UNIVERSITY OF HONG KONG

DOCTORAL THESIS

Robust Visual Learning under Imperfection: Navigating Limited Supervision and Label Uncertainty

Author:
Jichang LI

Supervisor:
Prof. Yizhou YU

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Department of Computer Science
Faculty of Engineering

September, 2024

Abstract of thesis entitled

Robust Visual Learning under Imperfection: Navigating Limited Supervision and Label Uncertainty

Submitted by

Jichang LI

for the degree of Doctor of Philosophy

at The University of Hong Kong

in September, 2024

This doctoral dissertation explores effective strategies for robust visual learning in the presence of imperfect data sources, with a particular focus on three key areas: Semi-supervised Domain Adaptation (SSDA), Learning with Noisy Labels (LNL), and Federated Learning with Noisy Labels (F-LNL). Each area presents distinct challenges stemming from limited supervision and label uncertainty, necessitating methods to alleviate these challenges and enhance model adaptability and accuracy.

In the realm of SSDA, this research introduces two state-of-the-art methods: Cross-domain Adaptive Clustering (CDAC) and Graph-based Adaptive Betweenness Clustering (G-ABC). These methodologies harness the presence of a few target labels to facilitate precise feature adaptation between source and target domains. Specifically, CDAC employs a proposed version of the adversarial adaptive clustering loss, as well as an adapted version of pseudo labeling, to enhance feature clustering across domains, thereby improving both inter-domain adaptation and intra-domain generalization. Similarly, G-ABC, constructed upon a refined graph structure, proposes adaptive betweenness clustering to align semantic features across domains by establishing connections based on semantic consistency and feature similarity. Both methods enhance feature alignment between domains, thereby bolstering model generalization to the target domain. Empirical evaluations conducted on diverse datasets such as DomainNet, Office-Home, and Office-31 demonstrate the superior performance of these methods over existing state-of-the-art SSDA algorithms.

To tackle the pervasive issue of label noise in LNL tasks, this study proposes a novel algorithm called Neighborhood Collective Estimation (NCE), comprising two steps: 1) Neighborhood Collective Noise Verification, which categorizes all training samples into either a clean or noisy subset, and 2) Neighborhood Collective Label Correction, which corrects the labels of noisy samples. To this end, NCE enhances predictive reliability by contrasting candidate samples against their feature-space nearest neighbors, enriching predictive information and mitigating biases in noisy label identification and correction. The efficacy of NCE is demonstrated through superior performance on benchmarks including CIFAR-10, CIFAR-100, and Clothing1M.

In the domain of F-LNL, this thesis introduces the FedDiv framework to address challenges originating from data heterogeneity and noise heterogeneity. Leveraging complementary knowledge learned from all clients, this FedDiv succeeds in decreasing the adverse effects of label noise across local clients while preserving data privacy. To be specific, FedDiv proposes global noise filtering and predictive consistency-based sampling to enhance the robustness and stability of learning in the decentralized scenarios. The effectiveness of FedDiv has been demonstrated by empirical evaluations on benchmark datasets such as CIFAR-10, CIFAR-100, and Clothing1M under various label noise settings for both IID and non-IID data partitions.

In conclusion, this dissertation contributes robust methodologies that advance the frontier of machine learning by enabling models to learn effectively from imperfect data. These methodologies, inspired by the adaptability of the human visual system, demonstrate significant progress in handling real-world data complexities. Extensive experimental validations set new benchmarks for robust learning in the presence of data imperfections.

[465 Words]

Robust Visual Learning under Imperfection: Navigating Limited Supervision and Label Uncertainty

by

Jichang LI

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy

at

University of Hong Kong
September, 2024

Declaration

I, Jichang LI, declare that this thesis titled, " Robust Visual Learning under Imperfection: Navigating Limited Supervision and Label Uncertainty ", which is submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy, represents my own work except where due acknowledgement have been made. I further declared that it has not been previously included in a thesis, dissertation, or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Signed: 

Date: September, 2024

For Mama and Papa

Acknowledgements

I would like to thank...

Life's duration is uncertain, seemingly long yet possibly fleeting. Throughout my four-year doctoral journey, amid the steady increase in my weight, I often lamented the fading youth. These swift years were marred by procrastination and minimal learning, with my knowledge acquisition seemingly at a standstill. However, the fortune of encountering my doctoral advisors, Prof. Yizhou YU and Prof. Guanbin LI, along with every other individual I met during this journey, especially those who provided invaluable assistance, marked a significant treasure in my life. Here, I would like to express my sincere gratitude to all of them! THANK YOU VERY MUCH!

Jichang LI
Department of Computer Science
The University of Hong Kong
September, 2024

List of Publications

JOURNALS:

[1] **Jichang Li**, Guanbin Li, and Yizhou Yu. Adaptive Betweenness Clustering for Semi-Supervised Domain Adaptation. *IEEE Transactions on Image Processing*, 2023 (IEEE TIP-2023).

[2] **Jichang Li**, Guanbin Li, and Yizhou Yu. Inter-Domain Mixup for Semi-Supervised Domain Adaptation. *Pattern Recognition*, Elsevier, 2023 (Elsevier PR-2023).

CONFERENCES:

[1] **Jichang Li**, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-Domain Adaptive Clustering for Semi-Supervised Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021 (CVPR-2021).

[2] **Jichang Li**, Guanbin Li, Feng Liu, and Yizhou Yu. Neighborhood Collective Estimation for Noisy Label Identification and Correction. In *European Conference on Computer Vision* 2022 (ECCV-2022).

[3] **Jichang Li**, Guanbin Li, Hui Cheng, Zicheng Liao and Yizhou Yu. FedDiv: Collaborative Noise Filtering for Federated Learning with Noisy Labels. In *The Thirty-Eighth AAAI Conference on Artificial Intelligence*, 2024 (AAAI-2024).

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ABC	**A**daptive **B**etweenness **C**lustering
ADBC	**A**cross-**D**omain **B**etweenness **C**lustering
a.k.a	**A**lso **K**nown **A**s
C-LNL	**C**entralized **L**earning with **N**oisy **L**abels
CUNR	**C**onfidence **U**ncertainty based **N**ode **R**emoval
CSS	**C**lass-wise **S**imilarity **S**core
CDAC	**C**ross-**D**omain **A**daptive **C**lustering
DA	**D**omain **A**daptation
DNN	**D**eep **N**eural **N**etwork
Eq.	**Eq**uation
F-LNL	**F**ederated **L**earning with **N**oisy **L**abels
FNF	**F**ederated **N**oise **F**ilter
G-ABC	**G**raph-based **A**daptive **B**etweenness **C**lustering
GMM	**G**aussian **M**ixture **M**odels
KL	**K**ullback–**L**eibler Divergence
KNN	**K**-**N**earest-**N**eighbors
LNL	**L**earning with **N**oisy **L**abels
NCE	**N**eighborhood **C**ollective **E**stimation
NCLC	**N**eighborhood **C**ollective **L**abel **C**orrection
NCNV	**N**eighborhood **C**ollective **N**oise **V**erification
PCS	**P**redictive **C**onsistency based **S**ampler
PDEP	**P**rediction **D**issimilarity based **E**dge **P**runing
SGD	**S**tochastic **G**radient **D**escent
SSL	**S**emi-**S**upervised **L**earning
SOTA	**S**tate-**o**f-**t**he-**a**rt
SSDA	**S**emi-**S**upervised **D**omain **A**daptation
UDA	**U**nsupervised **D**omain **A**daptation
WDBC	**W**ithin-**D**omain **B**etweenness **C**lustering

Chapter 1

Introduction

In the evolving field of machine learning, the pursuit of a system of robust visual learning capable of functioning effectively under imperfect conditions has become a critical area of study. Traditional supervised learning paradigms often assume the availability of large quantities of accurately labeled data. However, this assumption rarely holds in real-world scenarios, where as illustrated in Table 1.1, data is often with the challenges of limited supervision, label uncertainty and decentralization. This thesis addresses these challenges through innovative approaches in Semi-supervised Domain Adaptation (**SSDA**), Learning with Noisy Labels (**LNL**), and Federated Learning with Noisy Labels (**F-LNL**). As shown in Table 1.2, each of these tasks addresses one or two of the challenges we have discussed in Table 1.1. Collectively, these areas explore how to efficiently utilize imperfect data, ensuring both scalability and accuracy in real-world applications.

Challenge	Content	Reasons
Limited Supervision	Data is frequently unlabeled	It is simply too expensive to obtain labels for complex tasks and much of the data are sourced from the internet.
Label Uncertainty	Data is Imperfectly labeled	Detailed annotations can often be ambiguous.
	There is domain gap between datasets	There are distribution discrepancies across data from multiple sources.
Data decentralization	Data is decentralized	Data is often distributed across multiple decentralized sources.

Table 1.1: Challenges of data imperfection in visual learning.

Task	Limited Supervision	Label Uncertainty	Data Decentralization
SSDA	Semi-supervised learning in target domain	Domain bias between source and target domains	-
LNL	-	Label noise in training data	-
F-LNL	-	Label noise in local data of each client	Decentralized data in local clients

Table 1.2: Core challenges addressed in each task.

SSDA plays a pivotal role in leveraging a limited number of target labels to reduce domain shift between the source and target domains, thus facilitating precise adaptation of the model to the target domains [129, 71, 176]. This thesis presents two distinct

works under SSDA, each proposing methodologies to minimize domain discrepancies and effectively utilize mixed labeled and unlabeled data in training. These approaches are designed to enhance the adaptability and performance of learning models when confronted with data from similar but distinct distributions.

LNL tackles the inevitable issue of label noise presented in training datasets, which can significantly degrade the performance of learning algorithms if unaddressed [76, 161, 96, 138]. This research introduces robust training strategies that are resistant to the misguiding influence of incorrect labels. By developing algorithms that can identify and mitigate the impact of such errors, this work ensures that learning models remain reliable even when trained with compromised data quality.

F-LNL extends the challenges of label noise to the federated setting, where data privacy and distribution add layers of complexity [167, 80]. This research innovates on integrating robust noise-handling mechanisms with federated learning protocols, allowing for the collaborative training of models across multiple decentralized data sources without compromising data privacy. These mechanisms are crucial for applications in environments where data cannot be centralized due to privacy concerns, such as in hospitals [27], etc.

The unifying theme of this thesis is the robustness of visual learning in the face of limited supervision and label uncertainty. By addressing the scarcity of sample labels and the unreliability of available labels, this study pushes the performance boundaries of visual learning tasks under the conditions of data imperfection. Each of the four works included in this thesis contributes to a deeper understanding of real-world data and its application to practical system tasks corresponding to machine learning.

Moreover, this research not only advances theoretical methodologies but also demonstrates significant empirical successes across various domains, underscoring the practical usefulness and applicability of the proposed methods. The integration of these methodologies into a cohesive framework showcases the potential for significant improvements in machine learning applications, where data imperfections are the norm rather than the exception.

In conclusion, this thesis encapsulates a journey through the challenges and innovations in learning from imperfect data. The outcomes of this research contribute to the broader vision of creating adaptable, efficient, and robust machine learning systems capable of overcoming the barriers posed by real-world data conditions, thus paving the way for the next generation of machine learning applications.

1.1 Problems and Solutions

1.1.1 Semi-supervised Domain Adapation

Problem definition. Semi-supervised domain adaptation (SSDA) extends unsupervised domain adaptation (UDA) by integrating a small subset of labeled samples from

the target domain, potentially leading to significant improvements in model performance. However, SSDA poses distinct challenges, primarily centered on achieving effective inter-domain adaptation while preserving intra-domain generalization [112]. Current SSDA methods have predominantly concentrated on either sample-wise or distribution-wise feature alignment across domains [38, 62, 13], often overlooking the critical aspect of class-wise sub-distributions within the target domain. This oversight may result in feature mismatching and reduced generalization performance on novel test data from target domains.

Moreover, the scarcity of target labels in SSDA, relative to the abundance in the source domain, tends to bias the feature representation towards the source domain, consequently compromising the discrimination capacity of the model within the target domain, as illustrated in [129]. These challenges underscore a significant gap in achieving robust feature alignment and representation in SSDA, underscoring the need for novel approaches capable of effectively leveraging the sparse labeled target data while comprehensively addressing the discrepancies between domains.

Solution 1: Cross-domain Adaptive Clustering (`CDAC`). This proposed approach introduces a novel adaptation scheme for handling the SSDA tasks, focusing on cluster-wise feature alignment to address the challenges posed by sparse labeled data in the target domain. `CDAC` creatively utilizes the proposed adversarial adaptive clustering loss to enhance both intra-domain and inter-domain feature alignment. By grouping features of unlabeled target data into clusters, `CDAC` aligns these clusters with corresponding clusters in the source domain, thus achieving a cluster-wise feature adaptation across domains. To be more specific, this scheme employs minimax training on the parameters of a feature extractor and a classifier to ensure that the intra-domain adaptation aligns the unlabeled target features into well-defined clusters guided by the limited number of labeled target samples. Simultaneously, inter-domain adaptation is facilitated through a classifier trained to maximize the proposed adversarial adaptive clustering loss, aligning the cluster-wise feature distributions between the domains.

Moreover, `CDAC` incorporates an adapted version of pseudo labeling to amplify the number of labeled samples in each class within the target domain, thereby stabilizing and enhancing the cluster cores. This method not only addresses the feature mismatching by producing more robust and powerful cluster cores for each class but also ensures that these clusters are discriminatively aligned across domains, significantly leading to improved performance of the SSDA model on benchmark datasets such as DomainNet [118], Office-Home [149] and Office-31 [128].

Solution 2: Graph-based Adaptive Betweenness Clustering (`G-ABC`). Built upon a refined graph structure, this framework presents a unique approach to SSDA by harnessing the power of semantic transfer. To be specific, `G-ABC` first constructs a graph to capture the pairwise associations between labeled samples from both domains and unlabeled target samples, using pairwise label similarity as the basis for connectivity. This graph facilitates the propagation of semantic label information across domains,

addressing the challenge of partial feature alignment inherent in SSDA. To refine the connectivity and enhance the model's robustness, G-ABC implements two novel strategies: Confidence Uncertainty based Node Removal (CUNR) and Prediction Dissimilarity based Edge Pruning (PDEP). These strategies selectively eliminate unreliable connections, thereby focusing the learning process on more probable and meaningful associations.

Within this refined graph structure, the proposed algorithm of Adaptive Betweenness Clustering operates to cluster unlabeled target samples towards labeled samples from either the source or the target domain, effectively achieving semantic transfer. This clustering is differentiated into across-domain betweenness clustering and within-domain betweenness clustering, each tailored to optimize semantic alignment and enhance the discriminative power of the model in the target domain. By facilitating a balanced transfer of label information and promoting a domain-invariant feature space, G-ABC significantly improves the generalization capability of SSDA models. This is evidenced by its superior performance on established benchmarks such as DomainNet [118], Office-Home [149] and Office-31 [128], outperforming existing state-of-the-art SSDA methods.

Connections and Differences between CDAC and G-ABC. Both CDAC and G-ABC propose clustering to achieve the goal of cross-domain feature alignment under semi-supervised domain adaptation. However, CDAC conducts clustering on unlabeled target samples and uses adversarial training to drive cross-domain alignment. In contrast, G-ABC introduces adaptive betweenness clustering, which connects unlabeled target samples with labeled source and target data to facilitate semantic transfer and thus achieve domain adaptation. By comparing Table 2.1, Table 2.1 and Table 2.3 with Table 3.1, Table 3.2 and Table 3.3, G-ABC demonstrates superior performance than CDAC, likely due to its building on a refined graph that utilizes CUNR and PDEP to remove unreliable nodes and edges, leading to a more significant improvement in overall performance.

1.1.2 Learning with Noisy Labels

Problem definition. Learning with noisy labels (LNL) is a critical challenge in the training of deep neural networks, particularly when relying on large datasets that may contain inaccurately labeled data. This issue stems from the dependence of these models on high-quality annotations for achieving optimal performance in tasks like image classification. The presence of noisy labels can lead to the deterioration of model accuracy as the network might learn the noise instead of the underlying true patterns. As showcased by [41, 76, 111, 183], this problem is exacerbated in real-world training datasets where some classes may have disproportionately high levels of label noise, complicating the training process further.

Moreover, traditional label noise handling techniques often suffer from confirmation bias [142, 3], where the model reinforces its own erroneous label predictions

throughout training iterations, leading to a compounding of errors and a decrease in the reliability of the output. The complexities inherent in LNL require innovative approaches capable of robustly identifying and correcting noisy labels while avoiding biases associated with initial incorrect label assumptions.

Solution: Neighborhood Collective Estimation (NCE). The proposed NCE approach innovatively addresses the challenges of learning with noisy labels by enhancing the predictive reliability by contrasting candidate samples against their feature-space nearest neighbors, enriching predictive information and mitigating biases in noisy label identification and correction. NCE comprises two key steps: Neighborhood Collective Noise Verification (NCNV) and Neighborhood Collective Label Correction (NCLC).

In NCNV, the identification of noisy labels is enhanced through the analysis of label consistency within the feature-space neighborhood of the the candidate sample. By comparing the given label of the candidate with the predicted label distributions of its nearest neighbors, NCNV effectively distinguishes between noisy and clean labels. This method reduces reliance on potentially biased individual predictions by leveraging a broader context provided by contrastive neighbors, which improves the detection accuracy of label noise.

Following noise identification, NCLC focuses on correcting noisy labels using a weighted aggregate of labels from neighboring clean samples. This relabeling strategy ensures that corrections are not solely based on the initial, possibly erroneous labels but are informed by a consensus from similar samples. The integration of collective intelligence from the neighborhood mitigates the risk of reinforcing incorrect labels, thus addressing the common issue of confirmation bias in traditional noise handling methods.

Additionally, to reinforce the robustness of the learning process, NCE incorporates established semi-supervised learning techniques like mixup regularization [186] and consistency regularization [64]. These techniques use a combination of clean and corrected labels to train the model, further enhancing its ability to generalize from imperfect data.

Overall, NCE not only provides a systematic framework for handling noisy labels but also integrates effectively with existing machine learning pipelines, significantly improving the accuracy and reliability of models trained on noisy datasets. Extensive validation on benchmarks such as CIFAR-10 [60], CIFAR-100 [60], Clothing-1M [162] and Webvision-1.0 [86] confirms that NCE outperforms existing state-of-the-art LNL approaches, making it a valuable tool for researchers and practitioners dealing with noisy labels in real-world datasets.

1.1.3 Federated Learning with Noisy Labels

Problem definition. Federated Learning with Noisy Labels (F-LNL) introduces a complex scenario where a global model is trained across multiple decentralized clients, each

possessing data of varying quality and label accuracy, as illustrated in [167]. Unlike traditional paradigms of centralized learning with noisy labels [76, 138], F-LNL contends with not only data heterogeneity, where different clients have data distributions that are not identically distributed, but also noise heterogeneity, where the distribution and severity of label noise vary significantly across clients. These issues are compounded by the inherent challenges of federated settings, such as preserving privacy and minimizing data leakage. The presence of noisy labels can significantly impede the learning process, resulting in sub-optimal model performance or bias towards erroneous data. Moreover, the inability to directly inspect or modify data at the client level further complicates effective noisy label management, hindering the implementation of standard noise mitigation techniques. This context necessitates novel approaches capable of robustly handling diverse types and levels of noise while ensuring that collaborative learning yields a coherent and accurate global model.

Solution: `FedDiv`**.** The proposed framework, `FedDiv`, aims to tackle the challenges of the F-LNL task by introducing a robust algorithm for handling label noise among federated clients while preserving privacy. More specifically, `FedDiv` initially designed a Federated Noise Filter (FNF) to be applied to each client, thereby categorizing local privacy data as either clean or noisy. This filter operates by modeling the global noise distribution across all clients using a Gaussian Mixture Model (GMM) [21]. To accomplish this, each client first learns the parameters of their own local GMM based on the loss values observed from their training data, reflecting the likelihood of samples being mislabeled. Subsequently, these local GMM parameters obtained from all clients are aggregated at the server to construct a global GMM. Thus, this global GMM model serves as a federated noise filter, enabling label noise filtering at each client. This process aids in optimizing the noise learning model by distilling complementary knowledge learned from all clients without compromising their data privacy.

After identifying noisy labels, `FedDiv` removes or relabels these noisy samples on each client. For relabeling, the sample labels of the candidates with high confidence predicted by the global neural network model are used as their pseudo-labels, ensuring that only reliable predicted labels are reintroduced into the training process. This method not only cleans the local data from privacy clients but also enhances the ability of the global network model to generalize by learning from local datasets with cleaner sample labels.

To further enhance the robustness of the local training process, `FedDiv` incorporates a Predictive Consistency based Sampler (PCS). This component ensures that the training data used locally is consistent with predictions made by the global model. PCS uses counterfactual reasoning [47, 117, 153] to refine the predictions for local samples, aiming to make them more reliable and less prone to noise memorization. This strategy helps in maintaining the integrity and stability of local model updates before they are aggregated to update the global model. `FedDiv` has been rigorously tested on multiple benchmark datasets such as CIFAR-10 [60], CIFAR-100 [60], and Clothing1M [163].

These tests have demonstrated its effectiveness in significantly outperforming existing F-LNL methods under various label noise settings and data distributions.

This proposed framework not only addresses noise and data heterogeneity effectively but also ensures that the federated learning process remains stable and efficient. In summary, `FedDiv` presents a comprehensive approach to handling noisy labels in federated learning environments. By effectively integrating noise filtering, predictive consistency sampling, and robust model aggregation, `FedDiv` ensures that the federated learning process is resilient to label noise, thereby improving the overall performance and reliability of the global model.

1.2 Main Contributions

This doctoral thesis highlight significant advancements in handling imperfect data across various domains of robust visual learning, including Semi-supervised Domain Adaptation (SSDA), Learning with Noisy Labels (LNL), and Federated Learning with Noisy Labels (F-LNL).

To sum up, the main contributions of this thesis are included as follows.

1) A robust approach named `CDAC` (Cross-domain Adaptive Clustering) for semi-supervised domain adaptation. For simultaneously reaching inter-domain and intra-domain adaptation, `CDAC` pioneers in devising an adversarial adaptive clustering loss to achieve this goal, where by conducting minimax training over a feature extractor and a classifier, thereby narrowing the gap between feature distributions across domains. Additionally, pseudo labeling is here adopted to produce robust cluster cores for each class during clustering, bridging the divide between limited supervision and label uncertainty.

2) An extended algorithm termed `G-ABC` (Graph-based Adaptive Betweenness Clustering) for semi-supervised domain adaptation. This algorithm addresses the challenges imposed by SSDA with achieving categorical domain alignment through cross-domain semantic alignment. Leveraging a refined graph structure that captures associations between unlabeled target samples and labeled data from both domains, Adaptive Betweenness Clustering is presented to facilitate semantic transfer across domains, promoting alignment between unlabeled target samples and both the source and target domains. The ability of `G-ABC` has been showcased to further enhance classification performance and generalization capability despite limited supervision and label uncertainty.

3) An effective method referred to as `NCE` (Neighborhood Collective Estimation) for learning with noisy labels. To effectively addresses the challenge in LNL tasks, two pivotal steps, namely Neighborhood Collective Noise Verification and Neighborhood Collective Label Correction, are introduced to conduct label noise identification and noisy label relabeling. To enhance the predictive reliability of

candidate samples, both steps use feature-space nearest neighbors, mitigating confirmation bias prevalent in traditional LNL approaches. This methodology not only reduces the impact of noisy labels through robust noise verification and label correction but also enhances model generalization under label uncertainty.

4) A novel framework designated `FedDiv` for federated learning with noisy labels. `FedDiv` effectively mitigates the negative impact of label noise across federated client models without compromising data privacy. To enhance label noise filtering and training stability on each client, `FedDiv` proposes leveraging a global noise filter and a Predictive Consistency-based Sampler. This dual framework prevents noise memorization and significantly boosts the classification performance of federated neural network models, demonstrating superiority in handling heterogeneous noise in decentralized data settings.

1.3 Thesis Structure

The remaining part of this thesis consist of five chapters, with the first four chapters detailing identified problems and proposed solutions, followed by an overall conclusion and future works in the final chapter. The remaining sections are organized as follows.

1) Chapters 2 and 3 initiate the exploration of the tasks of Semi-supervised Domain Adaptation (SSDA). In Chapter 2, the importance of achieving both inter-domain adaptation and intra-domain generalization concurrently in addressing SSDA tasks is first highlighted. To this end, the robust approach of *Cross-domain Adaptive Clustering* is then presented, which creatively utilizes adversarial training to conduct minimax training on the parameters of a feature extractor and a classifier, effectively achieving the desired goal. Following this, Chapter 3 introduces another SSDA algorithm for addressing the same tasks. It revisits the limitations of current SSDA models in strategies for domain alignment and proposes categorical feature alignment to enhance the model adaptation to the target domain. This is formulated as a novel framework of *Graph-based Adaptive Betweenness Clustering*. Both of the two SSDA approaches proposed in these chapters effectively enhance the adaptability and performance of the learning models when dealing with data from similar but distinct distributions.

2) Chapter 4 delves into the problem of label noise disrupting model stability in existing algorithms for visual learning and introduces the task of Learning with Noisy Labels (LNL) to mitigate this issue. In this chapter, the inadequacy of existing solutions for LNL in avoiding conformation biases is first demonstrated, which significantly degrade the performance of learning algorithms. To address this, a new framework for handling the LNL task - *Neighborhood Collective Estimation for Learning with Noisy Labels* - is then proposed to tackle the issue of

conformation biases. This chapter not only offers a systematic approach for handling noisy labels but also integrates effectively with existing pipelines for machine learning, thereby enhancing the accuracy and reliability of models trained on noisy training datasets.

3) In Chapter 5, the task of noisy label learning is extended into the federated scenario, namely Federated Learning with Noisy Labels (F-LNL). This chapter first analyzes the issues of training instability and noise memorization caused by label noise across clients in existing F-LNL methods, resulting in decreased performance of robust learning models in federated scenarios. Following this, a new framework for addressing F-LNL - *Collaborative Noise Filtering for Federated Learning with Noisy Labels* - is subsequently introduced to achieve federated noise filtering and predictive consistency-based sampling for stable federated model learning. This ensures resilience of the federated learning process to label noise.

4) Chapter 6 provides overall conclusions of the four proposed approaches for robust vision learning and discusses the limitations of this study as well as future research directions.

Chapter 2

Cross-Domain Adaptive Clustering for Semi-Supervised Domain Adaptation

This doctoral thesis begins the journey by looking into the problem of semi-supervised domain adaptation, a task of robust vision learning with limited supervision.

2.1 Introduction

Semi-supervised domain adaptation (SSDA) is a variant of the unsupervised domain adaptation (UDA) problem. With a small number of labeled samples in the target domain, SSDA has the potential to significantly boost performance in comparison to UDA. In general, domain adaptation needs to reduce inter-domain gap (i.e. differences in feature distributions between two domains) and decrease intra-domain gap (i.e. differences among class-wise sub-distributions in the target domain) in order to achieve inter-domain adaptation and intra-domain adaptation simultaneously [112].

Many existing domain adaptation approaches start with inter-domain adaptation, and guide their models to learn cross-domain sample-wise feature alignment [132, 22, 13], or distribution-wise feature alignment [38, 92, 62]. In the semi-supervised learning setting, adversarial learning is employed in [129, 104] to improve sample-wise feature alignment for inter-domain adaptation. However, such previous work ignores extra information indicated by class-wise sub-distributions in the target domain, and thus results in cross-domain feature mismatch during model training, thereby reducing model generalization performance on novel test data in the target domain.

Whereafter, much work on domain adaptation has turned to intra-domain adaptation [57, 39]. By optimizing class-wise sub-distributions within the target domain, the generalization performance of adaptation models can be improved. In the context of semi-supervised domain adaptation, the presence of few labeled target samples is utilized to help features of unlabeled target samples from different classes be guided

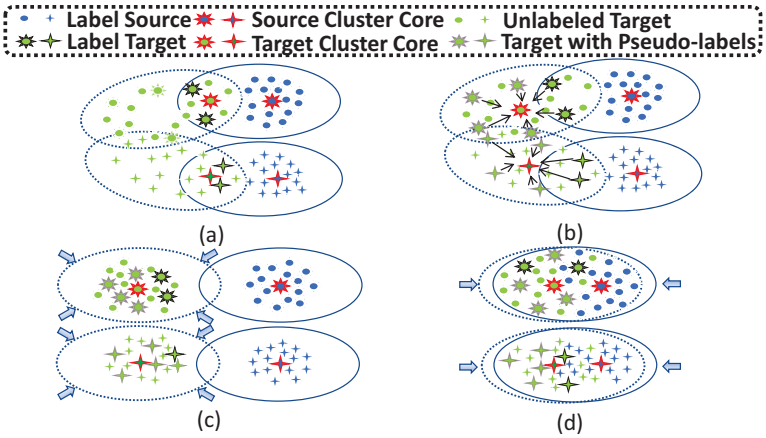


Figure 2.1: Conceptual overview of our Cross-domain Adaptive Clustering (CDAC) approach. (a) Supervision on labeled data from both source and target domains to ensure partial cross-domain feature alignment. (b) Pseudo labeling for giving pseudo-labels on unlabeled target samples to form enhanced target cluster cores with higher robustness and power. (c) Minimization of the adversarial adaptive clustering loss for grouping features from the target domain into clusters. (d) Maximization of the adversarial adaptive clustering loss to facilitate cross-domain cluster-wise feature alignment.

to aggregate in the corresponding clusters to form perfect class-wise sub-distributions in the target domain, which reduces the possibility of feature mismatch across domains. However, a model trained with supervision on few labeled target samples and labeled source data just can ensure partial cross-domain feature alignment because it only aligns the features of labeled target samples and their correlated nearby ones with the corresponding feature clusters in the source domain [59]. Also, the trained model cannot produce a highly discriminative feature representation for the target domain because the learned feature representation is biased to the sample discrimination of the source domain due to the existence of a much larger scale of labeled samples than those of the target domain [129]. These could lead to disconnection between the labeled and unlabeled target samples as well as misalignment between unlabeled target samples and the source domain.

In this paper, we propose a novel approach called **C**ross-**d**omain **A**daptive **C**lustering (CDAC), as Figure 2.1 shows, to address the aforementioned problem. It first groups features of unlabeled target data into clusters and further performs cluster-wise feature alignment across the source and target domains rather than sample-wise or distribution-wise feature alignment. In this way, our approach achieves both inter-domain adaptation and intra-domain adaptation simultaneously. More specifically, our proposed approach performs minimax optimization over the parameters of a feature extractor and a classifier. For intra-domain adaptation, the features of unlabeled target samples are guided by labeled target samples to form clusters corresponding to the classes of labeled samples by minimizing an adversarial adaptive clustering loss with respect to the parameters of the feature extractor. For inter-domain adaptation, the classifier is

trained to maximize the same loss defined on unlabeled target samples so that cluster-wise feature distribution in the target domain is aligned with the corresponding feature distribution in the source domain.

In addition, we apply pseudo labeling to unlabeled samples in the target domain and retain pseudo-labels with high confidence. In the SSDA setting, since only a very small number (typically one or three) of target samples from each class are labeled, it is hard for such few samples to form a stable and accurate cluster core. Pseudo labeling expands the number of "labeled" samples in each class in the target domain, and thus produces a more robust and powerful cluster core for each class. Such an enhanced cluster core can attract unlabeled samples from the corresponding class towards itself in the target domain using the adversarial adaptive clustering loss. Therefore, our pseudo labeling technique assists adversarial learning, and helps our SSDA model reach higher performance.

In summary, our main contributions of the proposed Cross-domain Adaptive Clustering (`CDAC`) approach are as follows.

- We introduce an adversarial adaptive clustering loss to perform cross-domain cluster-wise feature alignment so as to solve the SSDA problem.
- We integrate an adapted version of pseudo labeling to enhance the robustness and power of cluster cores in the target domain to facilitate adversarial learning.
- Extensive experiments on benchmark datasets, including DomainNet [118], Office-Home [149] and Office-31 [128], demonstrate that our proposed `CDAC` approach achieves the state-of-the-art performance in semi-supervised domain adaptation.

2.2 Related Work

2.2.1 Adversarial Learning for UDA

Most domain adaptation algorithms attempt to achieve feature distribution alignment between domains by minimizing the domain shift between the source domain and the target domain, so that the knowledge learned from the source data can be transferred to the target domain and improve its classification performance [114, 34]. Adversarial learning is one of the mainstream solutions [190, 144, 11]. Saito *et al.* [131] proposed to train task-specific classifiers and maximize their output discrepancy to detect target samples that are far from the support of the source distribution, then learn to generate target features near the support to fool the classifiers. [150, 112] introduce entropy-based adversarial training to enhance high-confident predictions in the target domain. Moreover, in order to overcome the issue of mode collapse caused by the separate design of task and domain classifiers, Tang *et al.* [139] proposed discriminative adversarial learning to promote the joint distribution alignment within both feature-level and class-level.

Different from previous sample-wise adversarial learning based domain adaptation methods, we first propose adaptive cluster-wise feature alignment to achieve both inter-domain and intra-domain adaptation. This method can greatly alleviate the situation that the model produces feature representations with bias towards the source domain caused by the dominance of most labeled source samples during model training, and can reduce the difficulty of exploring the decision boundary of the classifier by improving the cohesion of unlabeled samples in the target domain, so as to improve the performance of the model in a two-pronged manner.

2.2.2 Pseudo Labeling on UDA

Pseudo labeling, a.k.a self-training, is often used in semi-supervised learning, aiming to give reliable pseudo-labels to unlabeled data through an ensemble of output predictions from multiple models and assist model training to improve performance and its generalization [67, 37]. In the field of semi-supervised image classification, the reliability of pseudo-labels is usually improved by integrating the output predictions of one model with multiple augmented inputs [8, 64], outputs of different models [160], or multiple predictions of the same model in different training stages [182, 3, 29]. In previous researches, pseudo labeling is also proved to be effective in domain adaptation, e.g. [112] proposed entropy-based ranking function to separate the target domain data into an easy and hard split followed by employing self-supervised adaptation from easy to hard for decreasing intra-domain gap. To avoid introducing noise from pseudo labeling, [39] constructed a robust Gaussian-Uniform mixture model in spherical feature space to guarantee the correctness of given pseudo-labels from unlabeled target data.

In this work, pseudo labeling is employed to give pseudo-labels for unlabeled target data with high probabilistic confidence and thus expand the number of "labeled" samples in each class of the target domain, resulting in a more robust and powerful cluster core for each class to facilitate adversarial learning.

2.2.3 Semi-supervised Domain Adaptation

Semi-supervised domain adaptation (SSDA) is a relatively promising form of transfer learning, which intents to leverage a small number of labeled samples (e.g, one or few samples per class) in the target domain and give full play to their potential to greatly improve the performance of domain adaptation. Recently, SSDA has recently attracted wide attentions [129, 123, 56, 69, 59, 177] from researchers. [129, 123] first proposed to solve SSDA by aligning the features from both domains by means of adversarial learning. [56] proposed to reduce intra-domain discrepancy within the target domain to attract unaligned target sub-distributions towards the corresponding source sub-distributions so as to improve feature alignment across domains. In addition, [112] proposed to decompose SSDA into a semi-supervised learning (SSL) problem in the target domain and an unsupervised domain adaptation (UDA) problem across domains, and then train two classifiers using Mixup and Co-training methods, so as to bridge the

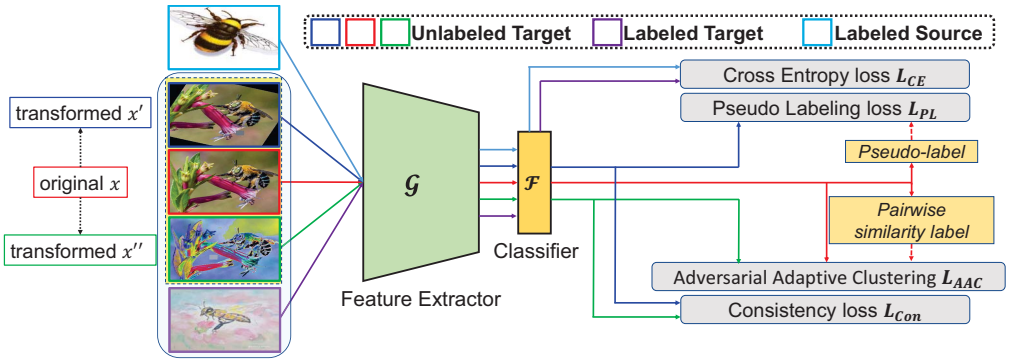


Figure 2.2: Outline of our model architecture and training procedure. Arrows with various colors represent data flows for different types of samples from both source and target domains. The feature extractor G uses Alexnet or Resnet-34 as the backbone network and the classifier F is an unbiased linear network with a normalized layer, which is shared by both domains. As shown, an image x from unlabeled target data is first fed to the feature extractor and the classifier and then its prediction is constructed to pairwise similarity label and pseudo-label, which are employed as targets for its two different transformed versions, x' and x'', to train the model with the adversarial adaptive clustering loss and the proposed pseudo labeling loss, respectively.

gap and exchange expertise between the source and target domains. Furthermore, [69] proposed to explore the optimal initial weights for the adaptation model using online meta-learning. Most of the previous approaches solve SSDA based on sample-wise feature alignment. In this work, we take an attempt to use adaptive cluster-wise feature alignment affiliated with pseudo labeling to achieve both inter-domain and intra-domain adaptation.

2.3 Methodology

In this section, we first introduce the background and notations of SSDA, and then present our proposed Cross-domain Adaptive Clustering (CDAC) approach, which contains an adversarial adaptive clustering loss and a pseudo labeling loss. Finally, we summarize the overall loss used in our work. An outline of our model architecture and training procedure is shown in Figure 2.2.

2.3.1 Semi-supervised Domain Adaptation

Semi-supervised domain adaptation seeks a classifier for a target domain when given labeled data $\mathcal{S} = \left(x_i^s, y_i^s\right)_{i=1}^{N_s}$ from a source domain as well as both unlabeled data $\mathcal{U} = \left\{\left(x_i^u\right)\right\}_{i=1}^{N_u}$ and labeled data $\mathcal{L} = \left\{\left(x_i^l, y_i^l\right)\right\}_{i=1}^{N_l}$ from the target domain. $\mathcal{S}, \mathcal{U}$ and $\mathcal{L}$ represent three subsets of available data in this problem, and they contain N_s, N_u and N_l instances, respectively. In the semi-supervised setting, N_l is much smaller than N_s and N_u, and only contains one shot or few shots per class. Each data point $x_i^s(x_i^l)$ from $\mathcal{S}(\mathcal{L})$ has its associated label $y_i^s(y_i^l)$, while any data point x_i^u from $\mathcal{U}$ has none.

Our work aims to make our SSDA model trained using $\mathcal{S}$, $\mathcal{U}$ and $\mathcal{L}$ perform well on test data from the target domain.

Our network consists of two components, i.e. a feature extractor G, parameterized by θ_G, and a classifier F, parameterized by θ_F, as in existing work [129, 56, 59]. The classifier F is an unbiased linear network with a normalization layer, which maps features from the feature extractor G into a spherical feature space. This similarity-based feature space is more suitable for decreasing the feature variance of samples sharing the same class label [129, 59]. These are commonly used model settings for the SSDA problem [129, 56, 59].

The feature of an input image x, $G(x)$, is fed into the classifier F to obtain the probabilistic prediction as follows:

$$p(x) = \sigma(F(G(x))), \tag{2.1}$$

where $\sigma(\cdot)$ is the softmax function. For convenience, we often abbreviate $p(x)$ as $\mathbf{p}$, i.e. $\mathbf{p} = p(x)$.

To train our model with supervision from all labeled data from both source and target domains, we follow the practices of existing work on SSDA [129, 123, 56, 69, 59, 177], and include the following standard cross-entropy loss in the training loss,

$$\mathcal{L}_{CE} = -\sum_{(x,y)\in\mathcal{S}\cup\mathcal{L}} y \log(p(x)). \tag{2.2}$$

2.3.2 Adversarial Adaptive Clustering

The key idea in our work is the introduction of an adversarial adaptive clustering loss into semi-supervised domain adaptation to group features in the target domain into clusters and further perform cross-domain cluster-wise feature alignment to achieve inter-domain adaptation and intra-domain adaptation simultaneously. Underlying assumptions are that features of sample images form clusters and samples from the same cluster should have similar features and share the same class label. This loss first computes pairwise similarities among features of unlabeled samples in the target domain, then forces the class labels predicted by the classifier for such samples with pairwise feature similarities to be consistent. The latter is achieved by training the model with a binary cross-entropy loss, where binary pairwise feature similarities are used as groundtruth labels. This loss can force the features from the target domain to form clusters.

In detail, the above approach requires setting up connections on the basis of a similarity measure between sample pairs (x_i^u, x_j^u) from the same mini-batch. According to the above assumption, for a pair of similar samples, we set a pairwise pseudo-label $s_{ij} = 1$ (i.e. pairwise connection between paired samples); otherwise, $s_{ij} = 0$ for dissimilar samples. According to [42], pairwise feature similarity can be measured using the

indices of feature elements rank ordered according to their magnitudes. If two samples share the same top-k indices in their respective lists of rank ordered feature elements, the paired samples belong to the same class with a high confidence and thus $s_{ij} = 1$; otherwise, $s_{ij} = 0$. Therefore, we can formulate pairwise similarity label as follows,

$$s_{ij} = 1\{\text{top}k\left(G\left(x_i^u\right)\right) = \text{top}k\left(G\left(x_j^u\right)\right)\}, \tag{2.3}$$

where top$k$ $(\cdot)$ denotes the top-k indices of rank ordered feature elements and we set $k = 5$. And $1\{\cdot\}$ is an indicator function.

Then we establish pairwise comparisons among unlabeled target data using the binary cross-entropy loss, which utilize the above pairwise feature similarity labels of sample pairs in a mini-batch as targets, i.e. our adversarial adaptive clustering loss $\mathcal{L}_{AAC}$ can be written as follows,

$$\begin{aligned}\mathcal{L}_{AAC} = -\sum_{i=1}^{M}\sum_{j=1}^{M} & s_{ij}\log(\mathbf{p}_i^T\mathbf{p}'_j) \\ & + (1 - s_{ij})\log(1 - \mathbf{p}_i^T\mathbf{p}'_j),\end{aligned} \tag{2.4}$$

where M is the number of unlabeled target samples in each mini-batch and $\mathbf{p}_i = p\left(x_i^u\right) = \sigma(F(G(x_i^u)))$ represents the prediction of an image x_i^u in the mini-batch. Also, $\mathbf{p}'_i = p\left(x'_i\right) = \sigma(F(G(x'_i)))$ indicates the prediction of a transformed image x'_j, which is an augmented version of x_j^u using a data augmentation technique. The inner product $\mathbf{p}_i^T\mathbf{p}'_j$ in Eq. (2.4) is used as a similarity score, which predicts whether image x_i^u and the transformed version of image x_j^u share the same class label or not. Besides, as illustrated in [126], combining data augmentation techniques in samples of can massively improve the model performance.

What is the goal of Cross-domain Adaptive Clustering achieved using the $\mathcal{L}_{AAC}$ loss? Similar to [129], we also enforce supervision on labeled samples from the source and target domains and perform minimax training on unlabeled target domain samples to optimize the model, but we replace the conditional entropy loss with our adversarial adaptive clustering loss. In our work, directly minimizing the $\mathcal{L}_{AAC}$ loss makes features of similar samples in the target domain close but features of dissimilar ones distant so that features form clusters within the target domain. However, the learned feature representation in the target domain would be always biased towards the source domain because a large number of source labels dominate the supervision process. Thus direct minimization of $\mathcal{L}_{AAC}$ over unlabeled target domain data would make this worse and give rise to more severe overfitting. Therefore, we utilize a gradient reversal layer [31] to flip the gradients of $\mathcal{L}_{AAC}$ between the feature extractor and the classifier and, in this situation, the classifier is still enforced to ensure correct classification in the target domain. In other words, the maximization of $\mathcal{L}_{AAC}$ on unlabeled target domain data

would decrease the bias of feature representations towards the source domain and encourage the model to produce more domain-invariant features so as to facilitate cross-domain feature alignment. Thus, a preliminary loss function for adversarial learning in our network can be summarized as follows,

$$\begin{aligned} \theta_G^* &= \arg\min_{\theta_G} \mathcal{L}_{CE} + \lambda \mathcal{L}_{AAC}, \\ \theta_F^* &= \arg\min_{\theta_F} \mathcal{L}_{CE} - \lambda \mathcal{L}_{AAC}, \end{aligned} \tag{2.5}$$

where λ is a scalar hyper-parameter that controls the balance between the cross-entropy loss and the proposed adversarial adaptive clustering loss.

2.3.3 Pseudo Labeling for Unlabeled Target Domain Data

Due to the small number of labeled target domain samples in the SSDA problem, it is hard for the adversarial adaptive clustering loss to form stable and accurate cluster cores in the target domain during model training, which may negatively affect cross-domain cluster-wise feature alignment. To solve this problem, we apply pseudo labeling to unlabeled target samples and retain pseudo-labels with high confidence to expand the number of "labeled" samples in the target domain, thereby forming more robust cluster cores for different classes. Pseudo labeling is a classic technique for semi-supervised learning [3, 64], and utilizes the prediction capability of a model to generate artificial hard labels for a subset of unlabeled samples and then train the model with a supervised loss involving these artificial labels. In our work, we choose the progressive pseudo labeling technique in [64].

In the proposed pseudo labeling process, we first feed an image x_j^u from a mini-batch of unlabeled images into the current model, and the prediction $\mathbf{p}_j = p\left(x_j^u\right) = \sigma(F(G(x_j^u)))$ from the model is then converted to a one-hot hard label $\hat{y}_j^u = \arg\max(\mathbf{p}_j)$, which is used as a pseudo label in a supervised loss. Afterwards, the prediction $\mathbf{p}_j'' = p\left(x_j''\right)$ produced from another transformed image x_j'' for the same image x_j^u is obtained to increase the input diversity of our model. Therefore, in this section, our model is trained using the standard cross-entropy loss as follows,

$$\mathcal{L}_{PL} = -\sum_{j=1}^{M} 1\{\max(\mathbf{p}_j) \geq \tau\} \cdot \hat{y}_j^u \log(\mathbf{p}\left(x_j''\right)), \tag{2.6}$$

where $\mathbf{p}_j'' = p\left(x_j''\right) = \sigma(F(G(x_j'')))$ denotes the model prediction of the transformed image x_j'', and τ is a scalar confidence threshold that determines the subset of pseudo labels that should be retained for model training.

Our $\mathcal{L}_{PL}$ loss is employed to enhance the adversarial adaptive clustering loss. Once pseudo-labels with high confidence are identified and used for model training,

more robust cluster cores in the target domain can be established to make the feature clusters in the target domain better aligned with the source domain ones.

2.3.4 Overall Loss

The overall loss function for training our SSDA network can be summarized as follows,

$$\begin{aligned}\theta_G^* &= \arg\min_{\theta_G} \mathcal{L}_{CE} + \lambda \mathcal{L}_{AAC} + \mathcal{L}_{PL} + \mathcal{L}_{Con}, \\ \theta_F^* &= \arg\min_{\theta_F} \mathcal{L}_{CE} - \lambda \mathcal{L}_{AAC} + \mathcal{L}_{PL} + \mathcal{L}_{Con},\end{aligned} \tag{2.7}$$

where

$$\mathcal{L}_{Con} = w(t) \sum_{j=1}^{M} ||\mathbf{p}'_j - \mathbf{p}''_j||^2, \tag{2.8}$$

and $w(t) = \nu e^{-5\left(1-\frac{t}{T}\right)^2}$ is a ramp-up function used in [66] with the scalar coefficient ν, the current time step t and the total number of steps T in the ramp-up process. In order to improve the input diversity of our model, we have created two different transformed versions of each unlabeled image in the target domain to implement the adversarial adaptive clustering loss and the pseudo labeling loss, respectively. Therefore, we employ a consistency loss, $\mathcal{L}_{Con}$, to keep the model predictions on these two transformed images consistent.

2.4 Experiments

2.4.1 Experimental Setups

Benchmark datasets. We evaluate the efficacy of our proposed `CDAC` approach on several standard SSDA image classification benchmarks, including the DomainNet[1] [118], Office-Home[2] [149] and Office-31[3] [128]. DomainNet is initially a multi-source domain adaptation benchmark, and MME [129] borrows its subset as one of the benchmarks for SSDA evaluation. Similar to the setting of MME, we only select 4 domains, which are Real, Clipart, Painting, and Sketch (abbr. **R**, **C**, **P** and **S**), each of which contains images of 126 categories. Office-Home is a widely used UDA benchmark and consists of Real, Clipart, Art and Product (abbr. **R**, **C**, **A** and **P**) domains with 65 classes. Office-31 is a relatively small dataset contains three domains including DSLR, Webcam and Amazon (abbr. **D**, **W** and **A**) with 31 classes. For fair comparisons, the settings of our benchmark datasets refer to the existing SSDA approaches [129, 123, 59], including adaptation scenarios of each dataset, the number of labeled target data (typically 1-shot or 3-shot per class), sample selection strategies, etc.

[1] http://ai.bu.edu/M3SDA/
[2] http://hemanthdv.org/OfficeHome-Dataset/
[3] https://people.eecs.berkeley.edu/ jhoffman/domainadapt/

Table 2.1: Comparison results (Accuracy(%)) of CDAC ad the state-of-the-art SSDA algorithms on DomainNet under the settings of 1-shot and 3-shot using Alexnet and Resnet-34 as backbone networks.

Net	Method	R→C		R→P		P→C		C→S		S→P		R→S		P→R		Mean	
		1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
Alexnet	S+T [129]	43.3	47.1	42.4	45.0	40.1	44.9	33.6	36.4	35.7	38.4	29.1	33.3	55.8	58.7	40.0	43.4
	DANN [32]	43.3	46.1	41.6	43.8	39.1	41.0	35.9	36.5	36.9	38.9	32.5	33.4	53.5	57.3	40.4	42.4
	ENT [37]	37.0	45.5	35.6	42.6	26.8	40.4	18.9	31.1	15.1	29.6	18.0	29.6	52.2	60.0	29.1	39.8
	MME [129]	48.9	55.6	48.0	49.0	46.7	51.7	36.3	39.4	39.4	43.0	33.3	37.9	56.8	60.7	44.2	48.2
	Meta-MME [69]	-	56.4	-	50.2		51.9	-	39.6	-	43.7	-	38.7	-	60.7	-	48.8
	BiAT [56]	54.2	58.6	49.2	50.6	44.0	52.0	37.7	41.9	39.6	42.1	37.2	42.0	56.9	58.8	45.5	49.4
	APE [59]	47.7	54.6	49.0	50.5	46.9	52.1	38.5	42.6	38.5	42.2	33.8	38.7	57.5	61.4	44.6	48.9
	CDAC	**56.9**	**61.4**	**55.9**	**57.5**	**51.6**	**58.9**	**44.8**	**50.7**	**48.1**	**51.7**	**44.1**	**46.7**	**63.8**	**66.8**	**52.1**	**56.2**
Resnet-34	S+T [129]	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
	DANN [32]	58.2	59.8	61.4	62.8	56.3	59.6	52.8	55.4	57.4	59.9	52.2	54.9	70.3	72.2	58.4	60.7
	ENT [37]	65.2	71.0	65.9	69.2	65.4	71.1	54.6	60.0	59.7	62.1	52.1	61.1	75.0	78.6	62.6	67.6
	MME [129]	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
	UODA [123]	72.7	75.4	70.3	71.5	69.8	73.2	60.5	64.1	66.4	69.4	62.7	64.2	77.3	80.8	68.5	71.2
	Meta-MME [69]	-	73.5	-	70.3	-	72.8	-	62.8	-	68.0	-	63.8	-	79.2	-	70.1
	BiAT [56]	73.0	74.9	68.0	68.8	71.6	74.6	57.9	61.5	63.9	67.5	58.5	62.1	77.0	78.6	67.1	69.7
	APE	70.4	76.6	70.8	72.1	72.9	76.7	56.7	63.1	64.5	66.1	63.0	67.8	76.6	79.4	67.6	71.7
	CDAC	**77.4**	**79.6**	**74.2**	**75.1**	**75.5**	**79.3**	**67.6**	**69.9**	**71.0**	**73.4**	**69.2**	**72.5**	**80.4**	**81.9**	**73.6**	**76.0**

Table 2.2: Comparison results (Accuracy(%)) of CDAC ad the state-of-the-art SSDA algorithms on Office-Home under the setting of 3-shot using Alexnet and Resnet-34 as backbone networks.

Net	Method	R→C	R→P	R→A	P→R	P→C	P→A	A→P	A→C	A→R	C→R	C→A	C→P	Mean
Alexnet	S+T [129]	44.6	66.7	47.7	57.8	44.4	36.1	57.6	38.8	57.0	54.3	37.5	57.9	50.0
	DANN [32]	47.2	66.7	46.6	58.1	44.4	36.1	57.2	39.8	56.6	54.3	38.6	57.9	50.3
	ENT [37]	44.9	70.4	47.1	60.3	41.2	34.6	60.7	37.8	60.5	58.0	31.8	63.4	50.9
	MME [129]	51.2	73.0	50.3	61.6	47.2	40.7	63.9	43.8	61.4	59.9	44.7	64.7	55.2
	Meta-MME [69]	50.3	-	-	-	48.3	40.3	-	44.5	-	-	44.5	-	-
	BiAT [56]	-	-	-	-	-	-	-	-	-	-	-	-	56.4
	APE [59]	51.9	74.6	51.2	61.6	47.9	42.1	**65.5**	44.5	60.9	58.1	44.3	64.8	55.6
	CDAC	**54.9**	**75.8**	**51.8**	**64.3**	**51.3**	**43.6**	65.1	**47.5**	**63.1**	**63.0**	**44.9**	**65.6**	**56.8**
Resnet-34	S+T [129]	55.7	80.8	67.8	73.1	53.8	63.5	73.1	54.0	74.2	68.3	57.6	72.3	66.2
	DANN [32]	57.3	75.5	65.2	69.2	51.8	56.6	68.3	54.7	73.8	67.1	55.1	67.5	63.5
	ENT [37]	62.6	85.7	70.2	79.9	60.5	63.9	79.5	61.3	79.1	76.4	64.7	79.1	71.9
	MME [129]	64.6	85.5	71.3	80.1	64.6	65.5	79.0	63.6	79.7	76.6	67.2	79.3	73.1
	Meta-MME [69]	65.2	-	-	-	64.5	66.7	-	63.3	-	-	**67.5**	-	-
	APE [59]	66.4	**86.2**	**73.4**	**82.0**	65.2	66.1	**81.1**	63.9	80.2	76.8	66.6	79.9	74.0
	CDAC	**67.8**	85.6	72.2	81.9	**67.0**	**67.5**	80.3	**65.9**	**80.6**	**80.2**	67.4	**81.4**	**74.2**

Implementation details. Similar to previous SSDA work [129, 56], we choose Alexnet and Resnet-34 as our backbone networks. Firstly, the feature extractor is initialized with a pre-trained model on ImageNet[4] and the linear classification layer is initialized randomly, which has the same setting as [129, 123, 59, 56], such as architecture, output feature size, and so on. To balance multiple loss terms, we set λ in Eq. (2.7) to 1.0 and ν in Eq. (2.8) to 30.0. Also, we set the confidence threshold $\tau = 0.95$ in Eq. (2.6). We implement our experiments on the widely-used PyTorch[5] platform. Additionally, in each iteration, we first train our model with the standard cross-entropy loss only on labeled data from both source and target domains and then add our proposed losses on unlabeled target data to further optimize the model. Furthermore, we introduce RandAugment [19] as the data augmentation techniques used in this work. Finally, for fair comparisons, other experimental settings in our proposed CDAC, such as the optimizer, learning rate, mini-batch size, are the same as MME [129].

Baselines. We compare CDAC with previous state-of-the-art SSDA approaches, including **MME** [129], **UODA** [123], **BiAT** [56], **Meta-MME** [69], **APE** [59], **S+T**, **DANN** [32] and **Ent** [37]. Specifically, the model of the **S+T** method is trained using labeled source

[4]http://www.image-net.org/
[5]https://pytorch.org/

Table 2.3: Comparison results (Accuracy(%)) of CDAC ad the state-of-the-art SSDA algorithms on Office-31 under the settings of 1-shot and 3-shot on the Alexnet backbone network.

Net	Method	W→A		D→A		Mean	
		1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
Alexnet	S+T [129]	50.4	61.2	50.0	62.4	50.2	61.8
	DANN [32]	57.0	64.4	54.5	65.2	55.8	64.8
	CDAN [93]	50.4	60.3	48.5	61.4	49.5	60.8
	ENT [37]	50.7	64.0	50.0	66.2	50.4	65.1
	MME [129]	57.2	67.3	55.8	67.8	56.5	67.6
	BiAT [56]	57.9	68.2	54.6	68.5	56.3	68.4
	APE [59]	-	67.6	-	69.0	-	68.3
	CDAC	**63.4**	**70.1**	**62.8**	**70.0**	**63.1**	**70.0**

Table 2.4: Comparison results (Accuracy(%)) of CDAC and the state-of-the-art SSDA algorithms on DomainNet under the 5-shot and 10-shot settings using Resnet-34 as backbone.

Net	Method	R→C	R→P	P→C	C→S	S→P	R→S	P→R	MEAN
					5-shot				
Resnet-34	S+T [129]	64.5	63.1	64.2	59.2	60.4	56.2	75.7	63.3
	DANN [32]	63.7	62.9	60.5	55.0	59.5	55.8	72.6	61.4
	ENT [37]	77.1	71.0	75.7	61.9	66.2	64.6	81.1	71.1
	MME [129]	75.5	70.4	74.0	65.0	68.2	65.5	79.9	71.2
	APE [59]	77.7	73.0	76.9	67.0	71.4	68.8	80.5	73.6
	CDAC	**80.8**	**75.3**	**79.9**	**72.1**	**74.7**	**72.9**	**83.2**	**76.9**
					10-shot				
Resnet-34	S+T [129]	68.5	66.4	69.2	64.8	64.2	60.7	77.3	67.3
	DANN [32]	70.0	64.5	64.0	56.9	60.7	60.5	75.9	64.6
	ENT [37]	79.0	72.9	78.0	68.9	68.4	68.1	82.6	74.0
	MME [129]	77.1	71.9	76.3	67.0	69.7	67.8	81.2	73.0
	APE [59]	79.8	75.1	78.9	70.5	73.6	70.8	82.9	75.9
	CDAC	**83.1**	**77.2**	**81.7**	**74.3**	**76.3**	**74.6**	**84.7**	**78.9**

and target data only. In addition, **DANN** and Ent are both representative UDA methods and we re-train the models of DANN and Ent with an additional supervision loss by adding a few labeled target data.

2.4.2 Comparisons with State-of-the-Arts

Results on DomainNet, Office-Home and Office-31 under the settings of 1-shot and 3-shot with Alexnet and Resnet-34 as backbone networks are reported in Table 2.1, 2.2 and 2.3, respectively. As illustrated, our proposed CDAC significantly outperforms the state of the art throughout all experiments.

On DomainNet. As shown in Table 2.1, our CDAC significantly outperforms the existing approaches in all adaptation scenarios on DomainNet. Using Alexnet as the backbone, our method surpasses the existing best performing approach by 6.6% and 6.8% on average w.r.t the 1-shot and 3-shot settings respectively. Compared with the competing approaches using Resnet-34 as the backbone, CDAC also achieves the best results in all cases and surpasses the current best results by 6% and 4.3% in the settings of 1-shot and 3-shot. Note that **MiCo** proposed in [177] is an unpublished work concurrent with

Table 2.5: Ablation study results (Accuracy(%)) of CDAC using Resnet-34 as the backbone on DomainNet under the setting of 3-shot. In the UDA setting, the supervised cross-entropy loss $\mathcal{L}_{CE}$ refers to the model trained only with labeled source samples.

Net	Setting	$\mathcal{L}_{CE}$	$\mathcal{L}_{AAC}$	$\mathcal{L}_{PL}$	$\mathcal{L}_{Con}$	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Mean
Resnet-34	UDA	✓				57.8	61.4	58.1	53.4	58.2	49.6	72.3	58.6
	UDA	✓	✓			64.6	64.8	64.8	59.9	62.2	58.8	73.0	64.0
	UDA	✓		✓		68.0	73.2	68.3	61.8	67.0	63.1	76.5	68.2
	UDA	✓	✓	✓		76.9	73.9	73.9	66.2	70.2	69.0	79.3	72.8
	UDA	✓	✓	✓	✓	77.1	74.4	73.2	67.0	70.4	69.6	79.6	73.0
	SSDA	✓				60.0	62.2	59.4	55.0	59.5	50.1	73.9	60.0
	SSDA	✓	✓			69.4	68.1	68.3	62.8	65.6	62.0	76.9	67.6
	SSDA	✓		✓		76.7	73.6	76.3	66.9	70.3	69.3	80.4	73.4
	SSDA	✓	✓	✓		78.7	74.9	78.5	69.7	73.2	71.1	81.6	75.3
	SSDA	✓	✓	✓	✓	79.6	75.1	79.3	69.9	73.4	72.5	81.9	76.0

ours, and the average performance of our CDAC using Resnet-34 as the backbone is 0.4% higher than **MiCo** under the 3-shot setting.

On Office-Home and Office. To be consistent with the previous methods and achieve a fair comparison, we just employ Alexnet as the backbone on the Office-31 benchmark. As shown in Table 2.2 and Table 2.3, our CDAC outperforms all comparison methods w.r.t mean accuracy on both datasets. In addition, it is worth noting that our method using Alexnet as the backbone achieves superior performance for most adaptation scenarios on Office-Home, and consistently achieves the best performance on Office-31 w.r.t the adaptation scenarios of both "W→A" and "D→A".

Additional performance comparisons on DomainNet. We show additional comparisons with a varying number of labeled target domain samples of each category, i.e. 5-shot and 10-shot per class, on the DomainNet benchmark using Resnet-34 as the backbone network in Table 2.4. In comparison to the existing state-of-the-art SSDA approaches, the proposed method achieves better classification performance on DomainNet in all adaptation scenarios. Specifically, our CDAC method outperforms the previous best results by 3.3% and 3.0% on average under the 5-shot and 10-shot settings respectively.

2.4.3 Analysis

Ablation studies. We perform ablation studies on both SSDA and UDA settings to analyze the effectiveness of each loss term in our proposed CDAC, including $\mathcal{L}_{CE}$, $\mathcal{L}_{AAC}$, $\mathcal{L}_{PL}$ and $\mathcal{L}_{Con}$. All experiments are conducted on DomainNet using Resnet-34 as the backbone under the 3-shot setting. As shown in Table 2.5, we regard the model trained with the cross-entropy loss $\mathcal{L}_{CE}$ only on labeled samples from both domains as the baseline for SSDA. And then, by combining both $\mathcal{L}_{AAC}$ and $\mathcal{L}_{PL}$ with $\mathcal{L}_{CE}$, the trained model achieves 25.3% higher average performance than the baseline, while the classification accuracy is on average 17.6% (+$\mathcal{L}_{AAC}$) or 23.4% (+$\mathcal{L}_{PL}$) higher than the baseline when only one of them is used together with the cross-entropy loss. Furthermore, the model trained with all loss functions reaches the best classification performance compared with the baseline. Moreover, each loss term proposed in our approach used for

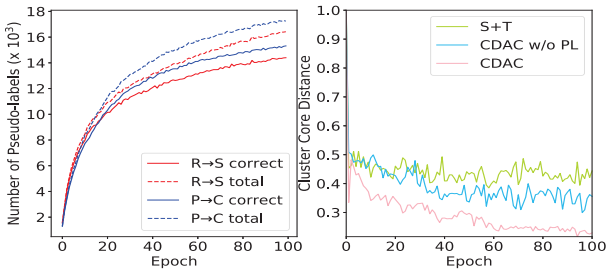


Figure 2.3: Left: The quantity and correctness of the proposed pseudo labeling technique on two adaptation scenarios of DomainNet (i.e., "R→S" and "P→C"), using Resnet-34 as the backbone under the setting of 3-shot and 1-shot, respectively. **Right:** Variation of Cluster Core Distance among different approaches during model training while class "axe" is taken as an example.

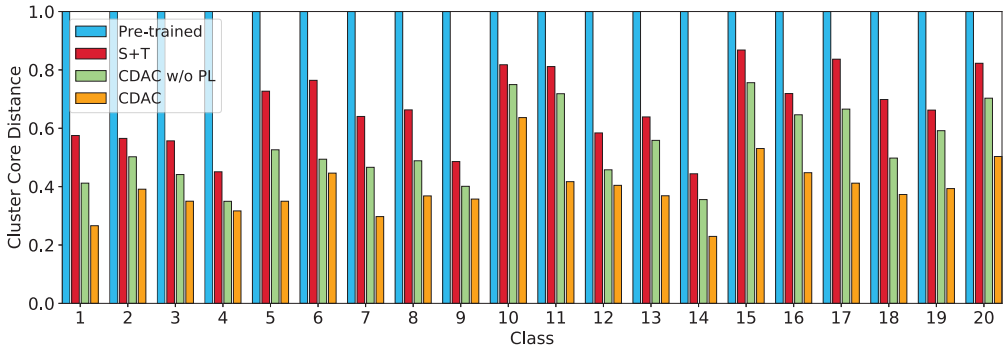


Figure 2.4: Comparison of final cluster core distances (CCDs) of 20 representative classes. Class 1-20 denote "see_saw", "speedboat", "sheep", "leaf", "raccoon", "feather", "laptop", "dog", "umbrella", "grapes", "streetlight", "foot", "butterfly", "axe", "eyeglasses", "goatee", "drums", "helmet", "asparagus" and "penguin". For each class, we show four results obtained from different approaches to indicate different approaches have different abilities to make target domain clusters and their corresponding source domain clusters closer. Apparently, our CDAC approach produces more discriminative features to help align cluster-wise feature distributions across domains.

unlabeled target examples also shows similar roles in improving classification performance under the UDA setting.

Effectiveness of Adversarial Adaptive Clustering. To evaluate the effectiveness of the adversarial adaptive clustering loss, we refer to [141, 95] and employ Cluster Core Distance (CCD) to measure the distance between the source and target domain feature clusters within the same class. Generally speaking, the more aligned cross-domain feature clusters are, the smaller the CCDs are. We compare our CDAC model with S+T and "CDAC w/o PL" (a degraded version of CDAC, which is trained with only $\mathcal{L}_{CE}$ and $\mathcal{L}_{AAC}$). As shown in the right of Figure 2.3, it can be observed that the CCDs of all three methods decrease gradually during model training and it demonstrates that the source and target domain clusters within each class become closer. And both "CDAC w/o PL" and CDAC can result in better feature alignment than S+T. The CCD obtained from the model trained with CDAC finally converges to the minimum value, indicating

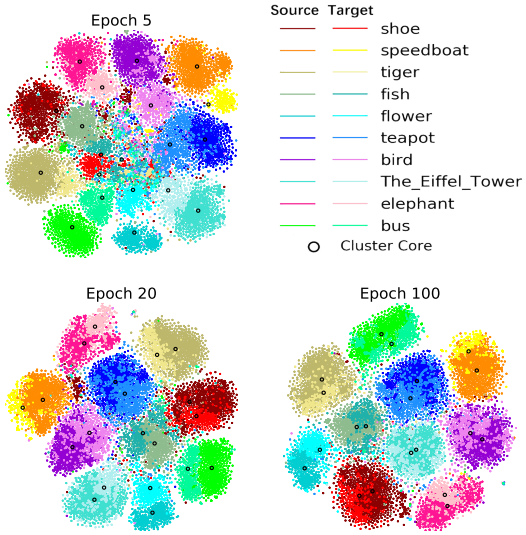


Figure 2.5: Visualization for feature distribution variations during model training with t-SNE. We choose 10 representative classes under the adaptation scenario "R→S" on DomainNet and their corresponding feature distributions from both source and target domains are displayed with different colors while black cycles represent cluster cores. We can observe that the final features have better cross-domain alignment than those at the beginning.

that CDAC overall shows the best classification performance. This demonstrates the effectiveness of the proposed adversarial adaptive clustering loss in guiding the model towards learning better cluster-wise feature alignment.

Additionally, inspired by [95, 141], Cluster Core Distance (CCD) is introduced to evaluate the effectiveness of the adversarial adaptive clustering loss, which measures the distance between the two source and target domain feature clusters within the same class. In details, the CCDs can be defined as $\{d_1^e, d_2^e, \ldots, d_k^e, \ldots, d_K^e\}$, where d_k^e denotes the Euclidean distance for class k in the e-th epoch during model training and K represents the number of classes in a dataset. To be fair, the CCD d_k^e at each epoch e is normalized by the initial CCD d_k^0, which is calculated using the initial model parameterized by pre-trained weights on ImageNet without any fine-tuning. In general, the more aligned cross-domain feature clusters are, the smaller CCDs are.

Therefore, We here use 2000 samples (1000 from source domain and 1000 from target domain, each includes 50 samples per class from 20 representative classes) in this validation experiment on DomainNet in the adaptation scenario, "R→S", under the 3-shot setting using Resnet-34 as the backbone. We compare our CDAC model with "Pre-trained", S+T and "CDAC w/o PL" models. Specifically, "Pre-trained" means that the model is parameterized using pre-trained weights on ImageNet without any further training, while the S+T model is trained with labeled samples in the source and target domains only. Also, "CDAC w/o PL" denotes a degraded version of our complete CDAC model, and is trained with the standard cross-entropy loss and the proposed

adversarial adaptive clustering loss without using any pseudo labels. We show the final CCDs of the above mentioned 20 representative classes in Figure 2.4, and it can be observed that in every class, the CCD of our complete CDAC model achieves the smallest value. These results further verify that our proposed approach can effectively perform cross-domain cluster-wise feature alignment and help improve the classification performance of SSDA models.

Effectiveness of Pseudo Labeling. The left subfigure in Figure 2.3 shows the quality and correctness of our proposed pseudo labeling technique in the model training process under two adaptation scenarios on DomainNet (i.e., "R→S" and "P→C" using Resnet-34 as the backbone under the setting of 3-shot and 1-shot, respectively). It displays that a large proportion of unlabeled data is given correct pseudo-labels (up to 59.9% and 63.8% of total training examples per epoch at the best performance, respectively), which demonstrates the effectiveness of the proposed pseudo labeling technique in CDAC.

Feature visualization. We report with t-SNE [98] to display the gradual process of cluster-wise feature alignment during model training using the adaptation scenario "R→S" of DomainNet under the setting of 3-shot with Resnet-34 as the backbone. As shown in Figure 2.5, we visualize the variations of the cluster and the corresponding cluster core of each class in the model training process. It can be observed that as the model optimization progresses, target features gradually converge towards target cluster cores, and each cluster in the target domain also gradually moves closer to their corresponding source cluster cores, showing a cluster-wise feature alignment effect. We take the "bus" class as an example. In Epoch 5, the feature distributions from both source and target domains are relatively far away. Then, as the model iterates, they gradually approach and finally achieve a perfect match at the last epoch.

2.5 Conclusions

We have presented a novel approach called Cross-domain Adaptive Clustering (CDAC) to solve the SSDA problem. CDAC consists of an adversarial adaptive clustering loss to guide the model training towards grouping the features of unlabeled target data into clusters and further performing cluster-wise feature alignment across domains. Furthermore, an adapted version of pseudo labeling is integrated into CDAC to enhance the robustness and power of cluster cores in the target domain to facilitate adversarial learning. Extensive experimental results, as well as ablation studies, have validated the virtue of our proposed method.

Chapter 3

Adaptive Betweenness Clustering for Semi-Supervised Domain Adaptation

In this chapter, another extended algorithm for dealing with the robust vision learning task of semi-supervised domain adaptation is proposed.

3.1 Introduction

Deep neural network (DNN) has led to a series of breakthroughs in computer vision tasks such as Image Classification [101, 10, 82, 159, 70], Semantic Segmentation [88, 33, 173, 166], Object Detection [191, 52, 189, 170], Medical Analysis [196, 197], etc. However, the impressive effectiveness of the training of deep network models remarkably depends on a large number of sample labels, necessitating laborious work in data annotation. An alternative solution comprises boosting the model for the domain of interest (a.k.a., target domain) by employing off-the-shelf labeled training samples from a relevant domain (a.k.a., source domain). Nonetheless, due to the distribution/domain gap, such a solution often cannot generalize well from the source domain to the target domain to deal with variant circumstances of domain gaps. Unsupervised domain adaptation (UDA), which aims to tackle the distribution gap and decrease the influence of domain shift, has thus gained significant attention for a long time [9, 107, 35, 46], [32, 62, 156].

Recently, semi-supervised domain adaptation (SSDA), a variant of the UDA task, has received wider attention [129, 59, 71]. With a few labeled target samples, SSDA can significantly enhance the adaptation model's performance w.r.t the target domain, compared to unsupervised domain adaptation. In this way, a small amount of annotated data in the target domain can be used to expand the semantic space, allowing a large number of samples of the same category from different domains to be clustered together at the feature level, so as to achieve partial categorical alignment.

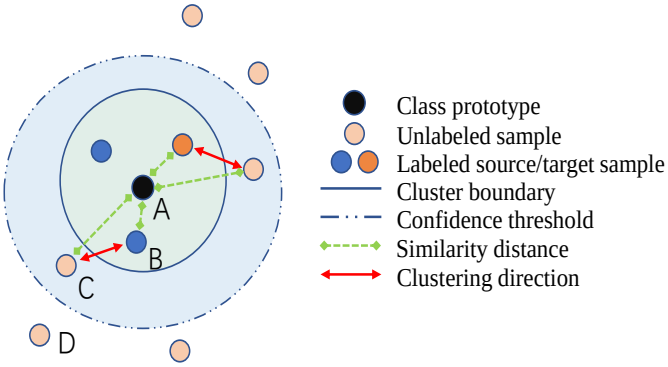


Figure 3.1: Conceptual illustration of G-ABC to showcase Adaptive Betweenness Clustering (ABC). The proposed `G-ABC` algorithm conducts sample clustering between a labeled point (e.g., "B") and an unlabeled point (e.g., "C"), when they have similarity distances within a confidence threshold to the same class prototype (e.g., "A") and they are with similar prediction distributions from the model. Herein, Point "A" guides the clustering process by serving as an intermediary (betweenness) point for Points "B" and "C". Point "D" is outside of the clustering range.

Despite the advantages of SSDA over UDA, SSDA also presents its own specific challenges. At first, in the SSDA scenario, a supervised model trained on a small number of labeled target samples and a large amount of labeled source data can only achieve partial cross-domain feature alignment, as it only aligns features of labeled target samples and their correlated nearby samples with the corresponding feature clusters in the source domain [59]. In addition, the trained SSDA model is incapable of producing highly discriminative feature representations for the target domain since the massive labels of the source domain dominate the supervision and causes the learned feature representations to be biased towards the discrimination of the source domain [129].

Preliminary SSDA works, such as [129, 59, 71, 122], have each proposed their individual solutions to tackle these challenges, and significant performance improvement has been witnessed. However, as with previous UDA studies focusing on global feature confusion at the domain level [113, 169, 32, 145], existing works are still unable to reach globally categorical domain alignment due to the scarcity of semantic label information for each category in the target domain. In other words, despite the presence of perfect domain-level alignment in feature confusion, it leads to label mismatch between massive unlabeled target data and the data of the source domain, hence compromising the model's performance.

Recent SSDA algorithms, such as [81, 94], have demonstrated that the semantic-aligned feature confusion across domains appears to work properly in semi-supervised domain adaptation, since during domain alignment, sample features from both domains with the same class will likely be aggregated into a same cluster. However, these methods achieve categorical domain alignment mostly through promoting semantic alignment between labeled samples across domains, with far less emphasis on employing a vast amount of unlabeled target samples.

In this paper, we present a novel SSDA approach, named **G**raph-based **A**daptive **B**etweenness **C**lustering (G-ABC), to tackle the challenges of the SSDA tasks. To achieve globally categorical domain alignment, we propose to enforce semantic transfer from labeled samples across domains to unlabeled target samples in order to promote cross-domain semantic alignment. Using the ground-truth sample labels from both domains as references, the trained SSDA model may thereby propagate semantic label information to the unlabeled target samples. In this way, a substantial amount of target label information is augmented through semantic propagation, thus significantly enhancing the generalization of the model to the target domain.

Specifically, we first construct a graph to capture the pairwise associations between unlabeled target samples and labeled samples from either the source or target domain, based on the pairwise label similarity of those paired samples. Then, we provide two connectivity refinement strategies, namely Confidence Uncertainty based Node Removal and Prediction Dissimilarity based Edge Pruning, to eliminate the noisy connectivity in the graph. In detail, the former degrades the connectivity towards unreliable unlabeled samples by removing unlabeled target instances with low predicted confidence, whereas the latter prunes the graph connections between samples with divergent probabilistic prediction distributions.

With the refined graph structure, we design a new clustering algorithm, namley Adaptive Bewteenness Clustering (ABC) as shown in Figure 3.1. To achieve semantic transfer, in this algorithm, we model the task of sample clustering between a labeled and an unlabeled sample as a binary pairwise-classification problem. The fundamental premise behind such an algorithm is to aggregate the feature representations of the paired samples that share the same class in the graph while separating those of different classes.

In particular, this algorithm involves two strategies to achieve semantic propagation, namely across-domain betweenness clustering (ADBC) and within-domain betweenness clustering (WDBC), by clustering the unlabeled target samples towards the labeled source or target domains. As a result, the ADBC strategy encourages alignment between unlabeled target samples and the source domain, whereas the WDBC scheme strengthens linkages between labeled and unlabeled target samples. Ultimately, semantic label information can be gradually transferred into unlabeled target instances as model training iterates. In this way, the rising balance of semantic label information of the source and target domains eliminates model bias toward the source domain and achieves globally categorical domain alignment, driving the model to generate more domain-invariant yet discriminative target features.

To sum up, our main contributions can be shown as follows:

- We propose a novel SSDA framework called **G**raph-based **A**daptive **B**etweenness **C**lustering (G-ABC) to tackle semi-supervised domain adaptation. To achieve globally categorical domain alignment, the proposed G-ABC conducts cross-domain

semantic alignment with semantic transfer from labeled samples of both domains to unlabeled target data.

- We construct a graph to characterize the associations between unlabeled target examples and labeled data of both domains. Two connectivity refinement strategies, namely Confidence Uncertainty based Node Removal and Prediction Dissimilarity based Edge Pruning, are further provided to decrease the noisy connectivity in the graph.
- Given the above-refined graph structure, we propose Adaptive Betweenness Clustering to impose semantic transfer across domains; in particular, we design across-domain betweenness clustering and within-domain betweenness clustering, respectively, to propagate semantic label information from labeled source and target domains to unlabeled target samples.
- We perform extensive experiments on three standard benchmark datasets, including DomainNet [118], Office-Home [149] and Office-31 [128], to verify the effectiveness of our proposed method, and the results show that our method outperforms all previous state-of-the-art SSDA methods by clear margins.

3.2 Related Work

3.2.1 Domain Adaptation

Domain adaptation (DA) addresses the problem of generalizing a model trained on a large number of labeled samples from the source domain to the target domain [48, 168, 201, 188, 188]. With the goal of decreasing the distribution gap across various domains, the most challenging issue of the DA problem is assisting the model in learning domain-invariant features. To accomplish domain adaptation, early classic algorithms to handle the domain adaptation tasks involve reducing the distribution discrepancies across domains assessed by Maximum Mean Discrepancy (MMD) [113, 169], sharing the identical cross-domain statistics (e.g., mean value and covariance) [136], and so on.

Tzeng et al. in [146], for instance, presented a method for leveraging MMD to drive the model to generate domain-invariant features by assessing the discrepancies between the model outputs of both domains. Long et al. optimized the adaptation model from a different perspective [92], i.e., employing multi-kernel MMD to evaluate the output differences of samples across domains at multiple model levels. In addition, recent advances trends favor adversarial domain alignment to expedite feature alignment across domains so that knowledge from classifiers trained on labeled source samples can be efficiently transferred to the target domain [32, 145, 135, 14, 12, 4, 174]. For example, Saito et al. conducted minimax training over unlabeled target samples to cluster these target features around domain-invariant class prototypes, thus imposing cross-domain feature alignment [129]. The aforementioned DA algorithms aimed at aligning source and target features at the domain level.

However, many related advances, such as [13, 115, 81, 194], demonstrated that decreasing the discrepancies of conditional distributions towards categorical domain alignments is preferred, resulting in improved adaptation between domains. To this end, it should be natural to incorporate semantic label information into adaptation. For instance, semantic alignment was proposed in [104, 83, 164] to achieve this. Motivated by it, we also emphasize semantic-level domain alignment, leveraging source and target labels as references, and encouraging semantic transfer from labeled source and target domains to unlabeled target samples.

3.2.2 Domain Adaptation Related to Graphs

The graphs employed in domain adaptation capture latent topological structures among the training data across domains, such that the learned relationships between domain samples can then be leveraged to encourage the model to better adapt reliable data structures from both domains. In general, samples from the source and target domains are represented by distinct graph structures. Thus, many previous DA algorithms relating to graphs, such as [187, 157, 181], first attempted to transfer knowledge learned on a labeled source graph to an unlabeled target graph. Based on the labeled graph in the source domain, these algorithms engaged source labels as the supervision signals, and the model therefore received training on both the source graph and the target graph respectively.

For example, Pilanci et al. employed frequency analysis to align two data graphs through which information can be transferred or shared [121, 120]. On the other hand, Ding et al. in [24] proposed constructing a cross-domain graph based on samples from both domains to capture the intrinsic structure in the shared space among the training data in order to concurrently enforce domain transfer and label propagation. In this way, domain-invariant feature learning and target discriminative feature learning are unified into the same framework, thus benefiting each other for more effective knowledge transfer.

Based on such an above observation, we also propose to hire the cross-domain graph to achieve categorical domain alignment. Nevertheless, earlier related works usually model graph Laplacian regularization to push graph nodes closer, but this unsupervised technique disregards the usage of sample labels to assist semantics alignment [195, 105, 152]. In our work, we design a novel clustering algorithm called Adaptive Betweenness Clustering to take full advantage of both source and target labels, thereby contributing to enhanced performance for the model to classify target samples.

3.2.3 Semi-supervised Domain Adaptation

Due to the availability of a few target labels, semi-supervised domain adaptation has the potential to significantly improve the classification performance of the model on the target domain in comparison to unsupervised domain adaptation [59, 69, 15, 176, 28]. Recent progress in SSDA, such as [129, 59, 56, 68, 122, 71] have primarily focused

on adversarial training to align cross-domain feature distributions. Here we mainly describe some related SSDA approaches that do not involve adversarial learning.

For example, Samarth et al. in [103] demonstrated that without the need for conventional adversarial domain alignment, self-supervision based pre-training and consistency regularization might be relied upon to produce a stronger classifier in the target domain, while Luo et al. in [94] developed "Relaxed cGAN" to transfer image styles from source samples to unlabeled target samples in order to help achieve domain-level distribution alignment. Besides, Yoon et al. in [28] also focus on style transferring for achieving better adaptation, generating assistant features by transferring intermediate styles between labeled and unlabeled samples.

In addition to bridging the gap and exchanging knowledge between the source and target domains, Yang et al. in [176] proposed decomposing the SSDA task into a semi-supervised learning (SSL) problem and an unsupervised domain adaptation problem. Specifically, the former is used to improve discrimination in the target domain, whereas the latter facilitates domain alignment. Such an algorithm trains two distinct classifiers utilizing Mixup and Co-training, respectively, in order to learn complementary features to each other, resulting in better domain adaptation. Similarly, [194] also performed adaptation by learning two classifier networks, trained for contradictory purposes. The first classifier groups target features to enhance intra-class density and increase categorical cluster gaps for robust learning, while the second, as a regularizer, disperses source features for a smoother decision boundary.

In this paper, we extend the motivation from [94, 81] and propose `G-ABC`, which achieves categorical domain alignment by providing increased access to unlabeled target samples during adaptation.

While there appears to be a superficial similarity between [180] and our proposed method in encouraging consistent predictions between features, we would like to clarify their differences as follows.

1. [180] merely encourages prediction consistency or similarity between a feature and its few neighbors, all of which are from the unlabeled target data. In contrast, our method, `G-ABC`, employs the proposed clustering technique called Adaptive Betweenness Clustering to group unlabeled samples toward labeled source or target instances. This is achieved by enforcing consistent probabilistic prediction distributions between two similar samples while forcing inconsistency otherwise. More specifically, using the ground-truth sample labels from both domains as references, `G-ABC` is more effective in propagating the semantic label information from the labeled source and target domains to the unlabeled target examples. Conversely, [180] cannot achieve this, as it only makes the unlabeled target features more compact in an unsupervised manner.

2. The pairwise label similarity we propose can be viewed as a more credible measure of pairwise relationships than the pairwise feature similarity used in [180].

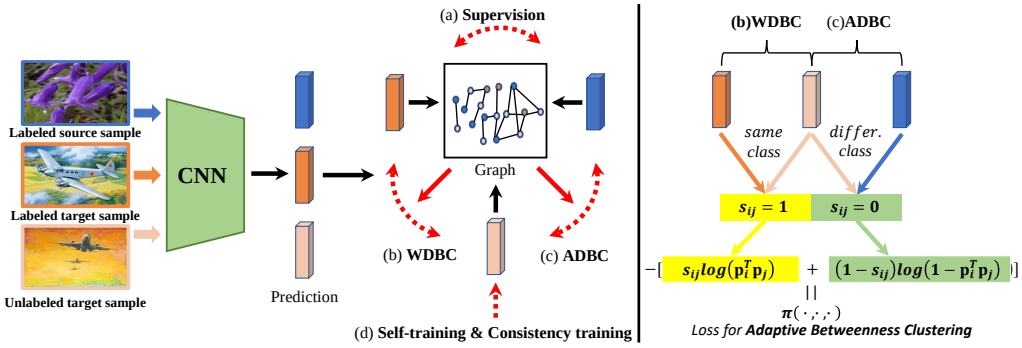


Figure 3.2: An overview of the proposed framework G-ABC and the training loss for Adaptive Betweenness Clustering. **Left:** (a) Supervision of labeled data from both source and target domains is applied to guarantee partially categorical domain alignment. (b) Within-domain betweenness clustering (WDBC) is used to determine the relationship between labeled and unlabeled target data. (c) Across-domain betweenness clustering (ADBC) is used to effectively align unlabeled target samples with the source domain. D) Auxiliary techniques for model optimization, including self-training, consistency training, etc. These four components together enable globally categorical domain alignment, progressively enhancing the model's performance, with (b) and (c) establishing reliable sample connectivity among training samples, represented by a graph. **Right:** Given a pairwise label s_{ij} between samples, the training loss of Adaptive Betweenness Clustering aims to bring samples from the same class closer together in the feature space when $s_{ij} = 1$, or to separate samples from different classes when $s_{ij} = 0$. This allows for semantic transfer from labeled source or target domains to unlabeled target samples. Note that the orange and light-orange samples belong to the same category, namely "plane", while the blue sample is from a different category, i.e. "flower".

This is especially true when the label information of labeled examples across domains can be trusted. Building upon this premise, we have taken two connectivity refinement strategies to build a reliable graph structure of pairwise relationships, which effectively mitigates the potential harm of noisy connectivity on model performance improvement. In contrast, the pairwise feature similarity introduced in [180] involves unsupervised sample matching, which makes it more challenging to accurately pair samples of the same class.

3.3 Methodology

In SSDA, we are given labeled samples from the source and target domains, denoted by $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ and $\mathcal{D}_l = \{(x_i^l, y_i^l)\}_{i=1}^{N_l}$, as well as unlabeled samples from the target domain, denoted as $\mathcal{D}_u = \{(x_i^u,)\}_{i=1}^{N_u}$, where N_s, N_l and N_u are the sizes of $\mathcal{D}_s$, $\mathcal{D}_l$ and $\mathcal{D}_u$, respectively. Our goal is to train an SSDA model using $\mathcal{D}_s$, $\mathcal{D}_l$ and $\mathcal{D}_u$, followed by evaluating the trained model on the target domain.

Like existing SSDA works, such as [129, 56, 59], we first parameterize the SSDA model by θ, made up of two components, namely a feature extractor F and a classifier G. The classifier G is an unbiased linear network with a normalization layer that maps

the extracted features from the feature extractor F into a spherical feature space. Here, the weight vectors of the classifier are denoted as $W = [w_1, w_2, \cdots, w_K]$, and these vectors can be regarded as the prototypes that represent K classes [129, 130]. Accordingly, samples with the same class label from the source or target domains are mapped nearby to the same class prototype in the feature space. As demonstrated in [129, 59], this has a considerable impact on minimizing the cross-domain feature variance of samples with the same class label. In a short, the normalized feature with temperature T of an input image x, $\mathbf{f} = \frac{1}{T}\frac{F(x)}{\|F(x)\|}$, is fed into the classifier G to obtain the probabilistic prediction as follows:

$$p(x) = \sigma(G(\mathbf{f})) = \sigma(W^T\mathbf{f}), \tag{3.1}$$

where $\sigma(\cdot)$ is a softmax function. $p(x)$ reflects the similarity scores, achieved by calculating the cosine distances, between the point x and the prototypes of distinct classes. For convenience, we often abbreviate $p(x)$ as $\mathbf{p}$, i.e., $\mathbf{p} = p(x)$.

Overview. In this paper, we propose Graph-based Adaptive Betweenness Clustering (`G-ABC`) to tackle semi-supervised domain adaptation. In detail, we first construct a graph to depict the pairwise associations between labeled samples from both domains and unlabeled examples from the target domain. Then, to degrade the noisy sample connectivity, we refine the original graph using two strategies: Confidence Uncertainty based Node Removal (CUNR) and Prediction Dissimilarity based Edge Pruning (PDEP). Afterwards, to associate complementary characteristics of the source and target labels with unlabeled target samples, we present Adaptive Betweenness Clustering to conduct semantic propagation, facilitating semantic alignment between domains. Finally, we leverage off-the-shelf and well-established techniques, such as pseudo-label selection, self-training [3, 64] and consistency training [165, 74, 160], to further optimize the model in order to achieve globally categorical domain alignment. An overview of the proposed method has been summarized in Figure 3.2.

3.3.1 Graph Construction

The goal of graph construction is to discover the sample connectivity of the training data with a graph $G = \langle V, E, A\rangle$. In particular, $V = \{v_i\}_{i=1}^{N_s+N_l+N_u}$ represents the collection of graph nodes consisting of labeled source instances from $\mathcal{D}_s$, labeled target samples from $\mathcal{D}_l$, and unlabeled target samples from $\mathcal{D}_u$, whereas E collects pairwise associations between a graph node of the unlabeled samples and the other node of the labeled examples. Then, the relationships between graph nodes given by the non-negative affinity matrix A can be calculated as follows,

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,N_l} & a_{1,N_l+1} & \cdots & a_{1,N_l+N_s} \\ a_{2,1} & \cdots & a_{2,N_l} & a_{2,N_l+1} & \cdots & a_{2,N_l+N_s} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{N_u,1} & \cdots & a_{N_u,N_l} & a_{N_u,N_l+1} & \cdots & a_{N_u,N_l+N_s} \end{bmatrix},$$

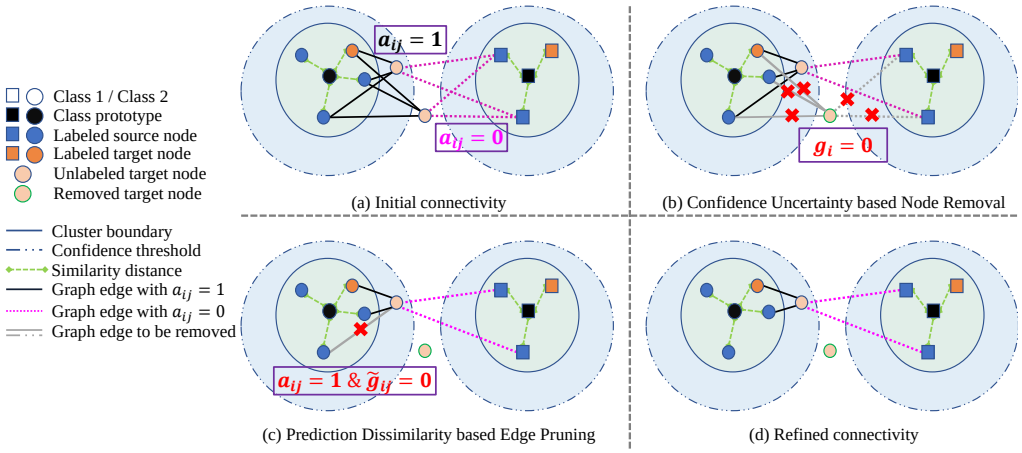


Figure 3.3: A diagram of G-ABC depicting graph construction and connectivity refinement. (a) demonstrates the initially constructed sample connectivity of the graph, while (d) presents the refined graph after the connectivity refinement process is performed. The connectivity refinement process is further illustrated by (b) and (c), which effectively eliminate noisy connectivity through the CUNR and PDEP strategies, respectively, resulting in a more reliable graph structure to represent the relationships between samples. The technical details of these four subdiagrams are as follows: (a) Using pairwise label similarities between samples, the initial connectivity between training examples in a graph is constructed; (b) Confidence Uncertainty based Node Removal (CUNR) reduces the connectivity towards unreliable unlabeled samples by removing nodes with low predicted confidence; (c) Prediction Dissimilarity based Edge Pruning (PDEP) further removes the connections between graph samples whose probabilistic prediction distributions are dissimilar; (d) A refined graph is obtained to properly capture the pairwise associations between samples.

where rows of this matrix denote the unlabeled samples of the target domain, while the first N_l columns and the last N_s columns of A refer to the labeled samples from the target and source domains, respectively. In addition, a_{ij} (i.e., the ij-th entry of A) is the weight of the edge connecting between graph nodes v_i and v_j, which encodes the mutual relationship of the sample pair.

Generally, the weight of a connectivity in a graph can be determined using cosine similarity between sample features [75, 6]. We propose in this paper that the pairwise associations can be established by comparing the given ground-truth labels of labeled samples to the model's predicted class labels of unlabeled samples; we refer to this as pairwise label similarity. Compared to pairwise feature similarity, pairwise label similarity can be viewed as a more credible measure of pairwise relationships, providing that the label information of labeled examples across domains is trustworthy. In addition, for the predicted label of the unlabeled target sample, we take measures to mitigate the noisy connectivity of the established graph (See Section 3.3.2). Afterwards, we can obtain the pairwise label similarity a_{ij} of an edge between graph nodes consisting of an unlabeled sample x_i (a.k.a, v_i), and a labeled sample x_j (a.k.a, v_j) affiliated

with its ground-truth class label y_j, as follows,

$$a_{ij} = 1\{\hat{y}_i = y_j\}, \tag{3.2}$$

where $\hat{y}_i = \arg\max_k(p(x_i)[k])$ is the predicted class label of the unlabeled target sample x_i, while $1\{\cdot\}$ is a binary indicator function. Notice that we illustrate the initial sample connectivity of the built graph in Figure 3.3(a).

3.3.2 Connectivity Refinement for Graph Unreliability

Once we obtain the initial graph, we introduce connectivity refinement to alleviate the unreliability of the graph. Due to high uncertainties caused by low predicted confidence, the graph nodes corresponding to unlabeled target samples are susceptible to receiving erroneous pseudo-labels, thereby resulting in noisy connectivity of the graph. In addition, as demonstrated in [151, 36, 91], labels and features should vary smoothly over the edges of the graph so as to well conduct semantic propagation.

To this end, features between nodes v_i and v_j of an edge should have a high degree of similarity when $a_{ij} = 1$, i.e., they should be neighbors in the spherical feature space. In this situation, the probabilistic prediction distributions of these two node samples predicted by the classifier ought to be similar. Consequently, there would be additional edges with noise in the graph, for which the probabilistic prediction distributions of both node samples from the model have a very low similarity.

To eliminate the connectivity with noise and unreliability in the graph G, we present two connectivity refinement strategies, namely Confidence Uncertainty based Node Removal and Prediction Dissimilarity based Edge Pruning, resulting in a refined graph that represents the pairwise relationship of the data structure with high reliability. An illustration is shown in Figure 3.3(b), (c), and (d).

Confidence Uncertainty based Node Removal (CUNR). This strategy degrades the connectivity towards unreliable unlabeled samples through the removal of unlabeled target instances with low-confidence predictions. We employ a sufficiently high confidence threshold $\tau \in [0, 1]$ to choose reliable candidate nodes from unlabeled target samples:

$$g_i = 1\{\max_k(p(x_i)[k]) > \tau\}, \tag{3.3}$$

where g_i is a binary indicator to preserve the node $v_i \in V$ corresponding to unlabeled samples on the target domain when $g_i = 1$ and to remove the node v_i when $g_i = 0$.

Prediction Dissimilarity based Edge Pruning (PDEP). As stated above, when $a_{ij} = 1$, the dissimilar prediction distributions between nodes of an edge in the graph might also lead to noisy connectivity in the graph, making a negative effect on semantic propagation. To remedy this, we first calculate the similarity score between the predicted label distributions of two nodes over an edge using dot product operation, and then

threshold the similarity scores with a scalar κ so as to obtain more credible graph connectivity. Therefore, we can formularize it as follows,

$$\tilde{g}_{ij} = \neg a_{ij} \vee 1\{\mathbf{p}_i^T \mathbf{p}_j > \kappa\}, \tag{3.4}$$

where $\mathbf{p}_i = p(x_i)$ and $\mathbf{p}_j = p(x_j)$. Besides, the binary indicator $\tilde{g}_{ij}$ serves to prune the graph edge connecting between nodes v_i (i.e., unlabeled target node x_i) and v_j (i.e., the sample x_j from labeled source or target data) when $\tilde{g}_{ij} = 0$; otherwise, the edge is preserved when $\tilde{g}_{ij} = 1$. In this manner, PDEP can effectively control the removal and preservation of the graph edge only when $a_{ij} = 1$. However, when $a_{ij} = 0$, PDEP becomes invalid, ensuring that the edge connecting nodes v_i and v_j in the graph will be consistently preserved.

Upon executing CUNR and PDEP, we can integrate Eq. (3.4) into Eq. (3.3) and then revise Eq. (3.3) for the rebuilt graph connectivity as follows,

$$g_i^j = g_i \cdot \tilde{g}_{ij}. \tag{3.5}$$

Here, the subscript i of g_i^j denotes the unlabeled target node x_i, and the superscript j indicates the other node x_j (from labeled source or target data) on the same edge as x_i.

Once we have the indicator g_i^j, we will be able to achieve more reliable connectivity between nodes in the graph G, hence improving the performance of semantic propagation.

3.3.3 Adaptive Betweenness Clustering

In this section, we conduct semantic transfer for categorical domain alignment using the updated graph that represents the reliable structure of the training data. Here, we propose a newly devised clustering algorithm for semantic propagation, called Adaptive Betweenness Clustering (ABC).

On the basis of the rebuilt graph, such an algorithm propagates and aggregates labels along graph edges. This allows the semantic label information of a labeled sample to be transferred to an unlabeled sample with clustering between samples. Using this approach in the spherical feature space provided by the prototypical classifier, the sample's predicted probability distributions indicate the cosine similarities between the feature and the prototypes for each category. Hence, this algorithm enforces a pair of samples with the same ground-truth (labeled) or predicted (unlabeled) class labels to have highly similar probability distributions. When the probability distribution of the latter is brought closer to that of the former, semantic propagation from labeled to unlabeled instances is achieved.

In specific, we first generate a sample pair from a labeled sample x_i and an unlabeled sample x_j by setting $s_{ij} = 1$ as a pairwise label if x_i and x_j belong to the same class, otherwise $s_{ij} = 0$ for different classes. Then, we adopt a binary cross-entropy

loss to draw samples from the same class closer together in the feature space while separating samples from other classes. Utilizing a pairwise label as a target, adaptive betweenness clustering thus can be computed with the following loss:

$$\pi(x_i, x_j, s_{ij}) = -[s_{ij} \log(\mathbf{p}_i^T \mathbf{p}_j) + (1 - s_{ij}) \log(1 - \mathbf{p}_i^T \mathbf{p}_j)], \tag{3.6}$$

where $\mathbf{p}_i = p(x_i)$ and $\mathbf{p}_j = p(x_j)$. As demonstrated in [126, 71], perturbations integrated into unlabeled target data can significantly improve the performance of the model; hence, we here augment unlabeled examples from the target domain for better propagation.

Due to domain shift in SSDA, the semantic information of labeled source samples, though is large in volume, is less correlated with unlabeled target examples, whereas the target label information of labeled samples has a greater correlation with unlabeled samples on the target domain, but is relatively scarce. Hence, to enable cross-domain semantic alignment, we propose across-domain betweenness clustering (ADBC) and within-domain betweenness clustering (WDBC) to propagate semantic label information from labeled instances on the source and target domains to unlabeled target data. With the distinct but complimentary characteristics of semantic information from source and target labels, semantic transfer can be conducted with the following losses with regard to unlabeled target samples:

$$\mathcal{L}^{abc} = \mathcal{L}^{wdbc} + \mathcal{L}^{adbc}, \tag{3.7}$$

$$\mathcal{L}^{wdbc} = \frac{1}{N_u} \sum_{i \in I} \frac{1}{N_l} \sum_{j \in P} g_i^j \cdot \pi(x_i, x_j, a_{ij}), \tag{3.8}$$

$$\mathcal{L}^{adbc} = \frac{1}{N_u} \sum_{i \in I} \frac{1}{N_s} \sum_{j' \in Q} g_i^{j'} \cdot \pi(x_i, x_{j'}, a_{ij'}), \tag{3.9}$$

where I, P and Q denote the collections of unlabeled samples from the target domain, and labeled ones from the target and source domains, respectively, whose sample indexes are denoted by $i \in \{1, 2, \cdots, N_u\}$, $j \in \{1, 2, \cdots, N_l\}$ and $j' \in \{N_l + 1, N_l + 2, \cdots, N_l + N_s\}$.

Here, to gain a better understanding of how Eqs. (3.2)-(3.6) work well in Eq. (3.8) (Eq. (3.9) follows the same rule as Eq. (3.8)), we provide the following detailed clarifications.

- According to Eq. (3.3), when $g_i > 0$ (i.e. $g_i = 1$), the model prediction remains confident. At this time, if $a_{ij} = 1$ (i.e. $s_{ij} = 1$ in Eq. (3.6)) and $\mathbf{p}_i^T \mathbf{p}_j > \kappa$ (in Eq. (3.4)), then $g_i^j > 0$ (i.e. $g_i^j = 1$ in Eq. (3.5)), and "$s_{ij} \log(\mathbf{p}_i^T \mathbf{p}_j)$" (i.e., the first positive term of Eq. (3.6)) contributes to the loss in Eq. (3.8). In this case, x_i and x_j should be close to each other in the feature space.
- Still under the condition that $g_i > 0$ (i.e. $g_i = 1$) in Eq. (3.3), when $a_{ij} = 0$, $\tilde{g}_{ij} = 1$ in Eq. (3.4)) always holds true, so $g_i^j > 0$ (i.e. $g_i^j = 1$). In this case,

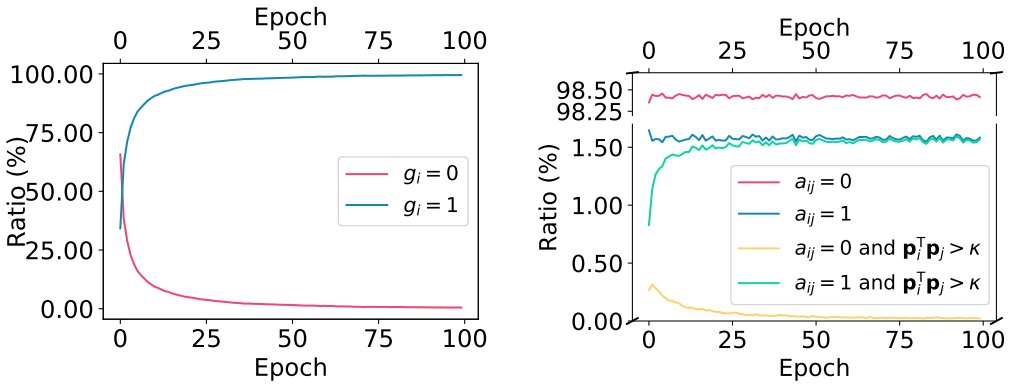


Figure 3.4: Empirical analysis of G-ABC to show the proportion of unlabeled target samples that meet the conditions proposed in the CUNR and PDEP strategies during the model training process. (a) and (b) are specified in the conditions introduced by Eqs. (3.2) and (3.3). For (b), it is important to note that due to the large number of classes in the dataset, only a small fraction of the sample pairs randomly generated in each epoch consist of an unlabeled target sample and a labeled source or target sample with the same label. Therefore, the proportion of unlabeled target domain samples satisfying "$a_{ij} = 1$" will be much smaller compared to the proportion of that satisfying "$a_{ij} = 0$". The experiment is performed with "R → P" on Office-Home using ResNet-34, under the 3-shot setup.

"$(1 - s_{ij}) \log(1 - \mathbf{p}_i^T \mathbf{p}_j)$", namely the second negative term in Eq. (3.6), contributes to the loss of Eq. (3.8), where x_i and x_j should be separated from each other.

- Finally, when $g_i < 0$ (i.e. $g_i = 0$), or when $a_{ij} = 0$ and $\mathbf{p}_i^T \mathbf{p}_j < \kappa$, then $g_j^i < 0$ (i.e. $g_i^j = 0$), and thus in Eq. (3.6), neither "$s_{ij} \log(\mathbf{p}_i^T \mathbf{p}_j)$" nor "$(1 - s_{ij}) \log(1 - \mathbf{p}_i^T \mathbf{p}_j)$" contributes to the loss for Eq. (3.8).

To provide a more intuitive demonstration, we conducted empirical analysis on the proportion of unlabeled target samples that meet the conditions specified in Eqs. (3.2) and (3.3) proposed in the CUNR and PDEP strategies during the model training process. The results are visualized in Figure 3.4. Figure 3.4(a) shows that as the model trains, the number of unlabeled samples satisfying the condition "$g_{ij} = 1$" gradually increases, indicating an increasing level of predicted confidence in the unlabeled target samples.

Moreover, in Figure 3.4(b), it can be observed that as the training progresses, the ratio of unlabeled target domain samples satisfying the condition "$a_{ij} = 1$ and $\mathbf{p}_i^T \mathbf{p}_j > \kappa$" approaches the ratio of that satisfying the condition "$a_{ij} = 1$". This demonstrates that the similarity between the prediction distributions of the unlabeled target samples and the labeled samples from the source or target domains increases gradually during the training process. On the contrary, the number of unlabeled target samples satisfying the condition "$a_{ij} = 0$ and $\mathbf{p}_i^T \mathbf{p}_j > \kappa$" decreases as the training progresses and may eventually approach zero.

These findings support our assumption that when two samples, one from the unlabeled target samples and the other from the labeled samples of the source or target domain, do not belong to the same class, their prediction distributions will be dissimilar.

3.3.4 Further Optimization of G-ABC based SSDA

After achieving cross-domain semantic alignment, we apply auxiliary techniques, such as pseudo-label selection, self-training, and consistency training, to further enhance the model training.

Pseudo-label Selection. Due to the scarcity of target labels, overfitting is likely to occur when the $\mathcal{L}^{wdbc}$ loss is applied. To mitigate this issue, we apply a pseudo-label selection strategy to unlabeled target samples and preserve pseudo-labels with high confidence to increase the number of target labels, hence enhancing the semantic label diversity on the target domain. In this work, pseudo-label selection employs the prediction capability of a model to generate artificial hard labels for a subset of unlabeled target samples; hence, a collection of pseudo-labeled target samples, denoted by $\mathcal{D}_{pu}$, can be obtained as follows,

$$\begin{aligned} \mathcal{D}_{pu} \leftarrow \{(x, \hat{y}) | \hat{y} = \arg\max_k (p(x)[k]), \\ \max(p(x)) > \tau', \forall x \in \mathcal{D}_u\}, \end{aligned} \tag{3.10}$$

where τ' denotes another higher confidence threshold than τ in Eq. (3.3). Noted that $\mathcal{D}_{pu}$ is only used in the $\mathcal{L}^{wdbc}$ loss, namely Eq. (3.8).

Self-training. According to [103, 64], we adopt self-training to boost the model's robustness against the selected samples in $\mathcal{D}_{pu}$. Specifically, we employ the progressive self-training technique described in [64], termed label consistency, in which the model is constrained to generate the same output when the selected images are augmented with slight perturbations. In practice, label consistency can be implemented through the following loss:

$$\mathcal{L}^{lab} = -\frac{1}{N_{pu}} \sum_{(x_i, y_i) \in \mathcal{D}_{pu}} p_y(y_i) \log(p(\mathbf{Aug}(x_i))), \tag{3.11}$$

where $N_{pu} = |\mathcal{D}_{pu}|$ indicates the sample size of $\mathcal{D}_{pu}$, $p_y(\cdot)$ represents the function to create a one-hot probability vector for a pseudo-label, and $\mathbf{Aug}(\cdot)$ denotes the function to perturb the input images.

Consistency Training. The pseudo-label selection mechanism fails to assign pseudo-labels to all unlabeled target samples. In accordance with [165], we can adequately leverage unlabeled target samples through consistency training, hence increasing the smoothness of the model. According to [165], we can achieve this by preserving the consistency of the model's output distributions between unlabeled target samples and

their perturbed counterparts using Kullback–Leibler (KL) divergence as follows,

$$\mathcal{L}^{con} = \frac{1}{N_u} \sum_{x_i \in \mathcal{D}_u} \tilde{p}(x_i) \log\left(\frac{\tilde{p}(x_i)}{p(\mathbf{Aug}(x_i))}\right), \tag{3.12}$$

$$\tilde{p}(x_i) = \text{Sharpen}(p(x_i)) = \frac{(p(x_i))^{\frac{1}{T'}}}{\sum_{k=1}^{K} (p(x_i)_k)^{\frac{1}{T'}}}, \tag{3.13}$$

where k indicates the k-th element of the target distribution vector $p(x_i^u)$ and T' is the temperature factor. It should be noticed that different from [165], we here use a "soft" distribution with a sharpening function Sharpen$(\cdot)$ proposed in [77, 8] to sharpen the observed probability distribution $\tilde{p}(x_i)$, thereby driving the model to generate lower-entropy predictions.

Training Objectives.The loss function for optimizing the model can be expressed as a combination of the cross-entropy loss $\mathcal{L}^{ce}$ over all labeled samples accessible across domains and additional losses as previously described, i.e.,

$$\mathcal{L}^{Overall} = \mathcal{L}^{ce} + \mathcal{L}^{lab} + \alpha\mathcal{L}^{con} + \beta\mathcal{L}^{abc}, \tag{3.14}$$

where α and β are scalar hyper-parameters for loss weights. The model is updated using stochastic gradient descent (SGD) for backpropagation.

3.4 Experiments

3.4.1 Experimental Setups

Datasets. The proposed `G-ABC` approach is evaluated on three widely used benchmark datasets, including DomainNet [118], Office-Home [149] and Office-31 [128]. To conduct a fair comparison, we follow all configurations of adaptation scenarios on different datasets as considered in [129, 122, 59, 71], while each category has one-shot or three-shot samples with labels available in the target domain during training.

DomainNet, consisting of 345 classes and six domains, is a large-scale benchmark dataset designed to evaluate multi-source domain adaption approaches. Following [129], we use a subset of the dataset proposed in [118] as one of our evaluation benchmarks. Similar to MME [129], we only chose 4 domains: Real (**R**), Clipart (**C**), Painting (**P**), and Sketch (**S**) (each comprises 126 categories of images), as other domains and categories may contain samples with excessive noise. In accordance with [129, 122, 59, 71], we construct experiments on seven adaptation scenarios employing these four domains.

Office-Home is a notable SSDA benchmark dataset containing numerous challenging adaptation scenarios. There are 65 classes in this collection, and the available domains are Real (**R**), Clipart (**C**), Art (**A**), and Product (**P**). To achieve a fair comparison, we apply 12 adaptation scenarios to this dataset compared to previous SSDA methods, including [129, 122, 59, 71, 176].

Office-31 consists of 31 object categories organized into three domains: Amazon (**A**), DSLR (**D**), and Webcam (**W**). These categories contain objects frequently seen in offices, such as keyboards, file cabinets, and laptops. Following previous SSDA efforts [129, 122, 59], we select Amazon as the source domain since only Amazon is a large domain with sufficient samples for each class, whereas Webcam and DSLR do not. Therefore, we only consider two adaptation scenarios on this relatively smaller SSDA dataset benchmark, i.e., "W → A" and "D → A".

Implementation. To be fair, we adhere to the conventional SSDA task configurations from earlier research [129, 122, 59]. Specifically, we first select AlexNet [61] and ResNet-34 [43] with pre-trained weights on ImageNet [61] as the backbone networks for all our experiments. However, the last layer of each backbone is replaced with a prototypical classifier based on cosine similarity, followed by an unbiased linear neural network that takes normalized features from the feature extractor as inputs. Here, we optimize the entire model using mini-batch stochastic gradient descent (SGD) with momentum. In addition, throughout each iteration, we first train the model under supervision on all labeled samples from both domains to generate representative prototypes of each class, followed by the proposed ADBC and WDBC stages to further enhance the model. Moreover, we use RandAugment [20] and Cutout [23] to generate perturbations for unlabeled target data used in Eq. (3.8), (3.9), (3.11), and (3.12).

For fair comparisons, the majority of the remaining experimental settings in our proposed method are identical to previous SSDA efforts like [129, 122, 59]. Similar to [129, 59], we implement all experiments on the PyTorch[1] platform. Besides, during each iteration, we randomly select four mini-batches from $\mathcal{D}_s$, $\mathcal{D}_l$, $\mathcal{D}_{pu}$, and $\mathcal{D}_u$, with batch sizes of 32, 32, 32, and 64 for AlexNet or 24, 24, 24, and 48 for ResNet-34. In addition, we employ the same learning rate schedule as [31], with the learning rate ξ_t at the t-th iteration set as follows:

$$\xi_t = \frac{\xi_0}{(1 + 0.0001 \times t)^{0.75}}, \tag{3.15}$$

where ξ_0 represents the initial learning rate. To balance numerous loss terms, we set α and β in Eq. (3.14) to 0.03 and 25.0. Then, based on [64, 71], we set the confidence threshold to $\tau = 0.95$ and $\tau' = 0.975$. In addition, we set the similarity threshold κ to 0.20. The value of temperatures involved in the construction of the model architecture and the sharpening function in Eq. (3.13) are set to 0.05 and 0.85, respectively. Due to the distinctness of each dataset and adaptation scenario, we set the total number of training epochs $\mathcal{T}$ to varying values. Note that $\mathcal{T} = 100$ is a common value setting for a variety of application scenarios.

To choose the hyper-parameters, such as α, β, τ and κ, similar to **MME** [129], we selected three labeled examples as the validation set for the target domain and utilized these validation examples to choose the value choice of these hyper-parameters when

[1] https://pytorch.org/

Table 3.1: Comparison results (%) of the proposed G-ABC and the state-of-the-art SSDA algorithms on DomainNet under the settings of 1-shot and 3-shot with both AlexNet (ANet) and ResNet-34 (RN-34) backbones. The top best methods are in **bold**. (Mean accuracy and standard variance over 3 trials)

Net	Method	R→C		R→P		P→C		C→S		S→P		R→S		P→R		Mean	
		1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
ANet	S+T [129]	43.3	47.1	42.4	45.0	40.1	44.9	33.6	36.4	35.7	38.4	29.1	33.3	55.8	58.7	40.0	43.4
	DANN [129]	43.3	46.1	41.6	43.8	39.1	41.0	35.9	36.5	36.9	38.9	32.5	33.4	53.5	57.3	40.4	42.4
	MME [129]	48.9	55.6	48.0	49.0	46.7	51.7	36.3	39.4	39.4	43.0	33.3	37.9	56.8	60.7	44.2	48.2
	Meta-MME [69]	-	56.4	-	50.2	-	51.9	-	39.6	-	43.7	-	38.7	-	60.7	-	48.8
	BiAT [56]	54.2	58.6	49.2	50.6	44.0	52.0	37.7	41.9	39.6	42.1	37.2	42.0	56.9	58.8	45.5	49.4
	APE [59]	47.7	54.6	49.0	50.5	46.9	52.1	38.5	42.6	38.5	42.2	33.8	38.7	57.5	61.4	44.6	48.9
	PAC [103]	55.4	61.7	54.6	56.9	47.0	59.8	46.9	52.9	38.6	43.9	38.7	48.2	56.7	59.7	48.3	54.7
	Relaxd-cGAN [94]	-	56.8	-	51.8	-	52.0	-	44.1	-	44.2	-	42.8	-	61.1	-	50.5
	ECACL-T [81]	56.8	62.9	54.8	58.9	**56.3**	60.5	46.6	51.0	**54.6**	51.2	45.4	**48.9**	62.8	67.4	53.4	57.7
	ECACL-P [81]	55.8	62.6	54.0	59.0	56.1	60.5	46.1	50.6	**54.6**	50.3	45.0	48.4	62.3	67.4	52.8	57.6
	S^3D [184]	53.5	56.5	51.8	52.2	49.1	53.9	40.1	44.4	44.9	48.7	39.9	39.2	61.7	65.4	48.7	51.5
	CLDA [133]	56.3	59.9	56.0	57.2	50.8	54.6	42.5	47.3	46.8	51.4	38.0	42.7	64.4	67.0	50.7	54.3
	CDAC [71]	56.9	61.4	55.9	57.5	51.6	58.9	44.8	50.7	48.1	51.7	44.1	46.7	63.8	66.8	52.1	56.2
	G-ABC (Ours)	**60.17±0.78**	**63.89±0.41**	**57.44±0.66**	**59.73±0.33**	55.98±0.93	**64.03±0.34**	**48.75±0.50**	**53.42±0.57**	54.11±0.61	**56.36±0.54**	**47.09±0.91**	48.17±0.35	**67.84±0.79**	**70.78±0.36**	**55.92**	**59.68**
RN-34	S+T [129]	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
	DANN [129]	58.2	59.8	61.4	62.8	56.3	59.6	52.8	55.4	57.4	59.9	52.2	54.9	70.3	72.2	58.4	60.7
	MME [129]	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
	UODA [122]	72.7	75.4	70.3	71.5	69.8	73.2	60.5	64.1	66.4	69.4	62.7	64.2	77.3	80.8	68.5	71.2
	Meta-MME [69]	-	73.5	-	70.3	-	72.8	-	62.8	-	68.0	-	63.8	-	79.2	-	70.1
	BiAT [56]	73.0	74.9	68.0	68.8	71.6	74.6	57.9	61.5	63.9	67.5	58.5	62.1	77.0	78.6	67.1	69.7
	APE [59]	70.4	76.6	70.8	72.1	72.9	76.7	56.7	63.1	64.5	66.1	63.0	67.8	76.6	79.4	67.6	71.7
	ELP [51]	72.8	74.9	70.8	72.1	72.0	74.4	59.6	63.3	66.7	69.7	63.3	64.9	77.8	81.0	69.0	71.6
	PAC [103]	74.9	78.6	73.0	74.3	72.6	76.0	65.8	69.6	67.9	69.4	68.7	70.2	76.7	79.3	71.4	73.9
	DECOTA[176]	79.1	80.4	74.9	75.2	76.9	78.7	65.1	68.6	72.0	72.7	69.7	71.9	79.6	81.5	73.9	75.6
	ECACL-T [81]	73.5	76.4	72.8	74.3	72.8	75.9	65.1	65.3	70.3	72.2	64.8	68.6	78.3	79.7	71.1	73.2
	ECACL-P [81]	75.3	79.0	74.1	77.3	75.3	79.4	65.0	70.6	72.1	74.6	68.1	71.6	79.7	82.4	72.8	76.4
	S^3D [184]	73.3	75.9	68.9	72.1	73.4	75.1	60.8	64.4	68.2	70.0	65.1	66.7	79.5	80.3	69.9	72.1
	UODAv2 [124]	77.0	79.4	75.4	76.7	75.5	78.3	66.5	70.2	72.1	74.2	70.9	72.1	79.7	82.3	73.9	76.2
	MCL [171]	77.4	79.4	74.6	76.3	75.5	78.8	66.4	70.9	74.0	74.7	70.7	72.3	82.0	83.3	74.4	76.5
	CLDA [133]	76.1	77.7	75.1	75.7	71.0	76.4	63.7	69.7	70.2	73.7	67.1	71.1	80.1	82.9	71.9	75.3
	CDAC [71]	77.4	79.6	74.2	75.1	75.5	79.3	67.6	69.9	71.0	73.4	69.2	72.5	80.4	81.9	73.6	76.0
	G-ABC (Ours)	**80.74±0.41**	**82.07±0.21**	**76.84±0.63**	**76.72±0.32**	**79.26±0.19**	**81.57±0.41**	**71.95±0.47**	**73.68±0.38**	**75.04±0.54**	**76.27±0.20**	**73.21±0.32**	**74.28±0.09**	**83.42±0.61**	**83.87±0.28**	**77.47**	**78.23**

the validation accuracy was at its highest. Also, during this process, we froze the other hyper-parameters while conducting experiments with a specific one.

Class-wise Similarity Score. To assess the effectiveness of Adaptive Betweenness Clustering, we define a Class-wise Similarity Score (CSS) to measure the average prediction similarity between two classes, where one class c originates from the unlabeled target domain and the other class c' comes from labeled source and target domains. In this way, we use $s(c, c')$ to define the CSS between the class c and c', and more specifically, $s(c, c')$ can be formulated as follows,

$$s(c, c') = \frac{1}{N_u^c} \sum_{i \in I^c} \frac{1}{N_l^{c'} + N_s^{c'}} \sum_{j \in P^{c'} \cup Q^{c'}} \mathbf{p}_i^T \mathbf{p}_j, \tag{3.16}$$

where I^c, $P^{c'}$ and $Q^{c'}$ denote the collections of unlabeled target samples of class c, labeled target samples of the class c', and labeled source samples of class c', respectively, each containing instances with sizes of N_u^c, $N_l^{c'}$ and $N_s^{c'}$. In general, the larger the average prediction similarity between classes c and c' is, the greater the CSS $s(c, c')$ is. At this point, a higher CSS score indicates that there is a greater similarity in predictions between unlabeled samples from class c in the target domain and the labeled source or target data from class c'. This implicitly suggests that these samples are close to each other in the feature space.

Baselines. We compare the classification performance of our proposed `G-ABC` algorithm to that of previous state-of-the-art SSDA algorithms, including **S+T** [129], **DANN** [129], **CDAN** [93], **MME** [129], **UODA** [122], **Meta-MME** [69], **BiAT** [56], **APE** [59], **ELP** [51], **PAC** [103], **Relaxed-cGAN** [94], **DECOTA** [176], **ECACL-T** [81], **ECACL-P** [81], **S^3D** [184], **UODAv2** [124], **MCL** [171], **CLDA** [133] and **CDAC** [71]. Note that S+T refers to an approach that trains the adaptation model solely with supervision on labeled samples from both domains, whilst DANN refers to the method presented in [32], but

Table 3.2: Comparison results (%) of G-ABC and the state-of-the-art SSDA algorithms on Office-Home under the setting of 3-shot with both AlexNet (ANet) and ResNet-34 (RN-34) backbones. The top best methods are in **bold**. (Mean accuracy and standard variance over 3 trials)

Net	Method	R→C	R→P	R→A	P→R	P→C	P→A	A→P	A→C	A→R	C→R	C→A	C→P	Mean
ANet	S+T [129]	44.6	66.7	47.7	57.8	44.4	36.1	57.6	38.8	57	54.3	37.5	57.9	50.0
	DANN [129]	47.2	66.7	46.6	58.1	44.4	36.1	57.2	39.8	56.6	54.3	38.6	57.9	50.3
	MME [129]	51.2	73.0	50.3	61.6	47.2	40.7	63.9	43.8	61.4	59.9	44.7	64.7	55.2
	Meta-MME [69]	50.3	-	-	-	48.3	40.3	-	44.5	-	-	44.5	-	-
	BiAT [56]	-	-	-	-	-	-	-	-	-	-	-	-	56.4
	APE [59]	51.9	74.6	51.2	61.6	47.9	42.1	65.5	44.5	60.9	58.1	44.3	64.8	55.6
	PAC [103]	**58.9**	72.4	47.5	61.9	**53.2**	39.6	63.8	**49.9**	60.0	54.5	36.3	64.8	55.2
	CLDA [133]	51.5	74.1	**54.3**	**67**	47.9	47	65.8	47.4	**66.6**	**64.1**	**46.8**	67.5	**58.3**
	CDAC [71]	54.9	75.8	51.8	64.3	51.3	**43.6**	65.1	47.5	63.1	63.0	44.9	65.6	56.8
	G-ABC (Ours)	55.12±0.71	**76.21±0.59**	53.20±0.45	64.59±0.43	50.45±0.70	41.76±0.51	**67.41±0.67**	47.51±0.87	62.07±0.93	63.52±0.97	42.72±0.81	**68.23±0.46**	57.73
RN-34	S+T [129]	55.7	80.8	67.8	73.1	53.8	63.5	73.1	54.0	74.2	68.3	57.6	72.3	66.2
	DANN [129]	57.3	75.5	65.2	69.2	51.8	56.6	68.3	54.7	73.8	67.1	55.1	67.5	63.5
	MME [129]	64.6	85.5	71.3	80.1	64.6	65.5	79	63.6	79.7	76.6	67.2	79.3	73.1
	Meta-MME [69]	65.2	-	-	-	64.5	66.7	-	63.3	-	-	67.5	-	-
	APE [59]	66.4	86.2	73.4	82.0	65.2	66.1	81.1	63.9	80.2	76.8	66.6	79.9	74.0
	Relaxed-cGAN [94]	68.4	85.5	73.8	81.2	68.1	67.9	80.1	64.3	80.1	77.5	66.3	78.3	74.2
	DECOTA [176]	**70.4**	87.7	74.0	82.1	68.0	69.9	81.8	64	80.5	79	68.0	83.2	75.7
	CLDA [133]	66.0	87.6	**76.7**	82.2	63.9	**72.4**	81.4	63.4	**81.3**	80.3	**70.5**	80.9	75.5
	CDAC [71]	67.8	85.6	72.2	81.9	67	67.5	80.3	65.9	80.6	80.2	67.4	81.4	74.2
	G-ABC (Ours)	70.02±0.18	**88.09±0.27**	75.96±0.48	**82.81±0.11**	**69.27±0.53**	70.54±0.42	**83.78±0.31**	**67.24±0.14**	80.37±0.10	**80.18±0.44**	69.22±0.25	**83.89±0.62**	**77.19**

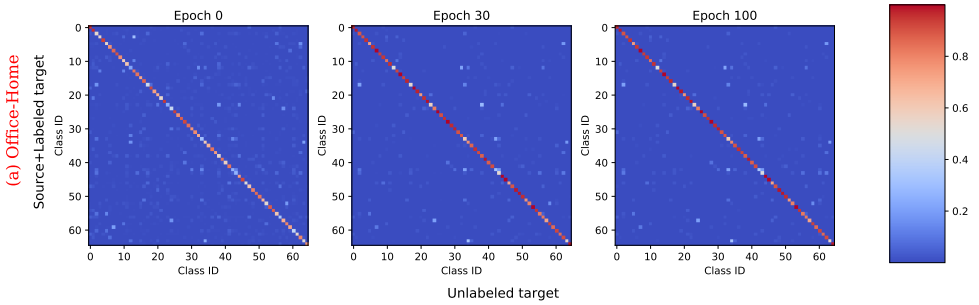
Table 3.3: Comparison results (%) of G-ABC and the state-of-the-art SSDA algorithms on Office-31 under the settings of 1-shot and 3-shot with the AlexNet backbone. The top best methods are in **bold**. (Mean accuracy and standard variance over 3 trials)

Method	W→A		D→A		Mean	
	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
S+T [129]	50.4	61.2	50.0	62.4	50.2	61.8
DANN [129]	57.0	64.4	54.5	65.2	55.8	64.8
MME [129]	57.2	67.3	55.8	67.8	56.5	67.6
BiAT [56]	57.9	68.2	54.6	68.5	56.3	68.4
APE [59]	-	67.6	-	69.0	-	68.3
PAC [103]	53.6	65.1	54.7	66.3	54.2	65.7
CLDA [133]	64.6	70.5	62.7	72.5	63.6	71.5
CDAC [71]	63.4	70.1	62.8	70.0	63.1	70.0
G-ABC (Ours)	**67.9±1.26**	**70.97±0.48**	**65.73±1.03**	**73.06±0.35**	**66.81**	**72.02**

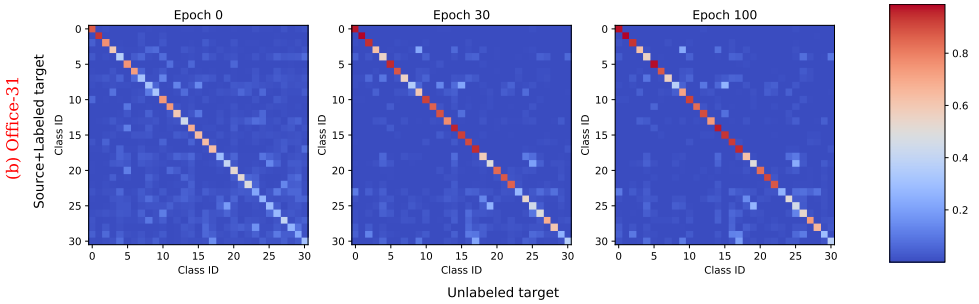
additionally applies a standard cross-entropy loss on a few labeled samples in the target domain.

3.4.2 Comparisons with State-of-the-Arts

On DomainNet. In order to highlight the advantages of the proposed algorithm, we compare our `G-ABC` strategy to numerous existing alternatives on the DomainNet benchmark. Table 3.1 presents the results of this dataset benchmark utilizing 1-shot and 3-shot settings with AlexNet and ResNet-34 as the corresponding backbone networks. As demonstrated, our proposed `G-ABC` method achieves more average performance gains than all existing approaches in the majority of DomainNet adaptation cases. Specifically, `G-ABC` improves the prior best-performing SSDA algorithm, i.e., **ECACL-T** and **DECOTA**, by mean accuracy margins of 2.3% and 3.1% while employing AlexNet and ResNet-34 as the backbones, respectively, for all adaptation scenarios under the 1-shot setting. In addition, the proposed method outperforms competing approaches in most of the adaptation scenarios defined on DomainNet with a 3-shot setting by outperforming the best available results (accuracy of 57.7% and 76.5% in **ECACL-T** and **MCL**) by 1.88% and 2.08% on average, when AlexNet and ResNet-34 serve as the backbone networks, respectively. These results demonstrate the effectiveness of our algorithm in dealing with SSDA tasks on DomainNet.



(a) "R → P" on Office-Home using ResNet-34



(b) "D→A" on Office-31 using AlexNet

Figure 3.5: The evaluation of Adaptive Betweenness Clustering involves analyzing the confusion matrices for each epoch on every dataset. Each element in these matrices is associated with a Class-wise Similarity Score, denoted as $s(c, c')$. This score, defined by Eq. (3.16) in Section 3.4.1, quantifies the similarity between two classes, c and c'. In this context, class c refers to the class whose samples are the unlabeled target data, while class c' includes classes from both the labeled source and target domains. A higher $s(c, c')$ score in each element suggests a greater similarity in predictions between the unlabeled samples from class c in the target domain and the labeled source or target data from class c'. The experiments are performed with (a): "R → P" on Office-Home using ResNet-34, and (b): "D→A" on Office-31 using AlexNet, respectively, both under the 3-shot setup.

On Office-Home. To validate the feasibility of the proposed `G-ABC` algorithm in SSDA, we also compare the results of our method to those of earlier methods on Office-Home. Similar to prior baselines [129, 122, 59, 176, 71], we conduct experiments on this dataset under the 3-shot setting, AlexNet and ResNet-34 as the backbones, and all 12 adaptation scenarios for Office-Home. Table 3.2 illustrates the classification accuracy of each adaptation scenario and the average performance of the proposed `G-ABC` algorithm on Office-Home, respectively. As demonstrated, the proposed method achieves the best average classification performance when utilizing the ResNet-34 backbone, and the accuracy surpasses the best baseline **DECOTA** by significant margins of 1.49%.

On Office-31. Aiming at further confirming the efficacy of the proposed `G-ABC` method, we conduct experiments comparing to the existing state-of-the-art approaches using the smallest benchmark dataset, namely Office-31. In order to retain consistency with prior approaches and to assure a fair comparison, we only use AlexNet as the backbone of this work. Table 3.3 shows that our method obtains the highest classification

Table 3.4: Ablation study results (%) of G-ABC on DomainNet under the setting of 3-shot with the ResNet-34 backbone. † denotes that the adaptation model is trained by $\mathcal{L}^{adbc}$ or $\mathcal{L}^{wdbc}$ loss without incorporating perturbations into the samples. ‡ denotes that the pseudo-labeled target samples are not used in the $\mathcal{L}^{wdbc}$ loss. Moreover, ♣ represents the removal of the first positive term, namely "$s_{ij}\log(\mathbf{p}_i^T\mathbf{p}_j)$", from Eq. (3.6) during training. Similarly, ♠ denotes the exclusion of the second negative term, namely "$(1-s_{ij})\log(1-\mathbf{p}_i^T\mathbf{p}_j)$", from Eq. (3.6) when model training goes on.

M-(#)	$\mathcal{L}^{ce}$	$\mathcal{L}^{adbc}$	$\mathcal{L}^{wdbc}$	$\mathcal{L}^{lab}$	$\mathcal{L}^{con}$	R→C	C→S	S→P	R→S	Mean
1	✓	-	-	-	-	60.0	55.0	59.5	50.1	56.2
2	✓	✓	-	-	-	78.0	67.6	71.0	69.5	71.5
3	✓	-	✓	-	-	78.1	68.8	70.8	70.8	72.1
4	✓	✓	✓	-	-	80.3	70.0	73.2	72.5	74.0
5	✓	-	-	✓	-	77.2	70.2	74.0	69.9	72.8
6	✓	-	-	-	✓	72.6	66.1	69.1	65.5	68.4
7	✓	-	-	✓	✓	78.3	71.5	74.3	71.0	73.8
8	✓	✓	-	✓	-	80.5	72.4	75.3	72.7	75.2
9	✓	-	✓	✓	-	81.0	73.1	76.0	73.1	75.8
10	✓	✓	✓	✓	-	81.6	73.2	75.9	73.8	76.1
11	✓	✓	✓	✓	✓	82.2	73.4	76.3	74.3	76.5
12	✓	†	†	✓	✓	81.5	72.1	75.3	72.5	75.4
13	✓	†	†	-	-	78.6	68.5	72.9	70.4	72.6
14	✓	✓	‡	✓	✓	82.0	72.9	75.7	73.2	75.9
15	✓	♣	♣	✓	✓	81.3	72.8	75.0	73.1	75.6
16	✓	♠	♠	✓	✓	79.8	72.0	75.5	72.4	74.9

performance in both "W → A" and "D → A" cases, with an average accuracy of 66.81% (+3.21%) under the 1-shot setting and 72.02% (+0.52%) under the 3-shot setting. In other words, our strategy outperforms the existing best-performing baseline, **CLDA**, by significant average accuracy margins in all adaptation scenarios, indicating the improved effectiveness of the proposed method on this dataset.

Discussion. It appears that superior performance gains have been observed on the DomainNet dataset compared to Office-Home. This is because Office-Home is a relatively simpler SSDA benchmark dataset. As shown in Table 3.2, it is evident that prior state-of-the-art (SOTA) methods have reached their performance limits on this dataset. Similar to these approaches, this saturation in performance makes it challenging for our method to achieve significant improvements compared to existing SOTA methods. In contrast, DomainNet has a larger domain shift, which poses challenges for domain adaptation methods and offers more room for improvement. As illustrated in Table 3.1, our method demonstrates more notable gains on DomainNet, as it is designed to effectively handle such complex domain shifts.

3.4.3 Ablation Analysis

We conduct extensive experiments to individually confirm the efficacy of each component of our proposed G-ABC approach. Specifically, Table 3.4 shows the ablation study results, where all experiments are performed in four adaptation scenarios on DomainNet using ResNet-34 as the backbone under the 3-shot setup. Furthermore, Figure 3.5 illustrates the evaluation of Adaptive Betweenness Clustering in both ADBC

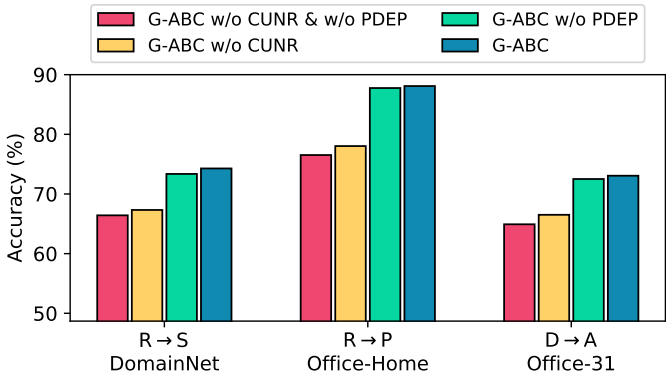


Figure 3.6: The impact of removing CUNR and PDEP on performance during graph construction. The experiments are performed on adaptation scenarios of "R→S" on DomainNet, "R→P" on Office-Home, and "D→A" on Office-31, respectively. All of them are conducted under a 3-shot setup, with the first two using ResNet-34 as network backbones, while the latter uses AlexNet.

and WDBC, while Fig 3.6 demonstrates the impact of removing CUNR and PDEP on graph construction and their influence on the model performance.

Effectiveness of ADBC and WDBC. To determine the effectiveness of $\mathcal{L}^{adbc}$ and $\mathcal{L}^{wdbc}$ proposed in our method, we first train the model using only labeled samples from both domains, serving as the baseline being depicted in row M-(1) of Table 3.4. According to Table 3.4, training the model with both ADBC and WDBC delivers greater classification performance gains than training the model with simply one of both. It can be observed that row M-(4) in Table 3.4 increases the baseline by an average of 17.9%, while the accuracy rates in row M-(2) and row M-(3) can only exceed the baseline by 15.4% and 16.0%, respectively, thereby confirming the validity of the ADBC and WDBC stages. In addition, when row M-(5) of Table 3.4 is considered as another baseline, a similar situation can be observed when contrasting among row M-(8), row M-(9) and row M-(10).

Effectiveness of Adaptive Betweenness Clustering. In order to further understand the role of the proposed clustering method, Adaptive Betweenness Clustering (ABC), in ADBC and WDBC, we employed the Class-wise Similarity Score (CSS), which is illustrated in detail in Section 3.4.1. The CSS is used to measure the similarity in predictions between unlabeled target samples and labeled source or target domains. As depicted in the heatmaps of Figure 3.5, the CSS scores between the same classes progressively increase, while those between different classes decrease throughout the model training process. This observation indicates that unlabeled target samples have a tendency to cluster together with labeled source and target data from the same class in the feature space, while samples from different classes become more distant from each other. This demonstrates the efficacy of Adaptive Betweenness Clustering in facilitating semantic propagation, thereby raising the performance of ADBC and WDBC.

To further validate the effectiveness of the Adaptive Betweenness Clustering loss,

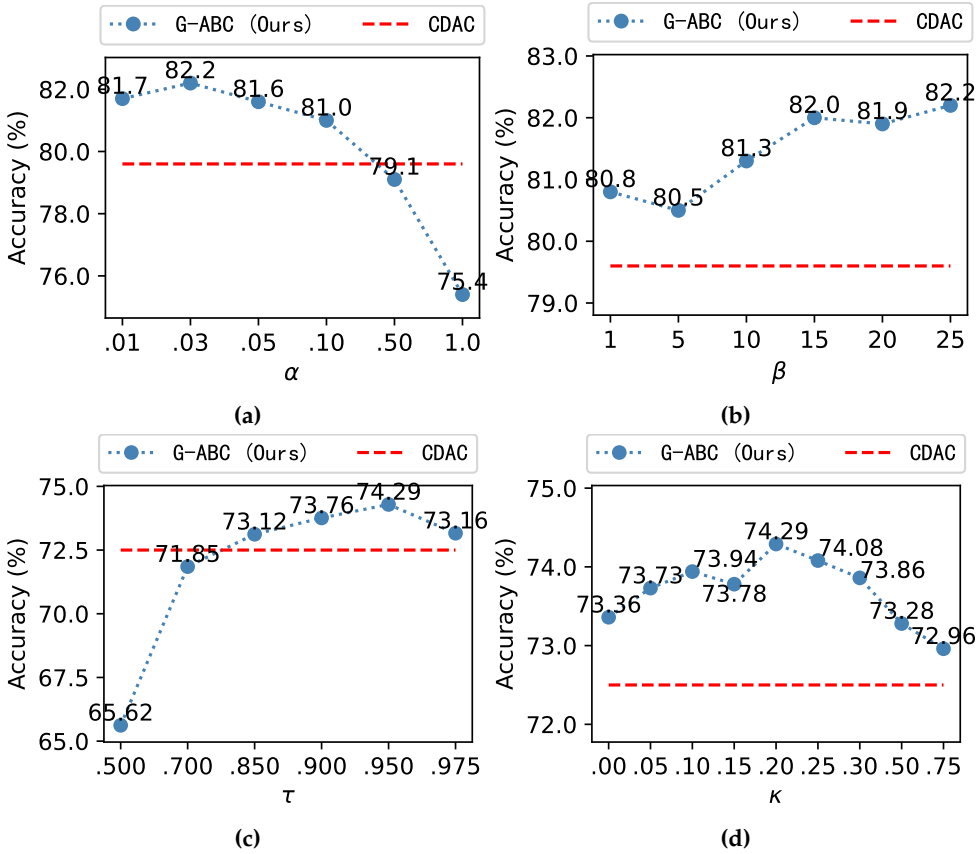
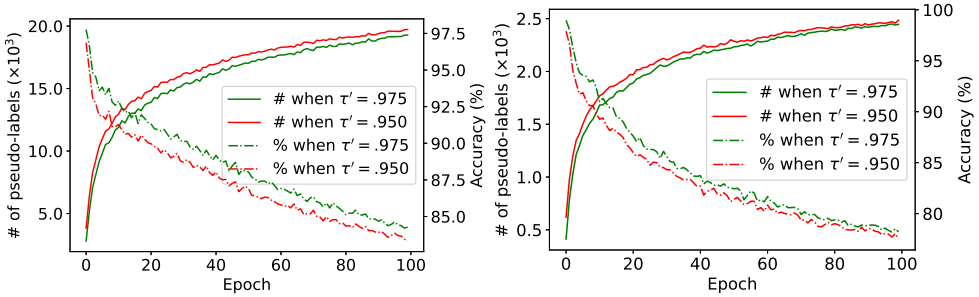


Figure 3.7: Sensitivity with respect to hyper-parameters α, β, τ and κ. The experiments are performed on DomainNet under the setting of the 3-shot using the ResNet-34 backbone, where (a) and (b) are with "R→C", and (c) and (d) are with "R→S". "**CDAC**" [71] is the best-performing baseline method in both adaptation scenarios. As illustrated, our method is not highly sensitive to changes in the hyper-parameters α, β, τ and κ. This is because, over a wide range, our `G-ABC` approach outperforms the baseline method "**CDAC**" significantly for all four hyper-parameters.

we also performed validation experiments on the two terms of the loss presented in Eq. (3.6), namely the first positive term "$s_{ij} \log(\mathbf{p}_i^T \mathbf{p}_j)$" and the second negative term "$(1 - s_{ij}) \log(1 - \mathbf{p}_i^T \mathbf{p}_j)$". According to Table 3.4, it can be observed that row M-(15), M-(16), and M-(7) correspond to the removal of individual terms or both terms from Eq. (3.6). By comparing these results with the classification performance of the full model indicated by row M-(11), it can be seen that removing either one or both terms leads to a decrease in model performance, with a more pronounced effect when both terms are removed. This finding indirectly verifies the effectiveness of the adaptive betweenness clustering loss.

Effectiveness of CUNR and PDEP. To confirm the impact of CUNR and PDEP during graph construction, we present the results of removing each of them individually, as well as both, on three SSDA benchmarks. As shown in Figure 3.6, the comparison between the full model `G-ABC` and its variants "G-ABC w/o CUNR" or "G-ABC w/o



(a) "R→S" on DomainNet **(b)** "D→A" on Office-31

Figure 3.8: The evolution of the numbers (#) and the accuracy (%) of the pseudo-labels over epochs while varying the confidence threshold τ'. The experiments are performed with (a): "R→S" on DomainNet using ResNet-34, and (b): "D→A" on Office-31 using AlexNet, both under the 3-shot setup.

PDEP" demonstrates that removing either CUNR or PDEP leads to a significant decline in the model's overall performance.

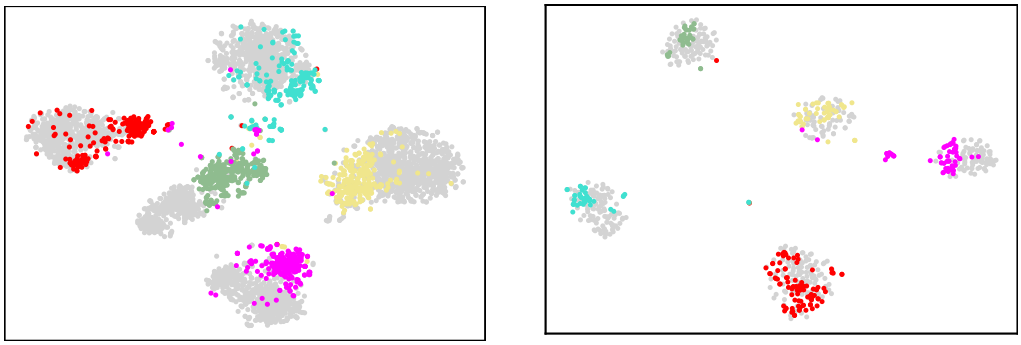
Moreover, when both are eliminated, referred to as "G-ABC w/o CUNR & w/o PDEP", the model's performance reaches its lowest point. This demonstrates that CUNR or PDEP effectively eliminates noisy connectivity while constructing a reliable graph structure, thereby enhancing the model's performance. In particular, CUNR provides a greater performance improvement than PDEP. This is because unlabeled target samples with lower confidence, which would be removed by CUNR during training, not only negatively impact samples of the same class but also affect samples from different classes, resulting in more substantial harm to the model's performance.

Effectiveness of Self-training. Examining the necessity of $\mathcal{L}^{lab}$, we should use experiments in row M-(4) of Table 3.4 in which the model is trained with $\mathcal{L}^{ce}$, $\mathcal{L}^{adbc}$ and $\mathcal{L}^{wdbc}$ as the baseline. As shown in Table 3.4, the average performance of row M-(10) with additional $\mathcal{L}^{lab}$ loss is 2.1% more than the baseline, indicating the necessity of this component for our proposed `G-ABC` approach.

Effectiveness of Consistency Training. Table 3.4 indicates the effectiveness of consistency training as well. Comparing row M-(10) (or row M-(6)) with row M-(11) (or row M-(7)), it is evident that consistency training for all unlabeled target data is beneficial.

Effectiveness of Pseudo-label Selection. In order to explore the impact of pseudo-label selection, we omit the pseudo-labeled target samples applying to Eq. (3.8). Table 3.4 demonstrates that in the comparison between rows M-(11) and M-(14), the average accuracy of row M-(14) is 0.6% less than that of row M-(11), revealing that pseudo-label selection is also effective for enhancing the performance of the model.

Effectiveness of Sample Perturbation. We propose to introduce perturbations into unlabeled target samples on both ADBC and WDBC. According to Table 3.4, by comparing row M-(4) and row M-(11) with row M-(13) and row M-(12), respectively, we



(a) "R → S" on DomainNet **(b)** "R → P" on Office-Home

Figure 3.9: Feature visualization using t-SNE. We randomly choose five classes with (a): "R → S" on DomainNet, and (b): "R → P" on Office-Home, respectively, with the ResNet-34 backbone and the 3-shot setup. Herein, data points in grey represent source samples, while brightly colored examples are from the target domain. The red, lightblue, purple, green, and yellow represent the categories of "Axe", "Bird", "Fence", "Shoe", and "Truck" on DomainNet, while these colors correspond to the categories of "Batteries", "Calendar", "Flowers", "Glasses", and "Monitor" within the Office-Home dataset.

observe that the performance of the mean accuracy decreases by 1.4% and 1.1%, respectively, indicating that it is necessary to include perturbations in our model training.

3.4.4 Further Analysis

We also investigate the hyper-parameter sensitivity to the loss weights α and β, the similarity threshold κ, as well as the hyper-parameter reasonability with respect to the confidence threshold τ'. In addition, we visualize the feature distributions across domains using t-SNE [98].

Hyper-parameter Sensitivity to α and β. In Figure 3.7(a) and (b), we highlight the influence of α and β. It can be observed that when $\alpha = 0.03$ and $\beta = 25.0$, the trained model achieves the highest performance in image classification. However, the accuracy decreases significantly when they are adjusted further from the optimal value. By introducing g_i^j in Eqs. (3.8) and (3.9) to remove graph nodes with noise sample connectivity, a larger proportion of unlabeled target samples do not actually participate in model updates but do contribute to the calculation of $\mathcal{L}_{abc}$, resulting in a smaller scale of $\mathcal{L}_{abc}$. Therefore, setting β to a higher value, i.e., 25.0, is advantageous for balancing the influence of $\mathcal{L}_{abc}$ and other loss items during model updates.

Hyper-parameter Sensitivity to τ and κ in CUNR and PDEP. We also conduct experiments to assess the sensitivity of our method to the hyperparameters τ and κ. Figure 3.7(c) and (d) illustrate that the classification performance of the model achieves best when τ and κ are set to 0.95 and 0.20, respectively; however, changing either of these parameters, especially on τ to less than 0.90 and κ to greater than 0.50, results in

a decline in accuracy. This is due to the fact that lighter CUNR causes a greater number of non-confident unlabeled target samples to be preserved in the graph, whereas higher PDEP causes an excessive number of node removals inside the graph, resulting in unreliable knowledge transfer between target samples.

Hyper-parameter Rationality to τ'. We conduct additional experiments to prove the validity of setting τ' to 0.975 as opposed to 0.95 by plotting variations in the quantity and accuracy of pseudo-labels for target samples whose expected probability are greater than τ'. As depicted in Figure 3.8, when τ' is set to 0.95, $\mathcal{D}_{pu}$ can collect significantly more pseudo-labeled target samples from $\mathcal{D}_u$. However, the large amount of noise contained in the pseudo-labels will also bring challenges to the model. This proves setting τ' to 0.975 is better than 0.95.

Feature Visualization. Using t-SNE for feature visualization, we present the feature distributions obtained by the proposed method for both domains in Figure 3.9. It can be observed that in the feature space, the learned features from different domains that belong to the same class are mapped nearby and clustered together, whereas those from distinct categories are significantly separated. This implies that the model trained using the proposed `G-ABC` approach is capable of producing domain-invariant and discriminative target features, thus contributing to the improved performance of the SSDA task.

3.5 Conclusions

This paper presents a novel SSDA method named Graph-based Adaptive Betweenness Clustering for achieving categorical domain alignment. It facilitates cross-domain semantic alignment by enforcing semantic transfer from labeled source and target data to unlabeled target samples. In this approach, a graph is first constructed to represent pairwise relationships between labeled examples from both domains and unlabeled target samples. Then, two strategies including Confidence Uncertainty based Node Removal and Prediction Dissimilarity based Edge Pruning are proposed to refine the connectivity in the graph to alleviate the influence of noisy edges. Provided with the refined graph, we present adaptive betweenness clustering to accomplish semantic transfer across domains with semantic propagation from labeled source or target examples to unlabeled samples on the target domain. Extensive experimental results as well as comprehensive analysis performed well on three benchmark datasets demonstrate the superiority of our proposed method, achieving new state-of-the-art results.

Chapter 4

Neighborhood Collective Estimation for Learning with Noisy Labels

This chapter delves into the problem of label noise disrupting model stability and robustness for visual learning models, thus introducing the task of learning with noisy labels.

4.1 Introduction

Deep neural networks (DNNs) have achieved significant success in computer vision tasks, such as image classification [148, 1, 85, 7, 71, 196], etc. However, they rely heavily on tremendous quantities of high-quality manual annotations. To alleviate the need for extensive human annotations while improving the generalization capability of deep neural networks, learning with noisy labels (LNL) has been proposed to effectively leverage large-scale yet poorly-annotated datasets while mitigating the effects of model overfitting to noisy labels.

To tackle the challenges imposed by LNL, previous works have proposed massive strategies [41, 138, 76, 111, 183], including noisy label correction [2, 90], noisy label or sample rejection[76, 183, 55, 54], and noisy sample reweighing [154, 127, 49]. The mainstream pipeline first uses noise verification strategies to separate the original training set into a clean set and a noisy set, which contain training samples with clean labels and noisy labels respectively, in order to diminish the effect of noisy labels during model training. Then, (un)supervised learning or semi-supervised learning (SSL) based techniques are adopted to correct noisy labels and further optimize the classification model by regarding the clean set and noisy set as labeled and unlabeled samples respectively. In this scheme, original noisy labels are simply discarded for their high chances to be incorrect, avoiding the negative effect of noisy label memorization in the trained model.

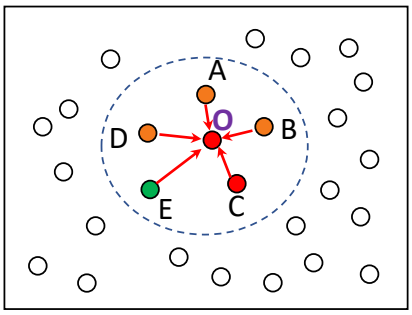


Figure 4.1: Conceptual illustration of our basic idea. Samples distributed within the dotted circle, including the candidate sample, Point O, and its nearest neighbors, i.e., Point A, B, C, D and E are close to each other in the feature-space neighborhood. Different colors indicate different labels (either predicted label or given groundtruth label). In the noise verification stage, a given label of the candidate (Point O) is considered noisy if there is a huge inconsistency between the label distributions of the candidate and its nearest neighbors; and otherwise, the candidate is considered as a clean sample. Likewise, in the noise correction stage, a noisy sample discards the given noisy label and is relabeled through a neighborhood collective estimation process involving its contrastive neighbors

In the context of learning with noisy labels, there may exist classes with imbalanced noisy or clean samples, especially in real-world noisy datasets such as Clothing-1M [162] and Webvision-1.0 [86]. For instance, there might be a relatively high proportion of noisy labels in some hard-to-annotate classes; on the other hand, a trained model may produce low-confident predictions on a relatively high proportion of hard-to-learn clean samples in some classes, making existing noise identification algorithms incorrectly identify them as noisy samples. As a result, noise accumulation may take place implicitly in such classes, making the trained model produce unreliable label predictions. The above scenarios could make an LNL algorithm fall into the so-called confirmation bias [142, 3], which causes the algorithm to favor incorrect training labels that have been confirmed with predicted labels in earlier training iterations. In this context, relying too much on the potentially biased label predictions for individual training samples would increase the risk of incorrectly identifying noisy labels in the noise verification stage. Moreover, confirmation bias also exists in the subsequent noise correction stage, where SSL or other methods, such as label-guessing [76, 108, 192] and label re-assignment [183], construct pseudo-labels for unlabeled samples in the noisy set using potentially biased label predictions. Apparently, model training in the optimization stage would strengthen this bias as more confident but incorrect predictions would defy new changes, and subsequently even deteriorate model performance in high noise ratio scenarios.

We are inspired by the premise of contrastive learning that samples from the same class should have higher similarity in the feature space than those from different classes [102, 109, 40]. Therefore, we approach learning with noisy labels from a different perspective and propose Neighborhood Collective Estimation (`NCE`), in which we re-estimate the predictive reliability of a candidate sample by contrasting it against its feature-space

nearest neighboring samples. Herein, we borrow the concept from contrastive learning, and then name such neighboring samples of the candidate as contrastive neighbors. Leveraging contrastive neighbors enriches the predictive information associated with the candidate and also makes such information relatively unbiased, thereby improving the accuracy of noisy label identification and correction. Figure 4.1 displays the basic idea of the proposed method.

Specifically, to abide by the mainstream LNL pipeline, we divide our method into two steps: 1) Neighborhood Collective Noise Verification (NCNV) to separate all training samples into a clean set and a noisy set, 2) Neighborhood Collective Label Correction (NCLC) to relabel noisy samples. In the NCNV stage, a candidate sample is considered noisy when there is a huge inconsistency between the one-hot vector of the given label of the candidate and the label distributions of its contrastive neighbors predicted using the trained model. In the NCLC stage, we only relabel noisy samples whose predicted label distribution is sufficiently similar to the given labels of neighboring clean samples, and the corrected label of a noisy sample is related to a weighted combination of the given labels of neighboring clean samples. Once we have identified clean samples and relabeled noisy ones, we leverage off-the-shelf and well-established techniques, such as mixup regularization [186] and consistency regularization [64], to perform further SSL-based model training.

In summary, the main contributions are as follows.

- We propose Neighborhood Collective Estimation for learning with noisy labels, which leverages contrastive neighbors to obtain richer and relatively unbiased predictive information for candidate samples and thus mitigates confirmation bias.
- Concretely, we design two steps called Neighborhood Collective Noise Verification and Neighborhood Collective Label Correction to identify clean samples and relabel noisy ones respectively.
- We evaluate our method on four widely used LNL benchmark datasets, i.e., CIFAR-10 [60], CIFAR-100 [60], Clothing-1M [162] and Webvision-1.0 [86], and the results demonstrate that our proposed method considerably outperforms state-of-the-art LNL methods.

4.2 Related Work

In this section, we focus on noise verification and label correction that are means involved in current dominant pipeline to address the LNL problem.

4.2.1 Noise Verification

Noise verification involves sample selection to choose and remove noisy labels within the training datasets. Proper noise verification strategies are necessary and several earlier works [41, 185, 55] have shown that samples with smaller cross-entropy loss are prone to hold clean labels, assuming that deep neural networks prefer to memorize simple patterns first rather than overfit to noisy labels. Also, some recently superior methods made efforts to model per-sample loss distributions with Beta Mixture Models (BMM) [97] or Gaussian Mixture Models (GMM) [119] to separate noisy labels from all the training samples [2, 76, 108, 192, 50, 178]. However, based on the predicted label distributions of individual candidate samples to identify the training samples, the above-stated noise verification strategies tend to fall into confirmation bias. Previous works have also attempted to identify noisy labels by leveraging neighborhood information. They either use neighborhood samples to remove noisy labels or re-weight them [158, 5, 111, 200, 161]. For example, Bahri et al. [5] proposed to identify noisy label by searching nearest neighbors based on the model predictions of a KNN classifier, while Zhu et al. [200] uses feature-space neighbors to help estimate a noise transition matrix. In our work, we employ neighborhood collective estimation to realize both the identification and correction of noise labels, and make the two promote each other, to achieve better noise label learning.

4.2.2 Label Correction

To alleviate the effect of noisy memorization, noisy labels are discarded simply, and then label correction is adopted to relabel unlabeled samples [96, 134, 76, 108, 183, 192]. This aims to give reliable pseudo-labels and support subsequent model training so as to achieve better performance. For example, SELFIE proposed by Song et al. [134] tried to perform label correction by considering model predictions from past selecting clean labels. Also, Li et al. [76] "co-guessed" pseudo-labels for unlabeled (noisy) samples via ensembling predictions of coupled networks, while Yao et al. [183] employed label re-assignment to provide pseudo-labels with the predictions of a temporally averaged model. Different from those as mentioned above, we correct noisy labels with the aid of neighboring labeled samples. This can relatively avoid confirmation bias that derives from model predictions at individual samples.

4.3 Methodology

Problem formulation. Learning with noisy labels seeks an optimal model trained with a large-scale noisy dataset $\mathcal{D}_{\textbf{train}} = \{(x_i, y_i)\}_{i=1}^{N}$, where N is the number of sample-label pairs and each pair consists of a training sample x_i and its associated label y_i over C classes while whether the given label is noisy or clean is unknown. During the training process, a sample is fed into a model being trained, that is parameterized by θ and

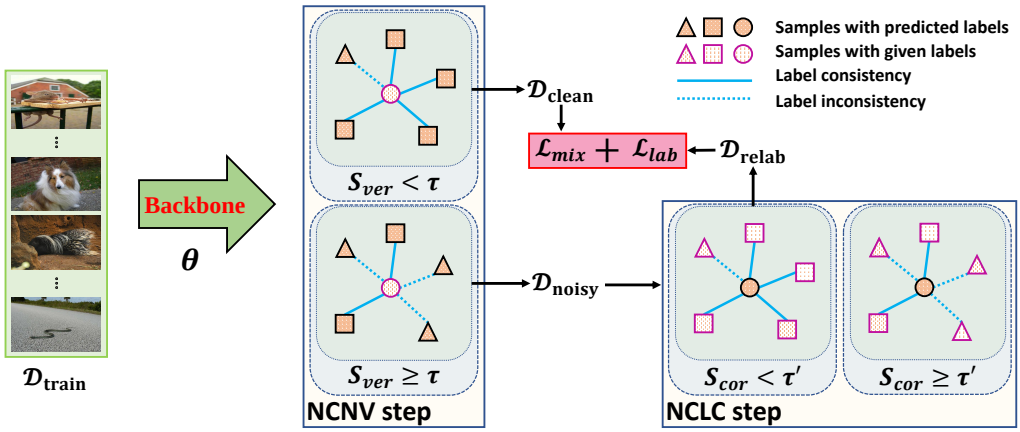


Figure 4.2: A Diagram of NCE for learning with noisy labels. Triangles and squares represent contrastive neighbors from two different classes while circles denote the candidate samples in various steps. We assume the candidates belong to the class represented by the squares. In this work, we design two steps called Neighborhood Collective Noise Verification (NCNV) and Neighborhood Collective Label Correction (NCLC) to identify clean samples and relabel noisy ones respectively. Both steps leverage contrastive neighbors to obtain richer and relatively unbiased predictive information for candidate samples and thus mitigate confirmation bias

contains a feature extractor Φ and a classifier with a softmax layer, to obtain its corresponding feature representation $\Phi(x_i)$ and class probabilities $p(y|x_i)$ respectively.

Contrastive neighbors. We contrast a candidate sample against its feature-space nearest neighbors to enrich and diversify predictive information of the candidate. Such nearest neighbors are called contrastive neighbors in this paper. First, to compute feature similarity between a candidate sample x_i and one of its feature-space neighbors x_j, we define a similarity function:

$$d(x_i, x_j) = \frac{\Phi(x_i)^\top \Phi(x_j)}{\|\Phi(x_i)\| \|\Phi(x_j)\|}, \tag{4.1}$$

where $d(\cdot, \cdot)$ denotes the cosine distance metric. Then, we set up a pairwise connection between the two samples and quantify the discrepancy between their label distributions through the Jensen-Shannon (JS) divergence as follows,

$$J(p_i, p_j) = \frac{1}{2} KL(p_i || \frac{p_i + p_j}{2}) + \frac{1}{2} KL(p_j || \frac{p_i + p_j}{2}), \tag{4.2}$$

where $KL(\cdot||\cdot)$ represents the Kullback-Leibler (KL) divergence, and for sample x_i (or x_j), in different contexts, p_i (or p_j) represents either its probabilistic label distribution predicted using a trained model or its given ground-truth label. $J(\cdot, \cdot)$ returns values in the range of [0,1], and the use of JS divergence allows us to measure the discrepancy between the probabilistic label distributions of different samples. $J(p_i, p_j) \to 0$ indicates that the label distributions of p_i and p_j are very similar while $J(p_i, p_j) \to 1$ means the label distributions of these two samples are of great difference.

Algorithm 1: Training procedure of NCE

Input: Dataset $\mathcal{D}_{\text{train}}$; Number of training epochs T_{tr}; Number of warm-up epochs T_{wu}; Learning rate η
Output: Optimal model parameter θ

for $t = 1 \dots T_{tr}$ **do**
  if $t < T_{wu}$ **then**
    // Step of Warm-up
    WarmUp($\mathcal{D}_{\text{train}}$, θ); ▷ Model initialization using a "WarmUp" function
  else
    // Step of NCNV
    Use Eq. (4.5) to split $\mathcal{D}_{\text{train}}$ into clean samples $\mathcal{D}_{\text{clean}}$ and noisy ones $\mathcal{D}_{\text{noisy}}$;
    // Step of NCLC
    Use Eq. (4.9) to relabel a subset of samples from $\mathcal{D}_{\text{noisy}}$ and form a new subset $\mathcal{D}_{\text{relab}}$;
    // Step of Fine-tuning
    Randomly sample mini-batches from $\mathcal{D}_{\text{clean}}$ and $\mathcal{D}_{\text{relab}}$;
    Update model parameter θ by applying SGD with η to Eq. (4.13);
  end if
end for

Overview. In this paper, we propose Neighborhood Collective Estimation (NCE) to tackle learning with noisy labels. In detail, we first propose Neighborhood Collective Noise Verification (NCNV) to identify noisy labels in $\mathcal{D}_{\textbf{train}}$ and divide $\mathcal{D}_{\textbf{train}}$ into clean subset $\mathcal{D}_{\textbf{clean}}$ and noisy subset $\mathcal{D}_{\textbf{noisy}}$. Then, we propose Neighborhood Collective Label Correction (NCLC) to relabel selected samples from $\mathcal{D}_{\textbf{noisy}}$ and form a new subset $\mathcal{D}_{\textbf{relab}}$. Finally, we leverage auxiliary techniques to perform model fine-tuning so as to further optimize our model. The diagram and the training procedure of our proposed model have been summarized in Figure 4.2 and Algorithm 2, respectively.

4.3.1 Neighborhood Collective Noise Verification

In an effort to identify label noise for the task of LNL, most recent research establish sample selection criteria on the basis of predicted label distributions of individual samples [41, 185, 55, 2, 76], thus it is hard for them to avoid confirmation bias. Aiming at mitigating such bias, we formulate a novel noise verification function that determines whether a candidate is a noisy sample or not through the estimation of its label inconsistency score, which measures the degree of inconsistency between the label distributions of the candidate sample and its contrastive neighbors. Specifically, given a candidate sample-label pair $(x^{(c)}, y^{(c)}) \in \mathcal{D}_{\textbf{train}}$, we first find its K nearest neighbors in the feature space using the cosine similarity in Eq. (4.1) and then declare them as contrastive neighbors, as formulated below.

$$\{x_k^{(c)}\}, k = 1, \cdots, K \leftarrow \textbf{KNN}(x^{(c)}; \mathcal{D}_{\textbf{train}}; K), \tag{4.3}$$

where $\textbf{KNN}(x^{(c)}; \mathcal{D}_{\textbf{train}}; K)$ is a function that returns K most similar samples in $\mathcal{D}_{\textbf{train}}$ for the candidate sample $x^{(c)}$. Note that $x^{(c)}$ is temporarily removed from $\mathcal{D}_{\textbf{train}}$ at this

moment.

Then, the neighborhood-based label inconsistency score for the given label of the candidate can be defined as follows,

$$S_{ver}(x^{(c)}, y^{(c)}) = \frac{1}{K}\sum_{k=1}^{K} J(p_y(y^{(c)}), p(y|x_k^{(c)})), \tag{4.4}$$

where $p_y(y^{(c)})$ is the one-hot vector for the given ground-truth label $y^{(c)}$ of the candidate sample and $p(y|x_k^{(c)})$ stands for the probabilistic label distribution of the k-th contrastive neighbor predicted using a classification model trained with all original samples including both clean and noisy ones. Here, instead of the model prediction at the candidate sample, we make use of model predictions at its contrastive neighbors, implicitly diversifying the predictive information of the candidate sample and making it relatively unbiased.

After computing the label inconsistency score for every candidate sample, we observe that if the given ground-truth label of a candidate sample is significantly different from the model prediction of its contrastive neighbor samples, i.e., of large inconsistency, then the given label is very likely to be a noisy label. Therefore, by setting a threshold τ, we can classify candidate sample $x^{(c)}$ as a noisy sample if $S_{ver}(x^{(c)}, y^{(c)}) \geq \tau$, and otherwise, a clean one. To this end, we can obtain $\mathcal{D}_{\textbf{clean}}$ and $\mathcal{D}_{\textbf{noisy}}$ as follows,

$$\begin{aligned} \mathcal{D}_{\textbf{clean}} &\leftarrow \{(x_i, y_i)\,|S_{ver}(x_i, y_i) < \tau, \forall\,(x_i, y_i) \in \mathcal{D}_{\textbf{train}}\}, \\ \mathcal{D}_{\textbf{noisy}} &\leftarrow \{(x_i,)\,|S_{ver}(x_i, y_i) \geq \tau, \forall\,(x_i, y_i) \in \mathcal{D}_{\textbf{train}}\}. \end{aligned} \tag{4.5}$$

4.3.2 Neighborhood Collective Label Correction

After the neighborhood collective noise verification (NCNV) stage, we treat samples from $\mathcal{D}_{\textbf{clean}}$ and $\mathcal{D}_{\textbf{noisy}}$ as labeled and unlabeled samples respectively by simply discarding noisy labels to prevent noise memorization in the resulted classification model. To leverage the unlabeled samples, some studies have taken pseudo-labeling based methods to mine discriminative cues for model training [76, 183, 111], yet all of them resort to model predictions at individual unlabeled samples, again tracing back to the unavoidable bias. On the contrary, we set up neighborhood collective label correction (NCLC) stage, which corrects noisy labels by relying on neighboring clean samples to obtain more reliable and relatively unbiased pseudo-labels.

As in the NCNV stage, we first find K contrastive neighbors for each noisy sample $x^{(u)} \in \mathcal{D}_{\textbf{noisy}}$ according to the ranked feature similarities between $x^{(u)}$ and its neighbors, as formulated below. At this time, we require all its contrastive neighbors to belong to the clean set $\mathcal{D}_{\textbf{clean}}$.

$$\{(x_k^{(u)}, y_k^{(u)})\}, k = 1, \cdots, K \leftarrow \textbf{KNN}(x^{(u)}; \mathcal{D}_{\textbf{clean}}; K), \tag{4.6}$$

where $(x_k^{(u)}, y_k^{(u)})$ is a sample-label pair from $\mathcal{D}_{\textbf{clean}}$. Unlike the NCNV stage, the ground-truth label information of contrastive neighbors is required in this stage.

Afterwards, we perform the following label consistency check between each candidate sample and its contrastive neighbors to mine those noisy samples that are similar to their neighboring samples in both the feature and label space,

$$S_{cor}(x^{(u)}) = \frac{1}{K}\sum_{k=1}^{K} J(p(y|x^{(u)}), p_y(y_k^{(u)})), \tag{4.7}$$

where $J(p(y|x^{(u)}), p_y(y_k^{(u)}))$ computes the discrepancy between the probabilistic label distribution of the candidate sample $x^{(u)}$ predicted using the trained classification model, and the one-hot vector for the given ground-truth label of its k-th contrastive neighbor. A large $S_{cor}(x^{(u)})$ indicates that the predicted label of the candidate sample is highly dissimilar to the clean and definite labels of its contrastive neighbors, suggesting that the candidate sample may lie near the decision boundary of the model. To be safe, we drop such candidate noisy samples if $S_{cor}(x^{(u)}) \geq \tau'$, where a second threshold τ' is used. In contrast, a candidate sample that satisfies $S_{cor}(x^{(u)}) < \tau'$ is more likely to be farther away from the decision boundary and could derive a more reliable pseudo-label from its contrastive neighbors. Therefore, we define a label correction function to generate a new label for such a noisy sample as follows,

$$\textbf{Correct}(x^{(u)}) = \arg\max_{c} \sum_{k=1}^{K} w(x^{(u)}; k) \cdot p_y(y_k^{(u)}), \tag{4.8}$$

where we use $w(x^{(u)}; k) = 1 - J(p(y|x^{(u)}), p_y(y_k^{(u)}))$ to approximate the probability that the candidate sample belongs to the same class as its k-th contrastive neighbor, and $c = 1, \cdots, C$ indicates the c-th component of a label distribution vector has the maximum value. For convenience, we set $\hat{y}^{(u)} = \textbf{Correct}(x^{(u)})$.

Finally, we define a new sample collection that contains all relabeled noisy samples as follows,

$$\mathcal{D}_{\textbf{relab}} \leftarrow \{(x_i, \hat{y}_i) \,|\hat{y}_i = \textbf{Correct}(x_i), S_{cor}(x_i) < \tau', \forall x_i \in \mathcal{D}_{\textbf{noisy}}\}. \tag{4.9}$$

4.3.3 Training Objectives

Once we have the clean set $\mathcal{D}_{\textbf{clean}}$ and relabeled set $\mathcal{D}_{\textbf{relab}}$ respectively from the NCNV and NCLC steps, we use both datasets together to further optimize the classification model through fine-tuning. Auxiliary techniques are incorporated during model optimization. Since the initial classification model trained using both clean and noisy samples memorizes noisy labels during its training process and Mixup [186] can effectively attenuate such noise memorization, we first employ the mixup regularization

to construct augmented samples through linear combinations of existing samples from $\mathcal{D}_{\textbf{clean}}$.

Given two existing samples (x_i, y_i) and (x_j, y_j) from $\mathcal{D}_{\textbf{clean}}$, an augmented sample $(\tilde{x}, \tilde{y})$ can be generated as follows,

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \tilde{y} = \lambda p_y(y_i) + (1-\lambda)p_y(y_j), \tag{4.10}$$

where $\lambda \sim Beta(\alpha)$ is a mixup ratio and α is a scalar parameter of Beta distribution. The cross-entropy loss applied to B augmented samples in each mini-batch is defined as follows,

$$\mathcal{L}^{mix} = -\sum_{b=1}^{B} \tilde{y}_b \log p(y|\tilde{x}_b). \tag{4.11}$$

In the NCLC stage, more reliable pseudo-labels are assigned to noisy samples farther away from the decision boundary. To leverage these relabeled samples during model optimization, we apply consistency regularization to them to further enhance the robustness of the model [25]. Label consistency is a good choice to achieve this goal because it encourages the fine-tuned model to produce the same output when there are minor perturbations in the input [64]. In practice, we enforce label consistency through the following loss:

$$\mathcal{L}^{lab} = -\sum_{b'=1}^{B'} p_y(y_{b'}) \log p(y|\textbf{Aug}(x_{b'})), \tag{4.12}$$

where B' relabeled samples $(x_{b'}, y_{b'}) \in \mathcal{D}_{\textbf{relab}}$ are chosen in each iteration, $p_y(y_{b'})$ is the one-hot vector of the pseudo-label of $x_{b'}$, $\textbf{Aug}(\cdot)$ denotes the function that perturbs the chosen samples using Autoaugment technique proposed in [18], and $p(y|\textbf{Aug}(x_{b'}))$ is the predicted label distribution of the perturbed sample. Proved by our experiments, this label consistency loss can be also applied to the selected clean samples from $\mathcal{D}_{\textbf{clean}}$, especially under low noise ratios, to better boost the performance of the model.

As stated above, the overall loss function for final model fine-tuning is a combination of the cross-entropy and label consistency losses,

$$\mathcal{L}^{overall} = \mathcal{L}^{mix} + \gamma \mathcal{L}^{lab}, \tag{4.13}$$

where γ is a trade-off scalar to balance those two loss terms.

4.4 Experiments

4.4.1 Experimental Setups

Implementations. We highlight the effectiveness of our proposed `NCE` method on four standard LNL benchmark datasets: CIFAR-10 [60], CIFAR-100 [60], Clothing-1M [162]

and Webvision-1.0 [86]. To be fair, we follow most details of the training and evaluation processes from the previous work DivideMix [76], such as network architectures, confidence penalty for asymmetric noise, and so on.

CIFAR-10 and CIFAR-100 are two classic synthetic datasets for the LNL problem. We follow DivideMix [76] to create the noisy types, i.e., "Symmetry" and "Asymmetry", and to set noise ratios, namely "0.20", "0.50", "0.80" and "0.90" for "Symmetry", and "0.40" for "Asymmetry". Similar to existing works [90, 76, 161], we also select PreAct Resnet [45] as the model backbone for CIFAR-10/CIFAR-100. Then we train it using a SGD optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4} respectively. To better initialize our model, we set a warm-up step to perform supervised training on the model over all available samples using a standard cross-entropy loss. For effectiveness, this step is assigned a training period $T_{wu} = 10$ (or 30) for CIFAR-10 (or CIFAR-100). For adapting to diverse scenarios, we empirically set τ to 0.75 on CIFAR-10 or 0.90 on CIFAR-100, while τ' are usually set as 2×10^{-3} and 1×10^{-2} on CIFAR-10 and CIFAR-100, respectively. With respect to other hyper-parameters that are involved in NCE on CIFAR-10/CIFAR-100, we set $K = 20, T_{tr} = 300, \gamma = 1.0, \eta = 0.02$, $B = 128$, $B' = 128$ and $\alpha = 4$.

Clothing-1M and Webvision-1.0 are two large-scale real-world noisy datasets. Clothing-1M contains one million samples grabbed from the online shopping websites and Webvision-1.0 only uses top-50 classes originating from the Google image Subset of Webvision [86]. For Webvision-1.0, the results are reported from testing our model on both the WebVision validation set and the ImageNet ILSVRC12 validation set [137]. Our experiments on both datasets employ similar hyper-parameter settings, e.g., $T_{wu} = 1$, $B = 32$, $B' = 32$, $\alpha = 0.5$ and $K = 20$. τ is set to 0.65 for Clothing-1M and 0.90 for Webvision-1.0. As shown in Table 4.2 and Table 4.3 in the main text, on these two datasets, our model only using clean samples in $\mathcal{D}_{\textbf{clean}}$ for training outperforms previous state-of-the-art methods. Then, we follow the practice in DivideMix [76] and also set $\gamma = 0.0$ so that there is no need to set τ'. The learning rate schedule is the same for both datasets, that is, after half training epochs, the initial learning rate is divided by 10. The initial learning rate for Clothing-1M and Webvision-1.0 is set to 0.002 and 0.01 respectively. In addition, we choose Resnet-50 [45] and Inception-Resnet-V2 [137] as the backbones for Clothing-1M and Webvision-1.0, respectively. We train the models using a SGD optimizer with a momentum of 0.9 and a weight decay of 1×10^{-3}. Moreover, the number of training epochs for Clothing-1M and Webvision-1.0 are $T_{tr} = 80$ and $T_{tr} = 100$, respectively.

Baselines. We compare NCE with the following state-of-the-art algorithms to address the LNL problem on CIFAR-10 and CIFAR-100: **Cross-Entropy** [76], **F-correction** [116], **Co-teaching+** [185], **PENCIL** [63], **LossModelling** [2], **DivideMix** [76], **ELR** [90], **ProtoMix** [79] and **NGC** [161]. Herein, **Cross-Entropy** trains the model only with a supervised cross-entropy loss over training samples along with given noisy labels, and its results are copied from DivideMix. Besides methods stated above, we perform our

Table 4.1: Comparison results (Test accuracy (%)) of the proposed NCE and existing state-of-the-art methods on the CIFAR-10 and CIFAR-100 datasets, with results reported using mean accuracy and 95% confidence interval over 3 trails.

Dataset	**CIFAR-10**					**CIFAR-100**			
Noise type	**Symmetric**				**Assymetric**	**Symmetric**			
Method/Noise ratio	**0.2**	**0.5**	**0.8**	**0.9**	**0.4**	**0.2**	**0.5**	**0.8**	**0.9**
Cross-Entropy [76]	86.8	79.4	62.9	42.7	85.0	62.0	46.7	19.9	10.1
F-correction [116]	86.8	79.8	63.3	42.9	87.2	61.5	46.6	19.9	10.2
Co-teaching+ [185]	89.5	85.7	67.4	47.9	-	65.6	51.8	27.9	13.7
PENCIL [63]	92.4	89.1	77.5	58.9	88.5	69.4	57.5	31.1	15.3
LossModelling [2]	94.0	92.0	86.8	69.1	87.4	73.9	66.1	48.2	24.3
DivideMix [76]	96.1	94.6	93.2	76.0	93.4	77.3	74.6	60.2	31.5
ELR [90]	95.8	94.8	93.3	78.7	93.0	77.6	73.6	60.8	33.4
ProtoMix [79]	95.8	94.3	92.4	75.0	91.9	79.1	74.8	57.7	29.3
NGC [161]	95.9	94.5	91.6	80.5	90.6	79.3	75.9	62.7	29.8
NCE (best)	**96.2** ±0.09	**95.3** ±0.12	**93.9** ±0.22	**88.4** ±0.98	**94.5** ±0.70	**81.4** ±0.37	**76.3** ±0.28	**64.7** ±0.56	**41.1** ±0.54
NCE (last)	96.0 ±0.22	95.2 ±0.23	93.6 ±0.30	88.0 ±1.21	94.2 ±0.96	81.0 ±0.27	75.3 ±0.07	64.5 ±0.86	40.7 ±0.42

Table 4.2: Comparison results (Test accuracy (%)) of the proposed NCE and existing state-of-the-art methods on the Clothing-1M dataset.

Meta-L. [78]	DivideMix [76]	ELR [90]	ELR+ [90]	NestedCoT. [16]	AugDesc [108]	NCE
73.5	74.8	72.9	74.8	74.9	75.1	**75.3**

comparison on Clothing-1M with previous methods, including **Meta-Learning** [78], **ELR+** [90], **NestedCoTeaching** [16] and **AugDesc** [108], where the augmentation strategy of our method on this dataset refers to that of AugDesc for comparison fairness. Moreover, we evaluate the proposed approach on Webvision-1.0 by newly adding **Decoupling** [99], **MentorNet** [55], and **Co-teaching** [41].

4.4.2 Comparisons with State-of-the-Arts

Synthetic noisy datasets. CIFAR-10 and CIFAR-100 are two representative synthetic LNL benchmark datasets and we report results on these datasets in Table 4.1. For fair comparison, we follow all the settings in [76, 161]. We can see that our NCE outperforms all existing state-of-the-art methods on CIFAR-10 and CIFAR-100 under all settings of symmetric (from 20% to 90%) and asymmetric (40% only) label noise ratio. In particular, on CIFAR-10, our method surpasses the best performing baselines by 7.9% and 1.1% at the highest symmetric and asymmetric noise ratios, respectively. In addition, in comparison to the performance of existing algorithms on CIFAR-100, NCE achieves the highest classification accuracy under all four noise ratio settings by exceeding the second best by 2.1%, 0.4%, 2.0% and 7.7%, respectively.

Real-world noisy datasets. To further verify the effectiveness of the proposed NCE method, we also conduct experiments on real-world noisy datasets, namely Clothing-1M and Webvision-1.0. Table 4.2 and Table 4.3 show performance comparisons between NCE and existing algorithms when these two are respectively used as the training set.

Table 4.3: Comparison results (top-1 and top-5 test accuracy (%)) of the proposed NCE and existing state-of-the-art methods on the Webvision and ImageNet ILSVRC12 validation sets. The models are trained on the training set of the Webvision-1.0 dataset.

Method	WebVision		ILSVRC12	
	top-1	top-5	top-1	top-5
F-correction [116]	61.1	82.7	57.4	82.4
Decoupling [99]	62.5	84.7	58.3	82.3
MentorNet [55]	63.0	81.4	57.8	79.9
Co-teaching [41]	63.6	85.2	61.5	84.7
DivideMix [76]	77.3	91.6	75.2	90.8
ELR [90]	76.3	91.3	68.7	87.8
ELR+ [90]	77.8	91.7	70.3	89.8
NGC [161]	79.2	91.8	74.4	91.0
NCE	**79.5**	**93.8**	**76.3**	**94.1**

Table 4.4: Ablation study of our method (NCE) on the CIFAR-10 and CIFAR-100 datasets under multiple label noise ratios. "repl." is an abbreviation for "replaced", and $\mathcal{L}^{ce}$ means the model is trained on the clean samples using a cross-entropy loss. (Only one of three trails is selected for comparison in our NCE method)

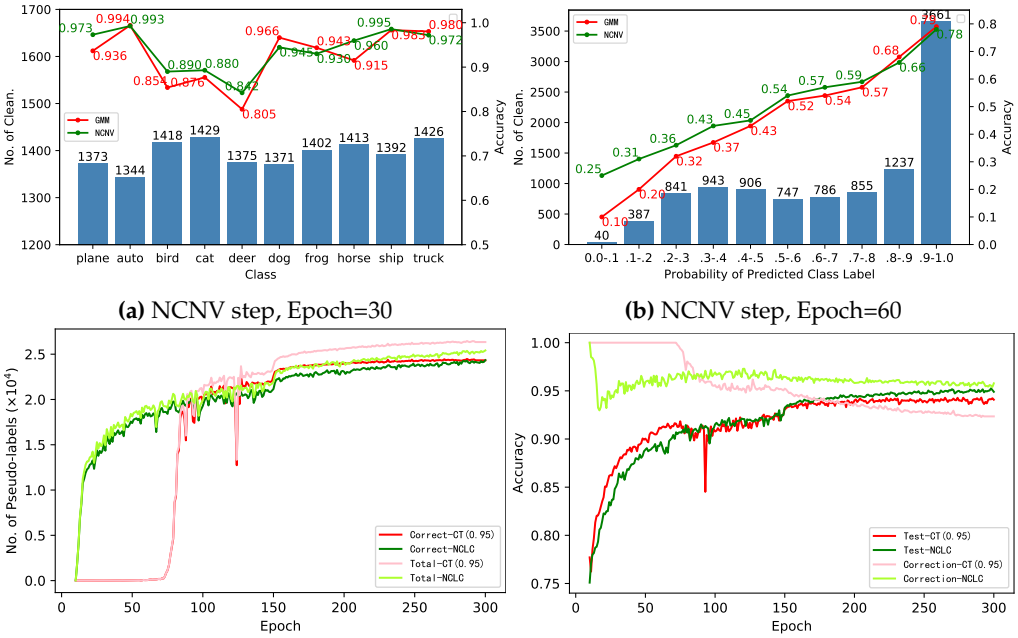
M-(#)	Dataset / Noise type / Method/Noise ratio	CIFAR-10 Symmetric 0.5	CIFAR-10 Symmetric 0.8	CIFAR-10 Assymetric 0.4	CIFAR-100 Symmetric 0.5	CIFAR-100 Symmetric 0.8	Mean
1	NCE	95.3	94.1	94.6	76.1	65.2	85.1
2	NCE repl. NCNV w/ GMM	94.8	79.0	89.7	75.8	56.8	79.2
3	NCE repl. NCLC w/ CT(0.95)	94.3	86.1	90.1	76.0	58.7	81.0
4	NCE repl. NCNV w/ GMM & w/o $\mathcal{L}^{lab}$	91.2	78.8	87.3	71.4	49.7	75.7
5	NCE w/o $\mathcal{L}^{lab}$	92.5	86.7	92.6	74.4	57.9	80.8
6	NCE repl. $\mathcal{L}^{mix}$ w/ $\mathcal{L}^{ce}$	93.3	78.5	89.0	73.2	55.2	77.8
7	NCE repl. perturbed w/ unperturbed in Eq. (4.12)	93.6	89.4	90.5	72.5	56.1	80.4

We can observe that NCE achieves the highest accuracy on Clothing-1M and an improvement of 0.2% over AugDesc, the best performing method among existing ones. Likewise, on the challenging Webvision-1.0, NCE again achieves higher performance than most existing methods in terms of top-1 and top-5 accuracy. These results further verify that our proposed approach can effectively perform well on the real-world noisy datasets.

4.4.3 Ablation Analysis

To provide insights on how effectively each component of our algorithm works, we conduct an ablation study by removing or replacing individual components. Results of this ablation study are summarized in Table 4.4 and Figure 4.3. Also, as displayed in and Figure 4.4, we perform feature visualization to further analyze the proposed algorithm. All experiments are performed on both CIFAR-10 and CIFAR-100 datasets.

Effectiveness of NCNV step. To examine the effectiveness of the NCNV step in identifying clean/noisy labels, we replace NCNV with a well-known GMM-based strategy proposed in DivideMix [76]. In Table 4.4, a comparison between row M-(1) and row M-(2) reveals that our NCNV step significantly outperforms the GMM-based strategy because the former is capable of identifying clean labels of harder samples. Specifically,



(a) NCNV step, Epoch=30 **(b)** NCNV step, Epoch=60

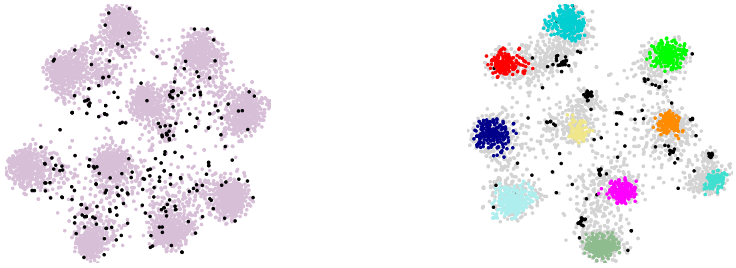
(c) NCNV step, Epoch=300 **(d)** NCLC step, Epoch=300

Figure 4.3: Analysis of ablation study results. **(a)** The accuracy of clean sample identification in various classes. **(b)** The accuracy of clean sample identification vs. the probability (confidence) of predicted class label. **(c)** The evolution of the numbers of pseudo-labels and correct pseudo-labels over epochs. **(d)** The evolution of label correction accuracy and test classification accuracy over epochs. The experiments for (a) and (b) are performed on CIFAR-10 and CIFAR-100 respectively with the same noise profile (Noise ratio: 0.80; Noise type: Symmetric). The blue bars represent the distribution of clean samples. (c) and (d) describe the same experiment, where we analyze the label correction performance of NCLC and Confidence Thresholding (i.e., CT(0.95)) on CIFAR-10 (Noise ratio: 0.50; Noise type: Symmetric)

Figure 4.3(a) and (b) show the power of our NCNV step in handling "hard" classes and "hard" samples in the clean subset. A class is considered "hard" when multiple methods have an overall low clean sample identification accuracy in the class, while a "hard" sample has a low probability (confidence) associated with its predicted class label. As Figure 4.3(a) shows, our method achieves higher sensitivity on "hard" classes, i.e. "cat", "bird" and "deer", where both methods have the lowest identification accuracy. In addition, Figure 4.3(b) also shows that our NCNV step works significantly better on "hard" samples, whose predicted class labels are associated with a low probability (confidence).

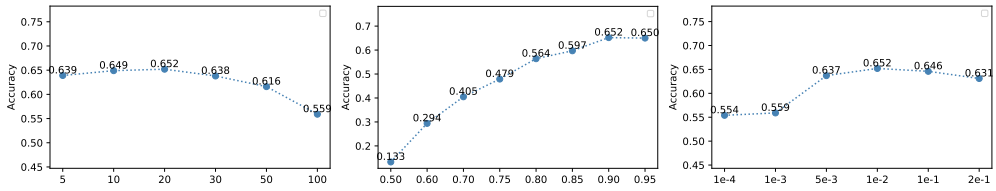
Effectiveness of NCLC step. To better understand the performance of the NCLC step in label correction, we replace NCLC with an existing label correction scheme, called Confidence Thresholding (CT) [64], which relabels such samples whose pseudo-labels have a confidence value exceeding a predefined threshold, e.g., 0.95. According to row M-(3) of Table 4.4, NCLC clearly outperforms CT under all noise ratio settings. In detail, Figure 4.3(c) and (d) reveal that CT works with few pseudo-labels in the early

(a) NCNV step, Epoch=30 **(b)** NCNV step, Epoch=60



(c) NCNV step, Epoch=300 **(d)** NCLC step, Epoch=300

Figure 4.4: Feature visualization using t-SNE. We choose 10 representative classes on CIFAR-100 (Noise ratio: 0.80; Noise type: Symmetric). (**a**)-(**c**) show how the distributions of misidentifications in the NCNV step evolve during model training. They are involved in samples from $\mathcal{D}_{\textbf{train}}$ corresponding to each representative class. In these subfigures, points in black are misclassified samples, such as clean (or noisy) samples misclassified as noisy (or clean) ones, in the training data, while samples in purple are correctly identified ones. The accuracy of training sample identification in (a)-(c) is 82.2%, 94.7% and 95.2%, respectively. (**d**) shows the feature distributions of unlabeled (noisy) samples in $\mathcal{D}_{\textbf{noisy}}$ corresponding to 10 classes in the NCLC step, and points in bright colors, black and grey respectively denote correctly relabeled samples, mis-relabeled ones and dropped ones



(a) K **(b)** τ **(c)** τ'

Figure 4.5: Sensitivity with respect to hyper-parameters K, τ and τ'. We conduct these experiments on CIFAR-100 with the same noise profile (Noise ratio: 0.80; Noise type: Symmetric). In this noise profile, our model achieves the best accuracy of 65.2% when we set $K = 20$, $\tau = 0.90$ and $\tau' = 0.01$.

epochs. This is because, at that moment, the model cannot fit the training samples well and thus unlabeled samples with low-confidence predictions (< 0.95) would not be

assigned pseudo-labels. Afterwards, although plenty of unlabeled samples are given pseudo-labels as model training goes on, the label correction accuracy drops at the same time. Ultimately, it leads to lower performance than NCLC, which, on the other hand, obtains more reliable pseudo-labels for unlabeled (noisy) points.

Necessity of mixup regularization. To verify the importance of the mixup regularization, we remove it from our algorithm and then perform standard supervision over clean samples. As shown in row M-(6) of Table 4.4, this change causes very serious performance degradation, indicating that the mixup regularization is able to effectively attenuate noise memorization.

Necessity of consistency regularization. To investigate the effectiveness of consistency regularization over unlabeled (noisy) samples, we conduct two experiments. First, we disable $\mathcal{L}^{lab}$, meaning that the model is only trained over all clean samples. By comparing row M-(5) with row M-(1) in Table 4.4, we observe that the performance under all noise ratios drops by 1.7% to 7.6%, suggesting that this consistency loss is important for the performance of the model, especially when the noise ratio is high. In the second experiment, we replace the perturbed samples used in Eq. (4.12) with unperturbed ones to examine the need of sample perturbations. As shown in row M-(7) of Table 4.4, the average accuracy drops considerably by 4.7%. This demonstrates that sample perturbations in Eq. (4.12) play a significant role in realizing the full potential of consistency regularization.

4.4.4 Further Analysis

Feature visualization. We use t-SNE [147] to visualize the feature distributions in both NCNV and NCLC steps. In Figure 4.4(a)-(c), we show how the distributions of misidentified samples across diverse classes evolve in the model training process. It can be observed that as model training proceeds, the number of misclassifications in the training data decreases gradually. The misclassifications are distributed near the boundaries of the clusters corresponding to the classes, showing a good noise verification effect. Furthermore, in the NVLC step, as illustrated in Figure 4.4(d), most well-relabeled samples are located in the core regions of the clusters, while the mis-relabeled points and dropped ones are closer to the boundaries of the clusters or peripheral areas between different clusters. This meets our assumption stated in Section 4.3.2 that a candidate sample in the NVLC step that satisfies Eq. (4.9) is more likely to be farther away from the decision boundary of the model and could derive a more reliable pseudo-label.

Hyper-parameter sensitivity. We also investigate the sensitivity of our proposed method to three key hyper-parameters, i.e., K, τ and τ'. Taking CIFAR-100 with (Noise ratio: 0.80; Noise type: Symmetric) as an example, Figure 4.5 shows that the model reaches a significantly high classification performance in this LNL case when we set K, τ and τ' to 20, 0.90, and 0.01 respectively; on the other hand, a probable decrease in accuracy ensues when we change any of those parameters. With achieving fair comparisons, we

follow DivideMix [76] to set other hyper-parameters that are involved in the training process or network architectures.

4.5 Conclusions

In this paper, we have introduced a novel method called Neighborhood Collective Estimation (NCE) to tackle the problem of learning with noisy labels. In this method, we re-estimate the predictive reliability of a candidate sample by contrasting it against its feature-space nearest neighbors. This can enrich and diversify predictive information associated with the candidate and also makes such information relatively unbiased. The accuracy of noisy label identification and correction can thus be improved, facilitating subsequent model training. In detail, NCE consists of two steps, 1) Neighborhood Collective Noise Verification (NCNV) for separating all training data into clean samples and noisy ones, and 2) Neighborhood Collective Label Correction (NCLC) for relabeling noisy samples. Extensive experiments and a thorough ablation study have confirmed the superiority of our proposed method.

Chapter 5

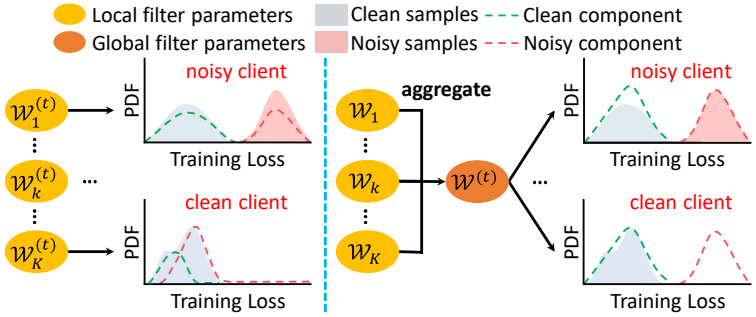
Collaborative Noise Filtering for Federated Learning with Noisy Labels

In this chapter, noisy label learning is extended to the study of federated learning, formulated as the task of federated learning with noisy labels.

5.1 Introduction

Compared to traditional Centralized Learning [82, 71, 74, 72, 73, 160, 159, 48], Federated Learning (FL) is a novel paradigm facilitating collaborative learning across multiple clients without requiring centralized local data [100]. Recently, FL has shown significant real-world success in areas like healthcare [106], recommender systems [175], and smart cities [193]. However, these FL methods presume clean labels for all client's private data, which is often not the case in reality due to data complexity and uncontrolled label annotation quality [140, 65]. Especially with the blessing of privacy protection, it is impossible to ensure absolute label accuracy. Therefore, this work centers on Federated Learning with Noisy Labels (F-LNL). In F-LNL, a global neural network model is fine-tuned via distributed learning across multiple local clients with noisy samples. Like [167], we here also assume some local clients have noisy labels (namely noisy clients), while others have only clean labels (namely clean clients).

Besides the private data on local clients, F-LNL faces two primary challenges: data heterogeneity and noise heterogeneity [58, 179]. Data heterogeneity refers to the statistically heterogeneous data distributions across different clients, while noise heterogeneity represents the varying noise distributions among clients. It has been demonstrated by [167, 58] that these two challenges in F-LNL may lead to instability during local training sessions. Previous studies in F-LNL [167, 58] have demonstrated that many FL approaches, such as [100, 84], are now in use to adequately address the first challenge. These approaches primarily focus on achieving training stability with convergence guarantees by aligning the optimization objectives of local updates and global



(a) Local Noise Filter (b) Federated Noise Filter

Figure 5.1: Conceptual comparison between local noise filtering and collaborative noise filtering. (a) Local noise filtering may have limited capabilities as each client develops its own local noise filter using its own private data only. Especially on clean clients, such filters would incorrectly identify a subset of clean samples to be noisy. (b) Collaborative noise filtering proposed by us significantly improves the performance of label noise filtering on each client as a federated noise filter is learned by distilling knowledge from all clients. PDF: Probability density function.

aggregation. However, they do not address label noise on individual clients. Therefore, to tackle noise heterogeneity, such F-LNL algorithms propose separating noisy data into clean and noisy samples, occasionally complemented by relabeling the noisy samples. This aims to mitigate the negative effects of noisy labels and prevent local model overfitting to such label noise, avoiding severe destabilization of the local training process.

Some F-LNL methods [58, 167] thus emphasize proposing local noise filtering, where each client develops its own noise filter to identify noisy labels. However, these noise filtering strategies omit the potential of learning prosperous knowledge from other clients to strengthen their capacities. Instead, they rely heavily on each client's own private data for training, thus leading to sub-optimal and inferior performance. For example, as shown in Figure 5.1(a), limited training samples available on each client impede the accurate modeling of their local noise distributions, significantly restricting noise filtering capabilities. In addition, if noise filtering and relabeling are not handled properly, overfitting of noisy labels can inevitably occur, leading to noise memorization and thus degrading global performance upon model aggregation. Developing strategies to prevent noise memorization [172] while enhancing training stability is critical, yet existing F-LNL algorithms have not succeeded in achieving this.

In this paper, we present a novel framework, `FedDiv`, to address the challenges of F-LNL. To perform label noise filtering per client, `FedDiv` consists of a global noise filter, called Federated Noise Filter (FNF), constructed by modeling the global distribution of noisy samples across all clients. Specifically, by fitting the loss function values of private data, the parameters of a local Gaussian Mixture Model (GMM) can be iteratively learned on each client to model its local noise distributions. These parameters are then aggregated on the server to construct a global GMM model that serves as a global noise filter, effectively classifying samples on each local client into clean or noisy. As

depicted in Figure 5.1(b), by leveraging collaboratively learned knowledge across all clients, FedDiv demonstrates a robust capability to fit the local label noise distributions within individual clean or noisy clients. This ability enhances noise filtering performance per client, and thus reduces training instability during local training sessions, while preserving data privacy.

After label noise filtering, we remove noisy labels from the identified noisy samples on each client and relabel those samples exhibiting high prediction confidence using the pseudo-labels predicted by the global model. To further prevent local models from memorizing label noise and improve training stability, we introduce a Predictive Consistency based Sampler (PCS) for identifying credible local data for local model training. Specifically, we enforce the consistency of class labels predicted by both global and local models, and apply counterfactual reasoning [47, 153] to generate more reliable predictions for local samples.

In summary, the contributions of this paper are as follows.

- We propose a novel one-stage framework, FedDiv, for addressing the task of Federated Learning with Noisy Labels (F-LNL). To enable stable training, FedDiv learns a global noise filter by distilling the complementary knowledge from all clients while performing label noise filtering locally on every client.
- We introduce a Predictive Consistency based Sampler to perform labeled sample re-selection on every client, thereby preventing local models from memorizing label noise and further improving training stability.
- Through extensive experiments conducted on CIFAR-10 [60], CIFAR-100 [60], and Clothing1M [163] datasets, we demonstrate that FedDiv significantly outperforms state-of-the-art F-LNL methods under various label noise settings for both IID and non-IID data partitions.

5.2 Related Work

Centralized Learning with Noisy Labels (C-LNL). Diverging from conventional paradigms for centralized learning, e.g. [72, 73], which operates on training samples only with clean labels, several studies have demonstrated the effect of methods to address C-LNL in reducing model overfitting to noisy labels. For instance, JointOpt [138] proposed a joint optimization framework to correct labels of noisy samples during training by alternating between updating model parameters and labels. As well, DivideMix [76] dynamically segregated training examples with label noise into clean and noisy subsets, incorporating auxiliary semi-supervised learning algorithms for further model training. Other strategies for handling C-LNL tasks include estimating the noise transition matrix [17], reweighing the training data [127], designing robust loss functions [26], ensembling existing techniques [70], and so on.

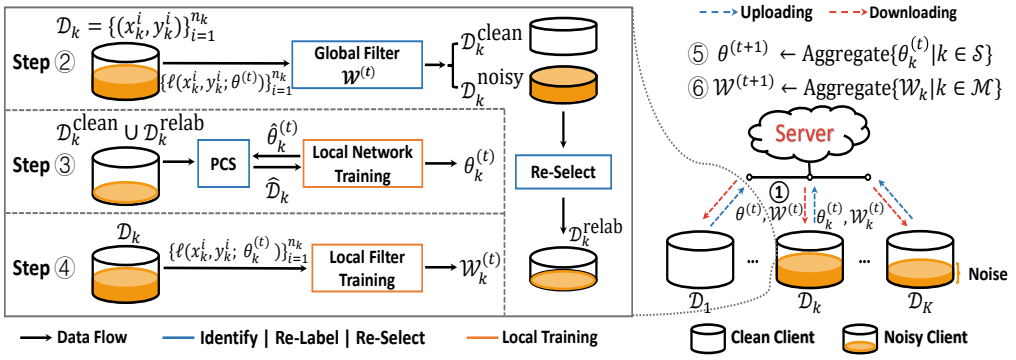


Figure 5.2: An overview of training procedure for the proposed FedDiv. In this work, the parameters of a local neural model and a local noise filter are simultaneously learned on each client during the local training sessions, while both types of parameters are aggregated on the server.

Given privacy constraints in decentralized applications, the server cannot directly access local samples of all clients to construct centralized noise filtering algorithms. Besides, a limited number of private data on local clients may also restrict the noise filtering capability. Hence, despite the success of existing C-LNL algorithms, they may no longer be feasible in federated settings [167].

Federated Learning with Noisy Labels. Numerous methods address challenges in federated scenarios with label noise. For instance, `FedRN` [58] detects clean samples in local clients using ensembled Gaussian Mixture Models (GMMs) trained to fit loss function values of local data assessed by multiple reliable neighboring client models. `RoFL` [179] directly optimizes using small-loss instances of clients during local training to mitigate label noise effects. Meanwhile, `FedCorr`[167] first introduces a dimensionality-based noise filter to segregate clients into clean and noisy groups using local intrinsic dimensionalities [96], and trains local noise filters to separate clean examples from identified noisy clients based on per-sample training losses.

However, existing F-LNL algorithms concentrate on local noise filtering, utilizing each client's private data but failing to exploit collective knowledge across clients. This limitation might compromise noise filtering efficacy, resulting in incomplete label noise removal and impacting the stability of local training sessions. Conversely, `FedDiv` proposes distilling knowledge from all clients for federated noise filtering, enhancing label noise identification in each client's sample and improving FL model training amidst label noise.

5.3 Methodology

In this section, we introduce the proposed one-stage framework named `FedDiv` for federated learning with noisy labels. In detail, we first adopt the classic FL paradigm,

Algorithm 2: The training procedure of FedDiv

Input: $\mathcal{M}$; $\{\mathcal{D}_k | k \in \mathcal{M}\}$; $\mathcal{T}$; T; ω; Initialized $\theta^{(0)}$; Initialized $\mathcal{W}^{(0)}$; Initialized $\{\mathcal{W}_k | k \in \mathcal{M}\}$; Initialized $\{\hat{p}_k | k \in \mathcal{M}\}$
Output: Global network model $\theta^{(\mathcal{T})}$

for $k \in \mathcal{M}$ **do**
  $[k] \leftarrow (\mathcal{D}_k, \hat{p}_k)$; ▷ Packaging
end for
// Step of Warm-up
for $t = 1$ to $\mathcal{T}_{wu}$ **do**
  Warm up $\theta^{(t)}$ for each client; $k \in \mathcal{M}$
end for
// Step of Usual training
for $t = 1$ to $\mathcal{T}$ **do**
  $\mathcal{S} \leftarrow$ Randomly select $\omega \times 100\%$ clients from $\mathcal{M}$;
  for $k \in \mathcal{S}$ **do**
    $(\mathcal{D}_k, \hat{p}_k) \leftarrow [k]$; ▷ Unpackaging
    Obtain $\mathcal{D}_k^{\text{clean}}$ and $\mathcal{D}_k^{\text{noisy}}$ via the federated noise filter model $\mathcal{W}^{(t)}$ using Eq. (5.7);
    Estimate $\hat{\delta}_k = |\mathcal{D}_k^{\text{noisy}}|/|\mathcal{D}_k|$;
    Obtain $\mathcal{D}_k^{\text{relab}}$ using Eq. (5.8);
    $\hat{\theta}_k^{(t)} \leftarrow \theta^{(t)}$;
    for $t' = 1$ to T **do**
      Obtain $\mathcal{D}_k^{\text{opt}}$ using Eqs. (5.9), (5.10), (5.11);
      Optimize $\hat{\theta}_k^{(t)}$ using Eq. (5.15);
    end for
    $\theta_k^{(t)} \leftarrow \hat{\theta}_k^{(t)}$;
    Update $\hat{p}_k^{(t)}$ using Eq. (5.12);
    Update $[k] \leftarrow (\mathcal{D}_k, \hat{p}_k^{(t)})$; ▷ Repackaging
    Obtain the local filter $\mathcal{W}_k^{(t)}$ using Algorithm 3;
    Upload $\theta_k^{(t)}$ and $\mathcal{W}_k^{(t)}$ to the server;
  end for
  Update the network $\theta^{(t+1)}$ using Eq. (5.1);
  for $k \in \mathcal{S}$ **do**
    $\mathcal{W}_k \leftarrow \mathcal{W}_k^{(t)}$;
  end for
  Update the global filter $\mathcal{W}^{(t+1)}$ using Eq. (5.5);
end for

namely `FedAvg` [100], to train a neural network model. On the basis of this FL framework, we propose a global filter model called Federated Noise Filter (FNF) to perform label noise filtering and noisy sample relabeling on every client. Then, to improve the stability of local training, a Predictive Consistency based Sampler (PCS) is presented to conduct labeled sample re-selection, keeping client models from memorizing label noise.

Let us consider an FL scenario with one server and K local clients denoted by $\mathcal{M}$. Each client $k \in \mathcal{M}$ has its own private data consisting of n_k sample-label pairs $\mathcal{D}_k = \{(x_k^i, y_k^i)\}_{i=1}^{n_k}$, where x_k^i is a training sample, and y_k^i is a label index over C classes. Here, we divide local clients into two groups: clean clients (with noise level $\delta_k = 0$ where $k \in \mathcal{M}$), which only have samples with clean labels, and noisy clients (with $\delta_k > 0$), whose private data might have label noise at various levels. Also, in this work, both IID and non-IID heterogeneous data partitions are considered.

As shown in Figure 5.2 and Algorithm 2, the training procedure of the t-th communication round performs the following steps:

Step ① : The server broadcasts the parameters of the global neural network model $\theta^{(t)}$ and the federated noise filtering model $\mathcal{W}^{(t)}$ to every client $k \in \mathcal{S}$, where $\mathcal{S} \subseteq \mathcal{M}$ is a subset of clients randomly selected with a fixed fraction $\omega = |\mathcal{S}|/K$ in this round.

Step ② : On every $k \in \mathcal{S}$, $\mathcal{W}^{(t)}$ is used to separate $\mathcal{D}_k$ into noisy and clean samples, and those noisy samples with high prediction confidence are assigned pseudo-labels predicted by $\theta^{(t)}$.

Step ③ (Local model training): Every client $k \in \mathcal{S}$ trains its local neural network using a subset of the clean and relabeled noisy samples selected using PCS to obtain its updated local parameters $\theta_k^{(t)}$. Here, we use $\hat{\theta}_k^{(t)}$ to denote the parameters of the local model being optimized during this training session.

Step ④ (Local filter training): Every client $k \in \mathcal{S}$ trains a local noise filter model with updated parameters $\mathcal{W}_k^{(t)}$ by fitting the per-sample loss function values of its private data $\mathcal{D}_k$. Such loss function values are evaluated using the logits of training samples in $\mathcal{D}_k$ predicted by $\theta_k^{(t)}$.

Step ⑤ : The server aggregates $\{\theta_k^{(t)} | k \in \mathcal{S}\}$ and then updates the global model as follows,

$$\theta^{(t+1)} \leftarrow \sum_{k \in \mathcal{S}} \frac{n_k}{\sum_{k \in \mathcal{S}} n_k} \theta_k^{(t)}. \tag{5.1}$$

Step ⑥ (Federated filter aggregation): The server collects $\{\mathcal{W}_k^{(t)} | k \in \mathcal{S}\}$ to update existing server-cached local filter parameters $\{\mathcal{W}_k | k \in \mathcal{S}\}$, and then aggregates all local filters $\{\mathcal{W}_k | k \in \mathcal{M}\}$ to obtain an updated federated noise filtering model $\mathcal{W}^{(t+1)}$.

This training procedure is repeated until the global model converges steadily or the predefined number of communication rounds $\mathcal{T}$ is reached. Each communication round involves local training sessions (Step ① - ④) on several randomly selected clients and the model aggregation phase (Step ⑤ - ⑥) on the server. The details of Step ② , Step ④ , and Step ⑥ are provided in Section 5.3.1, while the detailed description of Step ③ is given in Section 5.3.2 and Section 5.3.3.

5.3.1 Federated Noise Filter

Aiming at identifying label noise for the F-LNL task, we propose a Federated Noise Filter (FNF), which models the global distribution of clean and noisy samples across all clients. Motivated by [198], this FNF model can be constructed via the federated EM algorithm. Specifically, we first conduct local filter training to obtain locally estimated GMM parameters. This goal is reached by iteratively executing the standard EM algorithm [21] per client to fit its local noise distribution. Then, we perform federated filter aggregation to aggregate local GMM parameters received from all clients to construct a federated noise filter.

Local filter training. In general, samples with label noise tend to possess higher loss function values during model training, making it feasible to use mixture models to

Algorithm 3: Local filter training

Input: $\mathcal{D}_k$; $\theta_k^{(t)}$; $\mathcal{W}^{(t)}$

Output: Optimal local filter parameters $\mathcal{W}_k^{(t)}$

Initialize $\mathcal{W}_k^{(t)} = (\boldsymbol{\mu}_k^{(t)}, \boldsymbol{\sigma}_k^{(t)}, \boldsymbol{\pi}_k^{(t)})$ using $\mathcal{W}^{(t)}$

while $\mathcal{W}_k^{(t)}$ *is not converged* **do**

// E step:

$\gamma_{kg}(x, y; \theta_k^{(t)}) = \frac{\pi_{kg}^{(t)} \cdot \mathcal{N}(\ell(x,y;\theta_k^{(t)}); \mu_{kg}^{(t)}, \sigma_{kg}^{(t)})}{\sum_{g'=1}^{2} \pi_{kg'}^{(t)} \cdot \mathcal{N}(\ell(x,y;\theta_k^{(t)}); \mu_{kg'}^{(t)}, \sigma_{kg'}^{(t)})}$;

// M step:

$\mu_{kg}^{(t)} = \frac{\sum_{(x,y)\in\mathcal{D}_k} \gamma_{kg}(x,y;\theta_k^{(t)}) \cdot \ell(x,y;\theta_k^{(t)})}{\sum_{(x,y)\in\mathcal{D}_k} \gamma_{kg}(x,y;\theta_k^{(t)})}$;

$\sigma_{kg}^{(t)} = \frac{\sum_{(x,y)\in\mathcal{D}_k} \gamma_{kg}(x,y;\theta_k^{(t)}) \cdot [\ell(x,y;\theta_k^{(t)}) - \mu_{kg}^{(t)}]^2}{\sum_{(x,y)\in\mathcal{D}_k} \gamma_{kg}(x,y;\theta_k^{(t)})}$;

$\pi_{kg}^{(t)} = \frac{\sum_{(x,y)\in\mathcal{D}_k} \gamma_{kg}(x,y;\theta_k^{(t)})}{n_k}$;

end while

separate noisy samples from clean ones using per-sample loss values of the training samples [2, 76]. Therefore, for the k-th client in the t-th communication round, a local GMM can be built to model the local distribution of clean and noisy samples by fitting the per-sample loss distribution,

$$\{\ell(x, y; \theta_k^{(t)}) | (x, y) \in \mathcal{D}_k\}, \tag{5.2}$$

where $\ell(x, y; \theta_k^{(t)})$ is the cross-entropy loss of a sample-label pair $(x, y) \in \mathcal{D}_k$ when the local model $\theta_k^{(t)}$ is used for prediction. Then, we denote the two-component GMM model by $\mathcal{W}_k^{(t)} = (\boldsymbol{\mu}_k^{(t)}, \boldsymbol{\sigma}_k^{(t)}, \boldsymbol{\pi}_k^{(t)})$, where $\boldsymbol{\mu}_k^{(t)}$ and $\boldsymbol{\sigma}_k^{(t)}$ are vectors with entries $\mu_{kg}^{(t)}$ and $\sigma_{kg}^{(t)}$ denoting the mean and variance of the g-th Gaussian component, respectively. Here, we set $g = 1$ to represent the "clean" Gaussian component, i.e., the Gaussian component with a smaller mean (smaller loss), while $g = 2$ denotes the "noisy" one.

We further define a discrete variable z to represent whether a sample is clean or noisy. $\boldsymbol{\pi}_k^{(t)}$ denotes the prior distribution of z, i.e., $\pi_{kg}^{(t)} = P(z = g)$, where $\pi_{kg}^{(t)}$ should satisfy $\sum_{g=1}^{2} \pi_{kg}^{(t)} = 1$ and $0 \leq \pi_{kg}^{(t)} \leq 1$ for $g = 1, 2$. Thus, $P(\ell(x, y; \theta_k^{(t)}) | z = g)$ is modeled as a Gaussian distribution $\mathcal{N}(\ell(x, y; \theta_k^{(t)}); \mu_{kg}^{(t)}, \sigma_{kg}^{(t)})$. Then, the posterior $\gamma_{kg}(x, y; \theta_k^{(t)})$, which represents the probability of a sample x being clean ($g = 1$) or noisy ($g = 2$) given its loss value, can be computed as

$$\begin{aligned} \gamma_{kg}(x, y; \theta_k^{(t)}) &= P(z = g | x, y; \theta_k^{(t)}) \\ &= \frac{P(\ell(x, y; \theta_k^{(t)}) | z = g) P(z = g)}{\sum_{g'=1}^{2} P(\ell(x, y; \theta_k^{(t)}) | z = g') P(z = g')}. \end{aligned} \tag{5.3}$$

Afterwards, for each client k, leveraging its private data $\mathcal{D}_k$, updated local parameters $\theta_k^{(t)}$, and the federated filter parameters $\mathcal{W}^{(t)}$ received from the server, its optimal local filter parameters $\mathcal{W}_k^{(t)} = (\boldsymbol{\mu}_k^{(t)}, \boldsymbol{\sigma}_k^{(t)}, \boldsymbol{\pi}_k^{(t)})$ at current round t are derived through

the training of the local GMM models utilizing a standard EM algorithm [21], as illustrated by Algorithm 3. Noted that, $\mathcal{W}^{(t)}$ originated from the server here is used to initialize $\mathcal{W}_k^{(t)}$ for expediting convergence.

In the t-th communication round, once we have performed local filter training on the clients in $\mathcal{S}$, we upload the local filter parameters $\{\mathcal{W}_k^{(t)}|k \in \mathcal{S}\}$ to the server. Then, the server updates its cached local filter parameters corresponding to each client in $\mathcal{S}$ as follows,

$$\{\mathcal{W}_k \leftarrow \mathcal{W}_k^{(t)}|k \in \mathcal{S}\}, \tag{5.4}$$

where $\mathcal{W}_k$ is the server-cached version of the local noise filter on the k-th client. Note that `FedDiv` only sends three numerical matrices (i.e., $\boldsymbol{\mu}_k^{(t)}$, $\boldsymbol{\sigma}_k^{(t)}$, and $\boldsymbol{\pi}_k^{(t)}$) from each client to the server, and they merely reflect each client's local noise distributions instead of the raw input data, avoiding any risk of data privacy leakage.

Federated filter aggregation. After parameter uploading, the federated filter model can be constructed by aggregating the local filter parameters corresponding to all the clients $\{\mathcal{W}_k = (\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, \boldsymbol{\pi}_k)|k \in \mathcal{M}\}$ as follows,

$$\begin{aligned}
\mu_g^{(t+1)} &= \sum_{k\in\mathcal{M}} \frac{n_k}{\sum_{k\in\mathcal{M}} n_k}\mu_{kg}, \\
\sigma_g^{(t+1)} &= \sum_{k\in\mathcal{M}} \frac{n_k}{\sum_{k\in\mathcal{M}} n_k}\sigma_{kg}, \\
\pi_g^{(t+1)} &= \sum_{k\in\mathcal{M}} \frac{n_k}{\sum_{k\in\mathcal{M}} n_k}\pi_{kg},
\end{aligned} \tag{5.5}$$

where $\mathcal{W}^{(t+1)} = (\boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma}^{(t+1)}, \boldsymbol{\pi}^{(t+1)})$ will be used to perform label noise filtering on the selected clients at the beginning of the $(t+1)$-th communication round.

Label noise filtering. In the t-th communication round, once the k-th client has received the parameters of the global model $\theta^{(t)}$ and the federated filter model $\mathcal{W}^{(t)}$ from the server, the probability of a sample x from $\mathcal{D}_k$ being clean can be estimated through its posterior probability for the "clean" component as follows,

$$\mathbf{p}(\text{“clean”}|x, y; \theta^{(t)}) = P(z = 1|x, y; \theta^{(t)}). \tag{5.6}$$

Afterwards, we can divide the samples of $\mathcal{D}_k$ into a clean subset $\mathcal{D}_k^{\text{clean}}$ and a noisy subset $\mathcal{D}_k^{\text{noisy}}$ by thresholding their probabilities of being clean with the threshold 0.5 as follows,

$$\begin{aligned}
\mathcal{D}_k^{\text{clean}} &\leftarrow \{(x, y)\,|\mathbf{p}(\text{“clean”}|x, y; \theta_k^{(t)}) \geq 0.5, \forall\,(x, y) \in \mathcal{D}_k\}, \\
\mathcal{D}_k^{\text{noisy}} &\leftarrow \{(x,)\,|\mathbf{p}(\text{“clean”}|x, y; \theta_k^{(t)}) < 0.5, \forall\,(x, y) \in \mathcal{D}_k\}.
\end{aligned} \tag{5.7}$$

Noisy sample relabeling. We compute the noise level of the k-th client as $\hat{\delta}_k = |\mathcal{D}_k^{\text{noisy}}|/|\mathcal{D}_k|$,

while a client is considered a noisy one if $\hat{\delta}_k > 0.1$; and otherwise, a clean one. For an identified noisy client, we simply discard the given labels of noisy samples from $\mathcal{D}_k^{\text{noisy}}$ to prevent the model from memorizing label noise in further local training. In an effort to leverage these unlabeled (noisy) samples, we relabel those samples with high prediction confidence (by setting a confidence threshold ζ) by assigning predicted labels from the global model as follows,

$$\mathcal{D}_k^{\text{relab}} \leftarrow \{(x, \hat{y}) \,|\, \max(\mathbf{p}(x; \theta^{(t)})) \geq \zeta, \forall x \in \mathcal{D}_k^{\text{noisy}}\}, \tag{5.8}$$

where $\hat{y} = \hat{y}(x) = \arg\max \mathbf{p}(x; \theta^{(t)})$ is the pseudo-label for the sample x predicted by the global model $\theta^{(t)}$.

5.3.2 Predictive Consistency Based Sampler

During the t-th round on client k, upon obtaining the clean subset $\mathcal{D}k^{\text{clean}}$ and the relabeled subset $\mathcal{D}k^{\text{relab}}$, we integrate them into supervised local model training across T local epochs. However, the complete elimination of label noise among clients during noise filtering and relabeling is unattainable. On the other hand, relabeling inevitably introduces new label noise, causing instability in local model training, which further negatively affects the performance of the global model during aggregation. To tackle this, we propose a Predictive Consistency based Sampler (PCS) to reselect labeled samples for local training. Specifically, we observe enforcing the consistency of class labels respectively predicted by global and local models is a good strategy to achieve this goal. As training proceeds, the robustness of the model against label noise would be significantly increased (See below.), thus easily improving predictions' reliability of the samples having new label noise.

In addition, due to data heterogeneity in federated settings, especially for non-IID data partitions, local training samples owned by individual clients often belong to a smaller set of dominant classes. Thus, samples of dominant classes with newly introduced label noise would be better self-corrected during local model training, gradually leading to inconsistent predictions w.r.t those of the global model. However, such class-unbalanced local data would also contribute to the cause of the local model bias towards the dominant classes [155], which makes it more difficult for the local model to produce correct pseudo-labels for the samples that belong to minority classes. The proposed PCS strategy mitigates model bias to improve the reliability of class labels produced by local models. Here, we can de-bias the model predictions by improving causality via counterfactual reasoning [47, 117, 153], and therefore, the de-biased logit of a sample x from $\mathcal{D}_k^{\text{clean}}$ or $\mathcal{D}_k^{\text{relab}}$ is induced as follows,

$$F(x) \leftarrow f(x; \hat{\theta}_k^{(t)}) - \xi \log(\hat{p}_k), \tag{5.9}$$

where $F(x)$ is the de-biased logit later used for generating the de-biased pseudo-label, i.e., $\tilde{y}(x) = \arg\max F(x)$, and $\xi = 0.5$ is a de-bias factor. $f(x; \hat{\theta}_k^{(t)})$ is the original logit

for the sample x produced by the local model $\hat{\theta}_k^{(t)}$, currently being optimized. $\hat{p}_k$ represents the overall bias of the local model w.r.t all classes, which was previously updated according to Eq. (5.12) and cached on the k-th client during the last local training session.

Afterwards, PCS is used to re-select higher-quality and more reliable labeled training samples to perform local training as follows,

$$\mathcal{D}_k^{\text{resel}} \leftarrow \{(x, y) \,|\hat{y}(x) = \tilde{y}(x), \forall (x, y) \in \mathcal{D}_k^{\text{clean}} \cup \mathcal{D}_k^{\text{relab}}\}. \tag{5.10}$$

Similar to [167], we update the training dataset for further optimizing the local model as follows,

$$\hat{\mathcal{D}}_k = \begin{cases} \mathcal{D}_k^{\text{resel}}, & \text{if } \hat{\delta}_k \geq 0.1, \\ \mathcal{D}_k, & \text{if } \hat{\delta}_k < 0.1. \end{cases} \tag{5.11}$$

Once having the optimized $\theta_k^{(t)}$ from a local training session, we use it to update $\hat{p}_k$ with momentum as follows,

$$\hat{p}_k^{(t)} \leftarrow m\hat{p}_k + (1-m)\frac{1}{n_k}\sum_{x \in \mathcal{D}_k} \mathbf{p}(x; \theta_k^{(t)}), \tag{5.12}$$

where $m = 0.2$ is a momentum coefficient. We save $\hat{p}_k^{(t)}$ on the k-th client to update existing client-cached $\hat{p}_k$.

5.3.3 Objectives for Local Model Training

To enhance the model's robustness against label noise, we here use MixUp regularization [186] for local model training, further undermining the instability of training. Specifically, two sample-label pairs (x_i, y_i) and (x_j, y_j) from $\hat{\mathcal{D}}_k$ are augmented using linear interpolation, $\ddot{x} = \lambda x_i + (1-\lambda)x_j$ and $\ddot{y} = \lambda p_y(y_i) + (1-\lambda)p_y(y_j)$, where $\lambda \sim Beta(\alpha)$ is a mixup ratio, $\alpha = 1$ is a scalar to control its distribution, and $p_y(\cdot)$ is a function to generate a one-hot vector for a given label. Hence, on the k-th client in the t-th communication round, the local model $\hat{\theta}_k^{(t)}$ is trained with the cross-entropy loss applied to B augmented samples in one mini-batch as follows,

$$\mathcal{L}_{mix} = -\sum_{b=1}^{B} \ddot{y}_b \log \mathbf{p}(\ddot{x}_b; \hat{\theta}_k^{(t)}). \tag{5.13}$$

With the high heterogeneity of the non-IID data partitions, there could be only a limited number of categories on each client, and extensive experiments have shown that such a data partition forces a local model to predict the same class label to minimize the training loss. As in [138, 2], regularizing the average prediction of a local model over every mini-batch using a uniform prior distribution is a viable solution to

overcome the above problem, i.e.,

$$\mathcal{L}_{reg} = \sum_{c=1}^{C} \hat{\mathbf{q}}^c \log\left(\frac{\hat{\mathbf{q}}^c}{\mathbf{q}^c}\right), \text{ where } \mathbf{q} = \frac{1}{B}\sum_{b=1}^{B} \mathbf{p}(\ddot{x}_b; \hat{\theta}_k^{(t)}), \tag{5.14}$$

where $\hat{\mathbf{q}}^c = 1/C$ denotes the prior probability of a class c. $\mathbf{q}^c$ is the c-th element of the vector $\mathbf{q}$, which refers to the predicted probability of the class c averaged over B augmented training samples in a mini-batch.

Finally, on the k-th client in the t-th communication round, the overall loss function for optimizing the local model is defined as follows,

$$\mathcal{L}_{final} = \mathcal{L}_{mix} + \eta\mathcal{L}_{reg}, \tag{5.15}$$

where η is a weighting factor balancing $\mathcal{L}_{mix}$ and $\mathcal{L}_{reg}$. In the experiments, we set $\eta = 1$ when the data partition is non-IID; and otherwise, $\eta = 0$.

5.4 Experiments

5.4.1 Experimental Setups

To be fair, we here adopt the consistent experimental setups with `FedCorr` [167] to assess the efficacy of our proposed approach `FedDiv`.

Datasets and data partitions. We validate `FedDiv`'s superiority on three classic benchmark datasets, including two synthetic datasets namely CIFAR-10 [60] and CIFAR-100 [60], and one real-world noisy dataset, i.e., Clothing1M [163]. Like [167], we take both IID and non-IID data partitions into account on CIFAR-10 and CIFAR-100, but only consider non-IID data partitions on Clothing1M. Under the IID data partitions, each client is randomly assigned the same number of samples with respect to each class.

Additionally, we employ Dirichlet distribution [87] with the fixed probability p and the concentration parameter α_{Dir} to construct non-IID data partitions. Specifically, we begin by introducing an indicator matrix $\Phi \in R^{C\times K}$, and each entry Φ_{ck} determines whether the k-th client has samples from the c-th class. For every entry, we assign a 1 or 0 sampled from the Bernoulli distribution with a fixed probability p. For the row of the matrix Φ that corresponds to the c-th class, we sample a probability vector $q_c \in R^{Q_c}$ from the Dirichlet distribution with a concentration parameter $\alpha_{Dir} > 0$, where $Q_c = \sum_k \Phi_{ck}$. Then, we assign the k'-th client a $q_{ck'}$ proportion of the samples that belong to the c-th category, where k' denotes the client index with $\Phi_{ck} = 1, k = 1, \cdots, K$, and $\sum_{k'=1}^{|q_c|} q_{ck'} = 1$.

Hyper-parameter	CIFAR-10	CIFAR-100	Clothing1M
# of clients (K)	100	50	500
# of classes (C)	10	100	14
# of samples	50,000	50,000	1,000,000
Architecture	ResNet-18	ResNet-34	Pre-trained ResNet-50
Mini-batch size	10	10	16
Learning rate	0.03	0.01	0.001
$\mathcal{T}_{wu}$	5	10	2
$\mathcal{T}_{ft}$	500	450	50
$\mathcal{T}_{ut}$	450	450	50
$\mathcal{T}$	950	900	100
T	5	5	5
ω	0.1	0.1	0.02

Table 5.1: Hyper-parameter configurations of FedDiv across different datasets.

Label noise settings. Similar to [167], the noise level for the k-th client can be defined as follows:

$$\delta_k = \begin{cases} u \sim U(\tau, 1), & \text{probability of } \rho, \\ 0, & \text{probability of } 1-\rho. \end{cases} \tag{5.16}$$

Here, ρ signifies the probability of a client being noisy. For a noisy client with $\delta_k \neq 0$, the noise level is initially sampled at random from the uniform distribution $u \sim U(\tau, 1)$, with τ being its lower bound. Subsequently, $\delta_k \cdot 100\%$ of local examples are randomly selected as noisy samples, with their ground-truth labels replaced by all possible class labels.

Baselines. We compare `FedDiv` with existing state-of-the-art (SOTA) F-LNL methods, including `FedAvg` [100], `RoFL` [179], `ARFL` [179], `JointOpt` [138], `DivideMix` [76], and `FedCorr` [167]. Their experimental results reported in this paper are borrowed from [167]. Specifically, `FedAvg` is a classic FL algorithm, while `RoFL`, `ARFL` and `FedCorr` are three existing F-LNL methods. To better illustrate `FedDiv`'s superiority, we also report the classification performance of two C-LNL representatives, namely `JointOpt` and `DivideMix`, under both federated and centralized settings, respectively.

Implementations. We set ω and $\mathcal{T}$ to 950, 900 and 100, and 0.1, 0.1 and 0.02 on CIFAR-10, CIFAR-100 and Clothing1M, respectively, while we also set the confidence threshold $\zeta = 0.75$ for relabeling on all datasets. Note that, to enable faster convergence, we warm up local neural network models of each client for $\mathcal{T}_{wu}$ iterations (not training rounds; see [167]) using MixUp regularization [186]. Additionally, to be fair, most of our implementation details involving both local training and model aggregation are consistent with `FedCorr` [167] for each dataset under all federated settings and all label noise settings.

Similar to `FedCorr` [167], we select ResNet-18 [44], ResNet-34 [44] and Pre-trained ResNet-50 [44] as the network backbones for CIFAR-10, CIFAR-100 and Clothing1M, respectively. During the local model training sessions, we train each local client model over $T = 5$ local training epochs per communication round, using an SGD optimizer

with a momentum of 0.5 and a mini-batch size of 10, 10, and 16 for CIFAR-10, CIFAR-100, and Clothing1M, respectively. For each optimizer, we set the learning rate as 0.03, 0.01, and 0.001 on CIFAR-10, CIFAR-100, and Clothing1M, respectively. In addition, during data pre-processing, the training samples are first normalized and then augmented by hiring random horizontal flipping and random cropping with padding of 4. For most thresholds conducted on the experiments, we set them to default as in `FedCorr` [167], e.g. $\hat{\delta}_k = 0.1$ in Eq. (5.11), the probability of a sample being clean/noisy = 0.50 in Eq. (5.7), $\xi = 0.5$ in Eq. (5.9), etc. Additionally, we determine ζ in Eq. (5.8) using a small validation set, where $\zeta = 0.70$ meets the peak of validation accuracies.

To further enhance clarity, we present hyper-parameter summaries for each dataset in Table 5.1. Consistency in hyper-parameter settings is maintained across different label noise settings for both IID and non-IID data partitions on each dataset. It's worth noting that all experiments are conducted on the widely-used PyTorch platform[1] and executed on an NVIDIA GeForce GTX 2080Ti GPU with 12GB memory.

How to set ω, $\mathcal{T}$ and $\mathcal{T}_{wu}$. In this work, we streamlined the multi-stage F-LNL process proposed in `FedCorr` [167] into a one-stage process, avoiding the complexity of executing multiple intricate steps across different stages as in `FedCorr`. However, for fair comparisons, we maintained an equivalent number of communication rounds as in `FedCorr`. This totals $\mathcal{T}_{wu}$, encompassing federated pre-processing from `FedCorr`'s training iterations, and $\mathcal{T}$, covering both federated fine-tuning and usual training stages involving `FedCorr`. Notably, within $\mathcal{T}_{wu}$, we solely utilize MixUp regularization [186] to warm up the local neural network models for faster convergence.

Below, we will begin by introducing the multi-stage F-LNL pipeline proposed by `FedCorr`, followed by an analysis of fraction scheduling and the construction of the training rounds of `FedDiv`.

- **`FedCorr`.** `FedCorr` comprises three FL stages: federated pre-processing, federated fine-tuning, and federated usual training. During the pre-processing stage, the FL model is initially pre-trained on all clients for $\mathcal{T}_{wu}$ *iterations* (not training round) to guarantee initial convergence of model training. At the same time, `FedCorr` evaluates the quality of each client's dataset and identifies and relabels noisy samples. After this stage, a dimensionality-based filter [96] is proposed to classify clients into clean and noisy ones. In the federated fine-tuning stage, `FedCorr` only fine-tunes the global model on relatively clean clients for $\mathcal{T}_{ft}$ rounds. At the end of this stage, `FedCorr` re-evaluates and relabels the remaining noisy clients. Finally, in the federated usual training stage, the global model is trained over $\mathcal{T}_{ut}$ rounds using `FedAvg` [100] on all the clients, incorporating the labels corrected in the previous two training stages.
- **Fraction scheduling and communication rounds of `FedDiv`.** During FL training, a fixed fraction of clients will be selected at random to participate in local

[1] https://pytorch.org/

Method	Best Test Accuracy (%) ± Standard Deviation					
	ρ=0.4		ρ=0.6		ρ=0.8	
	τ=0.0	τ=0.5	τ=0.0	τ=0.5	τ=0.0	τ=0.5
FedAvg [100]	89.46±0.39	88.31±0.80	86.09±0.50	81.22±1.72	82.91±1.35	72.00±2.76
RoFL [179]	88.25±0.33	87.20±0.26	87.77±0.83	83.40±1.20	87.08±0.65	74.13±3.90
ARFL [30]	85.87±1.85	83.14±3.45	76.77±1.90	64.31±3.73	73.22±1.48	53.23±1.67
JointOpt [138]	84.42±0.70	83.01±0.88	80.82±1.19	74.09±1.43	76.13±1.15	66.16±1.71
DivideMix [76]	77.35±0.20	74.40±2.69	72.67±3.39	72.83±0.30	68.66±0.51	68.04±1.38
FedCorr [167]	94.01±0.22	94.15±0.18	92.93±0.25	92.50±0.28	91.52±0.50	90.59±0.70
FedDiv (Ours)	**94.42±0.29**	**94.30±0.19**	**93.67±0.22**	**93.41±0.21**	**92.98±0.60**	**91.44±0.25**

Table 5.2: Comparison results of FedDiv and existing SOTA methods on CIFAR-10 with IID setting at diverse noise levels.

Method	Best Test Accuracy (%) ± Standard Deviation		
	ρ=0.4	ρ=0.6	ρ=0.8
	τ=0.5	τ=0.5	τ=0.5
FedAvg [100]	64.41±1.79	53.51±2.85	44.45±2.86
RoFL [179]	59.42±2.69	46.24±3.59	36.65±3.36
ARFL [30]	51.53±4.38	33.03±1.81	27.47±1.08
JointOpt [138]	58.43±1.88	44.54±2.87	35.25±3.02
DivideMix [76]	43.25±1.01	40.72±1.41	38.91±1.25
FedCorr [167]	74.43±0.72	66.78±4.65	59.10±5.12
FedDiv (Ours)	**74.86±0.91**	**72.37±1.12**	**65.49±2.20**

Table 5.3: Comparison results of FedDiv and existing SOTA methods on CIFAR-100 with IID setting at diverse noise levels.

model training at the beginning of each round. Here, we set a fraction parameter ω to control the fraction scheduling, which is the same as the fine-tuning and usual training stages in `FedCorr`. However, during the pre-processing stage of `FedCorr`, every client must participate in local training exactly once in each iteration. Hence, any one client may be randomly sampled from all the clients with a probability of $\frac{1}{K} \cdot 100\%$ to participate in local model training, without replacement. As three stages of `FedCorr` have been merged into one in `FedDiv`, to ensure fairness in training, we convert the training iterations of the pre-processing stage into the training rounds we used, which gives us the corresponding training rounds $\mathcal{T}'_{wu} = (\mathcal{T}_{wu} \times K)/(w \times K) = \mathcal{T}_{wu}/w$. Therefore, in our work, the total number of communication rounds in the entire training process is $\mathcal{T}'_{wu} + \mathcal{T}$, where $\mathcal{T} = \mathcal{T}_{ft} + \mathcal{T}_{ut}$.

Model variants. We build the variants of `FedDiv` to evaluate the effect of the proposed noise filter as follows.

- `FedDiv(Degraded)`: Following [198], we here degrade the proposed federated noise filter by constructing the global noise filter using only the local filter parameters received in the current round instead of those from all clients.
- `FedDiv(Local filter)`: A local noise filter is trained for each client using its own private data to identify noisy labels within individual clients.

Method	Best Test Accuracy (%) ± Standard Deviation		
	p = 0.7 α_{Dir} = 10	p = 0.7 α_{Dir} = 1	p = 0.3 α_{Dir} = 10
FedAvg [100]	78.88±2.34	75.98±2.92	67.75±4.38
RoFL [179]	79.56±1.39	72.75±2.21	60.72±3.23
ARFL [30]	60.19±3.33	55.86±3.30	45.78±2.84
JointOpt [138]	72.19±1.59	66.92±1.89	58.08±2.18
DivideMix [76]	65.70±0.35	61.68±0.56	56.67±1.73
FedCorr [167]	90.52±0.89	88.03±1.08	81.57±3.68
FedDiv (Ours)	**93.18±0.42**	**91.95±0.26**	**85.31±2.28**

Table 5.4: Comparison results of FedDiv and existing SOTA methods on CIFAR-10 with non-IID setting at the noise level $(\rho, \tau) = (6.0, 0.5)$.

Dataset	**CIFAR-100**	**Clothing1M**
Noise level (ρ, τ)	(0.4, 0.0)	-
Method\(p, α_{Dir})	(0.7, 10)	-
FedAvg [100]	64.75±1.75	70.49
RoFL [179]	59.31±4.14	70.39
ARFL [30]	48.03±4.39	70.91
JointOpt [138]	59.84±1.99	71.78
DivideMix [76]	39.76±1.18	68.83
FedCorr [167]	72.73±1.02	72.55
FedDiv (Ours)	**74.47±0.34**	**72.96±0.43**

Table 5.5: Comparison results of FedDiv and existing SOTA methods on CIFAR-100 and Clothing1M under the non-IID data partitions.

5.4.2 Comparisons with State-of-the-Arts

Tables 5.2-5.5 summarize classification performance of `FedDiv` against state-of-the-art (SOTA) F-LNL methods on CIFAR-10, CIFAR-100, and Clothing1M across various noise settings for both IID and non-IID data partitions. Comparison results, based on mean accuracy and standard deviation over five trials, demonstrate `FedDiv`'s significant superiority over existing F-LNL algorithms, especially in challenging cases. For instance, in IID data partitions, Table 5.3 illustrates `FedDiv` outperforming `FedCorr` on CIFAR-100 by 6.39% in the toughest noise setting with $(\rho, \tau) = (0.8, 0.5)$. Similarly, for non-IID partitions in Table 5.4, `FedDiv` consistently surpasses `FedCorr` by 3.74% in the most challenging setting $(p, \alpha_{Dir}) = (0.3, 10)$ on CIFAR-10. Additionally, in Table 5.5, `FedDiv` exhibits a 0.41% improvement over `FedCorr` on Clothing1M, indicating its efficacy in real-world label noise distributions.

5.4.3 Ablation Analysis

To underscore the efficacy of `FedDiv`, we perform an ablation study to demonstrate the effect of each component.

Evaluation of federated noise filtering. To affirm the superiority of the proposed scheme for label noise filtering, we first compare `FedDiv` with our model variants `FedDiv(Local filter)` and `FedDiv(Degraded)`. As per Figure 5.3 and Table 5.6, the proposed noise filter exhibits a superior capacity of identifying label noise on both clean and noisy

Dataset **Noise level (ρ, τ)** **Method\(p, α_{Dir})**	**CIFAR-10** (0.6, 0.5) (0.3, 10)	**CIFAR-100** (0.4,0.0) (0.7, 10)
FedDiv (Ours)	85.31±2.28	74.47±0.34
FedDiv (Degraded)	83.22±2.61	73.06±0.93
FedDiv (Local filter)	81.34±3.65	71.37±0.76
FedDiv w/o Relab. & w/o PCS	82.17±3.06	73.09±1.89
FedDiv w/o PCS	82.83±2.59	73.66±0.96
FedDiv w/o $\mathcal{L}_{reg}$	83.60±3.65	72.43±1.29

Table 5.6: Ablation study results of FedDiv on CIFAR-10 and CIFAR-100, with varying noise levels under non-IID data partitions.

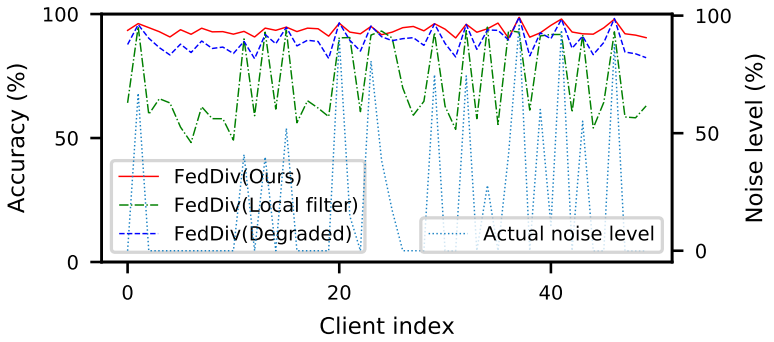


Figure 5.3: Accuracy comparisons of FedDiv in terms of noisy label identification at different clients, with the experiment conducted on CIFAR-100 with the non-IID data partition under the noise setting $(\rho, \tau) = (0.4, 0.0)$. (Best viewed zoomed in.)

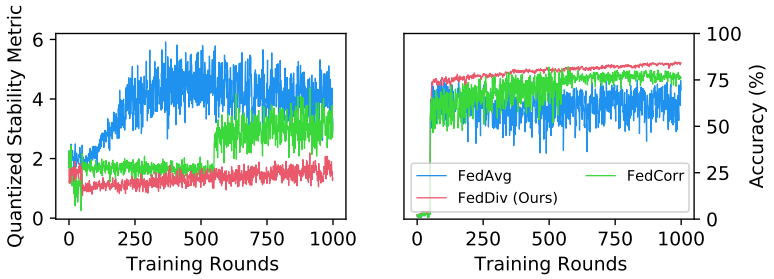


Figure 5.4: The evolution of quantized training stability *v.s.* test classification performance across epochs for various F-LNL algorithms. We quantitatively assess training stability in F-LNL by computing the average proximal regularization metric [84, 167] between the weights of local and global neural network models in the current training round. The experiments are conducted on CIFAR-10 with $(p, \alpha_{Dir}) = (0.3, 10.0)$ and $(\rho, \tau) = (6.0, 0.5)$.

clients, leading to considerably improved classification performance in comparison to its two variants.

Evaluation of relabeling and re-selection. To assess the efficacy of the proposed strategies for noisy sample relabeling and labeled sample re-selection, we systematically remove their respective components from the `FedDiv` framework. The results depicted

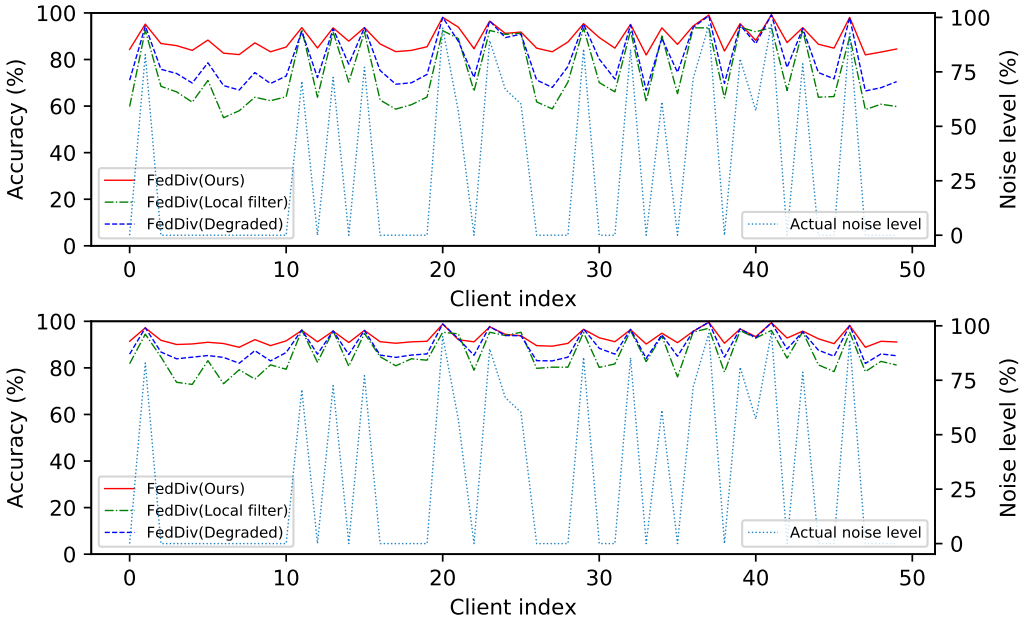


Figure 5.5: Accuracy comparisons of the proposed FedDiv in terms of noisy label identification *v.s.* different clients. The lightblue dotted line represents the actual noise level of each client, while the (dotted) lines in deep bright colors indicate the noise filtering performance with respect to different noise filters. The experiment is conducted on CIFAR-100 with the IID data partition under the noise setting $(\rho, \tau) = (0.4, 0.5)$. TOP: Evaluation in the 50-th communication round of the usual training stage; BOTTOM: Evaluation in the 500-th communication round of the usual training stage.

in Table 5.6 demonstrate a substantial decrease in accuracy across various noise settings for both types of data partitions. This indicates the importance of each individual component.

5.4.4 Further Analysis

Quantized training stability. To better grasp the motivation behind this approach, we propose using "Quantized training stability" to quantify the impact of data heterogeneity and noise heterogeneity [58, 179] on the training instability experienced during local training sessions. Technically, quantized training stability can be measured by the average proximal regularization metric between local and global model weights, denoted as $\theta_k^{(t)}$ and $\theta^{(t)}$ respectively, at round t. This is calculated by $\frac{1}{|\mathcal{S}|}\sum_{k\in\mathcal{S}}|\theta_k^{(t)} - \theta^{(t)}|^2$. As depicted in Figure 5.4, this instability results in notable discrepancies in weight divergence between local and global models, potentially hindering the performance enhancement of the aggregated model if left unaddressed. Additionally, considering the efficacy of different noise filtering strategies, our proposed federated noise filtering demonstrates superior performance in label noise identification per client, leading to decreased training instability during local training sessions and thus achieving higher classification performance of the aggregated model.

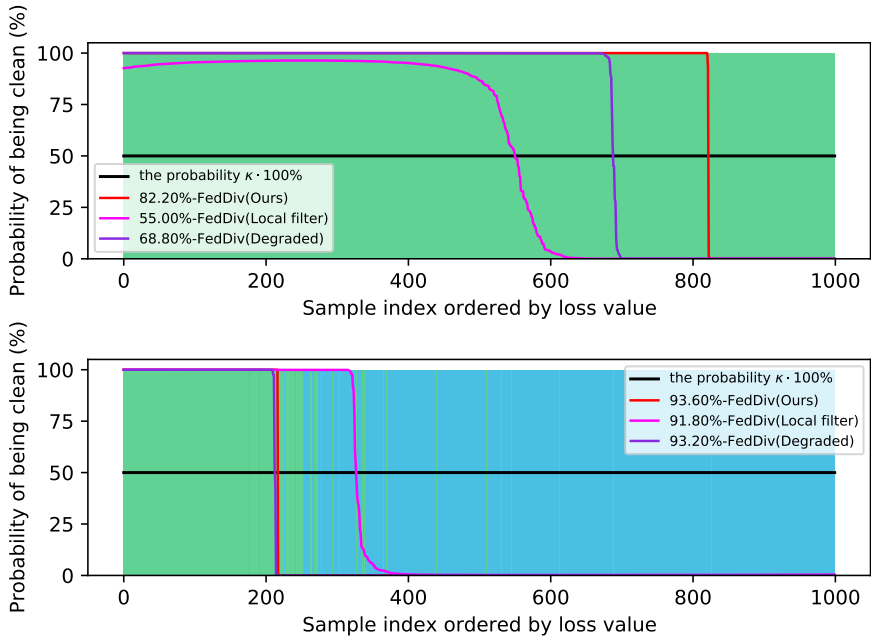


Figure 5.6: Two illustrations in performance of label noise filtering for three noise filters on a clean client (with the lowest accuracy of label noise filtering) and a noisy client. We show the probability distributions of samples being clean predicted by each filter, with the samples ranked according to the per-sample loss function values. In the legend, the percentages show the accuracy of noisy label identification with respect to each filter. In addition, as illustrated by the black line, a sample that is considered clean should have a predicted probability higher than $\kappa \cdot 100\%$. Furthermore, the green and blue bars represent, respectively, the distribution of the given **clean** and **noisy** samples. The evaluation is performed at the end of the 50-th communication round in the usual federated training stage. The experiment is conducted on CIFAR-100 with the IID data partition under the noise setting $(\rho, \tau) = (0.4, 0.5)$. TOP: Evaluation on the clean client; BOTTOM: Evaluation on the noisy client with $\delta = 0.725$. (**Best viewed zoomed in.**)

Further evaluation of federated noise filtering. To further verify the capability of our proposed label noise filtering strategy, we again compare `FedDiv` with `FedDiv(Local filter)` and `FedDiv(Degraded)` in Figure 5.5 and Figure 5.6. Both experiments are conducted on CIFAR-100 with the noise setting $(\rho, \tau) = (0.4, 0.5)$ for the IID data partition. Specifically, Figure 5.5 shows the accuracy of label noise filtering over all 50 clients at different communication rounds, while Figure 5.6 provides two examples to illustrate the noise filtering performance of different noise filters on clean and noisy clients.

As depicted in both Figure 5.5 and Figure 5.6, the proposed strategy consistently produces stronger label noise filtering capabilities on the vast majority of clients than the alternative solutions. Additionally, Figure 5.5 also shows that all these three noise filtering schemes significantly improve the label noise identification performance as model training proceeds, especially on clean clients—possibly because the network model offers greater classification performance—but ours continues to perform the best. These results once again highlight the feasibility of our proposed noise filtering strategy.

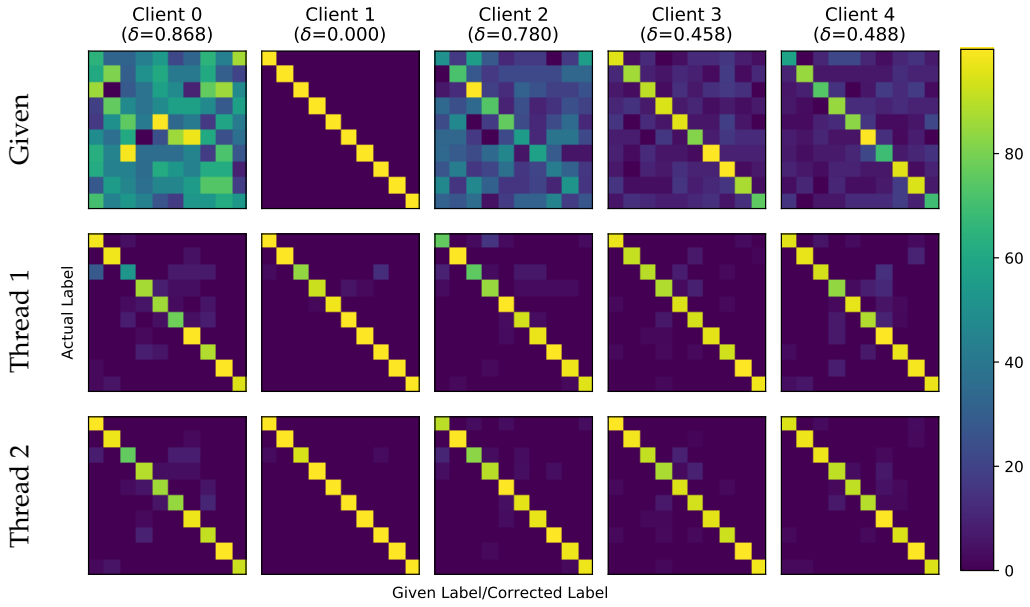
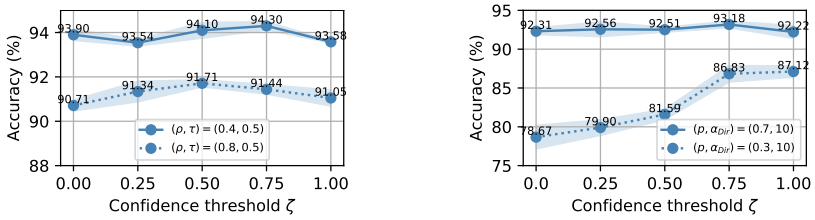


Figure 5.7: Performance evaluation of `FedDiv` in terms of label noise filtering, noisy sample relabeling and labeled sample re-selection on five representative clients. As indicated by the heat maps, three confusion matrices for each client are associated to the actual labels v.s. the given labels before processing, the corrected labels after label noise filtering and noisy sample relabeling (named Thread 1), and the corrected labels after labeled sample re-selection (named Thread 2), respectively. Note that, in practice, noisy label relabeling and labeled sample re-selection may not necessarily be conducted on clean clients (e.g., Client 1) during local model training, in accordance with Eq. (5.10). The experiment is conducted on CIFAR-10 with the IID data partition under the noise setting $(\rho, \tau) = (0.8, 0.5)$.



(a) IID (b) Non-IID

Figure 5.8: Sensitivity with respect to the hyper-parameter ζ. We show the test accuracy to illustrate the classification performance of the final FL model when we set ζ to 0.00, 0.25, 0.50, 0.75, and 1.00, respectively. We conduct these experiments on CIFAR-10 with both IID and non-IID data partitions. For the non-IID data partition, the noise level is set to $(\rho, \tau) = (0.6, 0.5)$.

Further evaluation of filtering, relabeling and re-selection. To emphasize each `FedDiv` thread's effectiveness in label noise filtering, noisy sample relabeling, and labeled sample re-selection, we compare confusion matrices before processing, after label noise filtering and noisy sample relabeling (Thread 1), and after labeled sample re-selection (Thread 2) in Figure 5.7. Figure 5.7 displays heat maps of these three confusion matrices on five representative clients. On clean or noisy clients with varying noise levels, each

thread gradually eliminates label noise, confirming the performance of each `FedDiv` component.

Hyper-parameter sensitivity. We analyze hyper-parameter sensitivity to the confidence threshold ζ. As shown in Figure 5.8, the proposed approach consistently achieves higher classification performance when ζ is set to 0.75. Therefore, $\zeta = 0.75$ is an excellent choice for setting the confidence threshold for noisy sample relabeling when training the FL model on CIFAR-10 and CIFAR-100 under different label noise settings for both IID and non-IID data partitions.

5.5 Conclusions

In this paper, we have presented `FedDiv` to handle the task of Federated Learning with Noisy Labels (F-LNL). It can effectively respond to the challenges in F-LNL tasks involving both data heterogeneity and noise heterogeneity while taking privacy concerns into account. On the basis of an FL framework, we first propose Federated Noise Filtering to separate clean samples from noisy ones on each client, thereby diminishing the instability during training. Then we perform relabeling to assign pseudo-labels to noisy samples with high predicted confidence. In addition, we introduce a Predictive Consistency based Sampler to identify credible local data for local model training, thus avoiding label noise memorization and further improving training stability. Experiments as well as comprehensive ablation analysis have revealed `FedDiv`'s superiority in handling F-LNL tasks.

Chapter 6

Conclusions and Future Works

In the ever-evolving world of machine learning, making the most of imperfect vision data is crucial for building robust systems for visual learning. This doctoral thesis explores how to effectively use flawed data when supervision is scarce or labels are uncertain. Traditional supervised learning relies on large, accurately labeled datasets, which are often not available in real-world scenarios where data may be unlabeled, partly labeled, or scattered across sources. To address these challenges, the thesis investigates innovative methods in tasks of Semi-supervised Domain Adaptation (SSDA), Learning with Noisy Labels (LNL), and Federated Learning with Noisy Labels (F-LNL), aiming for scalability and accuracy in practical applications.

These research areas focus on making the best use of imperfect vision data. Specifically, SSDA contributes to taking advantage of sparsely labeled target data for domain adaptation, often overlooked yet significant, thereby alleviating domain shifts between the source and target domains. As well, LNL tackles label noise, which can severely hamper learning if left unaddressed, while F-LNL extends these issues to federated learning settings, where data privacy and distribution complexities add further hurdles. These methods not only introduce novel theoretical concepts but also show promising practical effectiveness across different fields, demonstrating their usefulness in handling common imperfections in data for visual learning. Overall, this thesis presents new technical approaches and practical successes, offering solutions for common data imperfections in machine learning on vision tasks.

The contributions of each chapter can be summarized as follows.

1) SSDA is approached through two main methods: Cross-domain Adaptive Clustering (`CDAC`) and Graph-based Adaptive Betweenness Clustering (`G-ABC`). To be specific, the proposed `CDAC` method focuses on aligning features between clusters, using adversarial training to minimize differences between domains and improve model adaptability. Additionally, the `G-ABC` complements this by leveraging a refined graph structure to transfer semantics between domains, enhancing the ability of the model to generalize the target domains.

2) In LNL, the proposed framework termed Neighborhood Collective Estimation (`NCE`) is introduced, which mitigates label noise in the training dataset through

neighborhood collective noise verification and neighborhood collective label correction. This approach alleviates confirmation bias in candidate samples, thereby enhancing their predictive reliability, while preserving accuracy despite label noise.

3) The challenges of F-LNL are addressed with the proposed `FedDiv`. This method proposes federated noise filtering and predictive consistency based sampling to ensure robust learning across decentralized datasets while protecting data privacy. It is particularly useful in real-world applications.

Looking ahead, this research sets the stage for advancements in machine learning with imperfect data for vision tasks. These methods could be applied to other areas, such as unsupervised and reinforcement learning, where data imperfections are common. Integrating these approaches into broader machine learning frameworks could make visual learning systems more adaptable and efficient across various applications.

Furthermore, future research could explore the integration of large-scale language [143, 110], vision-language [125, 53], or multi-modal models [89, 199] to complement the methods developed here. By leveraging these advanced models, it may be possible to build even stronger visual learning systems capable of handling real-world data complexities with greater accuracy and efficiency.

In summary, this thesis presents valuable technical advances and practical applications across different fields. Its unified approach to handling imperfect data offers a promising path toward developing more adaptable, efficient, and robust visual learning systems capable of overcoming real-world challenges, while also opening up exciting avenues for future research.

Bibliography

[1] G. Algan and I. Ulusoy. "Image classification with deep learning in the presence of noisy labels: A survey". In: *Knowledge-Based Systems* 215 (2021), p. 106771.

[2] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness. "Unsupervised label noise modeling and loss correction". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 312–321.

[3] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. "Pseudo-labeling and confirmation bias in deep semi-supervised learning". In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–8.

[4] M. Awais, F. Zhou, H. Xu, L. Hong, P. Luo, S.-H. Bae, and Z. Li. "Adversarial robustness for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8568–8577.

[5] D. Bahri, H. Jiang, and M. Gupta. "Deep k-NN for noisy labels". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 540–550.

[6] Y. Bai, C. Wang, Y. Lou, J. Liu, and L.-Y. Duan. "Hierarchical connectivity-centered clustering for unsupervised domain adaptation on person re-identification". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 6715–6729.

[7] N. Bendre, H. T. Marín, and P. Najafirad. "Learning from few samples: A survey". In: *arXiv preprint arXiv:2007.15484* (2020).

[8] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. "Mixmatch: A holistic approach to semi-supervised learning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 5049–5059.

[9] S. Bickel, M. Brückner, and T. Scheffer. "Discriminative learning for differing training and test distributions". In: *Proceedings of the 24th international conference on Machine learning*. 2007, pp. 81–88.

[10] W. Boulila, M. Sellami, M. Driss, M. Al-Sarem, M. Safaei, and F. A. Ghaleb. "RS-DCNN: A novel distributed convolutional-neural-networks based-approach for big remote-sensing image classification". In: *Computers and Electronics in Agriculture* 182 (2021), p. 106014.

[11] Z. Cao, L. Ma, M. Long, and J. Wang. "Partial adversarial domain adaptation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[12] Z. Cao, L. Ma, M. Long, and J. Wang. "Partial adversarial domain adaptation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 135–150.

[13] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang. "Progressive feature alignment for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 627–636.

[14] M. Chen, S. Zhao, H. Liu, and D. Cai. "Adversarial-learned loss for domain adaptation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 2020, pp. 3521–3528.

[15] S. Chen, M. Harandi, X. Jin, and X. Yang. "Semi-supervised domain adaptation via asymmetric joint distribution matching". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[16] Y. Chen, X. Shen, S. X. Hu, and J. A. Suykens. "Boosting co-teaching with compression regularization for label noise". In: *CVPR Learning from Limited and Imperfect Data (L2ID) workshop*. 2021.

[17] D. Cheng, T. Liu, Y. Ning, N. Wang, B. Han, G. Niu, X. Gao, and M. Sugiyama. "Instance-dependent label-noise learning with manifold-regularized transition matrix estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16630–16639.

[18] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. "Autoaugment: Learning augmentation strategies from data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 113–123.

[19] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. "Randaugment: Practical automated data augmentation with a reduced search space. 2020 IEEE". In: *CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 3008–3017.

[20] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. "Randaugment: Practical automated data augmentation with a reduced search space". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[22] Z. Deng, Y. Luo, and J. Zhu. "Cluster alignment with a teacher for unsupervised domain adaptation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 9944–9953.

[23] T. DeVries and G. W. Taylor. *Improved Regularization of Convolutional Neural Networks with Cutout*. 2017. arXiv: 1708.04552 [cs.CV].

[24] Z. Ding, S. Li, M. Shao, and Y. Fu. "Graph adaptive knowledge transfer for unsupervised domain adaptation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 37–52.

[25] E. Englesson and H. Azizpour. "Consistency Regularization Can Improve Robustness to Label Noise". In: *arXiv preprint arXiv:2110.01242* (2021).

[26] E. Englesson and H. Azizpour. "Generalized jensen-shannon divergence loss for learning with noisy labels". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 30284–30297.

[27] X. Fang and M. Ye. "Robust Federated Learning With Noisy and Heterogeneous Clients". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10072–10081.

[28] Z. Fang, J. Lu, F. Liu, and G. Zhang. "Semi-Supervised Heterogeneous Domain Adaptation: Theory and Algorithms". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2023), pp. 1087–1105. DOI: 10.1109/TPAMI.2022.3146234.

[29] Z. Feng, Q. Zhou, G. Cheng, X. Tan, J. Shi, and L. Ma. "Semi-Supervised Semantic Segmentation via Dynamic Self-Training and Class-Balanced Curriculum". In: *arXiv preprint arXiv:2004.08514* (2020).

[30] S. Fu, C. Xie, B. Li, and Q. Chen. "Attack-resistant federated learning with residual-based reweighting". In: *arXiv preprint arXiv:1912.11464* (2019).

[31] Y. Ganin and V. Lempitsky. "Unsupervised domain adaptation by backpropagation". In: *International conference on machine learning*. PMLR. 2015, pp. 1180–1189.

[32] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. "Domain-adversarial training of neural networks". In: *The journal of machine learning research* 17.1 (2016), pp. 2096–2030.

[33] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez. "A survey on deep learning techniques for image and video semantic segmentation". In: *Applied Soft Computing* 70 (2018), pp. 41–65.

[34] J. Ghosn and Y. Bengio. "Bias learning, knowledge sharing". In: *IEEE Transactions on Neural Networks* 14.4 (2003), pp. 748–765.

[35] B. Gong, Y. Shi, F. Sha, and K. Grauman. "Geodesic flow kernel for unsupervised domain adaptation". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 2066–2073.

[36] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang. "Label propagation via teaching-to-learn and learning-to-teach". In: *IEEE transactions on neural networks and learning systems* 28.6 (2016), pp. 1452–1465.

[37] Y. Grandvalet and Y. Bengio. "Semi-supervised learning by entropy minimization". In: *Advances in neural information processing systems*. 2005, pp. 529–536.

[38] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. "A kernel two-sample test". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 723–773.

[39] X. Gu, J. Sun, and Z. Xu. "Spherical Space Domain Adaptation With Robust Pseudo-Label Loss". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9101–9110.

[40] B. Gunel, J. Du, A. Conneau, and V. Stoyanov. "Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning". In: *International Conference on Learning Representations*. 2020.

[41] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. "Co-teaching: Robust training of deep neural networks with extremely noisy labels". In: *NeurIPS*. 2018, pp. 8535–8545.

[42] K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman. "Automatically Discovering and Learning New Visual Categories with Ranking Statistics". In: *International Conference on Learning Representations*. 2020.

[43] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[44] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[45] K. He, X. Zhang, S. Ren, and J. Sun. "Identity mappings in deep residual networks". In: *European conference on computer vision*. Springer. 2016, pp. 630–645.

[46] S. Herath, M. Harandi, and F. Porikli. "Learning an invariant hilbert space for domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3845–3854.

[47] P. W. Holland. "Statistics and causal inference". In: *Journal of the American statistical Association* 81.396 (1986), pp. 945–960.

[48] D. Huang, J. Li, W. Chen, J. Huang, Z. Chai, and G. Li. "Divide and Adapt: Active Domain Adaptation via Customized Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7651–7660.

[49] L. Huang, C. Zhang, and H. Zhang. "Self-adaptive training: beyond empirical risk minimization". In: *Advances in Neural Information Processing Systems* 33 (2020).

[50] Z. Huang, G. Niu, X. Liu, W. Ding, X. Xiao, H. Wu, and X. Peng. "Learning with Noisy Correspondence for Cross-modal Matching". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 29406–29419.

[51] Z. Huang, K. Sheng, W. Dong, X. Mei, C. Ma, F. Huang, D. Zhou, and C. Xu. "Effective Label Propagation for Discriminative Semi-Supervised Domain Adaptation". In: *CoRR* abs/2012.02621 (2020). arXiv: 2012.02621.

[52] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa. "Cross-domain weakly-supervised object detection through progressive domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5001–5009.

[53] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. "Scaling up visual and vision-language representation learning with noisy text supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 4904–4916.

[54] L. Jiang, D. Huang, M. Liu, and W. Yang. "Beyond synthetic noise: Deep learning on controlled noisy labels". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4804–4815.

[55] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei. "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2304–2313.

[56] P. Jiang, A. Wu, Y. Han, Y. Shao, and B. Li. "Bidirectional Adversarial Training for Semi-Supervised Domain Adaptation". In: *Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence IJCAI-PRICAI-20*. 2020.

[57] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann. "Contrastive adaptation network for unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4893–4902.

[58] S. Kim, W. Shin, S. Jang, H. Song, and S.-Y. Yun. "FedRN: Exploiting k-Reliable Neighbors Towards Robust Federated Learning". In: *arXiv preprint arXiv:2205.01310* (2022).

[59] T. Kim and C. Kim. "Attract, Perturb, and Explore: Learning a Feature Alignment Network for Semi-supervised Domain Adaptation". In: *European Conference on Computer Vision*. Springer. 2020, pp. 591–607.

[60] A. Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: *Master's thesis, University of Tront* (2009).

[61] A. Krizhevsky, I. Sutskever, and G. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *NIPS*. 2012.

[62] A. Kumagai and T. Iwata. "Unsupervised domain adaptation by matching distributions based on the maximum mean discrepancy via unilateral transformations". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 4106–4113.

[63] Y. Kun and W. Jianxin. "Probabilistic End-to-end Noise Correction for Learning with Noisy Labels". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[64] A. Kurakin, C.-L. Li, C. Raffel, D. Berthelot, E. D. Cubuk, H. Zhang, K. Sohn, N. Carlini, and Z. Zhang. "FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence". In: *Advances in Neural Information Processing Systems*. 2020.

[65] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, et al. "The open images dataset v4". In: *International Journal of Computer Vision* 128.7 (2020), pp. 1956–1981.

[66] S. Laine and T. Aila. "Temporal Ensembling for Semi-Supervised Learning". In: *Proc. International Conference on Learning Representations (ICLR)*. 2017.

[67] D.-H. Lee. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks". In: *Workshop on challenges in representation learning, ICML*. Vol. 3. 2013.

[68] B. Li, Y. Wang, S. Zhang, D. Li, T. Darrell, K. Keutzer, and H. Zhao. "Learning Invariant Representations and Risks for Semi-supervised Domain Adaptation". In: *arXiv preprint arXiv:2010.04647* (2020).

[69] D. Li and T. Hospedales. "Online Meta-Learning for Multi-Source and Semi-Supervised Domain Adaptation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020.

[70] J. Li, G. Li, F. Liu, and Y. Yu. "Neighborhood collective estimation for noisy label identification and correction". In: *European Conference on Computer Vision*. Springer. 2022.

[71] J. Li, G. Li, Y. Shi, and Y. Yu. "Cross-Domain Adaptive Clustering for Semi-Supervised Domain Adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 2505–2514.

[72] J. Li, G. Li, and Y. Yu. "Adaptive Betweenness Clustering for Semi-Supervised Domain Adaptation". In: *IEEE Transactions on Image Processing* (2023).

[73] J. Li, G. Li, and Y. Yu. "Inter-Domain Mixup for Semi-Supervised Domain Adaptation". In: *Pattern Recognition* (2023).

[74] J. Li, S. Wu, C. Liu, Z. Yu, and H.-S. Wong. "Semi-supervised deep coupled ensemble learning with classification landmark exploration". In: *IEEE Transactions on Image Processing* 29 (2019), pp. 538–550.

[75] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen. "Heterogeneous domain adaptation through progressive alignment". In: *IEEE transactions on neural networks and learning systems* 30.5 (2018), pp. 1381–1391.

[76] J. Li, R. Socher, and S. C. Hoi. "DivideMix: Learning with Noisy Labels as Semi-supervised Learning". In: *International Conference on Learning Representations*. 2019.

[77] J. Li, R. Socher, and S. C. Hoi. "Dividemix: Learning with noisy labels as semi-supervised learning". In: *arXiv preprint arXiv:2002.07394* (2020).

[78] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli. "Learning to learn from noisy labeled data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5051–5059.

[79] J. Li, C. Xiong, and S. C. Hoi. "Learning From Noisy Data With Robust Representation Learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 9485–9494.

[80] J. Li, J. Pei, and H. Huang. "Communication-Efficient Robust Federated Learning with Noisy Labels". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, pp. 914–924.

[81] K. Li, C. Liu, H. Zhao, Y. Zhang, and Y. Fu. "ECACL: A holistic framework for semi-supervised domain adaptation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8578–8587.

[82] L. Li, J. Wang, J. Li, Q. Ma, and J. Wei. "Relation classification via keyword-attentive sentence mechanism and synthetic stimulation loss". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.9 (2019), pp. 1392–1404.

[83] S. Li, M. Xie, F. Lv, C. H. Liu, J. Liang, C. Qin, and W. Li. "Semantic concentration for domain adaptation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9102–9111.

[84] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. "Federated optimization in heterogeneous networks". In: *Proceedings of Machine Learning and Systems* 2 (2020), pp. 429–450.

[85] W. Li, F. Li, Y. Luo, P. Wang, et al. "Deep domain adaptive object detection: A survey". In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2020, pp. 1808–1813.

[86] W. Li, L. Wang, W. Li, E. Agustsson, and L. V. Gool. "WebVision Database: Visual Learning and Understanding from Web Data". In: *Arxiv Preprint* (2017).

[87] T. Lin, L. Kong, S. U. Stich, and M. Jaggi. "Ensemble distillation for robust model fusion in federated learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 2351–2363.

[88] B. Liu, J. Jiao, and Q. Ye. "Harmonic Feature Activation for Few-Shot Semantic Segmentation". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 3142–3153.

[89] H. Liu, C. Li, Q. Wu, and Y. J. Lee. "Visual Instruction Tuning". In: *Advances in Neural Information Processing Systems*. Vol. 36. Curran Associates, Inc., 2023, pp. 34892–34916.

[90] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda. "Early-Learning Regularization Prevents Memorization of Noisy Labels". In: *Advances in Neural Information Processing Systems* 33 (2020).

[91] Y. Liu, J. Lee, M. Park, S. Kim, and Y. Yang. "Transductive Propagation Network for Few-shot Learning". In: *CoRR* abs/1805.10002 (2018).

[92] M. Long, Y. Cao, J. Wang, and M. Jordan. "Learning transferable features with deep adaptation networks". In: *International conference on machine learning*. PMLR. 2015, pp. 97–105.

[93] M. Long, Z. Cao, J. Wang, and M. I. Jordan. "Conditional adversarial domain adaptation". In: *Advances in neural information processing systems* 31 (2018).

[94] Q. Luo, Z. Liu, L. Hong, C. Li, K. Yang, L. Wang, F. Zhou, G. Li, Z. Li, and J. Zhu. "Relaxed Conditional Image Transfer for Semi-supervised Domain Adaptation". In: *arXiv preprint arXiv:2101.01400* (2021).

[95] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang. "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2507–2516.

[96] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. Erfani, S. Xia, S. Wijewickrema, and J. Bailey. "Dimensionality-driven learning with noisy labels". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3355–3364.

[97] Z. Ma and A. Leijon. "Bayesian estimation of beta mixture models with variational inference". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.11 (2011), pp. 2160–2173.

[98] L. v. d. Maaten and G. Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[99] E. Malach and S. Shalev-Shwartz. “Decoupling" when to update" from" how to update"”. In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 960–970.

[100] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. “Communication-efficient learning of deep networks from decentralized data”. In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[101] A. Mikołajczyk and M. Grochowski. “Data augmentation for improving deep learning in image classification problem”. In: *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE. 2018, pp. 117–122.

[102] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Vol. 26. Curran Associates, Inc., 2013.

[103] S. Mishra, K. Saenko, and V. Saligrama. “Surprisingly Simple Semi-Supervised Domain Adaptation with Pretraining and Consistency”. In: *arXiv preprint arXiv:2101.12727* (2021).

[104] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto. “Few-shot adversarial domain adaptation”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 6673–6683.

[105] C. H. Nguyen and H. Mamitsuka. “Discriminative graph embedding for label propagation”. In: *IEEE transactions on neural networks* 22.9 (2011), pp. 1395–1405.

[106] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang. “Federated learning for smart healthcare: A survey”. In: *ACM Computing Surveys (CSUR)* 55.3 (2022), pp. 1–37.

[107] J. Ni, Q. Qiu, and R. Chellappa. “Subspace interpolation via dictionary learning for unsupervised domain adaptation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 692–699.

[108] K. Nishi, Y. Ding, A. Rich, and T. Hollerer. “Augmentation strategies for learning with noisy labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8022–8031.

[109] A. v. d. Oord, Y. Li, and O. Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018).

[110] OpenAI. *GPT-3.5B*. Tech. rep. 2021. URL: https://openai.com/research/gpt-3-5.

[111] D. Ortego, E. Arazo, P. Albert, N. E. O’Connor, and K. McGuinness. “Multi-Objective Interpolation Training for Robustness to Label Noise”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 6606–6615.

[112] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon. “Unsupervised Intra-domain Adaptation for Semantic Segmentation through Self-Supervision”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3764–3773.

[113] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. "Domain adaptation via transfer component analysis". In: *IEEE Transactions on Neural Networks* 22.2 (2010), pp. 199–210.

[114] S. J. Pan and Q. Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[115] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei. "Transferrable prototypical networks for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2239–2247.

[116] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. 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*. 2017, pp. 1944–1952.

[117] J. Pearl. "Causal inference in statistics: An overview". In: *Statistics surveys* 3 (2009), pp. 96–146.

[118] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. "Moment matching for multi-source domain adaptation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1406–1415.

[119] H. Permuter, J. Francos, and I. Jermyn. "A study of Gaussian mixture models of color and texture features for image classification and segmentation". In: *Pattern recognition* 39.4 (2006), pp. 695–706.

[120] M. Pilanci and E. Vural. "Domain adaptation on graphs by learning aligned graph bases". In: *IEEE Transactions on Knowledge and Data Engineering* (2020).

[121] M. Pilancı and E. Vural. "Domain Adaptation on Graphs via Frequency Analysis". In: *2019 27th Signal Processing and Communications Applications Conference (SIU)*. IEEE. 2019, pp. 1–4.

[122] C. Qin, L. Wang, Q. Ma, Y. Yin, H. Wang, and Y. Fu. "Contradictory Structure Learning for Semi-supervised Domain Adaptation". In: *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM. 2021, pp. 576–584.

[123] C. Qin, L. Wang, Q. Ma, Y. Yin, H. Wang, and Y. Fu. "Opposite Structure Learning for Semi-supervised Domain Adaptation". In: *arXiv preprint arXiv:2002.02545* (2020).

[124] C. Qin, L. Wang, Q. Ma, Y. Yin, H. Wang, and Y. Fu. "Semi-Supervised Domain Adaptive Structure Learning". In: *IEEE Transactions on Image Processing* 31 (2022), pp. 7179–7190. DOI: 10.1109/TIP.2022.3215889.

[125] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[126] S.-A. Rebuffi, S. Ehrhardt, K. Han, A. Vedaldi, and A. Zisserman. "Lsd-c: Linearly separable deep clusters". In: *arXiv preprint arXiv:2006.10039* (2020).

[127] M. Ren, W. Zeng, B. Yang, and R. Urtasun. "Learning to reweight examples for robust deep learning". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4334–4343.

[128] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. "Adapting visual category models to new domains". In: *European conference on computer vision*. Springer. 2010, pp. 213–226.

[129] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko. "Semi-supervised domain adaptation via minimax entropy". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 8050–8058.

[130] K. Saito, D. Kim, S. Sclaroff, and K. Saenko. "Universal domain adaptation through self supervision". In: *arXiv preprint arXiv:2002.07953* (2020).

[131] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. "Maximum classifier discrepancy for unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3723–3732.

[132] R. Shu, H. H. Bui, H. Narui, and S. Ermon. "A DIRT-T APPROACH TO UNSUPERVISED DOMAIN ADAPTATION". In: *Proc. 6th International Conference on Learning Representations*. 2018.

[133] A. Singh. "CLDA: Contrastive Learning for Semi-Supervised Domain Adaptation". In: *Advances in Neural Information Processing Systems* 34 (2021).

[134] H. Song, M. Kim, and J.-G. Lee. "Selfie: Refurbishing unclean samples for robust deep learning". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5907–5915.

[135] J.-C. Su, Y.-H. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker. "Active adversarial domain adaptation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 739–748.

[136] B. Sun and K. Saenko. "Deep coral: Correlation alignment for deep domain adaptation". In: *European conference on computer vision*. Springer. 2016, pp. 443–450.

[137] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[138] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa. "Joint optimization framework for learning with noisy labels". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5552–5560.

[139] H. Tang and K. Jia. "Discriminative Adversarial Domain Adaptation." In: *AAAI*. 2020, pp. 5940–5947.

[140] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman. "Learning from noisy labels by regularized estimation of annotator confusion". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 11244–11253.

[141] C. Tao, F. Lv, L. Duan, and M. Wu. "Minimax entropy network: Learning category-invariant features for domain adaptation". In: *arXiv preprint arXiv:1904.09601* (2019).

[142] A. Tarvainen and H. Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results".

In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 1195–1204.

[143] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023).

[144] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. "Adversarial Discriminative Domain Adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[145] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. "Adversarial discriminative domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7167–7176.

[146] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. "Deep domain confusion: Maximizing for domain invariance". In: *arXiv preprint arXiv:1412.3474* (2014).

[147] L. Van der Maaten and G. Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[148] J. E. Van Engelen and H. H. Hoos. "A survey on semi-supervised learning". In: *Machine Learning* 109.2 (2020), pp. 373–440.

[149] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. "Deep hashing network for unsupervised domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5018–5027.

[150] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez. "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019, pp. 2517–2526.

[151] F. Wang and C. Zhang. "Label propagation through linear neighborhoods". In: *IEEE Transactions on Knowledge and Data Engineering* 20.1 (2007), pp. 55–67.

[152] L. Wang, Z. Ding, and Y. Fu. "Adaptive graph guided embedding for multi-label annotation". In: *IJCAI*. 2018.

[153] X. Wang, Z. Wu, L. Lian, and S. X. Yu. "Debiased Learning from Naturally Imbalanced Pseudo-Labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14647–14657.

[154] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S.-T. Xia. "Iterative learning with open-set noisy labels". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8688–8696.

[155] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang. "Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 10857–10866.

[156] G. Wilson and D. J. Cook. "A survey of unsupervised deep domain adaptation". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 11.5 (2020), pp. 1–46.

[157] M. Wu, S. Pan, C. Zhou, X. Chang, and X. Zhu. "Unsupervised domain adaptive graph convolutional networks". In: *Proceedings of The Web Conference 2020*. 2020, pp. 1457–1467.

[158] P. Wu, S. Zheng, M. Goswami, D. N. Metaxas, and C. Chen. "A Topological Filter for Learning with Label Noise". In: *Advances in neural information processing systems* 33 (2020).

[159] S. Wu, G. Deng, J. Li, R. Li, Z. Yu, and H.-S. Wong. "Enhancing TripleGAN for semi-supervised conditional instance synthesis and classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10091–10100.

[160] S. Wu, J. Li, C. Liu, Z. Yu, and H.-S. Wong. "Mutual learning of complementary networks via residual correction for improving semi-supervised classification". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6500–6509.

[161] Z. F. Wu, T. Wei, J. Jiang, C. Mao, M. Tang, and Y. F. Li. "NGC: A Unified Framework for Learning with Open-World Noisy Data". In: (2021).

[162] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. "Learning from massive noisy labeled data for image classification". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2691–2699.

[163] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. "Learning from massive noisy labeled data for image classification". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2691–2699.

[164] W. Xiao, Z. Ding, and H. Liu. "Implicit Semantic Response Alignment for Partial Domain Adaptation". In: *Advances in Neural Information Processing Systems* 34 (2021).

[165] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le. "Unsupervised data augmentation for consistency training". In: *arXiv preprint arXiv:1904.12848* (2019).

[166] X. Xiong, S. Li, and G. Li. "Unpaired Image-to-Image Translation Based Domain Adaptation for Polyp Segmentation". In: *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2023, pp. 1–5.

[167] J. Xu, Z. Chen, T. Q. Quek, and K. F. E. Chong. "FedCorr: Multi-Stage Federated Learning for Label Noise Correction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10184–10193.

[168] R. Xu, G. Li, J. Yang, and L. Lin. "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1426–1435.

[169] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo. "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2272–2281.

[170] P. Yan, Z. Wu, M. Liu, K. Zeng, L. Lin, and G. Li. "Unsupervised domain adaptive salient object detection through uncertainty-aware pseudo-label learning".

In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 3. 2022, pp. 3000–3008.

[171] Z. Yan, Y. Wu, G. Li, Y. Qin, X. Han, and S. Cui. "Multi-level Consistency Learning for Semi-supervised Domain Adaptation". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Ed. by L. D. Raedt. Main Track. International Joint Conferences on Artificial Intelligence Organization, July 2022, pp. 1530–1536.

[172] E. Yang, D. Yao, T. Liu, and C. Deng. "Mutual Quantization for Cross-Modal Search With Noisy Labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7551–7560.

[173] J. Yang, R. Xu, R. Li, X. Qi, X. Shen, G. Li, and L. Lin. "An adversarial perturbation oriented domain adaptation approach for semantic segmentation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 12613–12620.

[174] J. Yang, C. Li, W. An, H. Ma, Y. Guo, Y. Rong, P. Zhao, and J. Huang. "Exploring robustness of unsupervised domain adaptation in semantic segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9194–9203.

[175] L. Yang, B. Tan, V. W. Zheng, K. Chen, and Q. Yang. "Federated recommendation systems". In: *Federated Learning*. Springer, 2020, pp. 225–239.

[176] L. Yang, Y. Wang, M. Gao, A. Shrivastava, K. Q. Weinberger, W.-L. Chao, and S.-N. Lim. "Deep co-training with task decomposition for semi-supervised domain adaptation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8906–8916.

[177] L. Yang, Y. Wang, M. Gao, A. Shrivastava, K. Q. Weinberger, W.-L. Chao, and S.-N. Lim. "MiCo: Mixup Co-Training for Semi-Supervised Domain Adaptation". In: *arXiv preprint arXiv:2007.12684* (2020).

[178] M. Yang, Z. Huang, P. Hu, T. Li, J. Lv, and X. Peng. "Learning with Twin Noisy Labels for Visible-Infrared Person Re-Identification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14308–14317.

[179] S. Yang, H. Park, J. Byun, and C. Kim. "Robust federated learning with noisy labels". In: *IEEE Intelligent Systems* 37.2 (2022), pp. 35–43.

[180] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui. "Generalized Source-free Domain Adaptation". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 8958–8967. DOI: 10.1109/ICCV48922.2021.00885.

[181] S. Yang, G. Song, Y. Jin, and L. Du. "Domain adaptive classification on heterogeneous information networks". In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 2021, pp. 1410–1416.

[182] S.-B. Yang and T.-L. Yu. "Pseudo-Representation Labeling Semi-Supervised Learning". In: *arXiv* (2020), arXiv–2006.

[183] Y. Yao, Z. Sun, C. Zhang, F. Shen, Q. Wu, J. Zhang, and Z. Tang. "Jo-SRC: A Contrastive Approach for Combating Noisy Labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 5192–5201.

[184] J. Yoon, D. Kang, and M. Cho. "Semi-supervised domain adaptation via sample-to-sample self-distillation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 1978–1987.

[185] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama. "How does disagreement help generalization against label corruption?" In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7164–7173.

[186] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. "mixup: Beyond Empirical Risk Minimization". In: *International Conference on Learning Representations*. 2018.

[187] Y. Zhang, G. Song, L. Du, S. Yang, and Y. Jin. "Dane: Domain adaptive network embedding". In: *arXiv preprint arXiv:1906.00684* (2019).

[188] Z. Zhang, W. Chen, H. Cheng, Z. Li, S. Li, L. Lin, and G. Li. "Divide and contrast: Source-free domain adaptation via adaptive contrastive learning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 5137–5149.

[189] G. Zhao, G. Li, R. Xu, and L. Lin. "Collaborative training between region proposal localization and classification for domain adaptive object detection". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer. 2020, pp. 86–102.

[190] H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J. Gordon. "Adversarial Multiple Source Domain Adaptation". In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018, pp. 8559–8570.

[191] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu. "Object detection with deep learning: A review". In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232.

[192] E. Zheltonozhskii, C. Baskin, A. Mendelson, A. M. Bronstein, and O. Litany. "Contrast to Divide: Self-Supervised Pre-Training for Learning with Noisy Labels". In: *arXiv preprint arXiv:2103.13646* (2021).

[193] Z. Zheng, Y. Zhou, Y. Sun, Z. Wang, B. Liu, and K. Li. "Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges". In: *Connection Science* 34.1 (2022), pp. 1–28.

[194] L. Zhong, Z. Fang, F. Liu, J. Lu, B. Yuan, and G. Zhang. "How does the combined risk affect the performance of unsupervised domain adaptation approaches?" In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12. 2021, pp. 11079–11087.

[195] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. "Learning with local and global consistency". In: *Advances in neural information processing systems* 16 (2003).

[196] H.-Y. Zhou, X. Chen, Y. Zhang, R. Luo, L. Wang, and Y. Yu. "Generalized radiograph representation learning via cross-supervision between images and free-text radiology reports". In: *Nature Machine Intelligence* 4.1 (2022), pp. 32–40.

[197] H.-Y. Zhou, Y. Yu, C. Wang, S. Zhang, Y. Gao, J. Pan, J. Shao, G. Lu, K. Zhang, and W. Li. "A transformer-based representation-learning model with unified processing of multimodal input for clinical diagnostics". In: *Nature Biomedical Engineering* (2023), pp. 1–13.

[198] C. Zhu, Z. Xu, M. Chen, J. Konečný, A. Hard, and T. Goldstein. "Diurnal or Nocturnal? Federated Learning of Multi-branch Networks from Periodically Shifting Distributions". In: *International Conference on Learning Representations*. 2022.

[199] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. "MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=1tZbq88f27.

[200] Z. Zhu, Y. Song, and Y. Liu. "Clusterability as an alternative to anchor points when learning with noisy labels". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12912–12923.

[201] J. Zhuang, Z. Chen, P. Wei, G. Li, and L. Lin. "Discovering Implicit Classes Achieves Open Set Domain Adaptation". In: *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2022, pp. 01–06.