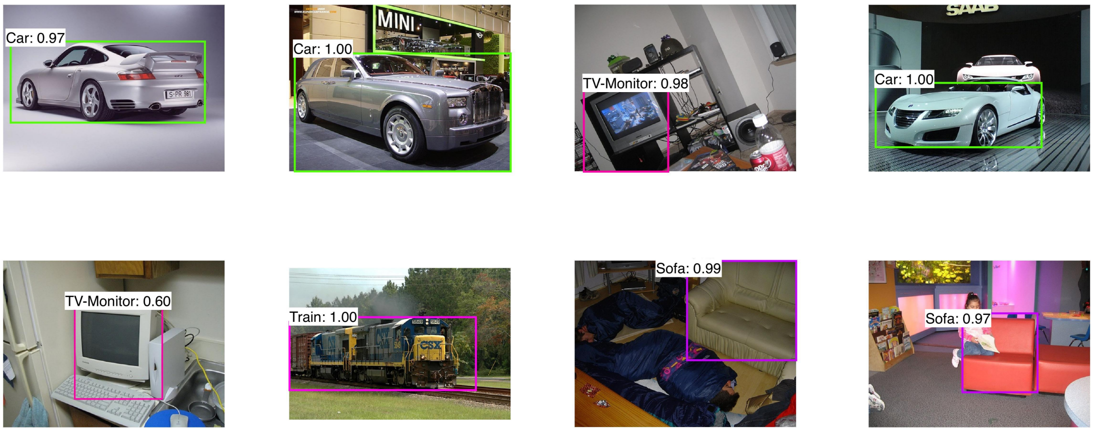
	aero	bike	bird	boat	botl	bus	car	cat	chair	cow	table	dog	hors	mbik	pers	plnt	shp	sofa	train	tv	mean
Zhang et al. (2017a)	47.4	22.3	35.3	23.2	13.0	50.4	48.0	41.8	1.8	28.9	27.8	37.7	41.6	43.8	20.0	12.0	27.8	22.9	48.9	31.6	31.3
Wang et al. (2014)	48.9	42.3	26.1	11.3	11.9	41.3	40.9	34.7	10.8	34.7	18.8	34.4	35.4	52.7	19.1	17.4	35.9	33.3	34.8	46.5	31.6
Kantorov et al. (2016)	**57.1**	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	**49.2**	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
Bilen and Vedaldi (2016)	46.4	58.3	35.5	25.9	14.0	66.7	53.0	39.2	8.9	41.8	26.6	38.6	44.7	59.0	10.8	17.3	40.7	49.6	56.9	50.8	39.3
Li et al. (2016)	54.5	47.4	**41.3**	20.8	17.7	51.9	**63.5**	46.1	**21.8**	**57.1**	22.1	34.4	**50.5**	61.8	16.2	**29.9**	40.7	15.9	55.3	40.2	39.5
Diba et al. (2017)	49.5	60.6	38.6	**29.2**	16.2	**70.8**	56.9	42.5	10.9	44.1	29.9	**42.2**	47.9	**64.1**	13.8	23.5	**45.9**	**54.1**	60.8	**54.5**	**42.8**
Vgg16-FRCNN	35.8	57.5	24.3	19.8	19.6	41.1	53.8	46.7	19.8	19.0	25.5	14.9	45.4	53.5	33.3	14.3	31.8	47.5	57.9	44.9	35.3
Res50-RFCN	41.0	51.6	28.6	16.9	23.5	49.5	46.7	47.4	14.6	24.1	23.7	16.4	41.9	53.8	25.7	14.4	28.4	33.7	57.2	47.4	34.3
Res101-RFCN	40.2	56.8	37.5	20.4	22.6	47.2	54.1	52.1	19.9	26.8	17.3	14.3	44.4	56.8	29.9	17.7	29.6	46.7	61.3	43.6	36.9
Final	42.4	**61.3**	39.4	23.5	**25.1**	50.1	57.3	**55.2**	18.8	26.4	22.4	17.0	48.2	56.3	**34.8**	19.2	30.6	49.0	**61.3**	51.0	39.5

3.5.5 Object Detection

We also conduct experiments on the object detection. It is often expensive and time-consuming to obtain amounts of labeled objects. Thus, how to use the collected small amount of labeled data together with large amounts of unlabeled samples in object detection is essential.

Object detection methods can be divided into a proposal based and proposal free types. Proposal based methods first determine bounding boxes of objects in each image and then make predictions on these given bounding boxes, while proposal free methods predict object bounding box and its class simultaneously. In this experiment, every bounding box instead of every image is viewed as a training sample. Two proposal based objected detection models, Fast RCNN (Girshick, 2015) and R-FCN (Dai et al., 2016), are adopted as base detectors, and VGG (Simonyan and Zisserman, 2014b), ResNet (He et al., 2016a) are the backbone networks for the detectors. Three combinations, Fast RCNN with VGG, R-FCN with ResNet50 and ResNet101, are treated as three separate views for each image. In the meanwhile, selective search (Uijlings et al., 2013), an unsupervised method, is used to generate proposals for both training and test images.

We evaluate our method on PASCAL VOC 2007 detection dataset (Everingham et al.,



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

Figure 3.4: Typical selected pseudo-labeled samples during training, where the bounding boxes with different colors indicate the generated pseudo-boxes by our method for different classes.

2010), which is one of the most widely used benchmarks in the object detection task. This dataset contains 10022 images annotated with bounding boxes for 20 object categories. It is officially split into 2501 training, 2510 validation, and 5011 testing images.

We randomly label 4 images for each class, which contain at least one bounding box belonging to the given class. It results in a total of 60 initial annotated images, and all the object bounding boxes in these 60 images are annotated. There are, in fact, an average of 4.2 images per class since some images have multiple bounding boxes.

For our proposed SPaCo method, classification and localization loss are employed to select unlabeled boxes during training. In the training phase, 2000 proposals for each image are generated using the selective search method, and all images are randomly flipped for data augmentation. γ is set to 0.3 for leveraging predictions from all views. The maximum iteration round is set to 5, and training epochs in each round is set to 9. We empirically use the learning rate of 0.001 for the first eight epochs and reduce it to 0.0001 for the last epochs. The momentum and weight decay are set as 0.9 and 0.0005, respectively.

Since there is rare work that only uses such few samples for object detection, we compare our proposed approach with recent state-of-art semi-supervised object detection methods which use full image-level labels from training. Li et al. (2016) decomposed this task into several steps to improve the detection accuracy. Wang et al. (2014) used the typical probabilistic latent semantic analysis to learn categories of images. Zhang et al. (2017a) simply used self-paced curriculum learning to detect objects from easy to hard. Kantorov et al. (2016) introduced context-aware guidance models to improve the localization. Bilen and Vedaldi (2016) proposed a weakly supervised detection network using selective search to generate proposals and train image-level classification based on regional features. Diba et al. (2017) employed location, segmentation and multi-sample network to solve this problem.

Table 3.8 summarizes the average precision (AP) of all competing methods on the PASCAL VOC 2007 test set. The competing methods usually use full image-level labels. In contrast, we use the same set of images but with fewer annotations: 60 annotated images and the others are free-labeled. Our proposed SPaCo method achieves 39.5% mAP, a competitive performance compared to state-of-art weakly supervised object detection algorithms. Moreover, results on some specific classes, e.g., bike, bottle and persons, even achieve the best performance.

We also display some pseudo-labeled images obtained by our method over each iteration in Figure 3.4. It is seen that the detector tends to choose images with relatively high classification confidence aggregated over the bounding boxes. After the detector is updated, it can gradually label objects in a more complicated situation. For instance, a rotated TV-Monitor is selected with higher confidence in iteration round 3 than the TV-Monitor sample selected in the first iteration round. Sofa overlapped with the person is also selected with higher confidence in the last iteration round. In contrast, the detector in the other three iterations fails to detect it.

3.6 Summary

In this chapter, we have proposed a unified self-paced co-training(SPaCo) model, which iteratively trains the classifier of each view and adds unlabeled samples into training with a “draw with replacement” learning manner. Two co-regularization terms, including hard and soft co-regularization terms, are introduced to define different strategies for unlabeled data. The rationality of our proposed SPaCo model is theoretically analyzed by PAC learning theory and SPL robustness explanation. The proof shows that the difference between views are utilized to boost the whole model performance. The experimental results also shows that the diversity between different views may result in prediction biases. In the toy data examples, both views are orthogonal to build a better classification model. From the person re-identification experiment, the model learned from ResNet and DenseNet architectures performs better than the model learned only from ResNet architectures. We can also make the model robust to the lousy view by directly imposing weights on views and learning it by a similar self-paced strategy. The weight can be learned from the weights on unlabeled examples among views. It is worth further developing such strategies to leverage different views. Moreover, when the supervised samples contain outliers or noise samples, it should be better to impose weights to suppress impacts of these noisy ones. This is also an important research topic, and we will consider it in the next chapter.

CHAPTER 4

NOISY LEARNING WITH SELF-REWEIGHTING FROM CLASS CENTROIDS

4.1 Introduction

Noisy labels are commonly encountered in practical computer vision and machine learning tasks. Existing datasets collected by search engines (Liang et al., 2016; Zhuang et al., 2017; Li et al., 2020) or annotated by crowdsourcing systems (Bi et al., 2014) usually contain a large number of noisy labels. In addition, there are also many erroneous labels even in manually annotated datasets as annotators may label data by mistake (Deng et al., 2009; Ma et al., 2020b; Wang et al., 2018; Han et al., 2019). Noisy labels in general handicap the model performance in two aspects: First, the increasing number of incorrectly annotated samples may lead to sampling effective samples insufficiently for training networks. Second, these noisy samples will harm the model optimization process by providing incorrect supervision signals. Therefore, learning with noisy data is a critical and challenging task (Raykar et al., 2010; Ren et al., 2018; Fang et al., 2019; Han et al., 2018).

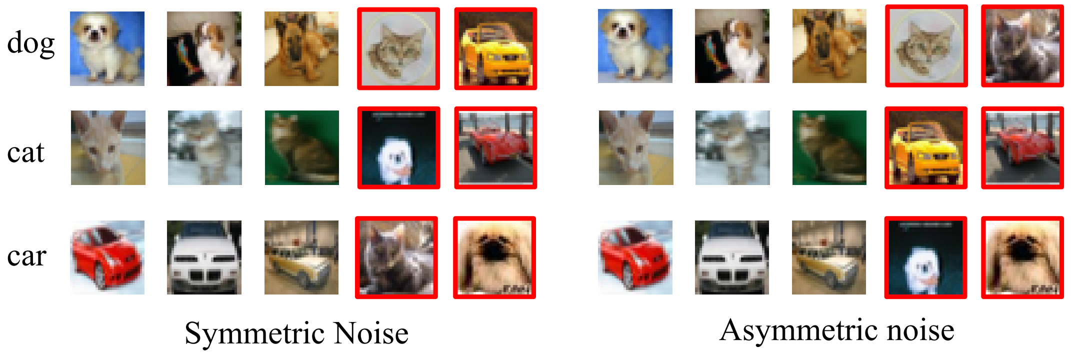


Figure 4.1: Training samples of different noise types. Noise samples are marked with red boxes. Falsely annotated labels with symmetric noise could belong to any other classes in the training set. In asymmetric noise, noise samples are only from a certain class.

A standard solution is to assign a dynamic weight to each sample when calculating the overall training loss. Impacts of noisy labeled data will be potentially reduced in training when they are weighted with smaller weights (Bi et al., 2014). For instance, assigning zero weights to falsely annotated samples in Figure 4.1 prevents a model from learning from fallacious supervision signals. Current methods generate sample weights solely based on the training losses (Shu et al., 2019; Ren et al., 2018; Ma et al., 2017). Precisely, the large training loss may imply that a sample is incorrectly annotated, and thus, a small weight will be assigned to the sample (Ren et al., 2018). However, a model often fits all data to diminish training losses where large weights might be assigned to noise samples. In this case, the assigned weights become unreliable as small weights should be produced.

We propose a novel reweighting method, namely self-reweighting from class centroids (SRCC), to ameliorate the weight assignment for noisy data. For each sample, we generate the weight by exploiting all training samples. Specifically, we first calculate the centroid of each class in the feature space. Then, the similarities between samples and class centroids are calculated to produce sample weights. Furthermore, the decision boundaries might still suffer distortions even after reweighting the noisy data. We thus leverage mixed inputs that are generated by linearly interpolating two random images and their labels to regularize the boundaries. Unlike the setting in MixUp (Zhang et al., 2017b), our data are noisy, and it is detrimental to train a model with directly interpolated labels. We leverage our learned robust class centroids to evaluate the confidence of the generated mixed data. The confidence of a mixed input is determined by the feature similarities between the mixed input and class centroids. In this fashion, assigning the sample weights of mixed inputs also takes all the data into account rather than two input labels that might be noisy. Our SRCC thus improves the reliability of sample weights and alleviates erroneous supervision signals caused by corrupted mixed inputs in training.

As the model optimization proceeds, sample features and the centroids of all classes will be updated. However, using all training samples to update class centroids at every training iteration requires a tremendous computational cost. This chapter proposes a momentum-based scheme to update class centroids online, where only the features of training samples in a batch are used to update the centroids. We update the class centroids and the model parameters alternatingly during the optimization. The effectiveness of our proposed method is analyzed to show the superiority of our algorithm. We have also conducted extensive experiments to validate the robustness of the proposed method. Experiments on both the synthetic and natural image recognition tasks demonstrate that our SRCC outperforms the state-of-the-art methods.

Above all, our contributions are summarized in the following three-fold aspects:

- We propose a simple yet effective self-reweighting from class centroids method (SRCC) to address samples with erroneous labels in deep network optimization. To reduce the impact of corrupted labels, we generate a robust sample weight for each sample based on its feature similarity to the class centroids.

- Our SRCC assigns the mixed data with weights based on their confidence in belonging to different classes, thus mitigating the problem of noisy mixed labels.

Our work is the first attempt to exploit mixed data with noisy labels to enhance deep networks to the best of our knowledge.

- Extensive experimental results on the CIFAR10, CIFAR100, Tiny-ImageNet, and Clothing1M datasets demonstrate that our method achieves promising classification performance and a plausible network generalization ability on the test set.

4.2 Preliminaries

4.2.1 Empirical Risk Minimization

Suppose training samples, $\{(x_1, y_1), \ldots, (x_N, y_N)\} \in \mathcal{D}$ are drawn i.i.d. from an unknown training distribution P, where x_i and y_i represent the i^{th} input image and the label, respectively. N indicates the number of the training samples. Let $\mathcal{F}(\theta)$ be a prediction function with parameters θ, mapping the input $x_i \in \mathcal{R}^d$ into the output label $\mathcal{F}(x_i;\theta) \in \mathcal{R}^K$. The objective of the risk minimization (RM) is:

$$\min_{\theta} \mathop{\mathbb{E}}_{(x_i,y_i)\sim P} \ell(\mathcal{F}(x_i;\theta), y_i), \tag{4.1}$$

where $\ell(\cdot)$ is the loss function. Eq. (4.1) is empirically approximated by the training data $\mathcal{D}$:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \ell(\mathcal{F}(x_i;\theta), y_i). \tag{4.2}$$

The empirical risk minimization (ERM) assigns the same weight to all training data during optimization. However, the noise labels will handicap the model optimization severely if they are treated equally as the clean data. To alleviate the impact of corrupted data, a sample reweighting scheme is introduced to the ERM optimization,

$$\min_{\theta} \sum_{i=1}^{N} v_i \ell(\mathcal{F}(\tilde{x}_i;\theta), \tilde{y}_i), \quad s.t. \sum_{i=1}^{N} v_i = 1, \tag{4.3}$$

where v_i is a weight ranging from 0 to 1 for sample x_i. It represents the confidence that sample x_i is correctly labeled. By assigning large weights to correctly labeled samples and small weights to noise data, we can reduce the impacts of inaccurate training losses caused by corrupted labels. Note that, the sample weights in previous studies are either determined by manually defined weight functions (Kumar et al., 2010; Ma et al., 2017, 2020a) or learned from extra clean data (Ren et al., 2018; Shu et al., 2019). To be specific, the weight of each sample is first calculated based on the sample loss and then normalized to ensure the sum of all the sample weights to be 1. However, since the labels are noisy and weights are solely computed based on sample losses, the generated sample weights might be unreliable and fail to tackle noisy data.

4.2.2 MixUp

In ERM, deep models are prone to overfit training examples. When noisy labels exist in the training data, overfitting will worsen the generalization performance of deep networks. As the number of clean samples decreases, the performance and generalization ability of deep models will degrade. To improve the model discriminative capacity, MixUp (Zhang et al., 2017b) feed mixed inputs that are linearly interpolated from two random images to our model. By doing so, the model is able to regularize the decision boundaries and thus boost the model generalization and classification performance. The objective function for the mixed data input is written as,

$$\begin{aligned}\min_{\theta} \mathop{\mathbb{E}}_{(x_i,y_i)\sim P} \mathop{\mathbb{E}}_{(x_j,y_j)\sim P} \mathop{\mathbb{E}}_{\lambda\sim \text{Beta}(\alpha,\alpha)} \\ \ell(\mathcal{F}(g_{mix}(x_i,x_j,\lambda);\theta), g_{mix}(y_i,y_j,\lambda)),\end{aligned} \tag{4.4}$$

where $g_{mix}(a,b,\lambda) = \lambda \cdot a + (1-\lambda) \cdot b$ is a mix function. Similar to (Zhang et al., 2017b), the coefficient λ follows the distribution Beta(α,α). The hyper-parameter α controls the interpolation weight between an image pair. When α is 0, we have the ERM principle. The objective in Eq. (4.4) is empirically estimated by minimizing the following mixed

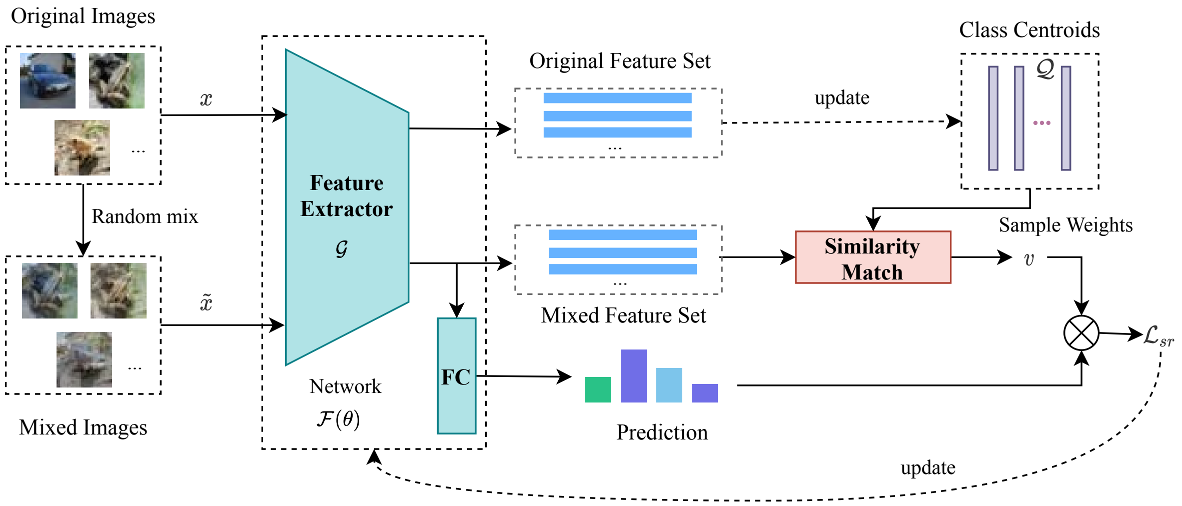


Figure 4.2: Framework of our self-reweighting from class centroids (SRCC). We use solid and dash lines to denote the forward and update operations. At each training step, we first extract features and calculate the class centroids for input images (the upper part of the figure). Then we randomly mix two images by linearly interpolating two original images. The weight of the mixed data is evaluated by the similarity between its feature and all class centroids. The reweighted losses are used to update the network. The class centroids and the network are iteratively updated to learn feature representations and classify images.

loss function:

$$\begin{aligned} &\min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \ell(\mathcal{F}(\tilde{x}_i;\theta), \tilde{y}_i), \\ \tilde{y}_i &= g_{mix}(y_p, y_q, \lambda), \\ \tilde{x}_i &= g_{mix}(x_p, x_q, \lambda), \end{aligned} \tag{4.5}$$

where (x_p, y_p) and (x_q, y_q) are vectors drawn from the N training samples randomly, and $\lambda \in [0,1]$. M indicates the number of mixed samples generated from the original samples.

4.3 Self-reweighting from Class Centroids

4.3.1 Framework

We design a self-reweighting strategy from class centroids for our training images. We intend to leverage more reliable information to generate a sample weight. The class centres are statistically more stable to noise labels than individual samples. Using the class centroids to generate the sample weight and confidence score, we can explore the relationship between the given sample and all the other training data rather than treating it as a single data point. The framework of our self-reweighting from class centroids is shown in Figure 4.2.

Although MixUp performs well on many tasks, the erroneous supervision caused by noise labels will limit its effectiveness. For example, if two samples (x_1, y_1) and (x_2, y_2) come from the same distribution containing false annotations, the ground-truth label for the mixed input $\tilde{x}_1 = g_{mix}(x_1, x_2, \lambda)$ does not correspond to $g_{mix}(y_1, y_2, \lambda)$. When the ground-truth labels of interpolated samples are inconsistent with the mixed ones, known as the manifold intrusion, a model will be trained with incorrect supervision signals. As shown in Figure 4.3, a mixed data point generated by two samples from two diagonal classes has a high probability of lying outside the original diagonal classes. These mixed inputs will degrade the model performance when a model is trained with MixUp. We also assign a sample weight to every mixed data during training to solve this issue. The self-reweighting objective is thus formulated as,

$$\begin{aligned} \min_{\theta} \quad & \mathcal{L}_{sr}(\mathcal{D};\theta) = \sum_{i=1}^{M} v(\tilde{x}_i)\ell(\mathcal{F}(\tilde{x}_i;\theta), \tilde{y}_i), \\ s.t. \quad & \sum_{i=1}^{M} v(\tilde{x}_i) = 1, \end{aligned} \tag{4.6}$$

where $\tilde{x}_i$ and $\tilde{y}_i$ indicate the mixed data and label as described in Eq. (4.5), and $v(\tilde{x}_i)$ denotes the sample weight of $\tilde{x}_i$. The higher $v(\tilde{x}_i)$ means that the mixed label is more reliable and closer to the ground-truth one. Otherwise, the mixed inputs are deemed as noise samples.

Since interpolating mixed labels are often inaccurate as seen in Figure 4.3, our SRCC measures the quality of mixed inputs to avoid assigning high weights to mixed ones with incorrect labels. Note that mixed data are used to train the network parameters and original images are exploited to produce mixed data as well as class centroids.

4.3.2 Sample Weight Generation

To obtain the sample weight for each mixed data, we first calculate the feature centroid of each class and then compute the confidence score for each mixed data. Here, we use ResNet architecture (He et al., 2016b) as the classifier network $\mathcal{F}(\theta)$. The features from the penultimate layer (*i.e.*, the last fully-connected layer before the classification layer) are used as our deep features, denoted by $\mathcal{G}(x_i)$. Therefore, the relationship between $\mathcal{F}(x_i;\theta)$ and $\mathcal{G}(x_i)$ is $\mathcal{F}(x_i;\theta) = f(\mathcal{G}(x_i))$, where f is a fully-connected layer followed by a softmax operation for classification. For each mixed example $\tilde{x}_i$, its feature is denoted as $\mathcal{G}(\tilde{x}_i)$. We use $\mathcal{Q}_c$ to represent the feature of the center of c^{th} class. The similarity between the mixed input and the c^{th} class centroid is defined as:

$$S_c(\tilde{x}_i) = \frac{e^{\mathcal{G}(\tilde{x}_i)^T \mathcal{Q}_c}}{\sum_{k=1}^{K} e^{\mathcal{G}(\tilde{x}_i)^T \mathcal{Q}_k}}, \tag{4.7}$$

where $S_c(\tilde{x}_i)$ denotes the similarity between the mixed data $\tilde{x}_i$ and the class centroid $\mathcal{Q}_c$, and K is the total class number.

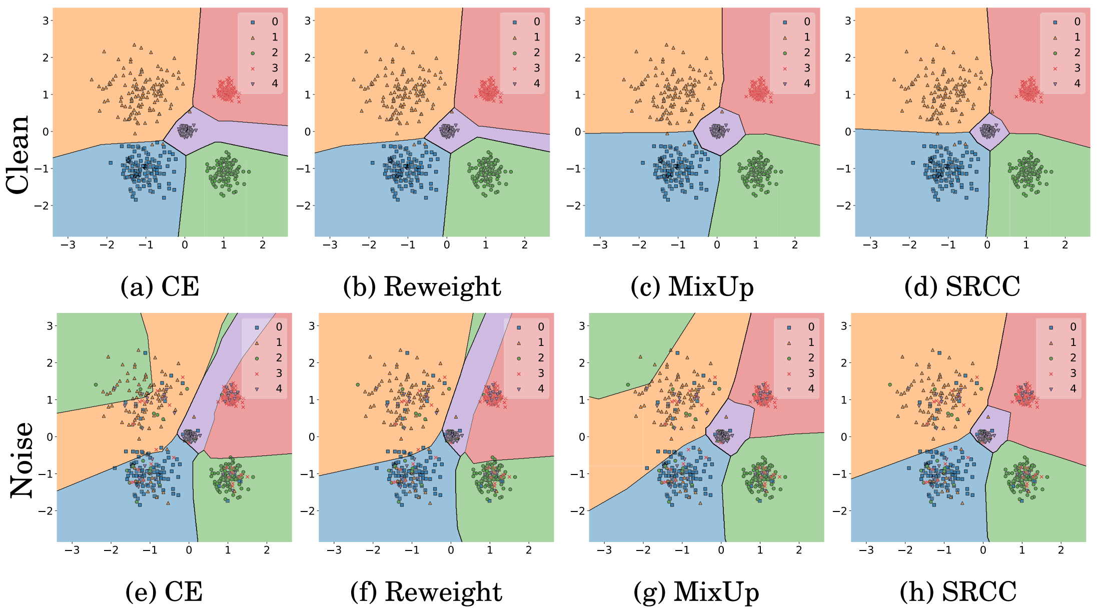
Then, we use normalized similarity scores with respect to all classes to measure mixed input confidence. As a mixed example is generated from examples from any two classes, we use $q(\tilde{x}_i) = g_{mix}(S_{y_1}(\tilde{x}_i), S_{y_2}(\tilde{x}_i), \lambda)$ to denote the confidence of a mixed input. To ensure that the sum of all the sample weights is equal to one, we normalize sample confidence as our sample weight,

$$v(\tilde{x}_i) = \frac{q(\tilde{x}_i)}{\sum_{m=1}^{M}(q(\tilde{x}_m))}. \tag{4.8}$$

To further reduce the impact of corrupted mixed inputs, we set the weight of the most unreliable sample to zero. This is achieved by using the normalization as follows:

$$v^{'}(\tilde{x}_i) = \frac{v(\tilde{x}_i) - v_{min}}{\sum_{m=1}^{M}(v(\tilde{x}_m) - v_{min})}, \tag{4.9}$$

where $v_{min} = \min_{m} \; v(\tilde{x}_m)$ is the minimum confidence score among all the training examples. In Section 4.4.4.5, we have provided detailed experiments on the scaling of sample



(a) CE (b) Reweight (c) MixUp (d) SRCC

(e) CE (f) Reweight (g) MixUp (h) SRCC

Figure 4.3: A toy experiment on synthetic data illustrates the effectiveness of our SRCC on regularizing the decision boundaries. The clean data (the first row) are generated from five Gaussian distributions with different means and standard deviations. The noisy data (the second row) are generated by randomly changing labels of examples in the clean data. The decision boundaries are displayed via Mlxtend (Raschka, 2018).

weights. By doing this, we alleviate the impact of the most unreliable mixed data and enlarge the range of sample weights. In other words, the reliable samples are assigned with higher weights. Some methods assign larger weights to the examples closer to the decision boundary instead of the class centroids. However, it would introduce more noisy samples in this way as noise annotations are more likely around the decision boundaries.

Compared to the individual sample weight generation methods (Ren et al., 2018; Shu et al., 2019), our confidence is more robust since the similarities between the sampled data and all the class centroids provide a more comprehensive manner to measure the position of the sample in the feature space. As presented in the second row of Figure 4.3, a training loss might be small for a mixed sample interpolated from two data points in the categories 1 (orange) and 2 (green) when their labels are mislabeled as category 3 (red). In this case, the network would classify this sample into the category 3 (red zone) and produces a small loss. Previous methods will assign a large sample weight for the mixed data point. Thus, the noisy samples would deviate the optimization process and degrade classification performance. In contrast, our method correlates the sample weight of one mixed example with all training samples via the class centroids. Although the loss for a single data might be small, the calculated distance between the mixed sample (interpolated from category 1 and 2) and the class centroid of category 3 will be larger than the distance between the sample and the class centroid of category 2 (green) or the category 4 (purple). Thus, the computed confidence of the mixed data will be small. We thus assign a small weight to the mixed data to avoid distraction in optimization.

4.3.3 Class Centroid Update

The weight of the mixed input is generated by similarities between the mixed feature and all class centroids. If the learned centroid of one class is close to its ground-truth feature centre, the similarity measurement is of high confidence to reflect the label correctness of the given samples. For the original sample x_i, we first extract the feature representation $\mathcal{G}(x_i)$. Suppose we have a model with parameters D and training samples N. A fully-connected network takes at least DN arithmetic operations to update the class centroids once. This requires BD (B denotes the batch size, $B << N$) steps to update parameters, consuming massive computational resources in each iteration for updating class centroids.

We propose a momentum-based scheme to update class centroids through batch samples. Specifically, at the t^{th} iteration, we first calculate the sample weight $v^t(x_i)$ for the original data x_i according to Eq. (4.7). We then update the class centroids as follows:

$$\mathcal{Q}_c^t = (1-\xi)\mathcal{Q}_c^{t-1} + \xi \sum_{i=1}^{B} v^t(x_i) \cdot \mathcal{G}^t(x_i) \cdot \mathcal{I}_c(y_i), \tag{4.10}$$

where t indicates the iteration step and ξ is set to the learning rate to control the momentum. $\mathcal{I}_c$ is an indicator function. It outputs 1 when the c^{th} position in the one-hot encoding label y_i is also 1. Otherwise, the indicator function outputs 0. It only takes

Algorithm 3 Self-reweighting from Class Centroids

```
Input: Dataset $\mathcal{D}$, Initiated parameters $\theta^0$, Initialted centers $\mathcal{Q}^0$, mixed parameter $\alpha$.
for $t = 1$ : num_iterations do
    Sample $(x_i, y_i)_{i=1}^B$ from $\mathcal{D}$.
    Extract features $\{\mathcal{G}^t(x_i)\}^B$ from original samples.
    Update class centroids using Eq. (4.10).
    Sample another B examples $(x_i, y_i)_{i=1}^B$ from $\mathcal{D}$.
    Calculate mixed samples $\{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^B$.
    Extract features $\{\mathcal{G}^t(\tilde{x}_i)\}_{i=1}^B$ for mixed inputs.
    Calculate weights $\{v^t(\tilde{x}_i)\}_{i=1}^B$ for mixed samples.
    Update model parameters using Eq. (4.6).
end for
```

BD arithmetic operations to update the class centroids in one iteration, the same as the computational cost in a model forward process.

After obtaining the updated class centroids $\mathcal{Q}^t$, we sample another batch of the original data to generate mixed images. Since the network may overly trust the mixed inputs if they are produced from the same data used for updating the centroids, we resample another batch of data to avoid this phenomenon. Then, we extract the features of the mixed samples $\mathcal{G}(\tilde{x}_i)$ as well as calculate the sample weights $v^t(\tilde{x}_i)$ to measure the training loss. Finally, the loss is backpropagated to update our model parameters.

4.3.4 Training

Algorithm 3 illustrates that our model parameters θ and class centroids $\mathcal{Q}$ are updated alternatingly in each iteration. We first sample B original examples from the dataset $\mathcal{D}$ and extract their features to update the class centroids. In the first few iterations, we assign all original samples with the same weight when updating class centroids since the initial network is not discriminative enough at first. Afterwards, we update class centroids by adopting our proposed reweighting mechanism and momentum based update strategy as indicated in Eq. (4.10). We then use the updated class centroids to obtain sample weights of mixed inputs. The mixed inputs are generated from the original images with a mixing coefficient λ which is randomly sampled from a beta distribution parameterized by α. The weights of mixed inputs are determined and normalized by Eq. (4.7) and Eq. (4.8). We update the model parameters θ by minimizing the objective in Eq. (4.6).

4.3.5 Analysis of SRCC

In this section, we analyze the effectiveness of our proposed method. For illustration, we consider a binary classification case and use the features $\{\mathcal{G}(x_i)\}_{i=1}^{N}$ from the last layer. We use $y_i = \{0,1\}$ to denote the label of x_i. Let $\mathcal{Q}_0$ and $\mathcal{Q}_1$ be the centroids for negative and positive sets, respectively. All samples are split into the positive set $\mathcal{P} = \{x_i | y_i = 1\}$ and the negative set $\mathcal{N} = \{x_i | y_i = 0\}$ based on the noisy labels. The samples in the noise set $\mathcal{S}$ are falsely annotated. Let $\mathbf{w}^t$ and $\mathbf{w}^*$ be the model parameters in the t^{th} step and the optimal parameter. For each sample in the noise set, we have

$$|y_i - \sigma({\mathbf{w}^*}^T \mathcal{G}(x_i))| \approx 1, \forall x_i \in \mathcal{S}, \tag{4.11}$$

where $\sigma(\cdot)$ is the sigmoid function. Let $\hat{y}_i^t = \sigma({\mathbf{w}^t}^T \mathcal{G}(x_i))$ be the predicted label in the t^{th} iteration. Here, the model with the optimal parameter $\mathbf{w}^*$ is able to output the clean labels for falsely annotated samples.

For simplicity, we use ℓ_i to denote the loss of sample x_i. The loss based sample reweighting methods fail to learn optimal parameter in the following theorem.

Theorem 4.1. *Suppose that $v_i = 1 - \epsilon$ holds when $\ell_i < \epsilon$ ($\epsilon > 0$ and $\epsilon^2 \approx 0$) in the loss sample reweighting algorithms. At the t^{th} iteration, if $|y_i - \hat{y}_i^t| = \epsilon_i$ and $\frac{\epsilon_i}{1-\epsilon_i} < \epsilon$ for every $\hat{y}_i^t$, and $\sum^{\mathcal{P}} \epsilon_i \mathcal{G}(x_i) - \sum^{\mathcal{N}} \epsilon_j \mathcal{G}(x_j) = \vec{0}$, the model parameter $\mathbf{w}^t$ will not converge to $\mathbf{w}^*$ after iterations.*

Proof: For the binary classification problem, the weighted loss function is then formulated as:

$$\mathcal{L}(\mathbf{w}) = -\sum_{i=1}^{N} v_i\{y_i \log \hat{y}_i + (1-y_i)log(1-\hat{y}_i)\},$$

where $\hat{y}_i = \sigma(\mathbf{w}^T \mathcal{G}(x_i))$ is the prediction by the model.

To update the model parameter, we simply use stochastic gradient descent (SGD) as follows:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \nabla_{\mathbf{w}^t} \mathcal{L}(\mathbf{w}),$$

where η denotes the learning rate. Taking the gradient of the loss function with respect to w, we obtain

$$\begin{aligned}\nabla_{\mathbf{w}^t} \mathcal{L}(\mathbf{w}) &= \sum_{i=1}^{N} \frac{\partial \mathcal{L}}{\partial \hat{y}_i^t} \frac{\partial \hat{y}_i^t}{\partial a_i^t} \nabla a_i^t(\mathbf{w}^t) \\ &= \sum_{i=1}^{N} \frac{v_i^t(\hat{y}_i^t - y_i)}{\hat{y}_i^t(1-\hat{y}_i^t)} \cdot \sigma(a_i^t)(1-\sigma(a_i^t)) \cdot \mathcal{G}(x_i) \\ &= \sum_{i=1}^{N} v_i^t(\hat{y}_i^t - y_i)\mathcal{G}(x_i),\end{aligned}$$

where we use a_i^t to denote the $\mathbf{w}^t \mathcal{G}(x_i)$ here. When $|\hat{y}_i^t - y_i| = \epsilon_i$, we have

$$\begin{aligned}\ell_i &= -y_i \log \hat{y}_i - (1-y_i)log(1-\hat{y}_i) \\ &= -log(1-\epsilon_i) \leq \frac{\epsilon_i}{1-\epsilon_i} < \epsilon.\end{aligned}$$

We then obtain the sample weight $v_i^t = 1-\epsilon$ based on the assumption in *Theorem* 4.1. The gradient with respect to $\mathbf{w}^t$ is then formed by:

$$\begin{aligned}\nabla_{\mathbf{w}^t} \mathcal{L}(\mathbf{w}) &= -\sum_{x_i \in \mathcal{P}} (1-\epsilon)\epsilon_i \mathcal{G}(x_i) + \sum_{x_j \in \mathcal{N}} (1-\epsilon)\epsilon_j \mathcal{G}(x_j) \\ &\approx -\sum_{x_i \in \mathcal{P}} \epsilon_i \mathcal{G}(x_i) + \sum_{x_j \in \mathcal{N}} \epsilon_j \mathcal{G}(x_j) = \vec{0}\end{aligned}$$

The model weight will not be updated when the model already fits training noise samples very well. It indicates that losses of noise samples will not be rectified if a model overfits the training samples. In contrast, our proposed method can still update $\mathbf{w}^t$ to approach $\mathbf{w}^*$ in the following theorem.

Theorem 4.2. *If* $\exists \mathcal{S}^{'} \subseteq \mathcal{S}$, *the condition* $\mathcal{Q}_{y_i}^T \mathcal{G}(x_i) < \mathcal{Q}_{1-y_i}^T \mathcal{G}(x_i)$ *satisfies for every* x_i *in* $\mathcal{S}^{'}$. *For the rest of* x_i, $(\mathcal{Q}_1 - \mathcal{Q}_0 - \mathbf{w}^t)^T \mathcal{G}(x_i) = 0$. *The model parameter* $\mathbf{w}^t$ *will converge to* $\mathbf{w}^*$ *after iterations.*

Proof: In our proposed method, the sample weight is evaluated by:

$$v_i^t = \frac{e^{\mathcal{Q}_{y_i}^T \mathcal{G}(x_i)}}{e^{\mathcal{Q}_0^T \mathcal{G}(x_i)} + e^{\mathcal{Q}_1^T \mathcal{G}(x_i)}}.$$

For the sample $x_i \in \mathcal{S}^{'}$, the sample weight v_i^t will less than 0.5 according to the equation. For other samples satisfying $(\mathcal{Q}_1 - \mathcal{Q}_0 - \mathbf{w}^t)^T \mathcal{G}(x_i) = 0$, the sample weight v_i^t is calculated by

$$v_i^t = \sigma((\mathcal{Q}_{y_i} - \mathcal{Q}_{1-y_i})^T \mathcal{G}(x_i)) = 1 - \epsilon_i.$$

The gradient of loss function with respect to $\mathbf{w}^t$ is now formulated as:

$$\nabla_{\mathbf{w}^t} \mathcal{L}(\mathbf{w}) \approx \sum_{\substack{x_i \in \mathcal{S}^{'} \\ y_i = 1}} (1 - v_i^t) \epsilon_i \mathcal{G}(x_i) - \sum_{\substack{x_j \in \mathcal{S}^{'} \\ y_j = 0}} (1 - v_j^t) \epsilon_j \mathcal{G}(x_j).$$

Note that we eliminate the zero items in above equations by using $\sum^{\mathcal{P}} \epsilon_i \mathcal{G}(x_i) - \sum^{\mathcal{N}} \epsilon_j \mathcal{G}(x_j) = \vec{0}$ and $\epsilon_i^2 \approx 0$. Then, we obtain

$$||\mathbf{w}^{t+1} - \mathbf{w}^*||^2 = ||\mathbf{w}^t - \eta \nabla_{\mathbf{w}^t} \mathcal{L}(\mathbf{w}) - \mathbf{w}^*||^2 =$$
$$||\mathbf{w}^t - \mathbf{w}^*||^2 + \eta^2 \nabla_{\mathbf{w}^t} \mathcal{L}(\mathbf{w})^2 - 2(\mathbf{w}^t - \mathbf{w}^*)^T \nabla_{\mathbf{w}^t} \mathcal{L}(\mathbf{w})).$$

The second term in above equation is nearly zero as $\epsilon_i^2 \approx 0$. For the samples in the set $\{x_i | x_i \in \mathcal{S}^{'}, y_i = 1\}$, we have $\mathbf{w}^{t^T} \mathcal{G}(x_i) > 0$ since $|\sigma(\mathbf{w}^{t^T} \mathcal{G}(x_i)) - y_i| = \epsilon_i < 0.5$, and $\mathbf{w}^{*^T} \mathcal{G}(x_i) < 0$ based on Eq. (4.11). For the samples in the set $\{x_j | x_j \in \mathcal{S}^{'}, y_j = 0\}$, we also have $\mathbf{w}^{t^T} \mathcal{G}(x_j) < 0$ and $\mathbf{w}^{*^T} \mathcal{G}(x_j) > 0$. The $1 - v_i^t > 0.5$ for x_i in $\mathcal{S}^{'}$. Therefore, the third term in above equation is larger than zero. We then have $||\mathbf{w}^{t+1} - \mathbf{w}^*||^2 < ||\mathbf{w}^t - \mathbf{w}^*||^2$. This shows that our proposed method is still able to update the model parameter when the loss based sample weighting methods fail.

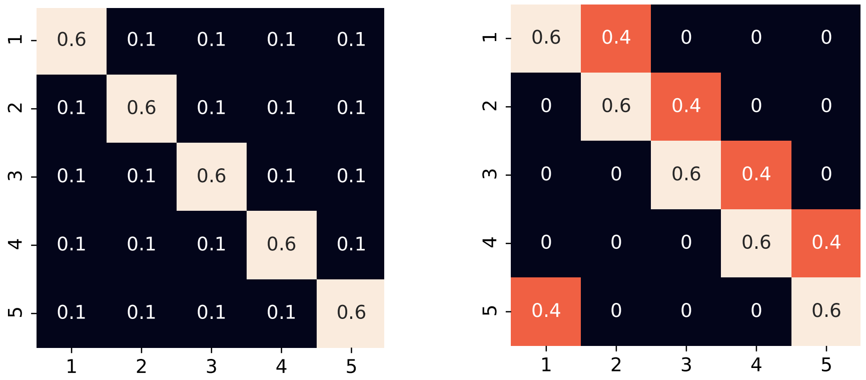
4.4 Experiments

4.4.1 Datasets

We testify the effectiveness of our proposed model on two benchmark datasets: CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), consisting of color images of 32 × 32 pixels arranged in 10 and 100 classes, respectively. There are 50,000 training and 10,000 test images in both datasets. We then evaluate our method on a larger dataset Tiny-ImagNet (Yao and Miller, 2015) which contains a training set of 100,000 images and a validation set of 10,000 images. These images are collected from 200 different classes of objects in ImageNet (Krizhevsky et al., 2012), and images are downsampled from the original resolution 256x256 pixels to 64x64 pixels. Instead of selecting a few clean examples as metadata to guide the learning process (Shu et al., 2019; Ren et al., 2018), we use all the training images without using any priors of clean data. We also conduct experiments on Clothing1M (Xiao et al., 2015), which is a large-scale dataset with real-world noisy labels and consists of 1M training images collected from online shopping websites.

4.4.2 Noise Setting

We test two types of label noise following the setting in Han et al. (2018). Symmetric noisy labels (Figure 4.4a) are produced by replacing a certain proportion of the labels of one class with other class labels uniformly (Han et al., 2018). In addition, we follow the setting in Shu et al. (2019) to generate asymmetric noisy data (Figure 4.4b), where



	1	2	3	4	5
1	0.6	0.1	0.1	0.1	0.1
2	0.1	0.6	0.1	0.1	0.1
3	0.1	0.1	0.6	0.1	0.1
4	0.1	0.1	0.1	0.6	0.1
5	0.1	0.1	0.1	0.1	0.6

(a) Symmetric

	1	2	3	4	5
1	0.6	0.4	0	0	0
2	0	0.6	0.4	0	0
3	0	0	0.6	0.4	0
4	0	0	0	0.6	0.4
5	0.4	0	0	0	0.6

(b) Asymetric

Figure 4.4: Transition matrices of different noise types at 40% noise rate (five classes contained).

labels are changed to another class in a pre-defined portion. As shown in Figure 4.4, we adopt different transitional matrices to produce noisy data of different noise types. We also set the noise rate to different levels following Wang et al. (2019) to measure the model robustness.

4.4.3 Implementation Details

We use several neural networks as the base classifiers for CIFAR-10 and CIFAR-100 datasets. ResNet32 (He et al., 2016b), Preact-ResNet18 (He et al., 2016c), MobileNet (Howard et al., 2017) and WRN28-10 (Zagoruyko and Komodakis, 2016) are selected as our backbone networks in our experiments. We train all the models by stochastic gradient descent (SGD) with the batch size of 128 and initial learning rate 0.1 following Zhang et al. (2017b). The learning rate decreases by 10 at the 50 epoch and 60 epoch, and models are trained for 70 epochs. For the Clothing1M dataset, we follow the previous work (Shu et al., 2019) and use ResNet-50 with ImageNet pre-trained weights. We fix the number of mixed training samples M in every training epoch. When M is large, it requires a longer time to train a model for one epoch. In our experiments, we set the number of mixed samples M to the number of original training samples for

Table 4.1: Classification accuracy on synthetic noisy data. Instances are sampled from the same distribution for five seeds.

Methods	seed 1	seed 2	seed 3	seed 4	seed 5
CE	89.67	91.83	89.33	89.67	91.50
Reweight	95.33	97.00	95.67	96.67	97.17
MixUp	97.33	97.33	98.33	98.00	97.33
SRCC	98.33	99.00	98.66	98.83	99.00

implementation efficiency and model performance. The default mixing parameter α is set to 1.0. We also analyze the effect of α. We run all the experiments on the Nvidia RTX-2080Ti card. We use the optimal hyperparameters reported in their original papers for all the compared methods.

4.4.4 Ablation Study

4.4.4.1 Regularization on Decision Boundaries

We conduct experiments on synthetic noise samples to demonstrate the effectiveness of our proposed SRCC on regularizing the decision boundaries. The training samples are generated from five Gaussian distributions centred at five 2D points as shown in Figure 4.3. There are in total 600 samples in the training set. We random flip labels of 40 examples to other classes to generate noisy data. Test samples are generated from the same distribution. We use a network with 2 hidden layers as the base classifier for all the methods. All the methods are trained with the batch size 64 for 200 epochs. In Figure 4.3, the decision boundaries of the compared methods are visualized by the Mlxtend tool (Raschka, 2018). As illustrated in Figure 4.3, our proposed SRCC better regularizes the decision boundaries of the network on both clean and noisy data. As a consequence, our SRCC outperforms the other methods on the classification accuracy, as indicated in Table 4.1.

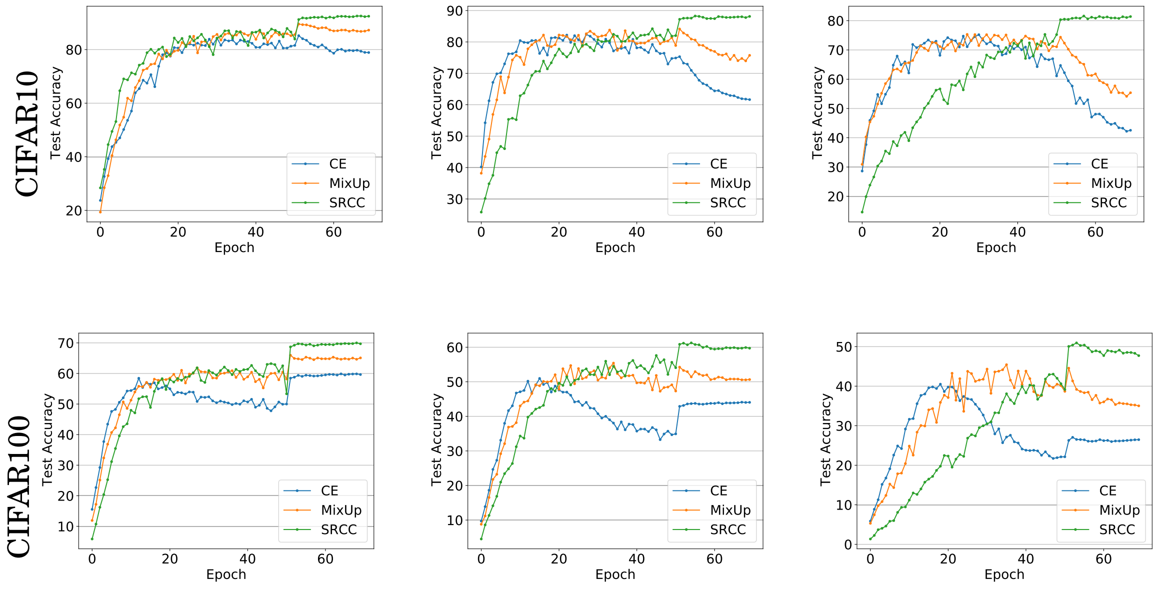
Table 4.2: Classification errors on CIFAR10 and CIFAR100 in different noise rates. Mean and standard deviation are reported.

Model	Methods	CIFAR10			CIFAR100		
		Symmetric Noise Rate			Symmetric Noise Rate		
		0.2	0.4	0.6	0.2	0.4	0.6
MobileNet	CE	13.53±0.41	18.03±0.28	25.69±0.67	47.95±0.90	55.10±0.22	66.66±0.93
	MixUp	12.90±0.16	16.40±0.57	23.24±0.41	38.30±0.69	47.14±0.77	60.84±0.80
	SRCC	**12.02**±0.12	**16.11**±0.54	**22.44**±0.69	**35.14**±0.39	**40.76**±0.29	**54.35**±0.49
Preact-ResNet18	CE	16.60±0.40	18.56±2.05	25.16±0.55	40.28±0.53	48.85±0.96	58.97±1.14
	MixUp	9.86±0.36	15.54±0.40	22.81±0.71	35.15±0.80	44.86±0.78	54.50±0.68
	SRCC	**8.50**±0.92	**12.02**±0.45	**19.26**±1.20	**30.48**±0.72	**38.29**±0.45	**49.02**±1.60
Wide-ResNet28	CE	14.53±0.42	17.40±0.37	25.38±0.35	35.00±0.58	45.80±0.93	56.73±0.97
	MixUp	8.83±0.46	13.69±0.48	18.98±1.61	28.80±0.55	37.45±0.81	45.65±0.17
	SRCC	**7.37**±0.36	**11.55**±0.97	**17.81**±0.79	**27.35**±0.21	**33.64**±0.39	**41.77**±0.81

4.4.4.2 Noise Rates and Architectures

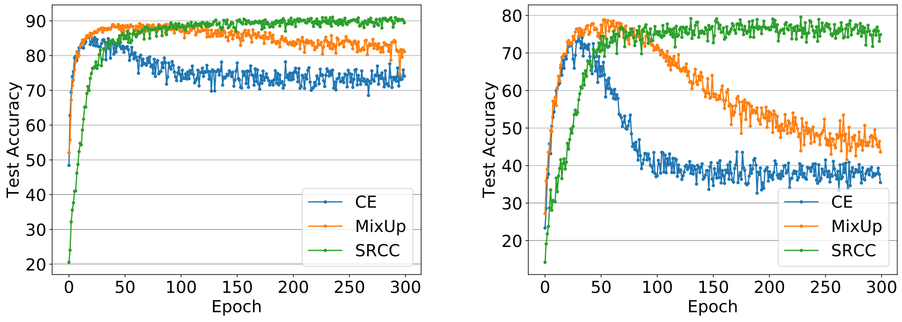
To demonstrate the robustness of our proposed method, we investigate the performance of our method for different backbones and noise rates. We conduct experiments on the CIFAR10, and each experiment is repeated five times using different random seeds. The mean and standard deviation of the top-1 error rate is reported. In particular, we compare our SRCC with the MixUp method (Zhang et al., 2017b) in the following settings: (1) The noisy data are generated with symmetric noise, where the noise rates are set to 0.2, 0.4 and 0.6, respectively. (2) Different classifier architectures, *i.e.*, MobileNet (Howard et al., 2017), Preact-ResNet18 (He et al., 2016c) and Wide ResNet (Zagoruyko and Komodakis, 2016), are adopted.

Table 4.2 indicates that different base classifiers armed with our SRCC algorithm achieve the lowest classification errors. Additionally, the error rates of Wide-ResNet28 are lower than those of Preact-ResNet18 and MobileNet. This indicates that a more robust base classifier also improves the model performance. All the classifiers trained with our SRCC achieve lower error rates than those trained with MixUp on both datasets in different noise rates. This manifests that the models trained with SRCC obtain better robustness against noisy data, demonstrating the superiority of our SRCC.



(a) Symmetric 0.2 (b) Symmetric 0.4 (c) Symmetric 0.6

Figure 4.5: Test accuracy vs. the number of epochs for SRCC and the compared methods.



(a) Symmetric 0.2 (b) Symmetric 0.6

Figure 4.6: Test accuracy vs. the number of epochs for SRCC and the compared methods trained with the fixed learning rate.

4.4.4.3 Convergence

To analyze the test accuracy behaviours of different losses during training, we plot the test accuracy in every iteration in Figure 4.5. Preact-ResNet18 is adopted as the base classifier for different methods. For the models trained with the CE loss, the performance decreases dramatically after reaching the highest test accuracy. After

Table 4.3: Classification errors on CIFAR10 for different hyper-parameter values α.

Methods	Noise Rate	
	0.2	0.4
MixUp ($\alpha = 0.2$)	11.13±0.38	16.43±0.72
MixUp ($\alpha = 0.5$)	11.82±4.23	15.85±0.82
MixUp ($\alpha = 1.0$)	10.22±0.62	15.25±0.41
SRCC ($\alpha = 0.2$)	9.35±0.32	13.38±0.34
SRCC ($\alpha = 0.5$)	**8.64**±0.49	11.81±0.23
SRCC ($\alpha = 1.0$)	8.70±0.78	**11.52**±0.27

50 training iterations, the learning rate is decayed by 10 times. There are noticeable performance gains in our method since most clean samples are used in our method which leads to superior convergence. This phenomenon indicates that the noise samples provide erroneous supervision in training, leading to inferior predictions for the test samples. Compared with the MixUp, our model achieves better performance on both CIFAR10 and CIFAR100.

To further investigate the causes of performance degradation, we conduct extra experiments on CIFAR10 with different symmetric noise rates. Instead of decaying the learning rate, we train all models for 300 epochs with the same learning rate 0.1. As shown in Figure 4.6, the performance of compared methods (*e.g.*, CE and MixUP) still decreases even if the learning rate is not decayed. This indicates that the performance degradation is caused by the fact that the noise examples provide false supervision and the compared methods cannot suppress these erroneous signals during training. The performance degradation becomes more severe when compared models are trained on data containing more noise samples. This shows that the performance of CE and MixUp is easily affected by noise samples, especially when there are many noise samples in the training data. Compared to these methods, the performance of our SRCC is stable, demonstrating the robustness and superiority of our proposed algorithm.

Table 4.4: Effects of reweighting strategies on CIFAR10 and CIFAR100.

Dataset	Methods	Noise Rate		
		0.2	0.4	0.6
CIFAR10	CE	83.40±0.40	81.44±2.05	74.84±0.55
	Reweight	88.41±0.56	85.79±1.19	75.41±2.58
	MixUp	90.14±0.36	84.46±0.40	77.19±0.71
	SRCC ($\mathcal{L}_c$)	91.22±1.09	87.71±0.62	80.55±1.15
	SRCC ($\mathcal{L}_{rc}+v_1$)	91.28±1.28	87.59±0.61	**80.81**±0.67
	SRCC ($\mathcal{L}_{rc}+v_2$)	**91.50**±0.92	**87.98**±0.45	80.74±1.20
CIFAR100	CE	59.72±0.53	51.15±0.96	41.03±1.14
	Reweight	59.55±0.55	53.11±0.21	42.51±0.58
	MixUp	64.85±0.80	55.14±0.78	45.50±0.68
	SRCC ($\mathcal{L}_c$)	67.80±0.14	58.72±0.86	48.65±0.61
	SRCC ($\mathcal{L}_{rc}+v_1$)	67.88±0.33	59.12±0.84	50.38±0.84
	SRCC ($\mathcal{L}_{rc}+v_2$)	**69.52**±0.72	**61.71**±0.45	**50.98**±1.60

4.4.4.4 Sensitivity of Hyper-parameter α

We evaluate the sensitivity of MixUp and SRCC with respect to different mixing coefficients controlled by α. The test error rates on CIFAR10 are reported in Table 4.3. Each model is run three times in this experiment. It can be seen that our SRCC is less sensitive to the hyper-parameter than MixUp. When α is set to 0.5, the performance variance of MixUp is much higher than ours. It implies that our proposed method is more robust to different parameters than MixUP.

4.4.4.5 Class Centroids

To study the effects of reweighting strategies for updating class centroids, we conduct experiments on CIFAR10 and CIFAR100 in different noise rates. We report test accuracy in Table 4.4. $\mathcal{L}_{rc}$ and $\mathcal{L}_c$ denote the class centroids with or without reweigthing original images. v_1 and v_2 denote that the weights of mixed inputs are normalized by Eq. (4.8) and Eq. (4.9) respectively. It shows that the models with reweighting outperform those with MixUp on both CIFAR10 and CIFAR100. Furthermore, we observe that a model updating class centroids with weighted features outperforms the one updating class

Table 4.5: Overall test accuracy of models with different centroids update strategies on CIFAR10.

Methods	Noise Rate		
	0.2	0.4	0.6
Identical sampling	91.02±1.71	87.35±0.67	79.04±2.52
Offline	**92.75**±0.46	**88.70**±0.89	79.62±2.64
SRCC	91.50±0.92	87.98±0.45	**80.74**±1.20

centroids with the mean of sample features. This also demonstrates that our SRCC can recognize reliable data when updating the class centroids.

We further investigate the effects of different centroids update strategies on the model performance. As shown in Table 4.5, we adopt three ways to update class centroids. The "Identical sampling" denotes the same batch samples model to produce mixed inputs and update class centroids. The "Offline" represents the model updating class centroids using all training samples after a training epoch. It is observed that the model that samples another batch of images for generating mixed inputs (*i.e.*, our SRCC) indeed outperforms the "Identical Sampling" model.

4.4.5 Comparisons with State-of-the-art

We compare our methods with different state-of-art methods. For fair comparisons on CIFAR10 and CIFAR100, ResNet32 is adopted as the base classifier by all the methods. For the Tiny-ImagnetNet dataset, we use Preact-ResNet18 as the base classifier for all the methods. The CE denotes that the model utilizes the cross-entropy loss to train the networks on noisy data. Forward (Patrini et al., 2017) corrects the prediction by a label transition matrix. Coteach (Han et al., 2018) adopts two models and an exchange loss for robust training. Meta-WeightNet (Shu et al., 2019) uses a simple network to learn a weighting function in a data-driven fashion, representing the state-of-the-art sample weighting methods. GCE (Zhang and Sabuncu, 2018) introduces a generalized cross-entropy loss for training deep neural networks with noisy labels. SL (Wang et al.,

Table 4.6: Comparisons with different state-of-the-art methods on CIFAR10 and CIFAR100. Mean and standard deviation of Top-1 Accuracy are reported. The relative degradation between the noise and clean cases is also reported in the parentheses. The best and second best results are marked in red and blue respectively.

Dataset	Methods	Clean	Symmetric Noise			Asymmetric Noise	
			Noise Rate			Noise Rate	
			0.2	0.4	0.6	0.2	0.4
CIFAR10	CE	92.89±0.32	76.83(16.06)±2.30	70.77(22.12)±2.31	63.21(29.68)±4.22	79.24(13.65)±1.33	69.92(22.97)±1.97
	Forward	91.85±0.15	87.83(4.12)±0.32	84.19(7.66)±0.21	78.92(12.93)±0.29	89.29(2.56)±0.50	82.32(9.53)±0.70
	Coteach	92.19±0.12	90.69(1.50)±0.12	85.30(6.89)±0.29	78.21(13.98)±2.58	82.22(9.97)±0.93	79.00(13.19)±1.27
	Meta-WeightNet	92.04±0.15	89.19(2.85)±0.57	86.10(5.94)±0.18	81.31(10.73)±0.37	90.33(1.71)±0.61	87.57(4.47)±0.23
	GCE	90.03±0.30	88.51(1.52)±0.37	85.48(4.55)±0.16	81.29(8.74)±0.23	88.55(1.48)±0.22	83.31(6.72)±0.14
	SL	89.31±0.29	88.38(0.93)±0.29	86.00(3.31)±0.23	81.19(8.12)±0.40	88.41(0.90)±0.28	82.87(6.44)±0.65
	Bi-Tempered	90.11±0.23	88.51(1.60)±0.31	84.93(5.18)±0.67	77.82(12.29)±0.79	88.23(1.88)±0.23	82.43(7.68)±0.23
	SRCC	92.41±0.17	90.52(1.89)±0.13	87.43(4.98)±0.42	81.59(10.82)±0.41	91.09(1.32)±0.25	87.89(4.52)±0.52
CIFAR100	CE	70.50±0.12	50.8(19.64)6±0.27	43.01(27.49)±1.16	34.43(36.07)±0.94	52.36(18.14)±0.17	41.23(29.27)±1.26
	Forward	68.52±0.36	61.27(7.24)±0.40	55.69(12.83)±0.32	45.15(23.37)±1.88	65.04(3.48)±7.71	46.77(21.75)±0.51
	Coteach	65.09±0.19	62.48(2.61)±0.28	53.88(11.21)±0.29	40.94(24.15)±0.87	57.55(7.54)±0.25	49.23(15.86)±0.78
	Meta-WeightNet	69.13±0.33	64.22(5.09)±0.28	58.64(10.67)±0.47	47.43(21.58)±0.76	64.22(5.09)±0.28	55.25(14.06)±0.47
	GCE	67.39±0.12	63.97(3.42)±0.43	58.33(9.06)±0.35	41.73(25.66)±0.36	62.07(5.32)±0.41	53.29(14.10)±0.09
	SL	62.82±1.44	60.74(2.08)±1.27	58.04(4.78)±1.79	46.77(16.05)±2.43	61.49(1.33)±0.85	51.21(11.61)±0.49
	Bi-Tempered	67.90±0.27	64.95(2.95)±0.22	59.83(8.07)±0.46	50.73(17.17)±0.50	61.25(6.65)±0.33	46.26(21.64)±0.36
	SRCC	69.31±1.16	65.76(3.55)±0.36	60.62(8.69)±0.68	49.23(20.08)±1.49	65.98(3.33)±0.50	54.86(14.45)±0.64

Table 4.7: Comparisons with different state-of-the-art methods in terms of test accuracy on Tiny-ImageNet.

Methods	Clean	Symmetric			Asymmetric	
		Noise Rate			Noise Rate	
		0.2	0.4	0.6	0.2	0.4
CE	58.59	44.39	37.14	31.19	47.01	34.06
Forward (Patrini et al., 2017)	58.52	46.51	37.16	29.72	48.16	34.51
Coteach (Han et al., 2018)	55.23	46.93	42.17	21.53	50.71	39.06
Meta-WeightNet (Shu et al., 2019)	57.55	51.33	46.68	39.91	51.23	37.72
GCE (Zhang and Sabuncu, 2018)	57.13	49.16	46.02	40.73	47.93	39.43
SL (Wang et al., 2019)	56.45	53.09	49.68	41.24	53.16	36.67
Bi-Tempered (Amid et al., 2019)	58.04	53.49	46.44	35.11	49.36	39.25
SRCC	59.77	54.24	50.64	41.56	54.45	40.91

2019) employs an asymmetric cross-entropy loss for robust learning with noisy labels. Bi-Tempered (Amid et al., 2019) introduces a robust bi-tempered logistic loss for training models with noisy labels.

As shown in Table 4.6, our proposed SRCC, in general, achieves the highest test accuracy. Bi-Tempered performs well for symmetric noisy datasets but fails to handle asymmetric noise. Although the results of Meta-WeigthNet for different noisy types are stable, it requires extra clean data to calibrate the sample weights during training. As

Table 4.8: Comparisons with state-of-the-art methods in terms of test accuracy on Clothing1M.

Method	CE	Forward (Patrini et al., 2017)	SL (Wang et al., 2019)	Meta-WeightNet (Shu et al., 2019)	SRCC
Accuracy	69.21	69.84	71.02	73.72	**73.99**

Meta-WeigthNet requires to feed-forward both training and validation data, and propagate gradients backwards three times, it takes at least several times the computational resources of baseline to update the network. In our SRCC, we only need extra forward operations to update the class centroids. We also summarize the average running time of a training epoch on CIFAR10. For one epoch, training the baseline, our SRCC and Meta-WeightNet, on average, cost 10.72s, 17.73s, and 81.27s, respectively. Therefore, Meta-WeightNet performs much slower than our approach during training.

For all compared methods, they either adopt different loss functions (such as Bi-temp) or introduce sample weights (such as Meta-WeightNet). The supervision signals of these methods are thus different from the standard CE loss. In practice, noise levels of training data are unknown to the compared algorithms. Therefore, we do not assume that the clean data are known in advance. Even though all samples are clean, the compared algorithms may also recognize them as noise ones, thus leading to performance degeneration. We also report the top-1 accuracy of different methods in the noise-free case. As suggested, we report the relative degradation in Table 4.6. Our method is comparable to the baseline in the noise-free case but our model is agnostic against the noise degrees of data. This implies that our proposed model identifies the clean samples accurately.

Furthermore, we also report the classification performance of different methods on Tiny-ImageNet in Table 4.7. Our proposed SRCC achieves the best classification accuracy when there is no noise in the training data and outperforms all the other methods when the training data contain different types of noisy examples.

To verify the effectiveness of the proposed method on real-world data, we further conduct experiments on Clothing1M. The results of other methods are reported from original papers. Table 4.8 shows the test accuracy of different methods. Our proposed method achieves the highest test accuracy on this dataset.

4.5 Summary

In this chapter, we introduced a novel reweighting method, dubbed self-reweighting from class centroids (SRCC), for learning with noisy labels. Our method exploits the class centres to measure the reliability of labels in computing the objective function, thus being more robust to corrupted labels. Furthermore, we also reweight class centroids to remove the noisy data in an online fashion. By doing so, we significantly reduce the computational cost while maintaining the effectiveness of the training process. The effectiveness of our proposed method is analyzed to show the advantage of our proposed method. Extensive experiments demonstrate that our SRCC achieves superior performance compared to the state-of-the-art on noisy data.

CHAPTER 5

Single Frame Supervision for Temporal Action Localization

5.1 Introduction

Recently, weakly-supervised temporal action localization (TAL) has attracted substantial interest. Given a training set containing only video-level labels, we aim to detect and classify each action instance in long, untrimmed testing videos. In the fully-supervised annotation, the annotators usually need to rollback the video for repeated watching to give the precise temporal boundary of an action instance when they notice an action while watching the video (Zhao et al., 2019). For the weakly-supervised annotation, annotators just need to watch the video once to give labels. They can record the action class once they notice an unseen action. This significantly reduces annotation resources: video-level labels use fewer resources than annotating the start and end times in the fully-supervised setting.

Despite the promising results achieved by state-of-the-art weakly supervised TAL work (Nguyen et al., 2018; Paul et al., 2018; Shou et al., 2018), their localization per-

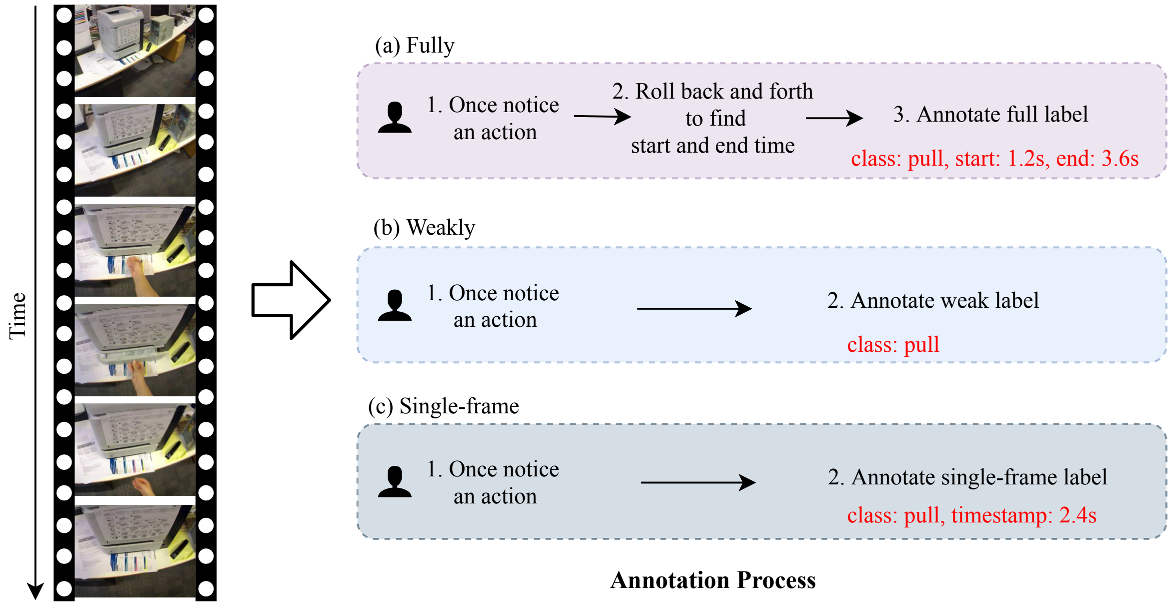


Figure 5.1: Different ways of annotating actions while watching a video. (a) Annotating actions in the fully-supervised way. The start and end time of each action instance are required to be annotated. (b) Annotating actions in the weakly-supervised setting. Only action classes are required to be given. (c) Annotating actions in our single frame supervision. Each action instance should have one timestamp. Note that the time is automatically generated by the annotation tool. Compared to the weakly-supervised annotation, the single frame annotation requires only a few extra pauses to annotate repeated seen actions in one video.

formance is still inferior to fully-supervised TAL work (Chao et al., 2018; Lin et al., 2018; Shou et al., 2017). In order to bridge this gap, we are motivated to utilize single frame supervision (Moltisanti et al., 2019): for each action instance, only one single positive frame is pointed out. The annotation process for single frame supervision is almost the same as it in the weakly-supervised annotation. The annotators only watch the video once to record the action class and timestamp when they notice each action. This significantly reduces annotation resources compared to full supervision.

In order to make full use of single frame supervision for our TAL task, we make three innovations to improve the model's capability in distinguishing background frames and action frames. First, we predict "actionness" at each frame which indicates the probability of being any actions. Second, based on the actionness, we investigate a novel background mining algorithm to determine frames that are likely to be the background and leverage these pseudo background frames as additional supervision. Third, when labeling pseudo action frames, besides the frames with high confidence scores, we aim to determine more pseudo action frames and thus propose an action frame expansion algorithm.

In addition, detecting precise start time and end time for many real-world applications is overkill. Consider a reporter who wants to find some car accident shots in an archive of street camera videos: it is sufficient to retrieve a single frame for each accident. The reporter can quickly truncate clips of desired lengths. Thus, in addition to evaluating traditional segment localization in TAL, we propose a new task called single frame localization, which requires only localizing one frame per action instance.

- To our best knowledge, this is the first work to use single frame supervision for the challenging problem of localizing temporal boundaries of actions. We show that the single frame annotation significantly saves annotation time.

- We find that single frame supervision can provide a vital cue about the background. Thus, from frames that are not annotated, we propose two novel methods to mine likely background frames and action frames, respectively. These likely background and action timestamps are further used as pseudo ground-truth for training.

- We conduct extensive experiments on three benchmarks, and the performances on both segment localization and single frame localization tasks are primarily boosted.

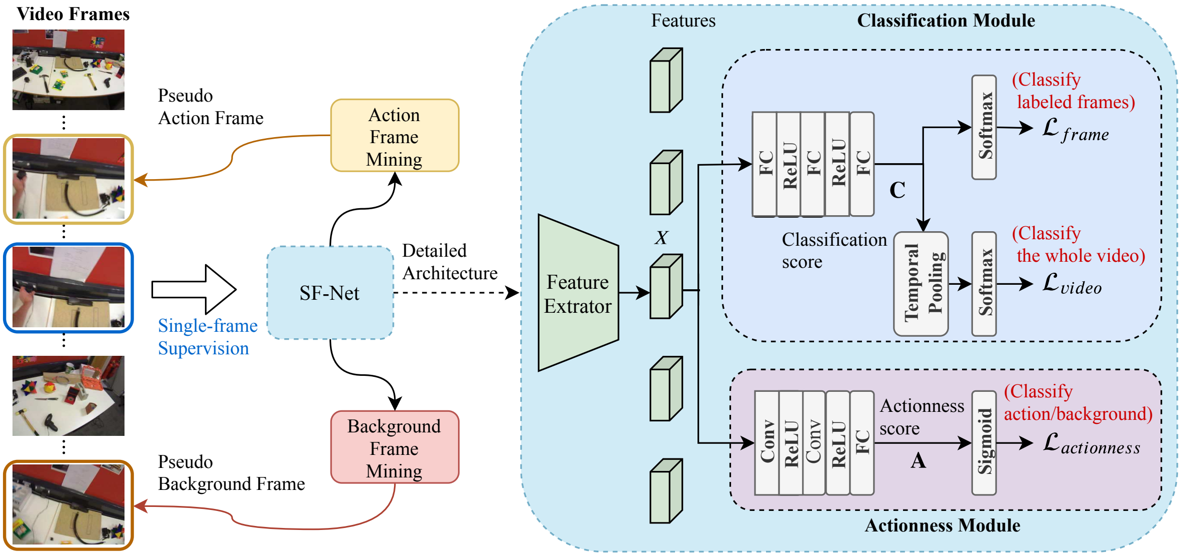


Figure 5.2: Overall training framework of our proposed SF-Net. Given single frame supervision, we employ two novel frame mining strategies to label pseudo action frames and background frames. The detailed architecture of SF-Net is shown on the right. SF-Net consists of a classification module to classify each labeled frame and the whole video, and an actionness module to predict the probability of each frame being action.

5.2 Single Frame Network

5.2.1 Problem Setup

A training video can contain multiple action classes and multiple action instances. Unlike the fully-supervised setting, which provides temporal boundary annotation of each action instance, in our single frame supervision setting, each instance only has one frame pointed out by the annotator with timestamp t and action class y. Note that $y \in \{1,\ldots,N_c\}$ where N_c is the total number of classes and we use index 0 to represent the background class.

Given a testing video, we perform two temporal localization tasks: (1) **Segment localization**. We detect each action instance's start time and end time with its action class prediction. (2) **Single frame localization**. We output the timestamp of each detected action instance with its action class prediction.

5.2.2 Framework

Our overall framework is presented in Figure 5.2. During training, learning from single frame supervision, SF-Net mines pseudo action and background frames. Based on the labeled frames, we employ three losses to jointly train a classification module to classify each labeled frame and the whole video, and an actionness module to predict the probability of each frame being action.

For a training batch of N videos, the features of all frames are extracted and stored in a feature tensor $X \in \mathcal{R}^{N \times T \times D}$, where D is the feature dimension, and T is the number of frames. As different videos vary in the temporal length, we simply pad zeros when the number of frames in a video is less than T.

The classification module outputs the score of being each action class for all frames in the input video. To classify each labeled frame, we feed X into three Fully-Connected (FC) layers to get the classification score $C \in \mathcal{R}^{N \times T \times N_c+1}$. The classification score C is then used to compute frame classification loss $\mathcal{L}_{frame}$. We also pool C temporally as described by Narayan et al. (2019) to compute video-level classification loss $\mathcal{L}_{video}$.

As shown in Figure 5.2, our model has an actionness branch of identifying positive action frames. Different from the classification module, the actionness module only produces a scalar for each frame to denote the probability of being contained in an action segment. To predict an actionness score, we feed X into two temporal convolutional layers followed by one FC layer, resulting in an actionness score matrix $A \in \mathcal{R}^{N \times T}$. We apply sigmoid on A and then compute a binary classification loss $\mathcal{L}_{actionness}$.

5.2.3 Pseudo Label Mining

5.2.3.1 Action Classification

We use cross entropy loss for the action frame classification. As there are NT frames in the input batch of videos and most of the frames are unlabeled, we first filter the

Algorithm 4 Action Frame Mining

```
Input: video classification activation $C \in \mathcal{R}^{T\times N_c+1}$, labeled action frame at time t
  belonging to action class $y$, expand radius $r = 5$, threshold $\xi = 0.9$.
Output: expanded frames set $\mathcal{S}$
gather classification score $C(t)$ for the anchor frame
$\mathcal{S} \leftarrow \{(t, y)\}$
function EXPAND(s):
  for $j \leftarrow 1; j \le r$ do
    $\hat{y}_{past} \leftarrow$ argmin $C(t + (j-1)s)$
    $\hat{y}_{current} \leftarrow$ argmin $C(t + js)$
    $\hat{y}_{future} \leftarrow$ argmin $C(t + (j+1)s)$
    if $\hat{y}_{past} == \hat{y}_{current} == \hat{y}_{future}$ and $C(t+js)_y \ge \xi C(t)_y$ then
      $\mathcal{S} \leftarrow (t + js, y)$
    end if
    $j \leftarrow j + s$
  end for
end function
EXPAND(-1)
EXPAND(1)
Return $\mathcal{S}$
```

labeled frames for classification. Suppose we have N_l labeled frames where $N_l \ll NT$. We can get classification activations of N_l labeled frames from C. These scores are fed to a Softmax layer to get classification probability $\mathbf{p}^l \in \mathcal{R}^{N_l\times N_c+1}$. The classification loss of annotated frames is formulated as:

$$\mathcal{L}^l_{frame} = -\frac{1}{N_l}\sum_i^{N_l} \mathbf{y}_i \log \mathbf{p}^l_i, \tag{5.1}$$

where the $\mathbf{p}^l_i$ denote the prediction for the i^{th} labeled action frame.

5.2.3.2 Mining Action Frames

With only a single label per action instance, the number of positive examples is relatively small. The model may be challenging to learn from these limited frames. To increase the temporal information available to the model, we design an action frame mining to introduce more frames into the training process. We treat each labeled action frame as an anchor for each action instance. We first set the expand radius r to limit the maximum expansion distance to the anchor frame at t. Then we expand the past from $t-1$ frame and the future from $t+1$ frame, separately. Suppose the action class of the anchor frame is represented by y_i. Suppose the current expanding frame has the same predicted label as the anchor frame, and the classification score at y_i class is higher than that score of the anchor frame multiplying a predefined value ξ. In that case, we then annotate this frame with label y_i and put it into the training pool. Otherwise, we stop the expansion process for the current anchor frame.

The action frame mining strategy is described in Algorithm 4. We treat the labeled action frame as the anchor frame and expand frames around it. We use a threshold ξ and the label consistency with neighbors to decide whether to add the unlabeled frame or not. The expanded frames are annotated with the same label as the anchor frame. As shown in Algorithm 4, we expand the frames at $t-1$ to the anchor frame. We first gather the classification score of three frames around $t-1$ frame. We then calculate the prediction classes for these three frames. If they all have the same predicted class and the classification score for the current frame at class y is above a threshold, we choose to put the current frame into the training frame set S. For all experiments in the chapter, we set $\xi = 0.9$ for fair comparisons.

5.2.3.3 Mining Background Frames

The background frames are also important and widely used to boost the model performance (Liu et al., 2019; Nguyen et al., 2019). Since there is no background label under the single frame supervision, our proposed model manages to localize background frames from all the unlabeled frames in the N videos. In the beginning, we do not have supervision about where the background frames are. Introducing the background class can avoid classifying a frame into one of the action classes. Our proposed background frame mining algorithm can offer us the supervision needed for training such a background class to improve the classifier's discriminability. Suppose we try to mine ηN_l background frames.

We first gather the classification scores of all unlabeled frames from C. The η is the ratio of background frames to labeled frames. These scores are then sorted along background class to select the top ηN_l scores $\mathbf{p}^b \in \mathcal{R}^{\eta N_l}$ as the score vector of the background frames. The pseudo background classification loss is calculated on the top ηN_l frames by

$$\mathcal{L}^b_{frame} = -\frac{1}{\eta N_l} \sum \log \mathbf{p}^b. \tag{5.2}$$

The background frame classification loss assists the model with identifying irrelevant frames. Different from background mining which either require extra computation source to generate background frames or adopt a complicated loss for optimization, we mining background frames across multiple videos and use the classification loss for optimization. The selected pseudo background frames may have some noises in the initial training rounds. As the training evolves and the classifier discriminability improves, we are able to reduce the noises and detect background frames more correctly. With the more correct background frames as supervision signals, the discriminability can be further boosted. In our experiments, we observed that this simple background mining strategy allows for better action localization results. We incorporate the background classification loss with the labeled frame classification loss to formulate the single frame classification loss

$$\mathcal{L}_{frame} = \mathcal{L}^l_{frame} + \frac{1}{N_c} \mathcal{L}^b_{frame} \tag{5.3}$$

where N_c is the number of action classes to leverage the influence from background class.

5.2.3.4 Actionness Prediction

In fully-supervised TAL, various methods learn to generate action proposals that may contain the potential activities (Xu et al., 2017; Lin et al., 2018; Chao et al., 2018). Motivated by this, we have designed the actionness module to make the model focus on frames relevant to target actions. Instead of producing the temporal segment in proposal methods, our actionness module produces the actionness score for each frame. The actionness module is in parallel with the classification module in our SF-Net. It offers extra information for temporal action localization. We first gather the actionness score $A^l \in \mathcal{R}^K$ of labeled frames in the training videos. The higher the value for a frame, the higher probability of that frame belongs to a target action. We also use the background frame mining strategy to get the actionness score $A^b \in \mathcal{R}^{\eta N_l}$. The actionness loss is calculated by,

$$\mathcal{L}_{actionness} = -\frac{1}{N_l}\sum \log \sigma(A^l) - \frac{1}{\eta N_l}\sum \log(1-\sigma(A^b)), \tag{5.4}$$

where σ is the sigmoid function to scale the actionness score to $[0,1]$.

5.2.4 Training

We employ video-level loss as described in Narayan et al. (2019) to tackle the problem of multi-label action classification at video-level. For the i^{th} video, the top-k activations per category (where $k = T_i/8$) of the classification activation $C(i)$ are selected and then are averaged to obtain a class-specific encoding $r_i \in \mathcal{R}^{C+1}$. We average all the frame label predictions in the video v_i to get the video-level ground-truth $q_i \in \mathcal{R}^{N_c+1}$. The video-level loss is calculated by

$$\mathcal{L}_{video} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N_c} q_i(j) \log \frac{\exp(r_i(j))}{\sum_{N_c+1}\exp(r_i(k))}, \tag{5.5}$$

where $q_i(j)$ is the j^{th} value of q_i representing the probability mass of video v_i belong to j^{th} class.

Consequently, the total training objective for our proposed method is

$$\mathcal{L} = \mathcal{L}_{frame} + \alpha\mathcal{L}_{video} + \beta\mathcal{L}_{actionness}, \tag{5.6}$$

where $\mathcal{L}_{frame}$, $\mathcal{L}_{video}$, and $\mathcal{L}_{actionness}$, denote the frame classification loss, video classification loss, and actionness loss, respectively. α and β are the hyper-parameters leveraging different losses.

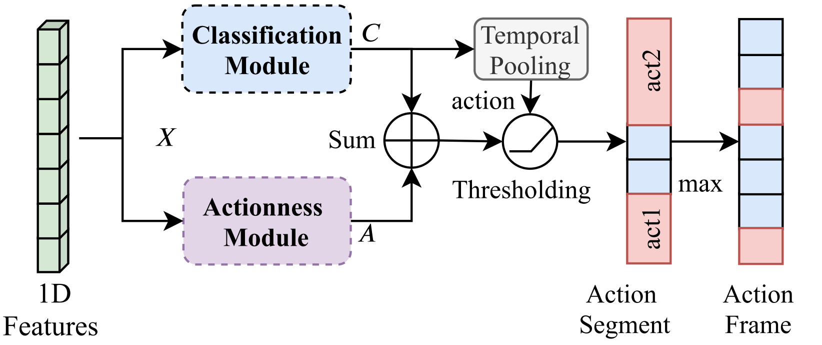


Figure 5.3: The inference framework of SF-Net. The classification module outputs the classification score C of each frame for identifying possible target actions in the given video. The action module produces the actionness score determining the possibility of a frame containing target actions. The actionness score together with the classification score are used to generate action segment based on the threshold.

5.2.5 Inference

During the test stage, we need to give the temporal boundary for each detected action. We follow previous weakly-supervised work (Nguyen et al., 2018) to predict video-level labels by temporally pooling and thresholding on the classification score. As shown in Figure 5.3, we first obtain the classification score C and actionness score A by feeding input features of a video to the classification module and actionness module. Towards segment localization, we follow the thresholding strategy in Nguyen et al. (2019) to keep the action frames above the threshold and consecutive action frames constitute an action segment. For each predicted video-level action class, we localize each action segment by detecting an interval that the sum of classification score and actionness score exceeds the preset threshold at every frame inside the interval. We simply set the confidence score of the detected segment to the sum of its highest frame classification score and the actionness score. Towards single frame localization, for the action instance, we choose the frame with the maximum activation score in the detected segment as the localized action frame.

5.3 Experiment

5.3.1 Datasets

THUMOS14 (Idrees et al., 2017) contain 1010 validation and 1574 test videos from 101 action categories. Out of these, 20 categories have temporal annotations in 200 validation and 213 test videos. The dataset is challenging, as it contains an average of 15 activity instances per video.

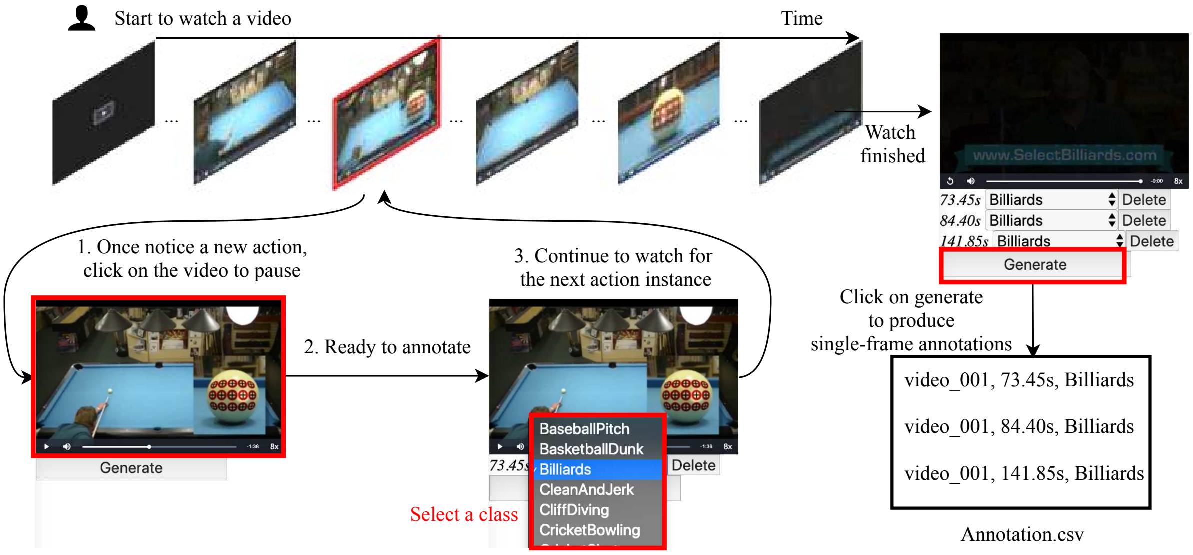
GTEA (Lei and Todorovic, 2018) include 28 videos of 7 fine-grained types of daily activities in a kitchen contained. An activity is performed by four different subjects, and each video contains about 1800 RGB frames, showing a sequence of 7 actions, including the background action.

BEOID (Damen et al., 2014) contain 58 videos and an average of 12.5 action instances is included in each video. The average length is about 60s, and there are 30 action classes in total. We randomly split the untrimmed videos in an 80-20% proportion for training and testing, as described in (Moltisanti et al., 2019).

5.3.2 Implementation Details

We use I3D network (Carreira and Zisserman, 2017a) trained on the Kinetics (Carreira and Zisserman, 2017b) to extract video features. For the RGB stream, we rescale the smallest dimension of a frame to 256 and perform the center crop of size 224×224. For the flow stream, we apply the TV-L1 optical flow algorithm (Zach et al., 2007). We follow the two-stream fusion operation in (Narayan et al., 2019) to integrate predictions from both appearance (RGB) and motion (Flow) branches. The inputs to the I3D models are stacks of 16 frames.

On all datasets, we set the learning rate to 10^{-3} for all experiments, and the model is trained with a batch size of 32 using the Adam (Kingma and Ba, 2015). Loss weight



```
video_001, 73.45s, Billiards
video_001, 84.40s, Billiards
video_001, 141.85s, Billiards
```

Figure 5.4: Interface for annotating a single frame. First step is to pause the video when annotators notice an action while watching the video. The second step is to select the target class for the paused frame. After annotating an action instance, the annotator can click the video to keep watching the video for the next action instance. Note that the time is automatically generated by the annotation tool. After watching a whole video, the annotator can press the generate button to save all records into a csv file.

hyper-parameters α and β are set to 1. The model performance is not sensitive to these hyper-parameters. For the hyper-parameter η used in mining background frames, we set it to 5 on THUMOS14 and set it to 1 on the other two datasets. The number of iterations is set to 500, 2000 and 5000 for GTEA, BEOID and THUMOS14, respectively.

5.3.3 Single Frame Annotation

We invite four annotators with different backgrounds to label single frames for all actions instances. Before annotating each dataset, four annotators have watched a few video examples containing different actions to become familiar with action classes. They are asked to annotate one single frame for each target action instance while watching the video by our designed annotation tool. Specifically, they must pause the video when they identify an action instance and choose the action class that the paused frame belongs to. Once they have chosen the action class, they need to continue watching the video and record the frames for the following target action instances. After watching the whole video, the annotator should press the generation button, and the annotation tool will then automatically produce the timestamps and action classes of all operated frames for the given video. The single frame annotation process is much faster than annotating the temporal boundary of each action in which the annotator often watches the video many times to define the start and end timestamp of a given action.

5.3.3.1 Annotation Guideline

Different people may have different understandings of what constitutes a given action. To reduce the ambiguity, we prepare a detailed annotation guideline, including clear action definitions and positive/negative examples with detailed clarifications for each action. For each action, we give (1) textual action definition for single frame annotation, (2) positive single frame annotations, and (3) segmented action instances which the annotator is familiar with.

5.3.3.2 Annotation Tool

Our annotation tool supports automatically recording timestamps for annotating single frames. This makes the annotation process faster when annotators notice an action and are ready to label the paused frame. The interface of our annotation tool is presented in Figure 5.4. After watching a whole video, the annotator can press the generate button, the annotation results will be automatically saved into a csv file. When annotators think they made a wrong annotation, they can delete it at any time while watching the video. We have shown the one annotation example in the supplementary file. We have uploaded a video in the supplementary file to show how to annotate single frame while watching the video.

Table 5.1: Single frame annotation differences between different annotators on three datasets. We show the number of action segments annotated by Annotator 1, Annotator 2, Annotator 3, and Annotator 4. In the last column, we report the total number of the ground-truth action segments for each dataset.

Datasets	Annotator 1	Annotator 2	Annotator 3	Annotator 4	# of total segments
GTEA	369	366	377	367	367
BEOID	604	602	589	599	594
THUMOS14	3014	2920	2980	2986	3007

5.3.3.3 Quality Control

We make two efforts to improve the annotation quality. First, each video is labeled by four annotators, and the annotated single frames of a view are randomly selected during experiments to reduce annotation bias. Secondly, we train annotators before annotating videos and make sure that they can notice target actions while watching the video.

5.3.3.4 Annotation Statistics

We also invite four annotators with different backgrounds to label a single frame for each action segment on three datasets. More details of the annotation process can be found in the supplementary material. In Table 5.1, we have shown the action instances of different datasets annotated by different annotators. The ground-truth in the Table denotes the action instances annotated in the fully-supervised setting. From the Table, we obtain that different annotators' number of action instances has a very low variance. The number of labeled frames is very close to the number of action segments in the fully-supervised setting. This indicates that annotators have a common justification for the target actions and hardly miss the action instance. They only pause once to annotate the single frame of each action.

We also present the distribution of the relative position of single frame annotation to the corresponding action segment. As shown in Figure 5.5, there are rare frames outside of the temporal range of action instances from the ground-truth in the fully-supervised

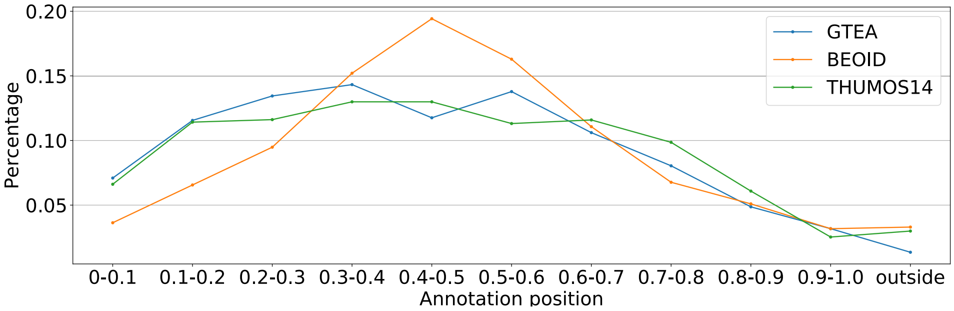


Figure 5.5: Statistics of human annotated single frame on three datasets. X-axis: single frame falls in the relative portion of the whole action; Y-axis: percentage of annotated frames. We use different colors to denote annotation distribution on different datasets.

setting. As the number of annotated single frames is almost the same as the number of action segments, we can infer that the single frame annotation includes all almost potential action instances. We obtain that annotators prefer to label frames near to the middle part of action instances. This indicates that humans can identify an action without watching the whole action segment. On the other hand, this will significantly reduce the annotation time compared with fully-supervised annotation as we can quickly skip the current action instance after single frame annotation.

5.3.3.5 Annotation Speed

To measure the required annotation resource for different supervision, we conducted a study on GTEA. Four annotators are trained to be familiar with action classes in GTEA. We ask the annotator to indicate the video-level, single frame, and temporal boundary labels of 21 videos lasting 93 minutes long. While watching, the annotator can skim, pause, and go to any timestamp. Each type of annotations are conducted separately where only video-level annotations are produced if the annotator is required to generate the video-level annotations. On average, the annotation time used by each person to annotate 1-minute video is 45s for the video-level label, 50s for the single frame label, and 300s for the segment label. The annotation time for the single frame label is close to the annotation time for the video-level label but much fewer than time for the

Table 5.2: Comparisons between different methods for simulating single frame supervision on THUMOS14. “Annotation” means that the model uses human annotated frame for training. “TS” denotes that the single frame is sampled from action instances using a uniform distribution, while “TS in GT” is using a Gaussian distribution near the mid timestamp of each activity. The AVG for segment localization is the average mAP from IoU 0.1 to 0.7.

Position	mAP@hit	Segment mAP@IoU			
		0.3	0.5	0.7	AVG
Annotation	**60.2**±0.70	**53.3**±0.30	28.8±0.57	9.7±0.35	**40.6**±0.40
TS	57.6±0.60	52.0±0.35	**30.2**±0.48	**11.8**±0.35	40.5±0.28
TS in GT	52.8±0.85	47.4±0.72	26.2±0.64	9.1±0.41	36.7±0.52

fully-supervised annotation.

5.3.4 Evaluation Metric

(1) **Segment localization**: We follow the standard protocol, provided with the three datasets, for evaluation. The evaluation protocol is based on mean Average Precision (mAP) for different intersection over union (IoU) values for the action localization task. (2) **Single frame localization**: We also use mAP to compare performances. Instead of measuring IoU, the predicted single frame is regarded as correct when it lies in the temporal area of the ground-truth segment, and the class label is correct. We use mAP@hit to denote the mean average precision of selected action frame falling in the correct action segment.

5.3.5 Ablation Study

5.3.5.1 Simulation of Single Frame Supervision

First, to simulate the single frame supervision based on ground-truth boundary annotations in the above three datasets, we explore the different strategies to sample a single frame for each action instance. We follow the strategy in Moltisanti et al. (2019) to generate single frame annotations with uniform and Gaussian distribution (**Denoted**

by TS and TS in GT). We report the segment localization at different IoU thresholds and frame localization results on THUMOS14 in Table 5.2. The model with each single frame annotation is trained five times. The mean and standard deviation of mAP is reported in the Table. Compared to models trained on sampled frames, the model trained on human annotated frames achieves the highest mAP@hit. As the action frame has the largest prediction score in the prediction segment, the model with higher mAP@hit can assist with localizing action timestamp more accurately when people need to retrieve the frame of target actions. When sampling frames are from near middle timestamps to the action segment (TS in GT), the model performs inferior to other models as these frames may not contain informative elements of complete actions. For the segment localization result, the model trained on truly single frame annotations achieves higher mAP at small IoU thresholds, and the model trained on frames sampled uniformly from the action instance gets higher mAP at larger IoU thresholds. It may be originated by sampled frames of uniform distribution containing more boundary information for the given action instances.

5.3.5.2 Module Design

To analyze the contribution of the classification module, actionness module, background frame mining strategy, and the action frame mining strategy, we perform a set of ablation studies on THUMOS14, GTEA and BEOID datasets. The segment localization mAP at different thresholds is presented in Table 5.3. We also compare the model with only weak supervision and the model with full supervision. The model with weak supervision is implemented based on Narayan et al. (2019). When the single frame annotations are adopted, large performance gain are obtained on all three datasets.

We observe that the model with single frame supervision outperforms the weakly-supervised model. Moreover, a significant performance gain is obtained on GTEA and BEOID datasets as the single video often contains multiple action classes, while action

Table 5.3: Segment localization mAP results at different IoU thresholds on three datasets. Weak denotes that only video-level labels are used for training. All action frames are used in the full supervision approach. SF uses extra single frame supervision with frame level classification loss. SFB means that pseudo background frames are added into the training, while the SFBA adopts the actionness module, and the SFBAE indicates the action frame mining strategy added in the model. For models trained on single frame annotations, we report mean and standard deviation results of five runs. AVG is the average mAP from IoU 0.1 to 0.7.

Dataset	Models	mAP@IoU				
		0.1	0.3	0.5	0.7	AVG
GTEA	Full	58.1	40.0	22.2	14.8	31.5
	Weak	14.0	9.7	4.0	3.4	7.0
	SF	50.0±1.42	35.6±2.61	**21.6**±1.67	**17.7**±0.96	30.5±1.23
	SFB	52.9±3.84	34.9±4.72	17.2±3.46	11.0±2.52	28.0±3.53
	SFBA	52.6±5.32	32.7±3.07	15.3±3.63	8.5±1.95	26.4±3.61
	SFBAE	**58.0**±2.83	**37.9**±3.18	19.3±1.03	11.9±3.89	**31.0**±1.63
BEOID	Full	65.1	38.6	22.9	7.9	33.6
	Weak	22.5	11.8	1.4	0.3	8.7
	SF	54.1±2.48	24.1±2.37	6.7±1.72	1.5±0.84	19.7±1.25
	SFB	57.2±3.21	26.8±1.77	9.3±1.94	1.7±0.68	21.7±1.43
	SFBA	**62.9**±1.68	36.1±3.17	12.2±3.15	2.2±2.07	27.1±1.44
	SFBAE	**62.9**±1.39	**40.6**±1.8	**16.7**±3.56	**3.5**±0.25	**30.1**±1.22
THUMOS14	Full	68.7	54.5	34.4	16.7	43.8
	Weak	55.3	40.4	20.4	7.3	30.8
	SF	58.6±0.56	41.3±0.62	20.4±0.55	6.9±0.33	31.7±0.41
	SFB	60.8±0.65	44.5±0.37	22.9±0.38	7.8±0.46	33.9±0.31
	SFBA	68.7±0.33	52.3±1.21	28.2±0.42	**9.7**±0.51	39.9±0.43
	SFBAE	**70.0**±0.64	**53.3**±0.3	**28.8**±0.57	**9.7**±0.35	**40.6**±0.40

classes in one video are fewer in THUMOS14. Both background frame mining strategy and action frame mining strategy boost the performance on BEOID and THUMOS14 by putting more frames into the training, the performance on GTEA decreases mainly due to that GTEA contains almost no background frame. In this case, it is not helpful to employ background mining and the actionness module to distinguish background against action. The actionness module works well for the BEOID and THUMOS14 datasets, although the actionness module only produces one score for each frame.

Table 5.4: The background η analysis on THUMOS14. AVG is the average mAP at IoU 0.1 to 0.7.

η	mAP@hit	mAP@IoU					
		0.1	0.3	0.5	0.6	0.7	AVG
0.0	44.4±0.56	58.6±0.55	41.1±0.80	20.2±0.69	12.9±0.58	7.3±0.10	31.7±0.47
1.0	57.7±0.41	68.3±0.37	51.1±0.57	28.2±0.52	17.7±0.09	9.4±0.31	39.3±0.13
3.0	60.6±1.36	71.0±1.21	53.8±0.71	29.3±1.14	18.9±0.88	9.4±0.43	41.1±0.80
5.0	60.6±0.85	70.6±0.92	53.7±1.21	29.1±0.39	19.1±1.31	10.2±0.84	41.1±0.78
7.0	60.9±0.56	70.7±0.08	54.3±1.18	29.5±0.13	19.0±0.50	10.1±0.27	41.3±0.44
9.0	60.2±1.12	70.3±0.83	53.4±0.8	29.6±0.58	18.8±0.99	10.1±0.37	41.0±0.60

Table 5.5: The loss coefficients analysis on THUMOS14. AVG is the average mAP at IoU 0.1 to 0.7.

parameter	mAP@hit	Segment mAP@IoU					
		0.1	0.3	0.5	0.6	0.7	AVG
$\alpha = 0.2$	61.9±0.34	71.6±0.73	54.2±1.31	29.3±0.47	18.4±0.62	9.7±0.35	41.3±0.56
$\alpha = 0.5$	61.9±0.68	71.8±0.36	54.4±0.68	30.2±0.41	19.3±0.92	10.2±1.14	41.9±0.47
$\alpha = 0.8$	60.7±0.95	71.0±0.40	53.8±0.64	29.4±0.26	19.0±0.23	10.0±0.25	41.2±0.22
$\beta = 0.2$	60.6±1.55	70.5±1.21	53.2±1.09	29.4±0.64	18.8±0.71	9.7±0.33	41.0±0.67
$\beta = 0.5$	60.2±0.69	70.5±0.55	53.7±0.71	29.4±0.16	18.8±0.47	10.0±0.34	41.1±0.42
$\beta = 0.8$	60.8±1.05	70.6±0.50	53.8±1.47	29.6±0.34	18.9±0.36	10.0±0.37	41.2±0.55

5.3.5.3 Background Ratio

Table 5.4 shows the results with respect to different background ratios η on THUMOS14. The mean and standard deviation of segment and frame metrics are reported. We ran each experiment three times. The single frame annotation for each video is randomly sampled from annotations by four annotators. From Table 5.4, we find that our proposed SF-Net boosts the segment and frame evaluation metrics on THUMOS14 dataset with background mining. The model becomes stable when the η is set in range from 3 to 9.

5.3.5.4 Loss Coefficients

We also conduct experiments to analyze the hyper-parameters of each loss item on the THUMOS14 in Table 5.5. The mean and standard deviation of segment and frame metrics are reported. We run each experiment three times. The single frame annotation

Table 5.6: Classification accuracy and class-agnostic localization AP on THUMOS14.

	Classification	Class-agnostic localization		
	mAP	AP@IoU=0.3	AP@IoU=0.5	AP@IoU=0.7
Ours w/o single frame	97.8	42.1	18.1	5.5
Ours w/ single frame	98.5	58.8	32.4	9.4

for each video is randomly sampled from annotations by four annotators. The default values of α and β are 1. Note that the main frame loss 5.2 is used in all experiments. We change one hyper-parameter and fix the other one in this experiment. From the Table 5.5, we observe that our model is not sensitive to these hyper-parameters. Our model achieves highest performance when the α is set to 0.2.

5.3.5.5 Classification & Localization Evaluation

We independently evaluate our single frame supervised and weakly-supervised models in terms of classification and localization. We adopt mean average precision (mAP) in Wang et al. (2016) to evaluate the video-level classification performance and AP at different IoU thresholds to evaluate the class-agnostic localization quality regardless of the action class. We report the video-level classification mAP in Table 5.6, showing only marginal gain as expected. THUMOS14 only contains one or two action classes in a single video, which makes the video easily classified into the target action category. We also evaluate boundary detection AP regardless of the label in Table 5.6, showing a large gain after adding single frame supervision.

5.3.5.6 Qualitative Results

We present the qualitative results on BEOID dataset in Figure 5.6. The first example has two action instances: *scan card* and *open door*. Our model localizes every action instance and classifies each action instance into the correct category. The temporal boundary for each instance is also close to the ground-truth annotation despite that we do not have

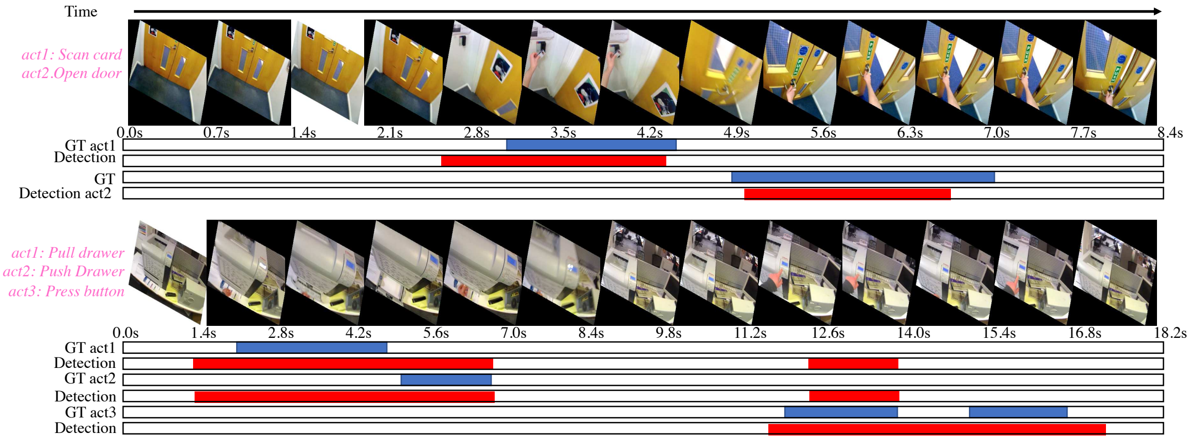


Figure 5.6: Qualitative Results on BEOID dataset. GT denotes the ground-truth and the action segment is marked with blue. Our proposed method detects all the action instances in the videos.

any temporal boundary information during training. For the second example, there are three different actions and total four action instances. Our SF-Net has detected all the positive instances in the videos. The drawback is that the number of detected segments for each action class is greater than the number of ground-truth segments. The model should encode the fine-grained action information from the target action area instead of the 1D feature directly extracted from the whole frame to better distinguish the actions of different classes. We will consider this in future work.

5.3.6 Comparisons with State-of-the-art

Experimental results on THUMOS14 testing set are shown in Table 5.7. Our proposed single frame action localization method is compared to existing methods for weakly-supervised temporal action localization and several fully-supervised ones. Our model outperforms the previous weakly-supervised methods at all IoU thresholds regardless of the choice of feature extraction network. The gain is substantial even though only one single frame for each action instance is provided. The model trained on human annotated frames achieves higher mAP at lower IoU compared to model trained on sampling frames

Table 5.7: Segment localization results on THUMOS14 dataset. The mAP values at different IoU thresholds are reported, and the column AVG indicates the average mAP at IoU thresholds from 0.1 to 0.5. * denotes the single frame labels are uniform sampled from the ground-truth annotations (see 5.3.5.1 for more details). # denotes single frame labels are manually annotated by human annotators.

Supervision	Method	mAP @IoU							
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	AVG
Full	S-CNN (Shou et al., 2016)	47.7	43.5	36.3	28.7	19.0	-	5.3	35.0
Full	CDC (Shou et al., 2017)	-	-	40.1	29.4	23.3	-	7.9	-
Full	R-C3D (Xu et al., 2017)	54.5	51.5	44.8	35.6	28.9	-	-	43.1
Full	SSN (Zhao et al., 2017b)	60.3	56.2	50.6	40.8	29.1	-	-	47.4
Full	Faster (Chao et al., 2018)	59.8	57.1	53.2	48.5	42.8	**33.8**	**20.8**	52.3
Full	BMN (Lin et al., 2019)	-	-	56.0	47.4	38.8	29.7	20.5	-
Full	P-GCN (Zeng et al., 2019)	**69.5**	**67.8**	**63.6**	**57.8**	**49.1**	-	-	**61.6**
Weak	Hide-and-Seek (Singh and Lee, 2017)	36.4	27.8	19.5	12.7	6.8	-	-	20.6
Weak	UntrimmedNet (Wang et al., 2017)	44.4	37.7	28.2	21.1	13.7	-	-	29.0
Weak	W-TALC (Ding and Xu, 2018)	49.0	42.8	32.0	26.0	18.8	-	6.2	33.7
Weak	AutoLoc (Shou et al., 2018)	-	-	35.8	29.0	21.2	13.4	5.8	-
Weak	STPN (Nguyen et al., 2018)	52.0	44.7	35.5	25.8	16.9	9.9	4.3	35.0
Weak	W-TALC (Paul et al., 2018)	55.2	49.6	40.1	31.1	22.8	-	7.6	39.7
Weak	Liu et al. (2019)	57.4	50.8	41.2	32.1	23.1	15.0	7.0	40.9
Weak	Nguyen et al. (2019)	**60.4**	**56.0**	**46.6**	**37.5**	**26.8**	**17.6**	**9.0**	**45.5**
Weak	3C-Net (Narayan et al., 2019)	59.1	53.5	44.2	34.1	26.6	-	8.1	43.5
Single frame simulation*	Moltisanti et al. (2019)	24.3	19.9	15.9	12.5	9.0	-	-	16.3
Single frame simulation*	SF-Net	68.3	62.3	52.8	**42.2**	**30.5**	**20.6**	**12.0**	51.2
Single frame#	SF-Net	**71.0**	**63.4**	**53.2**	40.7	29.3	18.4	9.6	**51.5**

uniformly from action segments. The differences are that the uniform sampling frames from ground-truth action segments contain more information about temporal boundaries for different actions. As there are many background frames in the THUMOS14 dataset, the single frame supervision assists the proposed model with localizing potential action frames among the whole video. Note that the supervised methods have the regression module to refine the action boundary, while we simply threshold on the score sequence and still achieve comparable results.

We conduct experiments on ActiviytNet1.2 by randomly sampling single frame annotations from ground-truth temporal boundaries. Table 5.8 presents the results on ActivityNet1.2 validation set. In this experiment, the annotations are generated by randomly sampling a single frame from ground-truth segments. We follow the standard evaluation protocal (Caba Heilbron et al., 2015) by reporting the mean mAP scores at different thresholds (0.5:0.05:0.95). Our proposed method can still obtain a performance

Table 5.8: Segment localization results on ActivityNet1.2 validation set. The AVG indicates the average mAP from IoU 0.5 to 0.95.

Supervision	Method	mAP @IoU			
		0.5	0.7	0.9	AVG
Full	CDC (Shou et al., 2017)	45.3	-	-	23.8
Full	SSN (Zhao et al., 2017b)	41.3	30.4	13.2	28.3
Weak	UntrimmedNet (Wang et al., 2017)	7.4	3.9	1.2	3.6
Weak	AutoLoc (Shou et al., 2018)	27.3	17.5	6.8	16.0
Weak	W-TALC (Ding and Xu, 2018)	37.0	14.6	-	18.0
Weak	Liu et al. (2019)	36.8	-	-	**22.4**
Weak	3C-Net (Narayan et al., 2019)	**37.2**	**23.7**	**9.2**	21.7
Single frame	SF-Net (**Ours**)	37.8	24.6	10.3	22.8

gain with single frame supervision on the large scale dataset.

5.4 Summary

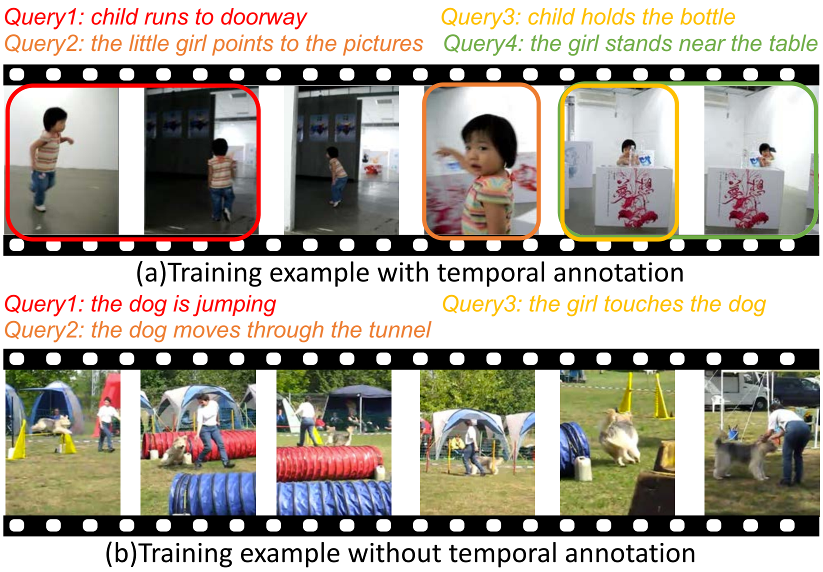
This chapter has investigated how to leverage single frame supervision to train temporal action localization models for both segment localization and single frame localization during inference. Our SF-Net uses single frame supervision by predicting actionness score, pseudo background frame mining and pseudo action frame mining. SF-Net significantly outperforms weakly-supervised methods in segment localization and single frame localization on three standard benchmarks. We demonstrate that the model can achieve impressive results even with data annotation containing much less information than the desired outputs.

CHAPTER 6

Weakly-supervised Moment Localization with Decoupled Consistent Concept Prediction

6.1 Introduction

Temporal localization of target events with natural language has attracted many interests recently (Anne Hendricks et al., 2017; Gao et al., 2017; Chen et al., 2018; Zhang et al., 2019). Given a natural language query, this task is to determine the start and the end timestamps of the described moment from a long video. The typical approach is to learn the correlations between the query sentence and the target video clip in a supervised way through temporal annotations (Anne Hendricks et al., 2017; Gao et al., 2017; Chen et al., 2018; Ning et al., 2018; Zhang et al., 2018), where the language description and the temporal location are given in each training sample (Figure 6.1). However, manually annotating temporal boundaries for a new large-scale dataset is extremely expensive and time-consuming (Zhao et al., 2017a). Learning with limited annotations is substantial in practical scenarios (Wang et al., 2020) which only requires a few samples for model learning. It is thus necessary to develop an algorithm that can automatically localize the


Query1: child runs to doorway
Query2: the little girl points to the pictures
Query3: child holds the bottle
Query4: the girl stands near the table
Query1: the dog is jumping
Query2: the dog moves through the tunnel
Query3: the girl touches the dog

(a)Training example with temporal annotation

(b)Training example without temporal annotation

Figure 6.1: Illustration of video moment retrieval task with natural language in the fully-supervised and weakly-supervised settings. The start and end timestamps of the video moment for each text query is given in the fully-supervised setting. These temporal annotations are not available in the weakly-supervised setting.

video moment using only the video-level texts.

In the weakly-supervised setting, modelling the relationship between the video clip and the query sentence is non-trivial due to the lack of supervision. Mithun et al. (2019) tackled this problem by learning a joint video-text embedding. A triplet ranking loss is used to maximize the similarities between the matched pairs, while the similarities of non-matched pairs are minimized. Gao et al. (2019) proposed to align and match video-sentence feature pairs to detect events in the video. However, when the supervision signals are absent, it is challenging to localize the visual representations through only sentence-level texts. On the contrary, the localization of atomic objects and actions, e.g., “hand”, “hat”, “put”, “turn”, is much easier. The localization for complex sentence queries could be achieved by exploiting the localization of atomic concepts in videos. As long as sufficient samples containing different atomic concepts are contained in the training dataset, the different video-language pairs could be exploited to build semantic connections between video clips and language descriptions. Gu et al. (2018) also showed that learning atomic components is helpful for higher-level event localization. The atomic components can be composed into complicated higher-level events where the simple components are naturally correlated. This chapter introduces the decoupled consistent concept prediction (DCCP) for weakly-supervised video localization with language queries.

Instead of directly learning correspondence between the query language and video clips, we decouple the task into the recognition of the objects and actions in both queries and videos. For example, in the text query "child holds the bottle", the nouns, i.e., "child", "bottle", and the verb "hold" are extracted. The language features of concepts and the video clip are mapped into the common embedding space for retrieving plausible video clips, which are further classified into the object or action categories. When the classification loss is minimized, the word and video clip features become more similar in the embedding space. The localization of the concept is thus resolved by matching features between word and video features. Our model can identify video moments for query sentences by leveraging the localisation of key concepts. To achieve this, we develop a concept pairing module in our DCCP framework to match the word with the video clips. In this module, the concept in the sentence is a probe to match moments in the video. The matching process is implemented in a pairing module with the attention mechanism, which has been widely used for relation reasoning in many natural language processing and computer vision tasks (Bahdanau et al., 2014; Vaswani et al., 2017). Our pairing module is fed with both concept language and video features and outputs the temporal localization map of concepts and the visual representation for concept classification. In addition, we construct a consistency loss to encourage that temporal localization of concepts in the same query language is overlapped. We make the following contributions in this chapter:

- A novel decoupled consistent concept prediction (DCCP) framework is proposed to

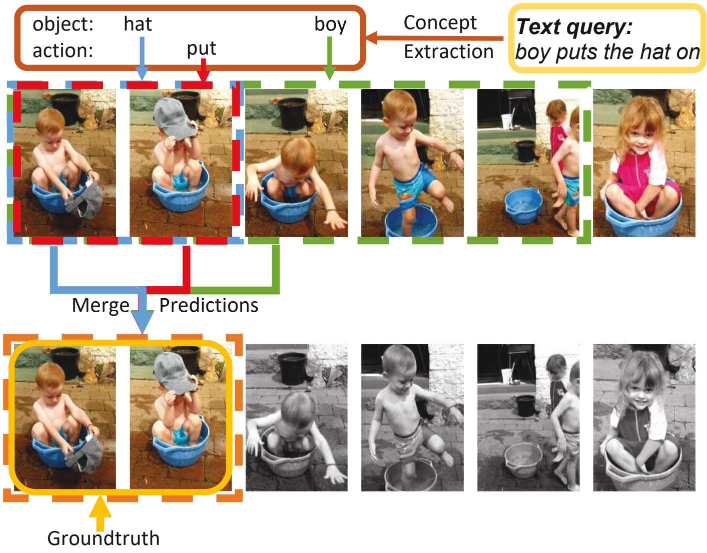


Figure 6.2: We demonstrate that localizing moments in a video with natural language can be well addressed through the localization of concepts in the video. The nouns and verbs in the query sentence are extracted to be localized in the video. The predictions from all concepts are merged to retrieve moments for the query sentence.

facilitate visual and language representation learning in the weakly-supervised video moment localization. Our framework decouples the moment localization with language into several sub-tasks of localizing key concepts.

- We introduce a concept pairing module in the DCCP framework for classification and a consistency loss for regularizing localization maps. Both designs effectively enhance the matching between the learned visual representation and the query concepts.

- Our framework achieves the state-of-the-art performance in the weakly-supervised setting on three standard moment retrieval datasets. Also, extensive ablations demonstrate the effectiveness of our proposed framework.

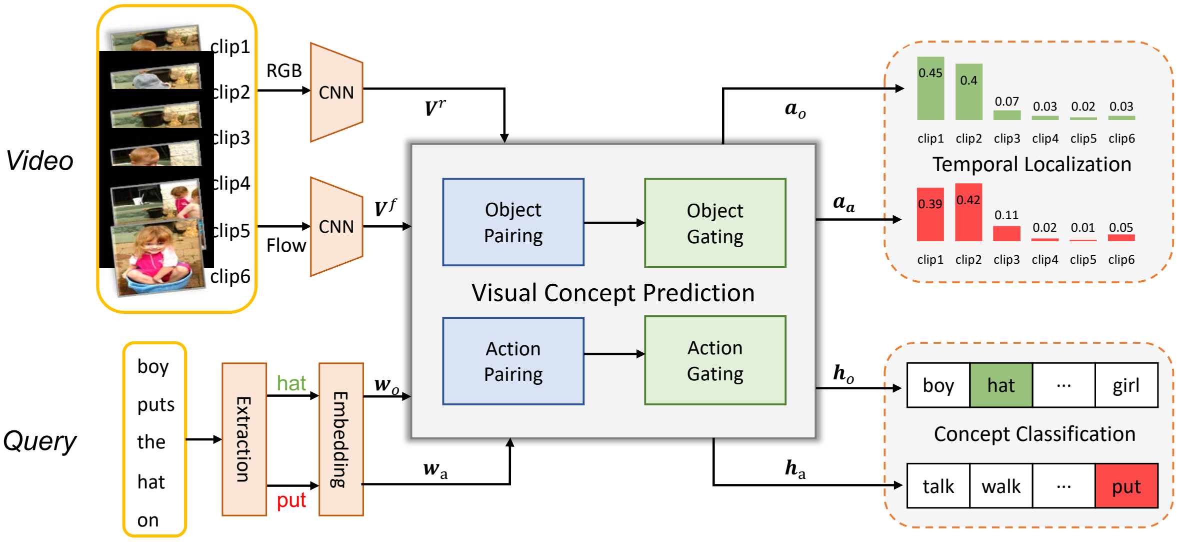


Figure 6.3: The framework of our decoupled consistent concept prediction (DCCP). The concepts in the query sentence are first extracted and then passed to the embedding layer. We introduce the concept prediction module to build semantic connections between the language and the video features. Details of the visual concept mining module are described in Section 6.2.2. The concept prediction module outputs the localization map and the visual embedding for each object and action concept, respectively. The localization map for the concept is the probability of each video clip containing that concept.

6.2 Decoupled Consistent Concept Prediction

6.2.1 Problem Setup

A video usually contains different visual objects (e.g., person, cat, train) and actions (e.g., run, shake) in different timelines. Events are highly related to these visual objects and actions, and the event descriptions also contain the corresponding object and action words. For instance, given a sentence in Figure 6.2, “the boy puts the hat on", the “boy" and “hat" are the objects and “put" is the action. The target video moment should also contain the visual concepts “boy" and “hat", and the action concept “put". When the query sentence becomes more complex, such as “the moment after the boy putting the hat on", the target video moment is still highly related to the moment “the boy puts the hat on”. Therefore, learning the semantic connections between videos and natural languages is significant for retrieving moments with natural language.

We focus on building the correlation between query sentences and videos through decoupled concepts in complicated query languages. We first extract the object and action concepts in the sentence and manage to localize these concepts in the video. These concepts can be extracted using POS tagger (Javed et al., 2018). Every words in the sentence would be tagged first via the POS tagger and the noun and verb words are extracted as the concepts. If the word embedding is similar to features of video clips containing the word concept, the temporal localization of the word could be determined by matching video features with the word embedding. As we use the pairing video clip for concept classification, features of video clips containing the target will be close to the language features in the embedding space after training. We construct such a model in Figure 6.3 where the visual representation of decoupled objects and actions are learned. In the meantime, the temporal localization of each concept is produced in the pairing module. The localization of all concepts is then used for retrieving moments for the query sentence.

The input video to our model is denoted as V. Each video is associated with descriptions $S = \{s_i\}_{i=1}^{M}$, where M is the number of sentence annotations, and s_j is a sentence description describing scenes in one of the video clips $V_{s_i} = \{f_t\}_{\tau^s}^{\tau^e}$. The start frame τ^s and end frame τ^e are unknown during training. The task is to predict the start $\hat{\tau}^s$ and the end $\hat{\tau}^e$ in videos for the natural language query s_i. Instead of directly localizing the moment with the sentence, we decouple the objects and actions from the query languages and localize the start and end timestamps of each concept in videos. As shown in Figure 6.3, we manage to utilize concept words to learn the visual representation in the given video. Once the visual embedding of the concept has been learned from the video, the moment containing that concept will be localized.

6.2.2 Framework

To learn the visual concept representation, we need first to localize the video moments containing that concept and then extract concept representation from localized video clips. We design a concept pairing module for localizing the concepts in the video and a feature gating block to mine visual concept embeddings for classification. The whole framework for concept prediction is displayed in Figure 6.4. There are concept pairing modules for matching the concept word with video clips and gating blocks for learning visual concept representation for each object and action concept.

6.2.2.1 Input Encoding

Along with the availability of large and well-labeled datasets in videos and languages, object and action concepts in videos and language can be well represented by off-the-shelf models. In this section, we adopt the existing vision and language models to encode the data.

Video Encoding. To encode both visual object and motion information in videos, we adopt *TVL1* (Zach et al., 2007) to extract optical flow and use RGB and flow images for extracting video features. For a long untrimmed video V, we split the video into H equal clips $V = \{v_i\}_{i=1}^{H}$, where H is the predefined number of clips. The start and end of the i^{th} clip are denoted by t_i^s and t_i^e, respectively. Then RGB features $\mathbf{V}^r = [\mathbf{v}_1^r, \cdots, \mathbf{v}_H^r]^T$ and flow features $\mathbf{V}^f = [\mathbf{v}_1^f, \cdots, \mathbf{v}_H^f]^T$ are produced to represent the visual encoding.

Language Encoding. We first extract all the nouns and verbs in all query sentences for the dataset. Afterwards, we sort these nouns and verbs based on the frequency in descending order. For the ease of the counting process, all nouns are singularized first, and all verbs are transformed into the present forms. The top N_o nouns and N_a verbs are preserved as object and action concepts for classification, respectively. The details of selecting the nouns and verbs are presented in Section 6.3.4.1. For each sentence

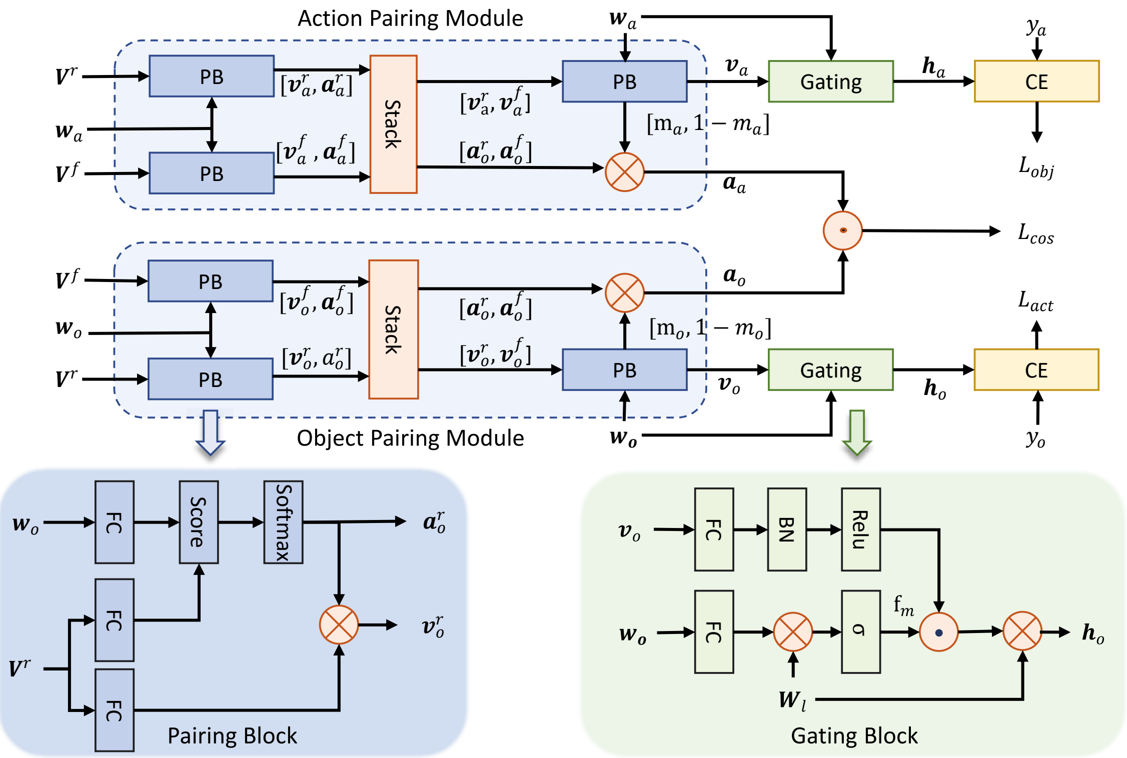


Figure 6.4: Illustration of our concept prediction module. The inputs include the object and action word embeddings, and the RGB and Flow video clip features. There are two types of concept pairing modules. Each module contains three pairing blocks ("PB"). "Gating" is the feature gating block to learn the visual concept embedding. $\odot$ and $\otimes$ are element-wise and matrix multiplication operations, respectively. σ denotes the sigmoid activation function. Two classification losses and one consistency loss are utilized for training the proposed model.

s, we extract the words belonging to the selected N_o objects or the N_a actions. The sentence and word features are extracted using the universal sentence encoder (Cer et al., 2018). We denote $\mathbf{w}_s$ as the sentence feature, $\mathbf{w}_o$ and $\mathbf{w}_a$ as the object and action word embedding, respectively.

6.2.2.2 Object and Action Pairing

The concept pairing module accepts word embeddings and video features as inputs, and produces the localization map and the visual feature for the query word. An intuitive solution to match the word with the correct video clip is to project features from different domains into the identical embedding space and select the video clip with the shortest distance to the concept word embedding as the matched video clip. This could be wrong when the target concept is contained in multiple video clips. Instead of directly selecting one video clip, we adopt the attention mechanism in the pairing module as it can leverage features from all video clips.

We adopt a word embedding as a probe to attend to video features to calculate the attention map and visual representation for the concept. As shown in Fig 6.4, the concept pairing module contains three pairing blocks. The first two blocks are used to learn RGB and Flow representations, and the third pairing block is to leverage features from previous blocks and output fused features for concept classification. Here we use the RGB feature $\mathbf{V}^r$ as input and denote $\mathbf{w}_o$ as the object word embedding for simple declaration.

For the k^{th} video clip in the RGB branch, the score containing the object concept is computed as follows:

$$a^r_{o,k} = \frac{e^{score(\mathbf{w}_o,\mathbf{v}^r_k))\}}}{\sum_{i=1}^{H} e^{score(\mathbf{w}_o,\mathbf{v}^r_i))\}}}, \tag{6.1}$$

where the function $score(\mathbf{w}_o, \mathbf{v}^r_k)$ calculates the pairing scores between the object and the k^{th} video clip. We adopt two designs to calculate pairing scores. We first map language and visual features into the embedding space. The inner product is then used in the first design to calculate the score:

$$score_{dot}(\mathbf{w}_o, \mathbf{v}^r_k) = w(\mathbf{w}_o\mathbf{W}_{po})^T(\mathbf{v}^r_k\mathbf{W}_{pv}) \tag{6.2}$$

where $\mathbf{W}_{po}$ and $\mathbf{W}_{pv}$ denote the projection matrix with respect to word and video features, respectively. w is the scalar for scaling scores. Another design is to calculate the matching score is by:

$$score_{add}(\mathbf{w}_o, \mathbf{v}^r_k) = \mathbf{w}_k{}^T\sigma(\mathbf{w}_o\mathbf{W}_{po} + \mathbf{v}^r_k\mathbf{W}_{pv}) \tag{6.3}$$

where σ denotes the sigmoid activation function. We can also use other non-linear functions, such as tanh. $\mathbf{w}_k$ denotes the weight parameter for weighting feature channels. The pairing block outputs the RGB visual feature $\mathbf{v}_o^r$ with respect to the object concept by averaging the features of video clips:

$$\mathbf{v}_o^r = \sum_k a_{o,k}^r \mathbf{v}_k^r \mathbf{W}_{pv}', \tag{6.4}$$

where $\mathbf{W}_{pv}'$ denotes the linear transformation matrix for the visual feature. The flow visual feature $\mathbf{v}_o^f$ is computed in the same way.

Given the word query, both RGB and Flow features could contain useful information, but we do not know which feature is more important for learning the visual concept representation. We stack both RGB ($\mathbf{v}_o^r$) and Flow ($\mathbf{v}_o^f$) features and feed it to the pairing block to learn a better visual representation for the given concept. The visual concept representation $\mathbf{v}_o$ is produced in terms of the query object word by:

$$\begin{aligned} \mathbf{v}_o &= m_o \mathbf{v}_o^r + (1 - m_o)\mathbf{v}_o^f, \\ m_o &= \frac{e^{score(\mathbf{w}_o, \mathbf{v}_o^r)\}}}{e^{score(\mathbf{w}_o, \mathbf{v}_o^r)} + e^{score(\mathbf{w}_o, \mathbf{v}_o^f)}}, \end{aligned} \tag{6.5}$$

We use the weight m_o to calculate the localization map $\mathbf{a}_o$ in terms of the given object by:

$$\mathbf{a}_o = m_o \mathbf{a}_o^r + (1 - m_o)\mathbf{a}_o^f. \tag{6.6}$$

Also, we obtain the action localization map $\mathbf{a}_a$ and the action feature $\mathbf{v}_a$ in the action pairing module in the same manner.

6.2.2.3 Feature Gating Block

The pairing module outputs the visual feature using the weighted average of all video features. Since a video clip may contain multiple concepts, the representation ability of the output feature for the single concept is limited. For instance, the weighted feature may contain both people and animal information, while only "people" exists in the query sentence. Using the weighted feature directly for the single target classification may impede the process of localizing the concept if the visual concept feature is not well learned.

To learn a more discriminative feature for classification., we employ a feature gating block in Figure 6.4. We take the object visual feature $\mathbf{v}^o$ as an example. The classification logit is calculated by:

$$\begin{aligned} \mathbf{h}_o &= \mathbf{W}_l(Relu(BN(\mathbf{W}_o\mathbf{v}_o)) \odot \mathbf{f}_m), \\ \mathbf{f}_m &= \sigma(\mathbf{W}_l\mathbf{W}_{ov}\mathbf{w}_o), \end{aligned} \tag{6.7}$$

where $\mathbf{W}_o$ and $\mathbf{W}_l$ are learnable parameters for the object visual feature, $\mathbf{W}_{ov}$ is the projection matrix for the object word feature, σ is a sigmoid operation, and BN denotes the batch normalization operation. $\mathbf{f}_m$ is a soft concept-oriented feature mask to force the visual feature focusing on dimensions highly related to the given word query.

The object prediction probability $\bar{\mathbf{y}}_o$ is calculated on $\mathbf{h}_o$ with a $softmax$ function. Similarly, the prediction $\bar{\mathbf{y}}_a$ for the action concept can also be calculated. All the parameters introduced for computing each prediction do not share weights.

6.2.3 Training

To train the whole network, we use the cross entropy loss for learning both object and action concept embeddings. We also introduce the consistency loss to relate the predictions from objects with the predictions from action pairing modules. During training, the object and action concepts are randomly sampled from the query sentence. Note that the number of objects and actions are determined before training and we use the classification layer with a fixed vocabulary. The classification loss for the object concept is calculated as follows:

$$\mathcal{L}_{obj} = \sum_{i=1}^{N_o} -y_o^i log \bar{y}_o^i, \tag{6.8}$$

where $y_o^i = 1$ if the query word is the i^{th} selected object word. The classification loss of action concept L_{act} is computed in the same way.

As the video clip related to the query sentence usually contains both object and action concepts, the localization map for both object and motion should also be correlated. If k^{th} video clip is the target clip for the query sentence, both a_o^k and a_m^k should be large to denote higher probabilities. We design the following consistency loss to regularize the localization map of objects and actions:

$$\mathcal{L}_{cos} = 1 - \mathbf{a}_o^T \mathbf{a}_a. \tag{6.9}$$

The consistency loss is to ensure that the localization maps of objects and actions should have overlap. Our training objective is then formulated as follows:

$$\mathcal{L} = \mathcal{L}_{obj} + \mathcal{L}_{act} + \beta \mathcal{L}_{cos}, \tag{6.10}$$

where β is the hyperparameter to leverage the consistency loss. When the loss function is minimized, the localization map will be updated to learn a good visual representation for the query concept.

6.2.4 Inference

The classification task is designed to localize correct moments by projecting the language features and the visual features into the same embedding space. During inference, our model only deployed the pairing module for localizing moments without classifying video clips. In addition, we only sample a single noun and verb from the query sentence as one training example to the model, while all the object and action words in the sentence

Algorithm 5 Proposal Generation

```
**Input:** $H$ video clips, the $i^{th}$ clip is represented by $(t_i^s,\ t_i^e,\ a_i)$.
$D \leftarrow \{(t_i^s, t_i^e, a_i)\}_{i=1}^{H}$
$C \leftarrow Combination(H, 2)$
**for** $(i, j)$ in $C$ **do**
    $s \leftarrow t_i^s,\ e \leftarrow t_j^e$
    $A \leftarrow [a_i, \cdots, a_j]$
    **if** $std(A) < \lambda$ and $mean(A) > 1/H$ **then**
        $a \leftarrow max(A) + \lambda * (j - i)$
    **else**
        $a \leftarrow min(A) - \lambda * (j - i)$
    **end if**
    $D \leftarrow D \cup \{(s, e, a)\}$
**end for**
**Output:** Proposal clips $D$
```

are used during inference. Suppose we have n concept words for the query sentence, the model will output n attention maps. We simply average all the results to produce the localization map for the query sentence.

For the i^{th} video clip, we use $(t_i^s,\ t_i^e,\ a_i)$ denote the start frame, end frame and prediction score, respectively. To generate proposals of longer video moments, we use the combination of H clips to create $\binom{H}{2}$ longer video clips. The score of each new clip is updated by merging predictions from video clips in the combination result. The detailed generation process is presented in Algorithm 5.

After generating more clip proposals, we have a set D containing several video clips with confidence scores. We then retrieve the video clip for the query sentence from the result set D. The video clip with the highest score is treated as the matched clip for the query sentence. The predicted start and end frames are calculated as follows:

$$\begin{gathered} \hat{t}^s = D_{k^*}[0],\ \hat{t}^e = D_{k^*}[1], \\ where\ k^* = \underset{k}{\operatorname{argmax}} D_k[2], \end{gathered} \tag{6.11}$$

where $D_k[i]$ denote the $i^{th}(i = 0, 1, 2)$ element of the k^{th} video clip in the set D.

6.3 Experiments

6.3.1 Datasets

We present experiments on three benchmark datasets for video moment retrieval with natural language to evaluate the performance of our proposed framework.

DiDeMo dataset for text to video moment retrieval was first introduced in Anne Hendricks et al. (2017). DiDeMo consists of 10,464 long videos; each video is 25-50 seconds and is split into six equal clips for a fair comparison. All the videos are collected from Flickr and contain a total of 26,892 moments. There are 33008, 4180, and 4022 video-sentence pairs for training, validation, and testing, respectively. The task in DiDeMo is to locate the video clip given a sentence query.

Charades-STA dataset (Gao et al., 2017) is built on the Charades (Sigurdsson et al., 2016) dataset, which contains 9,958 videos with an average length of 30 seconds. There are 16,128 video-sentence pairs contained in the Charades-STA dataset. The released annotations consist of 12,408 and 3,720 video-sentence pairs for training and testing, respectively.

ActivityNet Captions (Krishna et al., 2017) is a large-scale dataset of human activities. It contains 20k videos, amounting to 849 video hours with 100k total descriptions, each with its unique start and end time.

6.3.2 Implementation Details

We implement our model on Tensorflow (Abadi et al., 2016). For the languages in all datasets, we use pre-trained Universal Encoder (Cer et al., 2018) for extracting language features. We use the off-the-shelf tool NLTK (Loper and Bird, 2002) to extract nouns and verbs in the sentence and convert the nouns and verbs to the singular form and present tense, respectively. For DiDeMo dataset, we use pre-trained two-stream ConvNets (Simonyan and Zisserman, 2014a) as video clip encoder following Anne Hendricks et al. (2017). For the Charades-STA dataset, we use the extracted feature following Lin et al. (2020) which is released on the public website. We use a sliding window of length 192 frames with an overlap of 128 frames following Ning et al. (2018) to generate candidate video moments in the test set. For ActivityNet captions, we follow the setting in Gao et al. (2019) to split the video into five clips and only C3D (Tran et al., 2015) RGB feature are used. We use the gradient descent algorithm with a learning rate of 1.0×10^{-3} and moment 0.9 to train our proposed model for 20k steps. We set the batch size to 128 during the training period. The coefficient hyperparameter for the consistency loss β is set to 0.1, and the threshold λ is set to 0.05.

6.3.3 Evaluation Metric

For DiDeMo, we use the evaluation criteria following prior works in literature (Anne Hendricks et al., 2017). Specifically, the quality of an algorithm is judged by rank-based performance R@N (Recall at K), which calculates the percentage of test samples for which the correct result is found in the top-K retrievals to the query description. We present results for R@1, R@5, and R@10. The mean intersection over union (mIoU) is also adopted for evaluating DiDeMo dataset. Different from the DiDeMo dataset, which is to generate the clip number for a sentence query, the start and end time of the paired video clip is required to be produced in Charades-STA and ActivityNet Captions. We use R@1 and R@5 with different IoU thresholds to evaluate the proposed model.

6.3.4 Ablation Study

In this section, we conduct ablation studies on DiDeMo to demonstrate the effectiveness of each designed module in our proposed model. The localization map is also visualized for DiDeMo and Charades-STA.

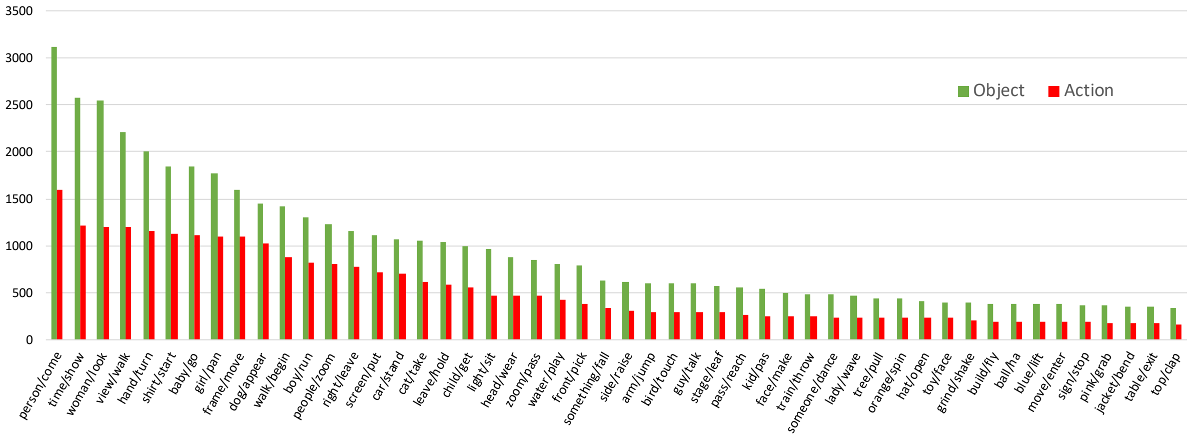


Figure 6.5: The number of top-50 nouns and verbs. We treat nouns in the sentence as objects and verbs as actions. We use green and red to distinguish the object and action.

Table 6.1: Comparisons on DiDeMo validation set in terms of vocabulary volume. Highest score is marked in bold.

words	Verb@10	Verb@50	Verb@100	Verb@200	Verb@400
	R@1 / R@5 / mIoU	R@1 / R@5 / mIoU	R@1 / R@5 / mIoU	R@1 / R@5 / mIoU	R@1 / R@5 / mIoU
Noun@10	17.13/67.01/29.84	16.74/67.34/28.87	17.15/67.96/28.63	17.40/68.18/29.75	16.94/68.11/28.09
Noun@50	17.66/67.75/31.69	18.10/**69.33**/30.75	16.10/68.49/27.76	17.00/68.32/28.53	14.88/65.57/25.89
Noun@100	16.95/67.99/29.83	**18.29**/67.48/31.41	17.32/69.18/29.45	16.19/68.27/28.36	17.44/68.42/29.14
Noun@200	17.36/68.37/29.76	18.25/67.08/**31.85**	17.05/67.91/29.20	16.98/69.23/28.56	17.00/68.70/29.28
Noun@400	17.60/68.37/29.56	17.66/67.39/30.68	17.58/67.63/29.59	15.66/63.11/27.04	15.86/68.58/27.15

6.3.4.1 Concept Selection

The object and action concepts are selected from all query sentences. We evaluate the model with different numbers of nouns and verbs. We sort all the nouns and verbs based on their frequencies in descending order and select the top-k words. Top 50 nouns and verbs are displayed in Figure 6.5. We remove the top-1 verb “be”, as it is not appropriate to be treated as an action concept. These nouns and verbs are the target object and action concepts required to be classified during training. We set the number of nouns and verbs to 10, 50, 100, 200, 400. The results are shown in Table 6.1. From the table, we observe that the proposed model achieves the highest mIoU with 200 nouns and 50 verbs. Our model performs more stable when the number of selected verbs is set to a medium value. Note that the number of training instances significantly decreased as more nouns

Table 6.2: Comparisons on DiDeMo validation set in terms of loss functions.

$\mathcal{L}_{obj}$	$\mathcal{L}_{act}$	$\mathcal{L}_{cos}$	R@1	R@5	mIoU
✓	✗	✗	15.00	62.94	30.01
✗	✓	✗	15.91	64.54	28.51
✗	✗	✓	10.23	56.84	16.81
✓	✓	✗	16.67	66.36	30.87
✓	✓	✓	18.25	67.08	31.85

Table 6.3: Comparisons on DiDeMo validation set in terms of video features.

Feature	R@1	R@5	mIoU
RGB	14.70	64.70	26.79
Flow	17.93	65.64	30.46
Average RGB + Flow	16.55	66.83	29.98
Attention RGB + Flow	18.25	67.08	31.85

or verbs are contained. Although localizing more concepts would help localize target moments, the imbalanced supervision makes the model hard for training and impacts the model performance. Therefore, there is a trade-off between the number of concepts and model performance. From Table 6.1, we observe that the proposed model performs stable when 200 nouns are selected. And the same situation happens when the model with 50 verbs are selected as action classes when the picked object classes are varied. This helps us choose the appropriate number of object and action concepts to get a good model. We use 200 nouns and 50 verbs as the default setting.

6.3.4.2 Loss Function

The loss function is trained to guide the model to find the correct localization of the given concept. In Table 6.2, we evaluate models with different loss combinations. First, we evaluate our model with the single loss to guide the training process. The model trained with a single consistency loss achieves the lowest performance. It is likely that a single consistency loss can not well guide the concept learning process. Then we validate the

Table 6.4: Comparisons on DiDeMo in terms of pairing score functions.

Score function	Val Set			Test Set		
	R@1	R@5	mIoU	R@1	R@5	mIoU
$l2$	12.60	54.49	20.84	13.13	52.27	22.11
$score_{dot}$	15.26	65.91	27.11	14.62	64.58	26.22
$score_{add}$ / σ	18.03	**67.96**	30.35	17.31	66.53	29.80
$score_{add}$/ tanh	**18.25**	67.08	**31.85**	**19.34**	**67.12**	**32.51**

proposed model using object and action concept classification losses together. The result is higher than the model using only a single object or verb classification loss. The model trained with both classification and consistency losses acquires the highest performance. It demonstrates that all designed losses are critical for localizing moments with natural languages.

6.3.4.3 Feature Fusion

We use both RGB and Flow features to localize moments in the experiment. In Table 6.3, we report the results of models with different RGB and Flow features. Our model with attention weights in Eq. (6.6) achieves much higher performance compared to models with RGB or Flow features only. This indicates that both RGB and Flow features contain useful information for localizing target moments.

6.3.4.4 Pairing Block

The pairing block is essential for matching the query word with video clips as the pairing result is used for visual concept mining during training and moment localization during inference. We compare different pairing blocks and evaluate them on both validation and test sets in Table 6.4. First, we use $l2$ to denote the model using Euclidean distance for pairing the query concept and video clips. The clip with minimum distance to the query feature is treated as the prediction clip. The $score_{dot}$ and $score_{add}$ denotes the

Table 6.5: Results with feature gating strategies on DiDeMo.

Configuration	Val Set			Test Set		
	R@1	R@5	mIoU	R@1	R@5	mIoU
W/O Gate	14.52	61.48	25.91	14.72	61.62	26.76
With Gate	18.25	67.08	31.85	19.34	67.12	32.51

model trained with score function Eq. (6.2) and Eq. (6.3), respectively. We also evaluate the model with the sigmoid and tanh activation functions. The models trained with the $score_{add}$ score function achieves better performance than models using the dot-based score function. Moreover, the activation function $sigmoid$ is inferior to $tanh$ on validation and testing set in terms of R@1 and mIOU metrics.

6.3.4.5 Feature Gating

The feature gating block is for learning a discriminative visual concept embedding, which implicitly improves the localization performance. We measure the contribution of the gating block in Table 6.5. Without any feature manipulation, the model performs inferior to models with feature gating. This validates the effectiveness of our proposed feature gating block.

6.3.4.6 Example Predictions

We qualitatively present a few examples illustrating the effectiveness of our weakly concept localization method. We qualitatively present a few examples illustrating the impact of our self-supervised concept localization method in Figure 6.6. Specifically, we display the localization map of every concept in the query sentence and present the final temporal localization with a green square. Note that there may be several ground-truth segments for one query sentence in DiDeMo, our model only outputs the segment with the highest localization score. The probabilities are produced by our DCCP model. We observe that the video clips can be localized through different concepts in the query

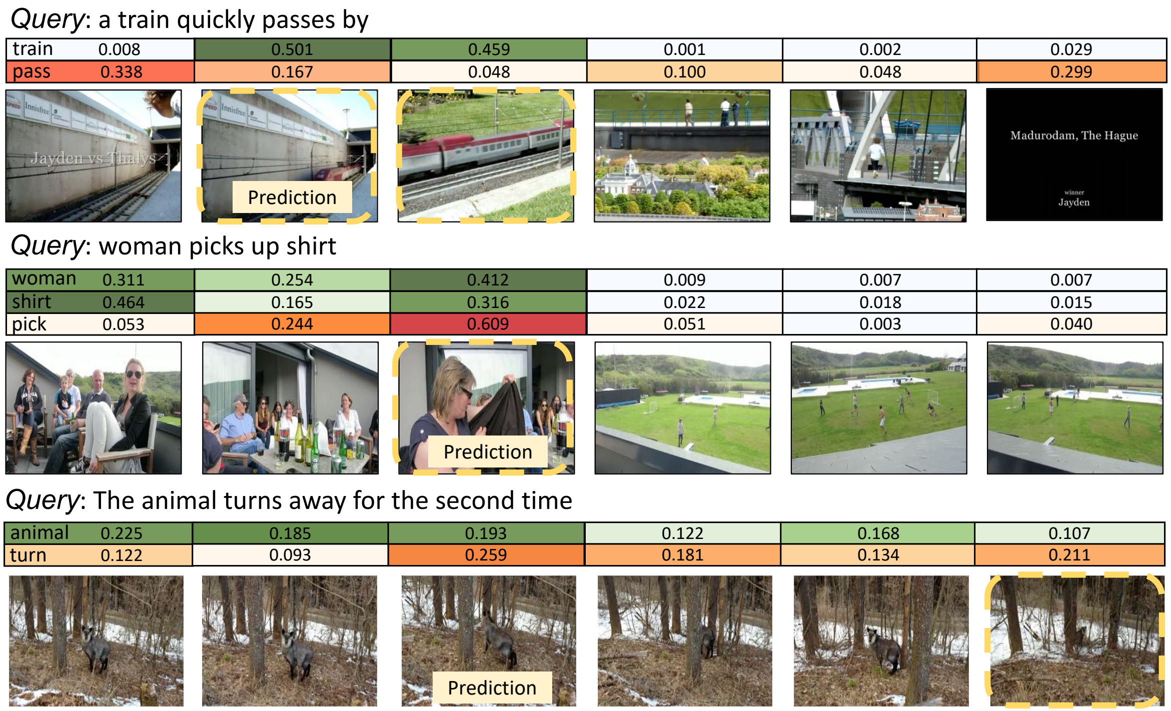

Query: a train quickly passes by
train 0.008 0.501 0.459 0.001 0.002 0.029
pass 0.338 0.167 0.048 0.100 0.048 0.299
Jayden vs Thalys
Prediction
Madurodam, The Hague
winner
Jayden
Query: woman picks up shirt
woman 0.311 0.254 0.412 0.009 0.007 0.007
shirt 0.464 0.165 0.316 0.022 0.018 0.015
pick 0.053 0.244 0.609 0.051 0.003 0.040
Prediction
Query: The animal turns away for the second time
animal 0.225 0.185 0.193 0.122 0.168 0.107
turn 0.122 0.093 0.259 0.181 0.134 0.211
Prediction

Figure 6.6: Natural language moment retrieval results on DiDeMo. Ground-truth moments are outlined with the yellow dashed line and retrieved moments are marked with the orange rectangle.

sentences.

For the first example, the moment “a train quickly passes by" is localized by the object concept “train". The localization of the action concept “pass" is wrong since “pass" is a metaphysical concept which makes it hard to be learned in the proposed network. However, Our proposed model successfully localizes two moments containing trains with high confidence and eventually retrieves the right moment for the query sentence. The second example contains three concepts in total, two of them are object concepts (woman and shirt) and the other one is action concept (pick). Since there are many people with clothes shown in videos, localizing moments becomes hard using predictions from only the object branch. Our algorithm detects that “woman" and “skirt" may be contained in the three video segments from the start. And through the localization of the action concept “pick", the target moment is localized accurately through the proposed method.

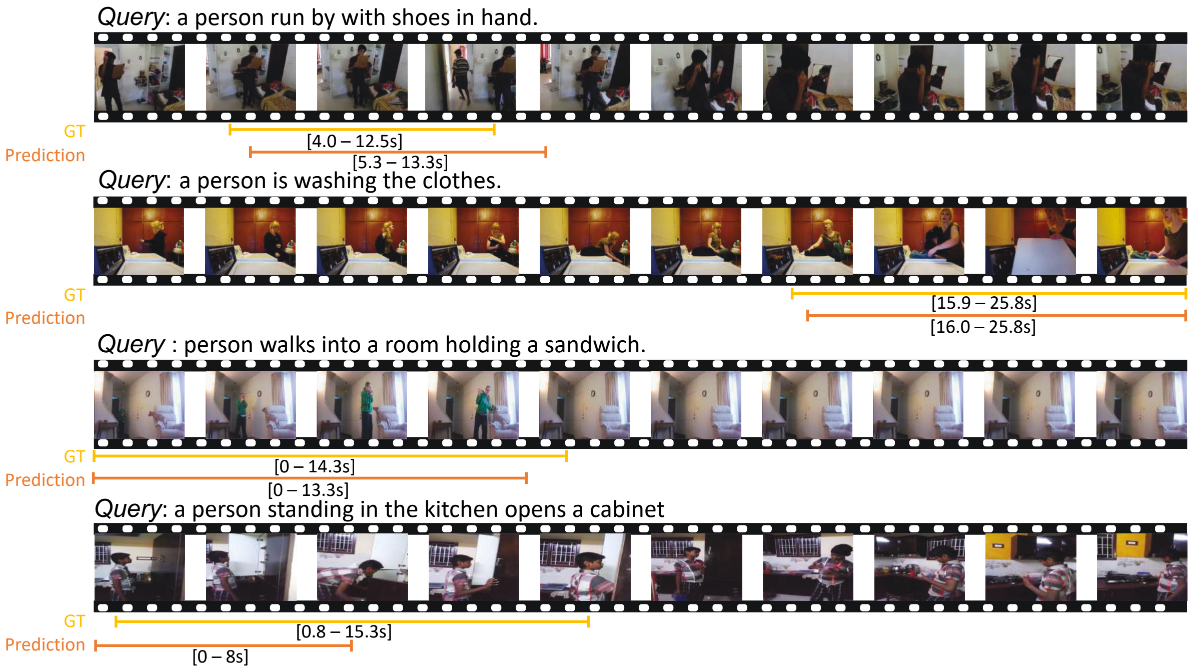

Query: a person run by with shoes in hand.
GT
Prediction
[4.0 – 12.5s]
[5.3 – 13.3s]
Query: a person is washing the clothes.
GT
Prediction
[15.9 – 25.8s]
[16.0 – 25.8s]
Query : person walks into a room holding a sandwich.
GT
Prediction
[0 – 14.3s]
[0 – 13.3s]
Query: a person standing in the kitchen opens a cabinet
GT
Prediction
[0.8 – 15.3s]
[0 – 8s]

Figure 6.7: Natural language moment retrieval results on Charades-STA by the proposed method. Ground-truth moments are marked in yellow and retrieved moments are marked in orange.

This not only validates that localizing moments with natural language can be solved by localizing concepts in the language, but also testifies the effectiveness of our model for localizing concepts. For the last examples, our model retrieves the first “turning'' moment instead of the second one. This is due to the difficulty of reasoning temporal information from language. Moreover, the task becomes more challenging to connect complicated language with the video moments for inferring the visual context. Our model is currently unable to handle such cases.

We also display a few examples on Charades-STA dataset in Figure 6.7. Specifically, we present the start and end timestamps of predictions and ground-truth given the query sentence. We obtain that our proposed model can output predictions with varied lengths. This validates the effectiveness of our merge algorithm on predictions. In each case, the target moment contains all concepts in the query sentence, and some concepts are only shown in the ground-truth video moment. For example, the “shoes”, “hand”, and “run”

Table 6.6: Comparisons on DiDeMo with different methods.

Method	Mode	R@1	R@5	mIoU
Random	-	3.75	22.50	22.64
CCA	Full	18.11	52.11	37.82
MCN (Anne Hendricks et al., 2017)	Full	19.88	62.39	33.51
TGN (Chen et al., 2018)	Full	28.23	79.26	42.97
ASST (Ning et al., 2018)	Full	30.53	77.34	47.14
MLLC (Hendricks et al., 2018)	Full	25.65	73.60	40.86
TGA (Mithun et al., 2019)	Weak	12.19	39.74	24.92
WSLLN (Gao et al., 2019)	Weak	18.40	54.40	27.40
VLANet (Ma et al., 2020c)	Weak	19.32	65.68	25.33
WSTAN (Wang et al., 2021a)	Weak	**19.40**	54.64	31.94
FSAN (Wang et al., 2021b)	Weak	**19.40**	57.85	31.92
DCCP(Ours)	Weak	19.34	**67.12**	**32.51**

concepts in the first case are only present in the ground-truth moment. The retrieval results in the second and third examples are longer than the length of predefined video clips (8s). It shows that adjacent clips have close prediction scores, and we retrieve the video moment more accurately with the final fused predictions. The prediction in the fourth case only gives a shorter video clip compared to the ground-truth clip. We believe it is difficult to capture without additional spatial and temporal reasoning as the moment of "washing hands" is not visible in the given video. Moreover, utilizing more cues from videos (e.g., audio, and context) may be helpful in localizing the target moment.

6.3.5 Comparisons with State-of-the-art

We first compare with other state-of-the-art supervised and weakly-supervised algorithms on DiDeMo in Table 6.6. All current supervised methods use temporal boundary annotations during training. The random result is generated by selecting a video clip randomly from the 21 video clips given a query sentence. Canonical correlation analysis (CCA) seeks vectors to maximize the correlation between sentence and video repre-

Table 6.7: Comparisons on ActivityNet Captions with different methods.

Method	Mode	R@1 IoU=0.5	R@1 IoU=0.3	R@5 IoU=0.5	R@5 IoU=0.3
Random	-	7.63	18.6	29.4	52.7
CTRL (Gao et al., 2017)	Full	14.0	28.7	-	-
QSPN (Xu et al., 2019)	Full	27.7	45.3	59.2	75.7
WSLLN (Wang et al., 2021a)	Weak	22.7	**42.8**	-	-
DCCP(Ours)	Weak	**23.2**	41.6	41.7	61.4

sentations. MCN (Anne Hendricks et al., 2017) integrates context features with the target clip feature to learn a better sentence-video match. CAL (Escorcia et al., 2019) aligns the video clips with language and demonstrates that data bias can be exploited using the temporal endpoint feature and temporal annotations. We report the result without the tef for a fair comparison. We also report results of ASST (Ning et al., 2018), TGN (Chen et al., 2018), and MLLC (Hendricks et al., 2018) methods. All these supervised methods utilize sentence and target video clips to build semantic connections. We report the weakly-supervised method TGA (Mithun et al., 2019), which employed the text-guided attention mechanism to minimize the distance of the video and language features. WSLLN (Gao et al., 2019) aligned the sentence feature with the detected clips. VLANet (Ma et al., 2020c) learned sharper attention by pruning out spurious candidate proposals. WSTAN (Wang et al., 2021a) aligned cross-modal semantic information by exploiting adjacent temporal networks in a multiple instance learning paradigm. FSAN (Wang et al., 2021b) learned token-by-clip cross-modal semantic alignment by an iterative cross-modal interaction module From Table 6.6, we can observe that our proposed model achieves the highest R@1 and R@5 results at IoU=0.5 compared to other weakly-supervised methods. Our proposed model is on par with the supervised CCA. Overall, our proposed model performs well even if no temporal annotations are provided.

We evaluate our method on ActivityNet Captions in Table 6.7. All the results are reported from the original papers. The “Random” result in the first line of the table is a

Table 6.8: Comparisons on Charades-STA with different methods.

Method	Mode	R@1	R@1	R@5	R@5
		IoU=0.7	IoU=0.5	IoU=0.7	IoU=0.5
Random	-	3.03	8.51	14.1	37.1
CTRL (Gao et al., 2017)	Full	7.15	21.4	26.9	29.1
CAL (Escorcia et al., 2019)	Full	12.0	-	23.0	-
ACL (Ge et al., 2019)	Full	12.2	30.5	35.1	64.8
SAP (Chen and Jiang, 2019)	Full	13.4	27.4	38.2	66.4
QSPN (Xu et al., 2019)	Full	15.8	35.6	45.4	77.0
TGA (Mithun et al., 2019)	Weak	8.84	19.9	33.5	65.5
WSLLN (Wang et al., 2021a)	Weak	9.21	25.3	30.5	68.8
SCN (Lin et al., 2020)	Weak	9.97	23.5	**38.8**	71.8
WSTAN (Wang et al., 2021a)	Weak	**12.2**	29.3	-	-
DCCP(Ours)	Weak	11.9	**29.8**	32.2	**77.2**

trivial baseline that we randomly select among candidate clips. CTRL (Gao et al., 2017) used a cross-model to retrieve moments with query sentences. QSPN (Xu et al., 2019) employed a multi-task loss to detect query sentences. All these methods used temporal annotations during training and trained a regression model to refine the boundary of the candidate clips. We observe that our model achieves similar performance compared to WSLLN on this dataset. This could be explained by the fact that the ActivityNet Captions contains many extremely long sentences and the scenes in videos are more complicated than the other two datasets.

We then compare our methods with the state-of-the-art supervised methods on Charades-STA, and the results are presented in Table 6.8. ACL (Ge et al., 2019) method is proposed to mine activities from both video and language modalities using activity concepts. SAP (Chen and Jiang, 2019) integrated semantic information in sentence query for better localizing the video activities. For the weakly-supervised methods, we also report results of TGA (Mithun et al., 2019), WSLLN (Gao et al., 2019), SCN (Lin et al., 2020) and WSTAN (Wang et al., 2021a). It is observed that our model outperforms other weakly-supervised methods in R@1 metrics. Although temporal annotations are significant for learning sentence-moment relationships, our proposed method achieves impressive performance compared to other supervised methods.

6.4 Summary

This chapter attempts to localize moments in videos for language in a weakly-supervised setting. We introduce a decoupled consistent concept prediction (DCCP) framework to build the relationship between video clips with decoupled concepts in the sentence. The concept prediction module is decoupled into object and action concept mining modules in DCCP. Each module consists of a pairing module to match the word with a video clip and learn a common embedding space for language and visual features. Both the object and action concept mining modules are optimized simultaneously by introducing a consistency loss. Experiments demonstrate the effectiveness of our proposed method and display an excellent performance for localizing moments with languages even compared to supervised methods. Temporal reasoning is essential for the complicated query sentence to localize the correct video moment. This requires comprehensive language reasoning on the temporal dimension, and we will consider it in future.

Chapter 7

Conclusion and Future Work

In this dissertation, we present that machine learning models can learn with imperfect data. We categorize the imperfect training data into three classes: 1) limited annotations; 2) noisy annotations; 3) weak annotations. Algorithms directly learned from these data usually perform poor in practical applications. We have designed several algorithms in chapters to show that models with impressive performance can be obtained with these imperfect data. In chapter 3, we studied the SSL and introduced a self-paced co-training (SPaCo) for various tasks where limited annotated samples and amounts of unlabeled samples are available. Chapter 4 solved the noisy training by proposing a self-reweighing scheme based on learned class centroids. We investigated two practical weak annotation problems in Chapters 5 and 6 and developed different networks for each task. Experiments demonstrate that our methods archive impressive performance when trained with weakly annotated data.

However, the model for noisy learning may not work on the weak annotations and vice versa. Annotated data of different types are processed separately in the present manuscript. In practice, even the weak annotations could have corrupted labels. Therefore, designing a unified algorithm for all types of imperfect supervision is thus a significant and challenging task and will attract more attention in the future. Also, fair classification (Fogliato et al., 2020) is also significant when imperfect annotations are available. It would be interesting to investigate how each category information supervise the learning process. In addition, the temporal information in videos and connections between multi-modal sources are not well encoded in this dissertation. It is also worth further studies on the temporal modeling and visual language learning.

Bibliography

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al.
Tensorflow: A system for large-scale machine learning.
In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

Steven Abney.
Bootstrapping.
In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 360–367. Association for Computational Linguistics, 2002.

Ehsan Amid, Manfred KK Warmuth, Rohan Anil, and Tomer Koren.
Robust bi-tempered logistic loss based on bregman divergences.
In *Advances in Neural Information Processing Systems*, pages 14987–14996, 2019.

Massih Amini, Nicolas Usunier, and Cyril Goutte.
Learning from multiple partially observed views - an application to multilingual text categorization.
In *Advances in Neural Information Processing Systems 22*, pages 28–36. Curran Associates, Inc., 2009.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell.
Localizing moments in video with natural language.
In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5803–5812, 2017.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio.
Neural machine translation by jointly learning to align and translate.
arXiv preprint arXiv:1409.0473, 2014.

Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu.
Understanding and improving early stopping for learning with noisy labels.
In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24392–24403. Curran Associates, Inc., 2021.
URL `https://proceedings.neurips.cc/paper/2021/file/cc7e2b878868cbae992d1fb743995d8f-Paper.pdf`.

Maria-Florina Balcan and Avrim Blum.
A discriminative model for semi-supervised learning.
J. ACM, 57:19:1–19:46, 2010.

Maria-Florina Balcan, Avrim Blum, and Ke Yang.
Co-training and expansion: Towards bridging theory and practice.
In *Advances in neural information processing systems*, pages 89–96, 2004.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston.
Curriculum learning.
In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

Wei Bi, Liwei Wang, James T Kwok, and Zhuowen Tu.
Learning to predict from crowdsourced data.
In *UAI*, pages 82–91, 2014.

Hakan Bilen and Andrea Vedaldi.
Weakly supervised deep detection networks.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016.

Avrim Blum and Tom Mitchell.
Combining labeled and unlabeled data with co-training.
In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles.
Activitynet: A large-scale video benchmark for human activity understanding.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

Joao Carreira and Andrew Zisserman.
Quo vadis, action recognition? a new model and the kinetics dataset.
In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017a.

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

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al.
Universal sentence encoder.
arXiv preprint arXiv:1803.11175, 2018.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018.

Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua.
Temporally grounding natural sentence in video.
In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 162–171, 2018.

Shaoxiang Chen and Yu-Gang Jiang.
Semantic proposal for activity localization in videos via sentence query.
In *AAAI*, 2019.

LI Chongxuan, Taufik Xu, Jun Zhu, and Bo Zhang.
Triple generative adversarial nets.
In *Advances in neural information processing systems*, pages 4088–4098, 2017.

Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou.
Large scale transductive svms.
Journal of Machine Learning Research, 7(Aug):1687–1712, 2006.

Jifeng Dai, Yi Li, Kaiming He, and Jian Sun.
R-fcn: Object detection via region-based fully convolutional networks.
In *Advances in neural information processing systems*, pages 379–387, 2016.

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov.
Good semi-supervised learning that requires a bad gan.
In *Advances in neural information processing systems*, pages 6510–6520, 2017.

Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio W Mayol-Cuevas.
You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video.
In *BMVC*, volume 2, page 3, 2014.

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

Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool.
Weakly supervised cascaded convolutional networks.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 914–922, 2017.

Li Ding and Chenliang Xu.
Weakly-supervised action segmentation with iterative soft boundary assignment.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6508–6516, 2018.

Quynh Ngoc Thi Do, Steven Bethard, and Marie-Francine Moens.
Facing the most difficult case of semantic role labeling: A collaboration of word embeddings and co-training.
In *COLING*, 2016.

Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell.
Temporal localization of moments in video collections with natural language.
arXiv preprint arXiv:1907.12763, 2019.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman.

The pascal visual object classes (voc) challenge.
International journal of computer vision, 88(2):303–338, 2010.

M. Fang, T. Zhou, J. Yin, Y. Wang, and D. Tao.
Data subset selection with imperfect multiple labels.
IEEE Transactions on Neural Networks and Learning Systems, 30(7):2212–2221, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine.
Model-agnostic meta-learning for fast adaptation of deep networks.
In Doina Precup and Yee Whye Teh, editors, *Proceedings of 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Riccardo Fogliato, Alexandra Chouldechova, and Max G'Sell.
Fairness evaluation in presence of biased noisy labels.
In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2325–2336. PMLR, 26–28 Aug 2020.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia.
Tall: Temporal activity localization via language query.
arXiv preprint arXiv:1705.02101, 2017.

Mingfei Gao, Larry Davis, Richard Socher, and Caiming Xiong.
WSLLN: Weakly supervised natural language localization networks.
In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1481–1487, Hong Kong, China, November 2019. Association for Computational Linguistics.

Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia.
Mac: Mining activity concepts for language-based temporal localization.
In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 245–253. IEEE, 2019.

Yixiao Ge, Feng Zhu, Dapeng Chen, Rui Zhao, and hongsheng Li.
Self-paced contrastive learning with hybrid memory for domain adaptive object re-id.

In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11309–11321. Curran Associates, Inc., 2020.
URL `https://proceedings.neurips.cc/paper/2020/file/821fa74b50ba3f7cba1e6c53e8fa6845-Paper.pdf`.

Kamran Ghasedi, Xiaoqian Wang, Cheng Deng, and Heng Huang.
Balanced self-paced learning for generative adversarial clustering network.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Ross Girshick.
Fast r-cnn.
In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

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

B. Han, I. W. Tsang, L. Chen, C. P. Yu, and S. Fung.
Progressive stochastic learning for noisy labels.
IEEE Transactions on Neural Networks and Learning Systems, 29(10):5136–5148, 2018.

B. Han, I. W. Tsang, L. Chen, J. T. Zhou, and C. P. Yu.
Beyond majority voting: A coarse-to-fine label filtration for heavily noisy labels.
IEEE Transactions on Neural Networks and Learning Systems, 30(12):3774–3787, 2019.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama.
Co-teaching: Robust training of deep neural networks with extremely noisy labels.
In *Advances in neural information processing systems*, pages 8527–8537, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.

Deep residual learning for image recognition.
2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Identity mappings in deep residual networks.
In *European conference on computer vision*, pages 630–645. Springer, 2016c.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell.
Localizing moments in video with temporal language.
In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1380–1390, 2018.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam.
Mobilenets: Efficient convolutional neural networks for mobile vision applications.
arXiv preprint arXiv:1704.04861, 2017.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger.
Densely connected convolutional networks.
2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, 2017.

Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah.
The thumos challenge on action recognition for videos ‚Äúin the wild‚Äù.
Computer Vision and Image Understanding, 155:1–23, 2017.

Syed Ashar Javed, Shreyas Saxena, and Vineet Gandhi.
Learning unsupervised visual grounding through semantic self-supervision.
arXiv preprint arXiv:1803.06506, 2018.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann.

Self-paced curriculum learning.
In *AAAI*, volume 2, pages 2694–2700, 2015.

Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev.
Contextlocnet: Context-aware deep network models for weakly supervised localization.
In *European Conference on Computer Vision*, pages 350–365. Springer, 2016.

Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Apostol Natsev, Mustafa Suleyman, and Andrew Zisserman.
The kinetics human action video dataset.
ArXiv, abs/1705.06950, 2017.

Diederik P. Kingma and Jimmy Ba.
Adam: A method for stochastic optimization.
In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles.
Dense-captioning events in videos.
In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al.
Learning multiple layers of features from tiny images.
2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton.
Imagenet classification with deep convolutional neural networks.
In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Abhishek Kumar and Hal Daume Iii.
A co-training approach for multi-view spectral clustering.
In *International Conference on International Conference on Machine Learning*, pages 393–400, 2011.

M Pawan Kumar, Benjamin Packer, and Daphne Koller.
Self-paced learning for latent variable models.
In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

Samuli Laine and Timo Aila.
Temporal ensembling for semi-supervised learning.
arXiv preprint arXiv:1610.02242, 2016.

Peng Lei and Sinisa Todorovic.
Temporal deformable residual networks for action segmentation in videos.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6742–6751, 2018.

Dong Li, Jia-Bin Huang, Yali Li, Shengjin Wang, and Ming-Hsuan Yang.
Weakly supervised object localization with progressive domain adaptation.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3512–3520, 2016.

Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li.
Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison.
In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1459–1469, 2020.

Guangxia Li, Kuiyu Chang, and Steven C. H. Hoi.
Multiview semi-supervised learning with consensus.
IEEE Transactions on Knowledge and Data Engineering, 24(11):2040–2051, 2012.

Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu.
Selective-supervised contrastive learning with noisy labels.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 316–325, June 2022.

Junwei Liang, Lu Jiang, Deyu Meng, and Alexander G Hauptmann.
Learning to detect concepts from webly-labeled video data.
In *IJCAI*, pages 1746–1752, 2016.

Kevin J. Liang, Samrudhdhi B. Rangrej, Vladan Petrovic, and Tal Hassner.
Few-shot learning with noisy labels.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9089–9098, June 2022.

Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang.

Bsn: Boundary sensitive network for temporal action proposal generation.
In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen.
Bmn: Boundary-matching network for temporal action proposal generation.
In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

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

Zhijie Lin, Zhou Zhao, Zhu Zhang, Qi Wang, and Huasheng Liu.
Weakly-supervised video moment retrieval via semantic completion network.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11539–11546, 2020.

Daochang Liu, Tingting Jiang, and Yizhou Wang.
Completeness modeling and context separation for weakly supervised temporal action localization.
In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Tongliang Liu and Dacheng Tao.
Classification with noisy labels by importance reweighting.
IEEE Transactions on pattern analysis and machine intelligence, 38(3):447–461, 2015.

Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei.
Gaussian temporal awareness networks for action localization.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 344–353, 2019.

Edward Loper and Steven Bird.
Nltk: The natural language toolkit.
In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang.
Smooth neighbors on teacher graphs for semi-supervised learning.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8896–8905, 2018.

Yueming Lyu and Ivor W. Tsang.
Curriculum loss: Robust learning and generalization against label corruption.
In *International Conference on Learning Representations*, 2020.
URL `https://openreview.net/forum?id=rkgt0REKwS`.

Fan Ma, Deyu Meng, Qi Xie, Zina Li, and Xuanyi Dong.
Self-paced co-training.
In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2275–2284. JMLR. org, 2017.

Fan Ma, Deyu Meng, Xuanyi Dong, and Yi Yang.
Self-paced multi-view co-training.
Journal of Machine Learning Research, 21(57):1–38, 2020a.
URL `http://jmlr.org/papers/v21/18-794.html`.

Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou.
Sf-net: Single-frame supervision for temporal action localization.
In *Computer Vision – ECCV 2020*, pages 420–437, Cham, 2020b. Springer International Publishing.

Minuk Ma, Sunjae Yoon, Junyeong Kim, Youngjoon Lee, Sunghun Kang, and Chang D Yoo.
Vlanet: Video-language alignment network for weakly-supervised video moment retrieval.
In *European Conference on Computer Vision*, pages 156–171. Springer, 2020c.

Naresh Manwani and PS Sastry.
Noise tolerance under risk minimization.
IEEE transactions on cybernetics, 43(3):1146–1151, 2013.

Deyu Meng, Qian Zhao, and Lu Jiang.
A theoretical understanding of self-paced learning.
Information Sciences, 414:319–328, 2017a.

Deyu Meng, Qian Zhao, and Lu Jiang.
A theoretical understanding of self-paced learning.
Information Sciences, 414:319–328, 2017b.

Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K. Roy-Chowdhury.
Weakly supervised video moment retrieval from text queries.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Davide Moltisanti, Sanja Fidler, and Dima Damen.
Action recognition from single timestamp supervision in untrimmed videos.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9915–9924, 2019.

Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao.
3c-net: Category count and center loss for weakly-supervised action localization.
In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han.
Weakly supervised action localization by sparse temporal pooling network.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6752–6761, 2018.

Phuc Xuan Nguyen, Deva Ramanan, and Charless C. Fowlkes.
Weakly-supervised action localization with background modeling.
In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

Kamal Nigam and Rayid Ghani.
Analyzing the effectiveness and applicability of co-training.
In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93. ACM, 2000.

Ke Ning, Linchao Zhu, Ming Cai, Yi Yang, Di Xie, and Fei Wu.
Attentive sequence to sequence translation for localizing clips of interest by natural language descriptions.
arXiv preprint arXiv:1808.08803, 2018.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu.

Making deep neural networks robust to label noise: A loss correction approach.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury.
W-talc: Weakly-supervised temporal activity localization and classification.
In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay.
Scikit-learn: Machine learning in Python.
Journal of Machine Learning Research, 12:2825–2830, 2011.

Sebastian Raschka.
Mlxtend: Providing machine learning and data science utilities and extensions to python,Äôs scientific computing stack.
The Journal of Open Source Software, 3(24), April 2018.
doi: 10.21105/joss.00638.
URL `http://joss.theoj.org/papers/10.21105/joss.00638`.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko.
Semi-supervised learning with ladder networks.
In *Advances in neural information processing systems*, pages 3546–3554, 2015.

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy.
Learning from crowds.
Journal of Machine Learning Research, 11(Apr):1297–1322, 2010.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun.
Learning to reweight examples for robust deep learning.
arXiv preprint arXiv:1803.09050, 2018.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen.
Improved techniques for training gans.
In *Advances in neural information processing systems*, pages 2234–2242, 2016.

Zheng Shou, Dongang Wang, and Shih-Fu Chang.
Temporal action localization in untrimmed videos via multi-stage cnns.
In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang.
Autoloc: Weakly-supervised temporal action localization in untrimmed videos.
In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–171, 2018.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng.
Meta-weight-net: Learning an explicit mapping for sample weighting.
In *Advances in Neural Information Processing Systems*, pages 1917–1928, 2019.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta.
Hollywood in homes: Crowdsourcing data collection for activity understanding.
In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 510–526. Springer, 2016.

Karen Simonyan and Andrew Zisserman.
Two-stream convolutional networks for action recognition in videos.
In *Advances in neural information processing systems*, pages 568–576, 2014a.

Karen Simonyan and Andrew Zisserman.
Very deep convolutional networks for large-scale image recognition.
arXiv preprint arXiv:1409.1556, 2014b.

Vikas Sindhwani and David S Rosenberg.
An rkhs for multi-view learning and manifold co-regularization.
In *Proceedings of the 25th international conference on Machine learning*, pages 976–983. ACM, 2008.

Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin.
Beyond the point cloud: from transductive to semi-supervised learning.
In *International Conference on Machine Learning*, pages 824–831, 2005a.

Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin.
A co-regularization approach to semi-supervised learning with multiple views.
In *Proceedings of ICML workshop on learning with multiple views*, pages 74–79. Citeseer, 2005b.

Krishna Kumar Singh and Yong Jae Lee.
Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization.
In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3544–3553. IEEE, 2017.

Antti Tarvainen and Harri Valpola.
Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results.
In *Advances in neural information processing systems*, pages 1195–1204, 2017.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri.
Learning spatiotemporal features with 3d convolutional networks.
In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders.
Selective search for object recognition.
International journal of computer vision, 104(2):154–171, 2013.

L. Valiant.
A theory of the learnable.
Communications of the ACM, 27(11):1134–1142, 1984.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz.

Interpolation consistency training for semi-supervised learning.
In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pages 3635–3641. AAAI Press, 2019.
ISBN 978-0-9992411-4-1.
URL `http://dl.acm.org/citation.cfm?id=3367471.3367546`.

Xiaojun Wan.
Co-training for cross-lingual sentiment classification.
In *ACL/IJCNLP*, 2009.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman.
Glue: A multi-task benchmark and analysis platform for natural language understanding.
arXiv preprint arXiv:1804.07461, 2018.

Chong Wang, Weiqiang Ren, Kaiqi Huang, and Tieniu Tan.
Weakly supervised object localization with latent category learning.
In *European Conference on Computer Vision*, pages 431–445. Springer, 2014.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Val Gool.
Temporal segment networks: Towards good practices for deep action recognition.
In *ECCV*, 2016.

Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool.
Untrimmednets for weakly supervised action recognition and detection.
In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4325–4334, 2017.

R. Wang, T. Liu, and D. Tao.
Multiclass learning with partially corrupted labels.
IEEE Transactions on Neural Networks and Learning Systems, 29(6):2568–2580, 2018.

Wei Wang and Zhi-Hua Zhou.
Analyzing co-training style algorithms.
In *European Conference on Machine Learning*, pages 454–465. Springer, 2007.

Wei Wang and Zhi-Hua Zhou.

A new analysis of co-training.
In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 1135–1142, 2010.

Wei Wang and Zhi-Hua Zhou.
Co-training with insufficient views.
In *ACML*, pages 467–482, 2013.

Wei Wang and Zhi-Hua Zhou.
Theoretical foundation of co-training and disagreement-based algorithms.
CoRR, abs/1708.04403, 2017.
URL `http://arxiv.org/abs/1708.04403`.

Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni.
Generalizing from a few examples: A survey on few-shot learning.
ACM Comput. Surv., 53(3), jun 2020.
ISSN 0360-0300.
doi: 10.1145/3386252.
URL `https://doi.org/10.1145/3386252`.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey.
Symmetric cross entropy for robust learning with noisy labels.
In *Proceedings of IEEE International Conference on Computer Vision*, pages 322–330, 2019.

Yuechen Wang, Jiajun Deng, Wengang Zhou, and Houqiang Li.
Weakly supervised temporal adjacent network for language grounding.
IEEE Transactions on Multimedia, 2021a.

Yuechen Wang, Wengang Zhou, and Houqiang Li.
Fine-grained semantic alignment network for weakly supervised temporal language grounding.
In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 89–99, 2021b.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang.
Learning from massive noisy labeled data for image classification.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

Huijuan Xu, Abir Das, and Kate Saenko.
R-c3d: Region convolutional 3d network for temporal activity detection.
In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017.

Huijuan Xu, Kun He, L Sigal, S Sclaroff, and K Saenko.
Multilevel language and vision integration for text-to-clip retrieval.
In *AAAI*, volume 2, page 7, 2019.

Xinxing Xu, Wen Li, Dong Xu, and Ivor W Tsang.
Co-labeling for multi-view weakly labeled learning.
IEEE transactions on pattern analysis and machine intelligence, 38(6):1113–1125, 2016.

Yoshihiro Yamada, Masakazu Iwamura, Takuya Akiba, and Koichi Kise.
Shakedrop regularization for deep residual learning.
arXiv preprint arXiv:1802.02375, 2018.

Leon Yao and John Miller.
Tiny imagenet classification with convolutional neural networks.
CS 231N, 2(5):8, 2015.

Han-Jia Ye, De-Chuan Zhan, Yuan Miao, Yuan Jiang, and Zhi-Hua Zhou.
Rank consistency based multi-view learning: A privacy-preserving approach.
In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 991–1000. ACM, 2015.

Guang Yu, Siqi Wang, Zhiping Cai, Xinwang Liu, Chuanfu Xu, and Chengkun Wu.
Deep anomaly discovery from unlabeled videos via normality advantage and self-paced refinement.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13987–13998, June 2022.

Shipeng Yu, Balaji Krishnapuram, Romer Rosales, and R. Bharat Rao.
Bayesian co-training.
Journal of Machine Learning Research, 12(3):2649–2680, 2011.

Xin Yu, Yurun Tian, Fatih Porikli, Richard Hartley, Hongdong Li, Huub Heijnen, and Vassileios Balntas.

Unsupervised extraction of local image descriptors via relative distance ranking loss.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

Zehuan Yuan, Jonathan C Stroud, Tong Lu, and Jia Deng.
Temporal action localization by structured maximal sums.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3684–3692, 2017.

Christopher Zach, Thomas Pock, and Horst Bischof.
A duality based approach for realtime tv-l 1 optical flow.
In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007.

Sergey Zagoruyko and Nikos Komodakis.
Wide residual networks.
arXiv preprint arXiv:1605.07146, 2016.

Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan.
Graph convolutional networks for temporal action localization.
In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

Bowen Zhang, Hexiang Hu, and Fei Sha.
Cross-modal and hierarchical modeling of video and text.
In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 374–390, 2018.

Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S. Davis.
Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Dingwen Zhang, Deyu Meng, Long Zhao, and Junwei Han.
Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning.
arXiv preprint arXiv:1703.01290, 2017a.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz.

mixup: Beyond empirical risk minimization.
arXiv preprint arXiv:1710.09412, 2017b.

Min-Ling Zhang and Zhi-Hua Zhou.
Cotrade: confident co-training with data editing.
IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 41(6): 1612–1626, 2011.

Zhilu Zhang and Mert Sabuncu.
Generalized cross entropy loss for training deep neural networks with noisy labels.
In *Advances in neural information processing systems*, pages 8778–8788, 2018.

Hang Zhao, Zhicheng Yan, Heng Wang, Lorenzo Torresani, and Antonio Torralba.
Slac: A sparsely labeled dataset for action classification and localization.
arXiv preprint arXiv:1712.09374, 2017a.

Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan.
Hacs: Human action clips and segments dataset for recognition and temporal localization.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8668–8678, 2019.

Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin.
Temporal action detection with structured segment networks.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017b.

L. Zheng, Y. Yang, and A. G. Hauptmann.
Person Re-identification: Past, Present and Future.
ArXiv e-prints, October 2016.

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian.
Scalable person re-identification: A benchmark.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1116–1124, 2015.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang.
Random erasing data augmentation.
In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Xiong Zhou, Xianming Liu, Chenyang Wang, Deming Zhai, Junjun Jiang, and Xiangyang Ji.
Learning with noisy labels via sparse regularization.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 72–81, October 2021.

Zhi-Hua Zhou.
Abductive learning: towards bridging machine learning and logical reasoning.
Science China Information Sciences, 62(7):76101, 2019.

Xiaojin Zhu, Bryan R. Gibson, and Timothy T. Rogers.
Co-training as a human collaboration policy.
In *AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, Usa, August*, 2012.

Bohan Zhuang, Lingqiao Liu, Yao Li, Chunhua Shen, and Ian Reid.
Attend in groups: a weakly-supervised deep learning framework for learning from web data.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1878–1887, 2017.