Improving Representation Learning from Data and Model Perspectives: Semi-Supervised Learning and Foundation Models

Yue Fan

A dissertation submitted towards the degree

Doctor of Engineering Science (Dr.-Ing.)

of the Faculty of Mathematics and Computer Science
of Saarland University

Saarbrücken, 2025.

Date of Colloquium:	04.07.2025
Dean of the Faculty:	Prof. Roland Speicher
Chair of the Committee:	Prof. Verena Wolf
Advisor:	Prof. Bernt Schiele
Reviewers:	Prof. Bernt Schiele
	Prof. Margret Keuper
	Prof. Bastian Leibe
Academic Assistant:	Dr. Jiahao Xie

Dedicated to Huimin Liu, for her unwavering love and support.

Abstract

In recent years, artificial intelligence (AI) has made impressive progress in various industries and everyday life. Its rapid advancements have been driven by the integration of large-scale data and sophisticated models. However, several significant challenges persist that hinder further progress. First, the success of modern AI systems relies heavily on large-scale labeled datasets; yet, acquiring such datasets is often costly, time-consuming, and impractical, particularly in sensitive domains like healthcare and finance, where privacy and regulatory issues complicate data collection. Second, although unlabeled data is typically abundant and more readily accessible, it presents its own set of challenges, including issues of imbalanced distribution, outliers, and domain shifts. These challenges complicate the effective utilization of unlabeled data, raising critical questions about how to extract robust representations from imperfect datasets. Third, there is a growing demand for versatile models capable of performing a wide range of tasks across diverse domains, motivated by the broader ambition of achieving Artificial General Intelligence (AGI). However, developing models that demonstrate task-agnostic representation learning and ensure transferability across modalities remains a substantial challenge, often limiting the applicability of existing solutions.

Therefore, this thesis aims to tackle the challenges of imperfect data and versatile model design by studying three key topics: standard semi-supervised learning (SSL), realistic SSL, and vision generalist models. In the first topic of standard SSL, we focus on improving two widely used methodologies: consistency regularization and pseudo-labeling. For augmentation-based consistency regularization, we propose explicitly regularizing the distance between feature representations, demonstrating that encouraging equivariant features leads to superior generalization performance compared to merely enforcing invariance. As for threshold-based pseudo-labeling, we introduce two innovative schemes for enhancement. The first is a self-adaptive thresholding approach that considers the current learning status of the model, while the second is a unified sample weighting framework that completely replaces traditional thresholding methods. Both methods achieve significant performance improvement over the previous state-of-the-art. In the second topic of realistic SSL, we begin by investigating realistic unlabeled data with imbalanced distributions or outliers. To address imbalanced SSL, we propose a novel co-learning framework that effectively decouples representation learning from classifier learning while maintaining a close coupling. Our method achieves state-of-the-art results across various benchmarks. For SSL with outliers, we introduce a simple but strong baseline that effectively leverages outlier data to enhance generalization. We also confront the challenge of unrealistic benchmarks by proposing a new benchmark for semi-supervised classification, which offers a fair testing ground to evaluate leading SSL methods across multiple domains, including natural language processing and audio. Additionally, we provide an open-source, modular, and extensible codebase to facilitate future developments in SSL. In the final topic of vision generalist models, we propose a diffusion-based approach that unifies four distinct types of vision tasks and demonstrates competitive performance compared to existing vision generalists.

In summary, this thesis advances the two mainstream techniques in standard SSL while investigating the challenges posed by realistic SSL, where we develop methods to deal with real-world unlabeled data and realistic evaluation. Additionally, we also take an initial step toward a unified model design for vision tasks.

Zusammenfassung

In den letzten Jahren hat die Künstliche Intelligenz (KI) in verschiedenen Industrien und im Alltag beeindruckende Fortschritte erzielt. Ihre rasanten Entwicklungen wurden durch die Integration von großskaligen Daten und fortschrittlichen Modellen vorangetrieben. Dennoch bestehen weiterhin mehrere bedeutende Herausforderungen, die weiteren Fortschritt behindern. Erstens hängt der Erfolg moderner KI-Systeme stark von großskaligen, annotierten Datensätzen ab. Das Sammeln solcher Datensätze ist jedoch oft teuer, zeitaufwendig und in sensiblen Bereichen wie dem Gesundheitswesen und der Finanzbranche unpraktisch, da Datenschutz- und Regulierungsfragen die Datenerhebung erschweren. Zweitens ist unlabeled Data zwar meist reichlich vorhanden und leichter zugänglich, bringt jedoch eigene Herausforderungen mit sich, wie unbalancierte Verteilungen, Ausreißer und Domänenverschiebungen. Diese Probleme erschweren die effektive Nutzung unannotierter Daten und werfen kritische Fragen auf, wie robuste Repräsentationen aus unvollkommenen Datensätzen extrahiert werden können. Drittens gibt es eine wachsende Nachfrage nach vielseitigen Modellen, die in der Lage sind, eine Vielzahl von Aufgaben über unterschiedliche Domänen hinweg zu lösen. Diese Entwicklung wird durch das übergeordnete Ziel der Erreichung von Artificial General Intelligence (AGI) motiviert. Die Entwicklung von Modellen, die eine aufgabenneutrale Repräsentationslernung ermöglichen und Transferfähigkeit zwischen Modalitäten gewährleisten, bleibt jedoch eine erhebliche Herausforderung, was die Anwendbarkeit bestehender Lösungen oft einschränkt.

Daher zielt diese Dissertation darauf ab, die Herausforderungen unvollkommener Daten und des Designs vielseitiger Modelle durch die Untersuchung von drei Schlüsselaspekten zu adressieren: Standard Semi-Supervised Learning (SSL), realistische SSL und generalistische Vision-Modelle. Im ersten Thema, dem Standard-SSL, liegt unser Fokus darauf, zwei weit verbreitete Methoden zu verbessern: Konsistenzregularisierung und Pseudo-Labeling. Für die auf Augmentation basierende Konsistenzregularisierung schlagen wir vor, die Distanz zwischen Merkmalsrepräsentationen explizit zu regularisieren. Wir zeigen, dass die Förderung von äquivarianten Merkmalen zu einer besseren Generalisierungsleistung führt, verglichen mit dem bloßen Erzwingen von Invarianz. Für das Schwellenwert-basierte Pseudo-Labeling führen wir zwei innovative Verbesserungen ein. Die erste ist ein selbstadaptiver Schwellenwertansatz, der den aktuellen Lernstatus des Modells berücksichtigt. Die zweite ist ein einheitliches Probengewichtungsframework, das traditionelle Schwellenwertmethoden vollständig ersetzt. Beide Ansätze erreichen signifikante Leistungsverbesserungen gegenüber dem bisherigen Stand der Technik. Im zweiten Thema, dem realistischen SSL, untersuchen wir zunächst realistische unannotierte Daten mit unbalancierten Verteilungen oder Ausreißern. Um das Problem des unbalancierten SSL zu lösen, schlagen wir ein neuartiges Co-Learning-Framework vor, das die Repräsentationslernung von der Klassifikationslernung effektiv entkoppelt und dennoch eine enge Kopplung aufrechterhält. Unsere Methode erzielt state-of-the-art Ergebnisse über verschiedene Benchmarks hinweg. Für SSL mit Ausreißern stellen wir eine einfache, aber leistungsstarke Baseline vor, die Outlier-Daten effektiv nutzt, um die Generalisierung zu verbessern. Wir stellen uns außerdem der Herausforderung unrealistischer Benchmarks, indem wir einen neuen Benchmark für semi-supervised Klassifikation vorschlagen, der eine faire Testgrundlage bietet, um führende SSL-Methoden über mehrere Domänen hinweg, einschließlich Natural Language Processing und Audio, zu bewerten. Zusätzlich stellen wir eine Open-Source-, modulare und erweiterbare Codebasis bereit, um zukünftige Entwicklungen im

Bereich SSL zu fördern. Im dritten Thema der generalistischen Vision-Modelle schlagen wir einen auf Diffusion basierenden Ansatz vor, der vier verschiedene Typen von Vision-Aufgaben vereint und eine wettbewerbsfähige Leistung im Vergleich zu bestehenden Vision-Generalisten demonstriert.

Zusammenfassend treibt diese Dissertation die zwei Haupttechniken im Standard-SSL voran, während sie die Herausforderungen des realistischen SSL untersucht. Dabei entwickeln wir Methoden, um mit realen unannotierten Daten und realistischen Evaluierungen umzugehen. Darüber hinaus unternehmen wir einen ersten Schritt in Richtung eines einheitlichen Modelldesigns für Vision-Aufgaben.

Acknowledgements

This PhD journey has been one of the most challenging and transformative experiences of my life. It has been a path filled with doubt and discovery, frustration and fulfillment. This acknowledgement is to all those who supported me along the way.

To my advisor, Bernt, thank you for being the compass in this intellectual wilderness. Your patience when I felt lost, your challenges when I got too comfortable, and your belief in the work—even when I couldn't see its value—kept me moving forward. I've learned far more than just science from you.

To my collaborators and labmates, thank you for the stimulating discussions, constructive critiques, and generous sharing of ideas. Your contributions not only improved the quality of the work but made the process more enjoyable and less lonely. The camaraderie we built—whether over experiments, coffee, or paper deadlines—made the lab a place of both growth and laughter. I am especially grateful for the sense of community you gave me.

To my friends outside of work, thank you for playing so many video games with me. Those moments brought laughter, presence, and balance into my life. I've come to realize I don't just love games—I love the people I played them with. You helped me rediscover the joy of friendship.

To my parents, thank you for your unconditional love and support, not just through this PhD, but throughout my entire life. We've had our arguments, but I've always known you are my home. I feel incredibly lucky to be your son, and I'm proud to have made you proud. I'm grateful you were able to watch my defense—I hope you see now that your son has grown into someone who can stand on his own.

To my dearest wife, no words can fully express what your support has meant to me. Moving to a new country, starting a new life, learning a new language, adapting to a new culture, all while enduring the pandemic—it was never easy. You gave up so much to be by my side. I know I wasn't always the best husband, and I wish I had treated you better during the hard times. I cannot imagine making it through this journey without you.

To the many kind strangers I met in different countries, from all walks of life—thank you for your small but unforgettable gestures of kindness. We may never meet again, but your light has stayed with me, and reminded me that this world is full of hope.

Though my PhD ends here, my story continues. I'm grateful that some of the work I did has already begun to help others—that means more to me than any title or degree. This journey has been long and difficult, but now, looking back, I realize that a PhD is not just about producing a thesis. It's about learning to seek help, and to offer it in return.

Contents

Introduction 1

Contents

In recent years, artificial intelligence (AI) has experienced rapid advancements, transforming industries and reshaping many aspects of everyday life. From assisting with creative tasks to accelerating scientific breakthroughs, AI systems have become deeply integrated into modern society. For instance, large language models such as ChatGPT [BMR+20, AAA+23] have proven useful as AI assistants, boosting productivity in tasks like writing, brainstorming, and coding [HSX+24]. In the realm of science, AlphaFold [JEP+21, VAD+22] has revolutionized biology by predicting protein structures with remarkable accuracy, speeding up progress in drug discovery, disease research, and biotechnology. Meanwhile, generative models like Stable Diffusion [RBL+22a, BDK+23] are driving innovation in entertainment and the creative arts by turning abstract ideas into vivid visuals, offering new tools to artists and designers.

At the core of these successes lie three fundamental pillars: data, models, and computing. Together, they form the foundation of AI's progress. A powerful AI system is typically trained on large-scale, high-quality datasets [GIF+24, SBV+22, NVNL+23, TBW+24], which are essential for achieving state-of-the-art performance [RKH+21a, DJP+24, DDM+23, CHL+24]. Equally crucial is the design and capacity of the models themselves, which must be capable of effectively learning from the available data [AZKB24, LLWL24]. However, both data collection and model development present significant challenges in practice:

- **The Challenge of Labeled Data.** The success of many modern AI systems has been built on large, labeled datasets, yet acquiring such datasets is often costly, time-consuming, and sometimes impractical. Data collection in sensitive domains, such as healthcare or finance, faces additional hurdles related to privacy, security, and data accessibility. Labeling data also comes with substantial logistical and economic challenges, especially as modern datasets continue growing in size. In contrast, unlabeled data is typically abundant and more readily accessible, raising a crucial question: how can we effectively utilize this wealth of unlabeled data? This challenge has sparked interest in semi-supervised learning (SSL) methods, which leverage both labeled and unlabeled data to improve model performance and hold promise for advancing AI in data-scarce settings.

- **Limitations of Unlabeled Data.** While unlabeled data is abundant, it often presents its own set of challenges, making it difficult to leverage effectively. For example, unlabeled data are frequently imbalanced, where certain classes or concepts are overrepresented, leading to biases in model training. Additionally, unlabeled data often contains outliers that do not belong to any class of interest, which can mislead or degrade model performance. Thus, a key challenge in modern semi-supervised learning is finding ways to extract robust, meaningful representations from imperfect and diverse unlabeled data.

- **The Need for Unified Models.** Model design plays an equally critical role in AI's success. A key trend in recent years is the pursuit of unified or generalist models that is capable of solving a wide range of tasks across diverse domains. This pursuit is motivated by the broader ambition to push AI toward Artificial General Intelligence (AGI), where models demonstrate versatile reasoning and adaptability. Despite notable progress and extensive research efforts in recent years [TLI$^+$23, DCLT18a, RWC$^+$19, RSR$^+$20], creating models that are both versatile and scalable remains a substantial challenge due to the requirement of task-agnostic representation learning and transferability across modalities.

In this thesis, we primarily focus on advancing semi-supervised learning by addressing challenges in leveraging unlabeled data in both standard and more realistic settings. Additionally, we take an initial step to explore unified model design in AI in the final chapter. The contributions of this research are outlined below.

- **Standard Semi-Supervised Learning.** In Part I, we examine and improve the two popular paradigms in standard semi-supervised learning, consistency regularization, and pseudo-labeling. Specifically:
  - In Chapter 3, we revisit the idea of enforcing invariant features in consistency regularization and improve it with a simple yet effective technique called FeatDistLoss which further regularizes the distance between feature representations.
  - In Chapter 4, we propose FreeMatch to improve the thresholding-based pseudo-labeling by a self-adaptive threshold which reflects the learning status of the model.
  - In Chapter 5, we introduce SoftMatch to overcome the inherent quantity-quality trade-off of pseudo-labeling by effectively leveraging the unconfident yet correct pseudo-labels.
- **Realistic Semi-Supervised Learning.** In Part II, we study and improve semi-supervised learning algorithms under realistic settings, where data is long-tail distributed, or contains outliers, or from different domains. Specifically:
  - In Chapter 6, we propose a novel co-learning framework CoSSL for imbalanced SSL, which decouples representation and classifier learning while coupling them closely via a shared encoder and pseudo-label generation.
  - In Chapter 7, we contribute a Simple but Strong Baseline, SSB, for open-set SSL. We find that incorporating pseudo-labels with high confidence into the training, irrespective of whether a sample is an inlier or outlier, improves the unlabeled data utilization ratio and, thus, the final performance.
  - In Chapter 8, we propose USB: a unified and challenging semi-supervised learning benchmark for classification with 15 tasks on CV, NLP, and Audio for fair and consistent evaluations. At that time, it was the first work to discuss whether current SSL methods that work well on CV tasks generalize to NLP and Audio tasks.

- **Foundation Models.** In Part III, we study the vision generalist models Specifically, in Chapter 9, we explore diffusion-based vision generalists, where we unify different types of dense prediction tasks as conditional image generation and re-purpose pre-trained diffusion models for it.

For the rest of this chapter, we discuss each topic and explain our contributions. Then, we provide an outline of the thesis with relevant publications

1.1 Standard Semi-Supervised Learning.

In machine learning, we traditionally distinguish between three types of learning paradigms: supervised learning, unsupervised learning, and semi-supervised learning (SSL). Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ be the output space. Given a training set with input data $\{X_i\}_{i=0}^n$, $X_i \in \mathcal{X}$ and labels $\{Y_i\}_{i=0}^n$, $Y_i \in \mathcal{Y}$, the goal of supervised learning is to construct a model f from the training set that maps from $\mathcal{X}$ to $\mathcal{Y}$. In contrast, unsupervised learning operates solely on the unlabeled training data $\{X_i\}_{i=0}^n$, seeking to uncover underlying structures in the data, such as clustering similar data points. SSL is a hybrid setting that combines aspects of both paradigms. It leverages a training set containing both labeled data $L = \{X_i, Y_i\}_{i=0}^n$ and unlabeled data $U = \{X_i\}_{i=n+1}^{n+m}$ to improve model performance. SSL can be further categorized into two subtypes: **transductive and inductive learning** [Vap98]. Given a labeled training set L and an unlabeled test set U, transductive learning aims to predict the labels only for the unlabeled test points, whereas the goal of inductive learning is to train a model capable of generalizing across the entire input space $\mathcal{X}$.

As the core concept of SSL is to effectively leverage the unlabeled data to improve the model performance, an important question is when does semi-supervised learning work? In fact, the effectiveness of SSL hinges on three core assumptions:

- **Smoothness assumption:** The label function is smoother in high-density regions than in low-density regions. Therefore, data points that are closer to each other in high-density regions are more likely to share a label.

- **Cluster assumption:** Data points from the same cluster are likely to have the same label. As clusters are often defined by sets of points that can be connected via many paths through high-density regions, this assumption implies that the decision boundary should lie in a low-density region.

- **Manifold assumption:** High-dimensional data often lie on low-dimensional manifolds. This assumption helps mitigate the curse of dimensionality by simplifying the data representation.

SSL has become increasingly relevant in the deep learning era, driven by two key factors. First, obtaining large quantities of labeled data is time-consuming and costly, particularly in specialized domains like medical or satellite image classification [DG22], where high-quality expert annotation is prohibitively difficult to require. Second, unlabeled data is abundant and readily available, for example, the vast amounts of images and text accessible online [CWC+22, TSF+16]. How to properly leverage the knowledge embedded in this unlabeled data is an important topic for both academia and industry. In fact, many studies have shown that combining a small amount of labeled data with a larger set of unlabeled data can lead to substantial performance improvements [BCG+19, BCC+20, SBL+20, LXH21, BRS+22].

In recent years, numerous SSL approaches have been proposed, which can broadly be divided into two categories: consistency regularization and pseudo-labeling (or self-training).

- **Consistency regularization** exploits the idea that a model should produce consistent predictions for perturbed versions of the same input. Perturbations can take various forms, such as input noise through data augmentation or stochasticity during the model inference, like dropout [SHK$^+$14]. A particularly successful strategy [XDH$^+$19] is data-augmentation-based consistency regularization, which produces data points from two sets of distinct augmentation strategies (often a strong augmentation and a weak augmentation) and enforces a consistency loss between them. This idea has inspired many recent state-of-the-art SSL methods [SBL$^+$20, CTF$^+$23, WCH$^+$23].

- **Pseudo-labeling**, also known as self-training or self-teaching, is one of the earliest forms of SSL [CSZ09]. It aims to generate artificial labels for unlabeled data so that the model can utilize them the same way as labeled data. Traditionally, self-training involves alternation between training the model on labeled data and augmenting the labeled set with high-confidence predictions on the unlabeled data. Recent methods [Lee13, SBL$^+$20, ZWH$^+$21] integrate pseudo-label generation with model training in an end-to-end manner, often using a threshold to filter the pseudo-labels based on confidence scores. And how to designing effective thresholds is crucial for the success of these methods.

Furthermore, there are many excellent works around generative models and graph-based methods. We refer to Chapter 2 for a more comprehensive introduction to SSL methods.

Despite the notable progress in SSL, significant challenges remain. Addressing these challenges is key to unlocking the full potential of SSL in real-world applications. In the following sections, we will delve into these challenges and outline our contributions to advancing semi-supervised learning.

1.1.1 Challenges

- **Perturbation in Consistency Regularization.** A major challenge in consistency regularization is to develop effective perturbations without breaking the clustering assumption. For instance, in image classification, strong data augmentations are commonly applied to perturb input images and encourage the model to produce consistent predictions. However, these strong augmentations can sometimes generate images that diverge significantly from the original semantics. It remains unclear whether enforcing invariance to such perturbations always benefits the model's generalization. Overly diverse perturbations risk breaking the assumption that points within the same high-density cluster share the same label, potentially degrading performance rather than improving it.

- **Quantity-Quality Tradeoff in Pseudo-Labeling.** In threshold-based pseudo-labeling, the central idea is to train the model using pseudo-labels whose prediction confidence exceeds a predefined threshold, while discarding the rest. However, this approach inherently introduces a quantity-quality tradeoff [CTF$^+$23] that complicates the learning process. On one hand, a high confidence threshold ensures that only high-quality pseudo-labels are used, but this comes at the cost of discarding a significant number of potentially correct but low-confidence labels, limiting the quantity of training data. On the other hand, lowering the threshold increases the number of pseudo-labels utilized, but it also introduces noisy, incorrect labels that can mislead the model. Striking the right balance between the quantity and quality of pseudo-labels remains an open problem, and finding adaptive mechanisms to tune this threshold effectively is critical for improving pseudo-labeling performance.

1.1.2 Contributions

In this section, we summarize our contributions to addressing the above main challenges of standard SSL.

In Chapter 3, we tackle the first challenge, *perturbation in consistency regularization*, by revisiting and improving the data-augmentation-based consistency regularization. Specifically, we show that while encouraging invariance results in good performance, encouraging equivariance to differently augmented versions of the same image consistently results in even better generalization performance. Therefore, we propose FeatDistLoss, which regularizes the distance between feature representations from differently augmented images of the same class while enforcing the same semantic class label. The final model, CR-Match, which combines FeatDistLoss with other strong techniques, defines a new state-of-the-art across a wide range of settings of standard SSL benchmarks.

In Chapter 4, we address the second challenge, *quantity-quality trade-off in pseudo-labeling*, via a self-adaptive and parameter-free threshold adjusting scheme that respects the model's learning status. We first discuss why thresholds should reflect the model's learning status and provide some intuitions for designing a threshold-adjusting scheme. Based on the analysis, we propose a novel approach, FreeMatch, which consists of Self-Adaptive Thresholding (SAT) and Self-Adaptive class Fairness regularization (SAF). SAT is a threshold-adjusting scheme that is free of setting thresholds manually and SAF encourages diverse predictions. Extensive results demonstrate the superior performance of FreeMatch on various SSL benchmarks, especially when the number of labels is very limited.

In Chapter 5, we address the second challenge, *quantity-quality trade-off in pseudo-labeling*, by developing a unified sample weighting framework called SoftMatch. We first identify that the inherent trade-off in previous methods mainly stems from the lack of careful design on the distribution of pseudo-labels, which is imposed directly by the weighting function. Then, we propose SoftMatch to effectively leverage the unconfident yet correct pseudo-labels, fitting a truncated Gaussian function to the distribution of confidence, which overcomes the trade-off. We further propose Uniform Alignment to resolve the imbalance issue of pseudo labels while maintaining their high quantity and quality. Finally, we demonstrate that SoftMatch outperforms previous methods on various image and text evaluation settings. We also empirically verify the importance of maintaining the high accuracy of pseudo-labels while pursuing better unlabeled data utilization in SSL.

1.2 Realistic Semi-Supervised Learning.

While standard semi-supervised learning (SSL) has made significant progress in leveraging unlabeled data, it often falls short in real-world applications due to two key limitations. First, standard SSL typically assumes clean, balanced data [OOR+18]. However, real-world data collection frequently results in imperfections, especially in large-scale unlabeled datasets that are challenging to clean [SVB+21]. In practice, unlabeled data may be imperfect in several ways. It could be long-tailed [ZKH+23], with some classes overrepresented and others underrepresented. It might include out-of-distribution (OOD) data or outliers [YZLL24], which, if not handled properly, can skew learning. Additionally, internal domain shifts may occur

[WD18], where the distribution of the data differs between training and test sets. These factors can severely limit the effectiveness of standard SSL methods when deployed in real-world scenarios. Second, existing evaluation benchmarks [OOR+18, ZWH+21] do not adequately reflect real-world challenges. They are often designed for specific computer vision (CV) tasks and fail to generalize across diverse domains. SSL methods also tend to overfit to toy datasets like CIFAR-10 and CIFAR-100 [KH+09a] due to extensive hyperparameter tuning, which limits their practical utility.

Semi-Supervised Learning with Imperfect Data. In this thesis, we explore semi-supervised learning under realistic conditions by addressing two major challenges: imbalanced SSL and open-set SSL.

- **Imbalanced Semi-Supervised Learning.** In real-world scenarios, training data is often long-tailed, with a few classes having abundant samples while most classes are under-represented. In SSL, this imbalance exists in both labeled and unlabeled data, making it difficult to perform well across all classes. Typically, models overfit to the majority classes, leading to poor generalization on minority classes. Imbalanced SSL seeks to develop classifiers that perform well across a wide range of test data distributions, including those that differ significantly from the training data. This is because uniform class distribution is insufficient to reflect the diversity of real-world applications [HHC+21], where users may have different needs. In practice, we need models capable of generalizing across varying and potentially unknown distributions.

- **Open-Set Semi-Supervised Learning.** Another critical challenge in real-world SSL is the presence of outliers or OOD data within unlabeled datasets. It is impractical to remove all outliers, as doing so would require labeling the unlabeled data. Therefore, open-set SSL addresses this challenge by considering a more realistic setting where the unlabeled data contains samples from unknown classes that do not appear in the labeled set. At test time, the model must accurately classify inlier samples while also identifying both seen and unseen outliers. While most current methods focus on filtering out outliers [YIIA20a, SKS21, HFC+21], an ideal solution would be to leverage the information from these outliers rather than discarding them. Thus, effective detection and utilization of outliers remains an open problem in SSL.

Semi-Supervised Learning with Realistic Evaluation. The current landscape of SSL evaluation is also in need of improvement. Existing benchmarks, such as Realistic SSL Evaluation [OOR+18] and TorchSSL [ZWH+21], while useful, are no longer actively maintained. These benchmarks primarily focus on a limited set of SSL algorithms and CV tasks, which limits their relevance in today's rapidly evolving field. Additionally, most of these benchmarks involve training models from scratch, which is computationally expensive and time-consuming due to the slow convergence of SSL algorithms [AFIW18]. There is a pressing need for a new benchmark that is both environmentally sustainable and cost-effective, allowing for fair comparisons across SSL methods and domains. This would enable the SSL community to continuously update algorithms and foster further advancements.

Despite the progress in the field, realistic semi-supervised learning presents significant challenges. In the following sections, we will delve deeper into these issues and outline our contributions toward addressing them.

1.2.1 Challenges

- **Imbalanced Semi-Supervised Learning.** While methods for long-tailed recognition have shown success in handling class imbalance [CBHK02, HM13, HG09, HLLT16, BMM18], they are not equipped to exploit the benefits of unlabeled data. This results in suboptimal performance, particularly when labeled data is scarce. On the other hand, SSL techniques are designed to leverage unlabeled data but often struggle when confronted with class imbalance. In fact, some standard SSL methods, when applied to imbalanced datasets, perform worse than simpler re-balancing approaches that completely ignore unlabeled data [KHP$^+$20b]. This raises a fundamental challenge in SSL: how to develop methods that can simultaneously address both the imbalance and the effective use of unlabeled data. Resolving this challenge is crucial for unlocking the full potential of SSL in real-world scenarios.

- **Open-Set Semi-Supervised Learning.** Most open-set SSL methods rely on a dual-task framework [CZLG20, GZJ$^+$20a, YIIA20a, SKS21, HFC$^+$21, PYJS22, HHLY22, HHYY22, HYG22], combining an inlier classifier with an outlier detector. These methods attempt to filter out outliers, training the classifier exclusively on inliers. However, this filtering process often discards a significant portion of inliers alongside out-of-distribution (OOD) data, leading to suboptimal classification performance due to the reduced utilization of the available unlabeled data. Furthermore, the shared feature encoder between the classifier and the detector can result in conflicting objectives, negatively impacting detection performance. Consequently, effectively identifying and utilizing outliers remains a major challenge in SSL.

- **Unrealistic benchmark.** As previously noted, current SSL benchmarks focus primarily on computer vision tasks, such as CIFAR-10/100, SVHN, STL-10, and ImageNet classification. This narrow scope restricts the broader understanding of SSL's potential, especially in other domains like natural language processing (NLP) and audio, where labeled data is equally scarce. Furthermore, with the growing adoption of the pre-training and fine-tuning paradigm, SSL has the potential to significantly reduce training costs. However, existing benchmarks fail to adequately support fair evaluation in this context, particularly when pre-trained models are involved. In addition, many SSL evaluation protocols (e.g., TorchSSL [ZWH$^+$21]) are computationally expensive, often requiring models to be trained from scratch [BCG$^+$19, BCC$^+$20, XDH$^+$20b, SBL$^+$20, XSY$^+$21, ZWH$^+$21]. For example, evaluating FixMatch [SBL$^+$20] using TorchSSL [ZWH$^+$21] demands approximately 335 GPU days (279 GPU days excluding ImageNet). Such resource-intensive evaluations can be prohibitive for academic research labs with limited computational resources, highlighting the need for more efficient and accessible benchmarking practices.

1.2.2 Contributions

In this section, we summarize our contributions to addressing the above main challenges of Realistic SSL.

In Chapter 6, we tackle the first challenge of imbalanced SSL by proposing a novel co-learning framework, CoSSL, which decouples representation and classifier learning while coupling them closely via a shared encoder and pseudo-label generation. Furthermore, we devise a novel Tail-class Feature Enhancement (TFE) method to increase the data diversity of tail classes by utilizing unlabeled data, leading to more robust classifiers. Together, our model achieves new state-of-the-art results on multiple imbalanced SSL benchmarks across a wide range of evaluation settings. Finally, we address the uniform test distribution issue by introducing new evaluation criteria that cover a large range of varying distributions.

In Chapter 7, we address the second challenge of open-set SSL by contributing a Simple but Strong Baseline, SSB, with three novel ingredients: (1) In contrast to detector-based filtering aiming to remove OOD data, we propose to incorporate pseudo labels with high inlier classifier confidence into the training, irrespective of whether a sample is an inlier or OOD. This not only effectively improves the unlabeled data utilization ratio but also includes many useful OOD data that can be seen as natural data augmentations of inliers. (2) Instead of directly sharing features between the classifier and detector, we add non-linear transformations for the task-specific heads and find that this effectively reduces mutual interference between them, resulting in more specialized features and improved performance for both tasks. (3) In addition, we propose pseudo-negative mining to further improve outlier detector training by enhancing the data diversity of OOD data with pseudo-outliers. Despite its simplicity, SSB achieves significant improvements in both inlier classification and OOD detection.

In Chapter 8, we tackle the third challenge of unrealistic benchmark of SSL by constructing a Unified SSL Benchmark (USB) for classification, which selects 15 diverse, challenging, and comprehensive tasks from CV, natural language processing (NLP), and audio processing (Audio). We systematically evaluate the dominant SSL methods, and also open-source a modular and extensible codebase for fair evaluation of these SSL methods. We further provide the pre-trained versions of the state-of-the-art neural models for CV tasks to make the cost affordable for further tuning. USB enables the evaluation of a single SSL algorithm on more tasks from multiple domains but with less cost. Specifically, on a single NVIDIA V100, only 39 GPU days are required to evaluate FixMatch [SBL$^+$20] on 15 tasks in USB while 335 GPU days (279 GPU days on 4 CV datasets except for ImageNet) are needed on 5 CV tasks with TorchSSL.

1.3 Foundation models.

In addition to data, the success of modern AI systems heavily relies on model design. Crafting an effective model architecture has become as crucial as having access to large-scale training data. In particular, learning a general perception model capable of handling diverse modalities and tasks is considered a significant milestone on the path to artificial general intelligence. The field of artificial intelligence has made remarkable strides toward building generalized model frameworks. In particular, autoregressive transformers [VSP$^+$17] have become the dominant unified approach in Natural Language Processing (NLP), addressing a wide variety of tasks with a single model architecture [TLI$^+$23, DCLT18a, RWC$^+$19, RSR$^+$20]. This success has not yet been fully realized in computer vision (CV), where the diversity of tasks and output formats creates additional challenges. As a result, state-of-the-art computer vision models often feature complex, task-specific designs that limit their ability to share features across tasks, thereby restricting knowledge transfer and scalability.

This discrepancy between NLP and CV has fueled increasing interest in developing unified approaches for vision tasks. Recent efforts in this area have sought to bridge the gap by proposing models that can handle diverse vision tasks under a common framework. For example, Pix2Seq [CSL$^+$21] introduced the idea of leveraging an autoregressive transformer to tackle vision tasks using next-token prediction, where traditionally complex outputs such as bounding boxes or segmentation masks are cast as sequences of discrete tokens. This idea has been further extended by models such as Unified-IO [LCZ$^+$22], which uses vector quantization to encode image features and output predictions such as segmentation masks or depth maps in a tokenized form. Similarly, OFA (One For All) [WYM+22] unified various cross-modal and unimodal tasks in a simple sequence-to-sequence learning framework, achieving competitive results with significantly fewer training resources.

However, despite these advancements, building vision generalist models remains a difficult and unresolved problem. The challenges stem from the inherent diversity of visual data and tasks, requiring models to handle both structured and unstructured outputs, while maintaining high performance across all tasks. Below, we discuss the primary challenges in building such models and outline our contributions toward addressing these issues.

1.3.1 Challenges

One of the most significant challenges in creating a vision generalist model is accommodating the diversity of task output formats. Unlike natural language processing (NLP), where various tasks such as text generation, classification, and translation can often be framed within a common sequence-to-sequence input-output structure, computer vision (CV) tasks tend to require highly specialized output representations. For instance, image classification tasks predict a single semantic label for each image, object detection involves regressing bounding box coordinates alongside class labels, and segmentation tasks require dense, per-pixel predictions. Each of these tasks has distinct output requirements, making the design of a unified model that excels across them a complex problem.

1.3.2 Contributions

In this section, we summarize our contributions to addressing the challenge above in building vision generalist models.

In Chapter 9, we tackle the above challenge by exploring diffusion-based vision generalists, where we unify different types of dense prediction tasks as conditional image generation and re-purpose pre-trained diffusion models [RBL+22b] for it. Our investigation reveals a list of interesting findings as follows: 1. Diffusion-based generalists show superior performance over the non-diffusion-based generalists on tasks involving semantics or global understanding of the scene. 2 We find conditioning on the image feature extracted from powerful pre-trained image encoders results in better performance than directly conditioning on the raw image. 3. Pixel diffusion [HJA20] is better than latent diffusion as it does not have the quantization issue while upsampling. 4. We observe that text-to-image generation pre-training stabilizes the training and leads to better performance. In experiments, we evaluate our method on four different types of tasks and show competitive performance to the other vision generalists.

1.4 Outline

In this section, we provide a summary of the thesis by briefly outlining each chapter and establishing connections between them. We also acknowledge any relevant publications and collaborations with other researchers.

Chapter 2, Related Work: This chapter surveys related works about improving representation learning from data and model perspectives. In particular, it focuses on the three directions of the thesis i.e., standard semi-supervised learning, realistic semi-supervised learning, and vision generalist model. We discuss how these works relate to the methods and contributions presented in this thesis. Discussions of related work specific to the following chapters are provided within each chapter.

Part I, Data perspective: standard semi-supervised learning

Chapter 3, Revisiting Consistency Regularization: In this chapter, we revisit the idea of data-augmentation-based consistency regularization in SSL. We find that while enforcing invariance by decreasing distances between features from differently augmented images leads to improved performance, encouraging equivariance instead, by increasing the feature distance, further improves performance. Based on this, we design a simple yet effective technique, FeatDistLoss, which imposes consistency and equivariance on the classifier and the feature level, to improve consistency regularization.

The content of this chapter corresponds to the GCPR 2021 publication with the title "Revisiting Consistency Regularization for Semi-Supervised Learning"[FKS21a]. This work was further extended to IJCV 2023 publication [FKS21b] with the same title. Yue Fan was the lead author of this paper under the supervision of Prof. Bernt Schiele. The journal extension also involved the supervision of Dr. Dengxin Dai. It is also a collaboration with Dr. Anna Kukleva.

Chapter 4, Self-Adaptive Thresholding: In this chapter, we study threshold-based pseudo-labeling in SSL and propose FreeMatch to adjust the thresholds in a self-adaptive manner according to the learning status of each class. Moreover, we introduce a self-adaptive class fairness regularization penalty to encourage the model for diverse predictions during the early training stage.

The content of this chapter corresponds to the ICLR 2023 publication with the title "FreeMatch: Self-adaptive Thresholding for Semi-supervised Learning" [WCH$^+$23]. Yidong Wang is the first author of this paper, under the supervision of Dr. Jindong Wang, Prof. Takahiro Shinozaki, and Prof. Xing Xie. It is also a collaboration with Yue Fan and Prof. Bernt Schiele from MPI-Informatics; Hao Chen, Prof. Marios Savvides, and Prof. Bhiksha Raj from CMU; Qiang Heng from North Carolina State University; Wenxin Hou from Microsoft STCA; Zhen Wu from Nanjing University. Yue Fan was involved in the weekly and more detailed discussions and contributed to the writing of the paper and the imbalanced SSL experiments.

Chapter 5, Unified Sample Weighting Framework: In this chapter, we tackle the quantity-quality trade-off in pseudo-labeling by developing a unified weighting framework. We propose SoftMatch to effectively leverage the unconfident yet correct pseudo-labels by fitting a truncated Gaussian function to the distribution of confidence.

The content of this chapter corresponds to the ICLR 2023 publication with the title "Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning" [CTF$^+$23]. Hao Chen is the first author of this paper, under the supervision of Prof. Marios Savvides and Prof. Bhiksha Raj. It is also a collaboration with Yue Fan and Prof. Bernt Schiele from MPI-Informatics; Ran Tao from CMU; Yidong Wang and Dr. Jindong Wang from Microsoft Research Asia. Yue Fan was involved in the weekly and more detailed discussions and contributed to the writing of the paper and the imbalanced SSL experiments.

Part II, Data perspective: realistic semi-supervised learning

Chapter 6, Imbalanced Semi-Supervised Learning: In this chapter, we study the problem of imbalanced SSL. We propose a novel co-learning framework CoSSL, which decouples representation and classifier learning while coupling them closely via a shared encoder and pseudo-label generation. Moreover, we propose Tail-class Feature Enhancement (TFE) for improved classifier learning for imbalanced SSL, which utilizes unlabeled data as a source of augmentation to enhance the data diversity of tail classes, leading to a more robust classifier.

The content of this chapter corresponds to the CVPR 2022 publication with the title "CoSSL: Co-Learning of Representation and Classifier for Imbalanced Semi-Supervised Learning" [FDS22]. Yue Fan was the lead author of this paper under the supervision of Prof. Bernt Schiele and Dr. Dengxin Dai. It is also a collaboration with Dr. Anna Kukleva.

Chapter 7, Open-Set Semi-Supervised Learning: In this chapter, we study the problem of open-set semi-supervised learning and contribute a simple but strong baseline, SSB, which effectively separates the feature space of inlier classification and outlier detection via non-linear transformations and effectively leverages outliers via confidence-based filtering. In addition, we propose pseudo-negative mining to further improve outlier detector training by enhancing the data diversity of OOD data with pseudo-outliers.

The content of this chapter corresponds to the ICCV 2023 publication with the title "SSB: Simple but Strong Baseline for Boosting Performance of Open-Set Semi-Supervised Learning" [FKDS23]. Yue Fan was the lead author of this paper under the supervision of Prof. Bernt Schiele and Dr. Dengxin Dai. It is also a collaboration with Dr. Anna Kukleva.

Chapter 8, Unified Semi-supervised Learning Benchmark: In this chapter, we address the issue of unrealistic benchmarks in SSL by introducing USB, a Unified Semi-supervised Learning Benchmark for Classification. It selects 15 diverse, challenging, and comprehensive tasks from CV, natural language processing (NLP), and audio processing (Audio). Furthermore, it also provide an open-source, modular, and extensible codebase for fair evaluation of SSL methods.

The content of this chapter corresponds to the NeurIPS 2022 publication with the title "USB: A unified semi-supervised learning benchmark for classification" [WCF$^+$22b]. Yidong Wang, Hao Chen, and Yue Fan are the co-first author of this paper, under the supervision of Prof. Bernt Schiele, Prof. Marios Savvides, Prof. Bhiksha Raj, Prof. Takahiro Shinozaki, Dr. Jindong Wang, Prof. Xing Xie, and Prof. Yue Zhang. It is also a collaboration with Wang Sun from Tsinghua University; Ran Tao from CMU; Wenxin Hou from Microsoft STCA; Linyi Yang from Westlake University; Renjie Wang, Zhi Zhou, Lan-Zhe Guo, Zhen Wu, and Prof. Yu-Feng Li from Nanjing University; Heli Qi and Satoshi Nakamura from Nara Institute of Science and Technology; Prof. Wei Ye from Peking University. Yue Fan was involved in the idea proposal, weekly and more detailed discussions, and contributed to the codebase implementation and the final paper writing.

Part III, Model perspective: building vision generalists

Chapter 9, Diffusion-Based Vision Generalist: In this chapter, we tackle the challenge of building vision generalist models by proposing a diffusion-based vision generalist, where we unify four types of dense prediction tasks as conditional image generation and repurpose pre-trained diffusion models for it. In addition, our exploration reveals a list of interesting findings for diffusion-based generalists.

The content of this chapter corresponds to the CVPR 2024 publication in the second workshop on foundation models with the title "Toward a Diffusion-Based Generalist for Dense Vision Tasks" [FXZ$^+$24]. Yue Fan was the lead author of this paper under the supervision of Dr. Yongqin Xian, Prof. Federico Tombari, and Prof. Bernt Schiele. The journal extension also involved the supervision of Dr. Dengxin Dai. It is also a collaboration with Dr. Xiaohua Zhai and Dr. Alexander Kolesnikov from DeepMind; and Dr. Muhammad Ferjad Naeem from Google.

1.5 Publications

The content of this thesis has previously appeared in the following publications, ordered as outlined above:

- [FKS21b] **Yue Fan**, Anna Kukleva, Dengxin Dai, and Bernt Schiele. "Revisiting Consistency Regularization for Semi-Supervised Learning", International Journal of Computer Vision (IJCV) 2023.
- [WCH$^+$23] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, **Yue Fan**, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, Xing Xie. "FreeMatch: Self-adaptive Thresholding for Semi-supervised Learning", In Proceedings of the International Conference on Learning Representations (ICLR), 2023.
- [CTF$^+$23] Hao Chen, Ran Tao, **Yue Fan**, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, Marios Savvides, "Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning", In Proceedings of the International Conference on Learning Representations (ICLR), 2023.
- [FDS22] **Yue Fan**, Dengxin Dai, Anna Kukleva, Bernt Schiele, "CoSSL: Co-Learning of Representation and Classifier for Imbalanced Semi-Supervised Learning", In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.
- [FKDS23] **Yue Fan**, Anna Kukleva, Dengxin Dai, Bernt Schiele, "SSB: Simple but Strong Baseline for Boosting Performance of Open-Set Semi-Supervised Learning", In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023.
- [WCF$^+$22b] Yidong Wang*, Hao Chen*, **Yue Fan***, SUN Wang, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Wei Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki, Bernt Schiele, Jindong Wang, Xing Xie, Yue Zhang (* Equal contribution), "Usb: A unified semi-supervised learning benchmark for classification", Advances in Neural Information Processing Systems (NeurIPS), 2022.
- [FXZ$^+$24] **Yue Fan**, Yongqin Xian, Xiaohua Zhai, Alexander Kolesnikov, Muhammad Ferjad Naeem, Bernt Schiele, Federico Tombari, "Toward a Diffusion-Based Generalist for Dense Vision Tasks", The second workshop on foundation models in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

Further contributions were made to the following works not discussed in this thesis:

- [WFN$^+$24] Haiyang Wang*, **Yue Fan***, Muhammad Ferjad Naeem, Liwei Wang, Yongqin Xian, Jan Eric Lenssen, Federico Tombari, Bernt Schiele (* Equal contribution), "TokenFormer: Rethinking Transformer Scaling with Tokenized Model Parameters", under review.

Related Work 2

Contents

In this chapter, we review literature from the perspectives of both model and data in AI. Our survey is divided into three main sections: standard semi-supervised learning (SSL), realistic SSL, and unified model design. Section 2.1 discusses key developments in standard SSL, including popular techniques such as consistency regularization, pseudo-labeling, generative models, and graph-based methods, before contrasting them with our proposed approaches. Section 2.2 focuses on realistic SSL, examining challenges posed by data imbalance, open-set conditions, and other factors that make standard SSL more vulnerable to real-world scenarios. Finally, in Section 2.3, we shift from a data-centric view to model design, examining related work on unified models and discussing their relevance to SSL.

2.1 Standard Semi-Supervised Learning

Semi-supervised learning (SSL) has emerged as a major field in deep learning, tackling scenarios where only a limited number of labeled examples are available, complemented by a substantial amount of unlabeled data. SSL methods aim to leverage this additional unlabeled data, uncovering latent patterns that can enhance learning performance. Below, we discuss some of the predominant approaches in SSL and their relations to our contributions. Note that while we primarily review SSL methods in image classification, the methods discussed here can be readily adapted to other domains, such as object detection, semantic segmentation, clustering, and regression.

2.1.1 Consistency regularization

Consistency regularization relies on the manifold or smoothness assumption, asserting that small, realistic perturbations of input data should not significantly alter a model's output. This idea has inspired a variety of methods to generate and leverage such perturbations. For instance, [TV17a] uses an exponential moving average (EMA) of the model as a teacher to produce alternative inputs for a student model. This idea has been further extended in [KWY+19], where they replace the EMA teacher with another student and force them to learn their own knowledge to avoid the performance bottleneck from the tight coupling. While [SJT16a, LA17] use random max-pooling and Dropout [SHK+14] to inject noise at the neural network level, [IPG+18] investigates the training process and improves generalization by averaging multiple points along the trajectory of stochastic gradient descent (SGD) with a cyclical learning rate, achieving flatter solutions than conventional SGD. [XDH+19, BCC+20, SBL+20, KMHK20] emphasize advanced data augmentation for introducing diverse perturbations. [BCG+19, VLK+19, BCC+20] use MixUp regularization [ZCDLP18] to promote convex behavior between examples. [GWL21] enforces label consistency with alpha-divergence, and [MMKI18a] introduces adversarial transformations to reinforce consistency, Among all methods, data augmentation-based noise injection [XDH+19] remains a particularly effective strategy, especially when combining advanced and weak augmentations to introduce substantial noise in unlabeled data, yielding considerable improvements [BCC+20, SBL+20]. For example, [SBL+20] incorporates pseudo-labeling by first generating pseudo-labels from weakly augmented images and then applying cross-entropy loss on the strongly augmented counterparts, enhancing consistency across transformations.

2.1.2 Pseudo-labeling

Unlike consistency regularization methods, pseudo-labeling approaches depend on high-confidence predictions, which are incorporated into the training dataset as labeled data. To generate these pseudo-labels for unlabeled data, various techniques have been proposed in the literature. For instance, EntMin [GB05] and Pseudo-label [Lee13] both use entropy minimization to select the most confident pseudo-label as a proxy ground truth for unlabeled instances. The Noisy Student method [XLHL20a] enhances pseudo-labeling by integrating data augmentation, dropout, and stochastic depth when training the student network, thereby improving robustness. Similarly, S4L [ZOKB19] employs data augmentation while also introducing a separate 4-class auxiliary task to boost performance. MPL [PXDL20] builds upon Pseudo-label [Lee13] by refining the teacher network's updates using feedback from the student network. Advances in data augmentation, meta-learning, and self-supervised learning, as well as powerful network architectures like EfficientNet [TL19] and SimCLR [CKNH20], have further strengthened self-training methods, enabling more accurate and reliable pseudo-labeling.

2.1.3 Generative models

Deep generative semi-supervised methods utilize generative models to enhance SSL by leveraging the distributional learning capabilities of generative approaches. A popular choice is Generative Adversarial Networks (GANs), which learn the real data distribution from unlabeled samples. For instance, CatGAN [Spr15] optimizes the discriminator to maximize mutual information between examples and their predicted class distributions, repurposing the discriminator as a classifier for SSL. GoodBadGAN [DYY+17] jointly trains a generator and classifier to solve a (K+1)-class problem, where the first K classes represent real data, and the (K+1)-th class represents synthetic images from the generator. Augmented BiGAN [KSF17] uses a pre-trained BiGAN generator to enrich SSL by injecting estimated tangents into the discriminator. MarginGAN [DL19] extends this by using a generator to maximize the margin of generated samples while the classifier minimizes the margin of fake images, addressing pseudo-label inaccuracy in SSL and improving overall accuracy. Structured GAN [DZL$^+$17] takes a distinct approach by conditioning sample generation on two independent latent variables (designated semantics and other variation factors) within a semi-supervised conditional generative modeling framework. Another prominent generative approach in SSL is based on Variational Autoencoders (VAEs). The foundational framework, M2 [KMJRW14], has inspired extensions such as ADGM [MSSW16] and ReVAE [JST$^+$20], which introduce auxiliary variables, albeit with different roles in each model. Infinite VAE [EADvdH17] combines several VAE models to boost overall performance, while Disentangled VAE [PVDMD$^+$17] and SDVAE [LPW$^+$19] tackle the semi-supervised VAE problem using various disentangling techniques. Under semi-supervised conditions with limited labeled data, these VAE-based methods focus on managing latent variables and label information to enhance performance effectively.

2.1.4 Graph-based methods

The idea of graph-based models for SSL is to perform label inference on a constructed similarity graph so that the label information can be propagated from the labeled samples to the unlabeled ones by incorporating both the topological and feature knowledge. A common solution is to find a function that is close to the given labels as possible, and smooth on the entire constructed graph. These two conditions can be further expressed in a general regularization framework in which loss function can be decomposed into a supervised loss term and a graph regularization term.

Traditional Graph-Based Methods: The choice of regularization differentiates various methods. [ZHS05] takes the directionality of the edges into consideration and incorporates the idea of naive random walk to perform label propagation on directed graphs. [GLT$^+$15] introduces deformed graph Laplacian (DGL) and provides the corresponding label prediction algorithm via DGL for SSL, where a new smoothness term that considers local information is added to the regularize. Motivated by the need to address the degeneracy of previous graph regularization methods when the label rate is meager, [CCTS20] proposes to replace the given label values with the assignment of sources and sinks like flow in the graph. Thus, a resulting Poisson equation based on the graph can be nicely solved.

GCN-Based Methods: With the recent success of GCN, there also exists a great number of GCN variants to enhance SSL. [LHW18] is the first to provide deep insights into GCN's success and failure on SSL tasks. Later, [JZL$^+$19] proposes to learn an optimal graph structure that best serves graph CNNs for semi-supervised learning by integrating both graph learning and graph convolution in a unified network architecture. [XCH$^+$20] proposes a Graph Inference Learning framework to boost the performance of semi-supervised node classification by learning the inference of node labels on graph topology.

Scalable and Uncertainty-Aware Graph Methods: [LBT$^+$18] introduces graph partition neural networks to handle extremely large graphs by alternating between locally propagating information between nodes in small subgraphs and globally propagating information between the subgraphs. [ZPCU19] adopts a Bayesian approach to incorporate uncertainty in the graph structure by viewing the observed graph as a realization from a parametric family of random graphs. [VYBT19] proposes to estimate label scores along with their confidences jointly in the GCN-based setting and uses these estimated confidences to determine the influence of one node on another during neighborhood aggregation, thereby acquiring anisotropic capabilities.

2.1.5 Connection to our work

In Chapter 3, we revisit the consistency regularization framework with data augmentation due to its recent impressive performance [LXH21, ZYH+22]. While efforts have been made to come up with more advanced and diverse augmentation strategies [BCC+20, SBL+20], we question whether it is always beneficial to make the model invariant to such strong perturbations. In strong contrast with the previous work, we find that while enforcing invariance by decreasing distances between features from differently augmented images leads to improved performance, encouraging equivariance instead, by increasing the feature distance, further improves performance. Based on this, we propose a novel feature distance loss to explicitly regularize representation learning at both the feature and the classifier level and show improved generalization performance. Moreover, as self-supervised learning has been shown to be helpful in the context of SSL [ZOKB19, BCC+20], we also incorporate an auxiliary self-supervised loss alongside training, which is different from [HFW+20, CKNH20, CKS+20a, REH+20], where self-supervised pre-training is used to initialize SSL and several training phases are involved.

In Chapter 4, we explore pseudo-labeling, a prominent paradigm in semi-supervised learning (SSL) [SBL+20], with a specific focus on threshold-based pseudo-labeling. Our initial investigation reveals that during early training stages, a low threshold is beneficial to encourage diverse pseudo-labels, maximizing the utilization of unlabeled data and accelerating convergence. However, a consistently low threshold as training progresses induces substantial confirmation bias. To address this, we propose FreeMatch, a method that dynamically adjusts the confidence threshold in a self-adaptive manner based on the model's current learning status. This approach differs from [SBL+20] as it introduces an adaptive threshold, and from [ZWH+21] as it avoids reliance on additional hyper-parameters. Additionally, inspired by the principle of distribution alignment [BCC+20], we incorporate a self-adaptive class fairness regularization penalty to promote diverse predictions during the early stages of training.

In Chapter 5, we delve further into pseudo-labeling, specifically examining the quantity-quality trade-off. A high confidence threshold [SBL+20] can ensure the quality of pseudo-labels but often discards a significant number of unconfident yet accurate labels, while a dynamically growing threshold [ZWH+21, XSY+21, BRS+22] increases pseudo-label utilization but inadvertently incorporates erroneous labels that may mislead training. To address this, we introduce a unified perspective on sample weighting formulation for threshold-based pseudo-labeling, demonstrating that existing methods differ mainly in their weighting functions and all face the quantity-quality trade-off challenge. We propose SoftMatch, which effectively leverages unconfident yet accurate pseudo-labels by fitting a truncated Gaussian function to the confidence distribution, enabling a flexible weighting function. Additionally, we propose Uniform Alignment to mitigate pseudo-label imbalance while preserving both high quantity and quality.

2.2 Realistic Semi-Supervised Learning

Recent advancements in standard semi-supervised learning (SSL) have shown significant progress in leveraging unlabeled data, helping models to generalize better with limited labeled examples. However, these methods often operate under the assumption that data is well-distributed, relevant, and clean. In real-world applications, data is rarely ideal; it tends to be imbalanced, may include outliers, and often exhibits distributional shifts that standard

SSL techniques do not account for. Furthermore, current SSL methods are typically evaluated on benchmarks that focus almost exclusively on curated, computer vision datasets. This narrow focus limits their applicability in more diverse, real-world settings and exposes a gap between research-driven SSL developments and practical deployment. In this section, we review related work in realistic SSL, emphasizing recent efforts to address these challenges through imbalanced SSL, open-set SSL, and the development of more representative SSL benchmarks.

2.2.1 Imbalanced SSL

Imbalanced semi-supervised learning (SSL) addresses a setting where the training data is only partially labeled, and both labeled and unlabeled data follow a long-tailed distribution. This setting generalizes traditional long-tailed recognition and standard SSL, presenting unique challenges for current algorithms. In this context, pseudo-labels generated by models trained on imbalanced data often skew heavily towards majority classes. Consequently, a primary approach in imbalanced SSL is to refine pseudo-labels to better represent the true distribution of the data. For instance, [KHP+20b] addresses bias in pseudo-labels by formulating a Lagrangian dual optimization problem that minimizes distortion from the original pseudo-labels, aligning them more closely with the true distribution of the training data. Similarly, [WSM+21] observes that standard SSL methods tend to exhibit high recall but low precision for majority classes, while minority classes display the reverse. To balance this, they propose a reverse sampling method for the unlabeled data, compensating for these biases. Additionally, [OKK21a] introduces a strategy to generate more balanced pseudo-labels by blending linear and semantic pseudo-labels in different proportions based on the unlabeled data distribution, then replacing the original linear pseudo-labels with these blended alternatives. [GL22] improves class balance by dynamically adjusting the pseudo-labeling threshold per class, using both the count and confidence of selected pseudo-labels to ensure equal representation across classes. Beyond refining pseudo-labels, there exist other methods that tackle class imbalance directly. For instance, [HJK20] introduces a suppressed consistency loss to protect decision boundaries for minority classes, preventing excessive smoothing in low-sample regions. [HKY+21] takes a progressive approach, gradually shifting the data distribution during training: initially prioritizing feature extraction, and later rebalancing to train the classifier effectively. Lastly, [LSK21] addresses model bias by introducing an auxiliary balanced classifier (ABC) with additional regularization terms, which helps rebalance the model's predictions across classes.

2.2.2 Long-tailed recognition

Since many challenges in imbalanced SSL mirror those of long-tailed distribution problems, many successful imbalanced SSL methods are directly inspired by long-tailed recognition techniques. Thus, it is valuable to also briefly discuss related work in standard long-tailed recognition. This field offers a variety of solutions, which can be broadly categorized as follows: 1. Re-sampling methods that aim to adjust the data distribution by either oversampling tail-class instances or undersampling head-class instances, balancing class representation within the training data [KJS20, LLK+21, MGR+18, PBS+20, SLH16, LWZ08, Wil72]. 2. Re-weighting methods that assign different weights to classes or instances, emphasizing tail classes to counterbalance their underrepresentation and intensify learning for these classes [CJL+19, HHC+21, HLLT16, TLZ+21, TWL+20, WZZ+21, WHL+20, ZCLJ21, HRCH21, LLW19, RD17, SGG16]. 3. Decoupling-based methods that separate representation learning from classifier learning and optimize them independently [KXR+20, WLK+20, ZLY+21, ZCLJ21]. 4. Metric learning methods that focus on learning an embedding space that accurately reflects the similarity of embedded features, often enforcing a larger margin for tail classes to reduce class overlap and improve minority classes [DGZ18, LSH+20b, WGY+19, ZFW+17, CZL+21]. 5. Transfer learning methods that transfer knowledge from head classes to tail classes, allowing tail classes to benefit from the richer information available in head-class representations [HJT+20, WRH17, LMZ+19]. 6. Ensemble-based methods that combine multiple expert models to make final predictions more balanced across classes [WLM+20, ZCT+21, ZCWC20]. 7. Knowledge distillation methods where a student model is trained in a more balanced manner by using guidance from an expert model [HWW21, XDH20a, ZCHP23]. 8. Grouping-based methods where data are divided into groups based on specific relationships or characteristics to reduce the imbalance [LWK+20, WSW+20]. These strategies, when adapted to the semi-supervised learning context, offer promising directions for addressing the unique challenges posed by imbalanced SSL.

2.2.3 Open-set SSL

The challenge of out-of-distribution (OOD) samples in unlabeled data, first identified by [OOR+18], reveals that standard SSL methods degrade in performance when OOD data is present. To address this, numerous approaches have been developed to mitigate the impact of OOD samples [CZLG20, GZJ+20a, YIIA20a, SKS21, HFC+21, PYJS22, HHLY22, HHYY22, HYG22]. Most existing methods aim to filter out OOD data, ensuring that the classifier is trained predominantly with in-distribution (ID) samples. For instance, [CZLG20] introduces soft target generation, which prevents catastrophic error propagation and enables effective learning from unconstrained, OOD-containing unlabeled data. [GZJ+20a] employs bi-level optimization to reduce the influence of OOD data through adaptive loss weighting, ensuring that performance is never compromised below a baseline of labeled-only learning. [YIIA20a] utilizes a joint optimization framework to iteratively update both the model parameters and OOD scoring, ultimately selecting ID samples with low OOD scores. [SKS21] integrates FixMatch [SBL+20] with novelty detection by applying one-vs-all (OVA) classifiers [SS21] and a consistency loss that enhances OVA-classifier stability under transformations, significantly improving outlier detection. [HFC+21] proposes a cross-modal matching module that detects outliers by generating a compatible feature space aligned with the main classification task. In a different vein, [HHLY22] leverages "recyclable" OOD data by identifying OOD samples closely related to ID data, using adversarial domain adaptation to project these samples into the ID feature space. Additionally, [ZKIC20] optimizes a teacher-student framework with an energy-discrepancy scoring function that enhances the security of sample selection, and [LCM+21] combines meta-learning with Weighted Batch Normalization to suppress OOD influence at the feature level.

2.2.4 Open-set recognition

Since open-set SSL shares many similarities with open-set recognition (OSR), as both address scenarios where unknown classes emerge during testing, it is valuable to discuss related work in open-set recognition. The OpenMax model [BB16] was one of the first deep-learning-based open-set classifiers to function without using background samples. It introduces the OpenMax layer, which estimates the probability that an input belongs to an unknown class, allowing for the rejection of "fooling" and unrelated open-set images presented to the system. An alternative approach, [LLCD19], proposes an improved instance representation method that clusters same-class instances while distancing instances of different classes. This representation enhances detection by creating larger separation zones where unknown classes can reside. The work in [CFG15] develops based on WiSARD [ADGF$^+$09], incorporating a rejection mechanism that flags inputs as outliers if the highest or best-matching class score falls below a predefined threshold. To enhance open-set rejection capabilities, [OP19b] employs a multi-task learning framework combining a classifier and a decoder network with a shared feature extractor. Here, reconstruction errors from the decoder network are utilized for open-set rejection. [OP19a] uses conditioned autoencoders for open-set identification by reconstructing inputs conditioned on class identity. Reconstruction errors are analyzed using Extreme Value Theory to define thresholds for distinguishing known and unknown samples. Recently, Generative Adversarial Networks (GANs) have been widely explored in OSR. G-OpenMax [GDCG17] extends OpenMax by generating unknown instances through GANs to simulate adversarial scenarios. Building on a similar concept, [NOF$^+$18] introduces a GAN-based method that generates examples close to, yet distinct from, the training set, training the model to detect outliers effectively. [YQLG17] introduces the adversarial sample generation (ASG) framework for open-category classification, generating both positive and negative samples from seen categories in an unsupervised manner via adversarial learning. Open-GAN [YHL$^+$19] also builds on adversarial learning by creating fake target samples with a generator and modifying the discriminator to accommodate multiple known classes alongside an unknown class. Additionally, Open Set Back Propagation (OSBP) [SYUH18] utilizes adversarial training for a more complex open-set framework that operates without requiring unknown source samples, pushing the boundaries of open-set recognition in deep learning.

2.2.5 Open-world SSL

Open-world SSL [CBL21, RKK$^+$22, RKS22] shares similarities with open-set SSL but also has several critical differences. Both approaches include unlabeled data from novel classes during training; however, the goal of open-world SSL is to classify inliers and discover new classes from out-of-distribution (OOD) data rather than rejecting them. Another key difference is that open-world SSL typically operates in a transductive learning setting, while open-set SSL necessitates generalization beyond the training distribution. [CBL21] first introduces the open-world SSL setting, along with an uncertainty adaptive margin mechanism that reduces the gap between intra-class samples and novel classes. Using a bi-level optimization framework, [RKK$^+$22] leverages pairwise similarity loss to exploit labeled set information, thereby implicitly clustering samples of novel classes. In [RKS22], a pseudo-labeling approach addresses SSL in the open-world setting by incorporating sample uncertainty and prior knowledge about class distributions, yielding reliable, distribution-aware pseudo-labels for both known and unknown class data. Finally, [WSKH24] introduces a framework that learns from all unlabeled data via self-supervision and an energy-based scoring system to identify known-class data accurately, making the approach well-suited for handling uncurated data in deployment.

2.2.6 SSL evaluation

Despite significant advancements in SSL, limited research has focused on establishing realistic benchmarks and evaluation protocols for assessing SSL methods. Current SSL benchmarks are primarily limited to basic computer vision tasks (e.g., CIFAR-10/100 [KH$^+$09a], SVHN

[NWC$^+$11b], STL-10 [CNL11a], ImageNet [DDS$^+$09]), which restricts their relevance in estimating generalization to diverse settings and real-world data. Besides, evaluation practices vary significantly across papers, with SSL methods often developed and tested under inconsistent experimental setups, making fair comparison and reliable assessment challenging. To address these gaps, [OOR$^+$18] proposes a unified reimplementation of widely-used SSL techniques, evaluating them across a suite of experiments intended to reflect real-world constraints. It covers four SSL algorithms and three classification tasks. Building on this, TorchSSL [ZWH$^+$21] extends the framework by reimplementing nine SSL algorithms and evaluating them on five classification tasks, further advancing efforts toward standardized, comprehensive SSL evaluation.

2.2.7 Connection to our work

In Chapter 6, we propose a novel co-learning framework, CoSSL, for tackling imbalanced SSL. CoSSL decouples representation learning from classifier learning, while coupling them through a shared encoder and pseudo-label generation. Our method is closely related to the decoupling-based methods [KXR$^+$20, WLK$^+$20, ZLY$^+$21, ZCLJ21] commonly used in long-tailed recognition, which also separate representation and classifier learning. However, CoSSL uniquely connects these two modules via a shared encoder, allowing them to exchange information and effectively bootstrap each other. Among existing imbalanced SSL methods, [LSK21] is the most closely related to CoSSL. The distinguishing features of CoSSL are: (1) the decoupling of representation and classifier training; (2) an active connection between the encoder and classifier modules through pseudo-labeling; and (3) tailored handling of tail classes by leveraging unlabeled data.

In Chapter 7, we introduce a Simple but Strong Baseline (SSB) for open-set SSL. Rather than filtering out OOD data, our approach applies a straightforward, confidence-based pseudo-labeling method to incorporate these samples, significantly improving classification outcomes. This approach diverges sharply from prior methods, which typically attempt to filter out OOD data before or during training [CZLG20, GZJ$^+$20a, YIIA20a, SKS21, HFC$^+$21, PYJS22, HHLY22, HHYY22, HYG22]. Additionally, rather than directly sharing features between the classifier and detector, we implement task-specific, non-linear transformations for each head. This separation effectively minimizes mutual interference, allowing for more specialized feature learning and enhancing performance for both tasks. To further strengthen the outlier detector, we introduce pseudo-negative mining, which improves OOD detector training by generating diverse pseudo-outliers, increasing data diversity and enabling more robust detection.

In Chapter 8, we introduce a unified and comprehensive semi-supervised learning benchmark for classification, spanning 15 tasks across computer vision (CV), natural language processing (NLP), and audio domains to facilitate fair and consistent evaluations. Unlike [OOR$^+$18] and [ZWH$^+$21], this benchmark is the first to explore the generalizability of SSL methods beyond CV tasks, examining whether existing SSL techniques that perform well in CV also translate effectively to NLP and audio. Furthermore, we implement an environmentally conscious and cost-efficient evaluation protocol by employing a pre-training and fine-tuning paradigm, substantially lowering the computational demands of SSL experiments. Following the practices in [OOR$^+$18] and [ZWH$^+$21], we have open-sourced a modular codebase along with configuration files, enabling easy reproduction of our reported results. Additionally, we provide comprehensive documentation and tutorials to support modifications. Our codebase is designed to be extensible, inviting contributions from the community, where new algorithms, models, configuration files, and results can be continually added to advance the field.

2.3 Vision generalist models

The success of modern AI systems is not only dependent on data but also critically shaped by model design. Recently, there has been significant interest in developing models that can tackle a wide variety of vision tasks within a unified framework. Inspired by the success of sequence-to-sequence modeling in Natural Language Processing (NLP) [TLI+23, DCLT18a, RWC+19, RSR+20], researchers have aimed to create generalist models for vision, integrating diverse tasks under a common approach.

Pixel-to-Sequence Frameworks: Several efforts in this direction demonstrate promising results. For instance, [CSL+21] introduces a language modeling approach for object detection, representing object attributes (e.g., bounding boxes and class labels) as sequences of discrete tokens, transforming object detection into a language modeling task conditioned on pixel inputs. Expanding on this, [CSL+22] extends the model to cover instance segmentation, keypoint detection, and image captioning, all mapped to a pixel-to-sequence framework. Similarly, [LCZ+22] develops a unified model that spans 90 different datasets, covering tasks like pose estimation, object detection, depth estimation, image generation, vision-language interaction, and NLP tasks. This model standardizes inputs and outputs as sequences of discrete tokens, further advancing task unification. [WWC+23] redefines the output of core vision tasks as images, and proposes an in-context learning generalist that unifies vision tasks as standard masked image modeling on the stitch of input and output image pairs. Based on the in-context learning generalist, [WZC+23] introduces a unified segmentation model via an in-context coloring problem with random color mapping for each data sample. The objective is to accomplish diverse tasks according to the context, rather than relying on specific colors. [WYM+22] presents a simple sequence-to-sequence framework unifying a range of cross-modal and unimodal tasks, including image generation, visual grounding, image captioning, classification, and language modeling. Moreover, [GYH+24, GPS+23] leverage the power of diffusion models and steer its functionality to an instruction-guided multi-task vision learner via instruction-tuning.

Task-Specific Customization within Unified Architectures: Alongside these approaches, another line of research aims for unified architectures with task-specific customization. For example, UViM [KSPB+22] addresses the high-dimensionality of vision outputs by using a guiding code—a short sequence encoded with task-specific information, allowing the main model to focus on task-specific predictions. Here, individual models are trained for each task, as the guiding code varies across tasks. Similarly, XDecoder [ZDY+23] unifies pixel-level segmentation, image-level retrieval, and vision-language tasks through a generic decoding procedure, predicting pixel-level masks and token-level semantics and combining these outputs differently for each task.

2.3.1 Connection to our work

In Chapter 9, we investigate diffusion-based vision generalists by unifying various dense prediction tasks under a conditional image generation framework, adapting pre-trained diffusion models for these tasks. Similar to [WWC+23], we reframe outputs from different vision tasks as RGB images, but unlike [WWC+23] using image inpainting, our model is trained purely as a conditional image generator, eliminating the need for additional in-context examples at test time. Our approach also differs from other diffusion-based vision generalists [GYH+24, GPS+23], by utilizing pixel diffusion rather than latent diffusion, which can suffer from quantization artifacts during upsampling. Furthermore, we enhance the image conditioning of the diffusion model by using powerful pre-trained vision encoders that extract semantically rich image features. This feature-based conditioning yields superior performance compared to models conditioned directly on raw images.

I

Standard Semi-Supervised Learning

In the first part of the thesis, we focus on standard semi-supervised learning (SSL) and investigate strategies to effectively leverage unlabeled data to enhance representation learning. Specifically,

in Chapter 3, we revisit the idea of augmentation-based consistency regularization and find that improving equivariance on strongly augmented images can provide even better performance rather than making the model invariant to all kinds of augmentations. To this end, we formulate FeatDistLoss to explicitly encourage equivariance between features from different augmentations while enforcing the same semantic class label.

in Chapter 4, we address the quantity-quality trade-off in pseudo-labeling and introduce a self-adaptive, parameter-free thresholding scheme that dynamically adjusts thresholds based on the current learning status. Our method yields consistent improvements across various settings. For scenarios with minimal supervision, we further introduce a class fairness objective to guide the model in effectively handling scarcely labeled classes.

in Chapter 5, we explore an alternative approach to threshold-based pseudo-labeling by fitting a sample-specific weight function. Here, we utilize a truncated Gaussian fit on the confidence distribution, which assigns smaller weights to potentially correct pseudo-labels with lower confidence. Additionally, we propose Uniform Alignment, a strategy to mitigate pseudo-label imbalance while maintaining both high quality and quantity.

3 Consistency Regularization for Semi-Supervised Learning

Contents

Semi-supervised learning (SSL) deals with scenarios where limited labeled data is available, and aims to enhance model performance by effectively leveraging the large amount of unlabeled data. Consistency regularization is one of the most widely-used techniques for SSL. Generally, the aim is to train a model that is invariant to various data augmentations. In this chapter, we revisit this idea and find that enforcing invariance by decreasing distances between features from differently augmented images leads to improved performance. However, encouraging equivariance instead, by increasing the feature distance, further improves performance. To this end, we propose an improved consistency regularization framework by a simple yet effective technique, FeatDistLoss, that imposes consistency and equivariance on the classifier and the feature level, respectively. Experimental results show that our model defines a new state of the art across a variety of standard semi-supervised learning benchmarks as well as imbalanced semi-supervised learning benchmarks. Particularly, we outperform previous work by a significant margin in low data regimes and at large imbalance ratios. Extensive experiments are conducted to analyze the method.

This Chapter is based on [FKS21b]. Yue Fan was the lead author of this paper and conducted all the experiments and wrote most parts of the paper.

3.1 Introduction

Deep learning requires large-scale and annotated datasets to reach state-of-the-art performance [RDS$^+$15a, LMB$^+$14]. As labels are not always available or expensive to acquire a wide range of semi-supervised learning (SSL) methods have been proposed to leverage unlabeled data [TV17a, LA17, MMKI18b, VLK$^+$19, BCG$^+$19, SBL$^+$20, XDH$^+$19, BCC$^+$20, AOA$^+$20a, Lee13, PXDL20, FOS20, BHB19, CKS$^+$20a].

Consistency regularization [BAP14a, LA17, SJT16a] is one of the most widely-used SSL methods. Recent work [SBL$^+$20, XDH$^+$19, KMHK20] achieves strong performance by utilizing unlabeled data in a way that model predictions should be invariant to input perturbations.

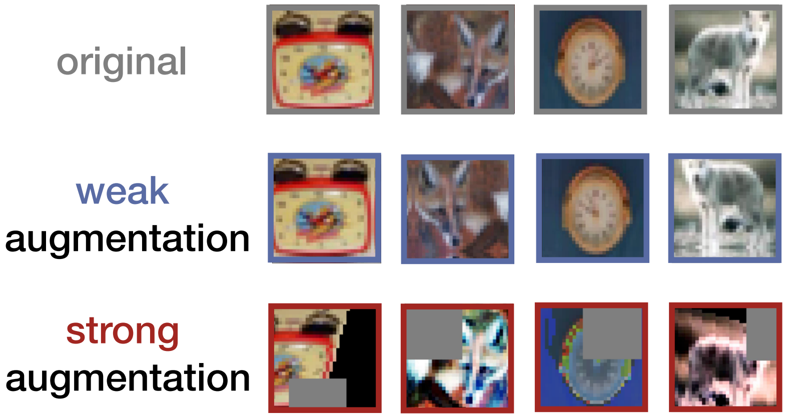


Figure 3.1: Examples of strongly and weakly augmented images from CIFAR-100 (please refer to Section 3.2.3 for details of strong and weak augmentation). The visually large difference between them indicates that it can be more beneficial if they are treated differently.

However, when using advanced and strong data augmentation schemes, we question if the model should be invariant to such strong perturbations. In Figure 3.1 we illustrate that strong data augmentation leads to perceptually highly diverse images. Thus, we argue that improving equivariance on such strongly augmented images can provide even better performance rather than making the model invariant to all kinds of augmentations. Moreover, existing works apply consistency regularization either at the feature level or at the classifier level. We find empirically that it is more beneficial to introduce consistency on both levels. To this end, we propose a simple yet effective technique, Feature Distance Loss (FeatDistLoss), to improve data-augmentation-based consistency regularization.

We formulate our FeatDistLoss as to explicitly encourage invariance or equivariance between features from different augmentations while enforcing the same semantic class label. Figure 3.2 shows the intuition behind the idea. Specifically, encouragement of equivariance for the same image but different augmentations (increase distance between stars and circles of the same color) pushes representations apart from each other, thus, covering more space for the class. Imposing invariance, on the contrary, makes the representations of the same semantic class more compact. In this work, we empirically find that increasing equivariance to differently augmented versions of the same image can lead to better performance especially when rather few labels are available per class (see section 3.4.3).

This chapter introduces the method *CR-Match* which combines FeatDistLoss with other strong techniques defining a new state-of-the-art across a wide range of settings of standard SSL benchmarks, including CIFAR-10, CIFAR-100, SVHN, STL-10, and Mini-Imagenet. More specifically, our contribution is fourfold. (1) We improve data-augmentation-based consistency regularization by a simple yet effective technique for SSL called *FeatDistLoss* which regularizes the distance between feature representations from differently augmented images of the same class as well as the classifier simultaneously. (2) We show that while encouraging invariance results in good performance, encouraging equivariance to differently augmented versions of the same image consistently results in even better generalization performance. (3) We provide comprehensive ablation studies on different distance functions and different augmentations with respect to the proposed FeatDistLoss. (4)

In combination with other strong techniques, we achieve *new state-of-the-art results* on most standard semi-supervised learning benchmarks as well as imbalanced semi-supervised learning benchmarks. In particular, our method outperforms previous methods by a significant margin

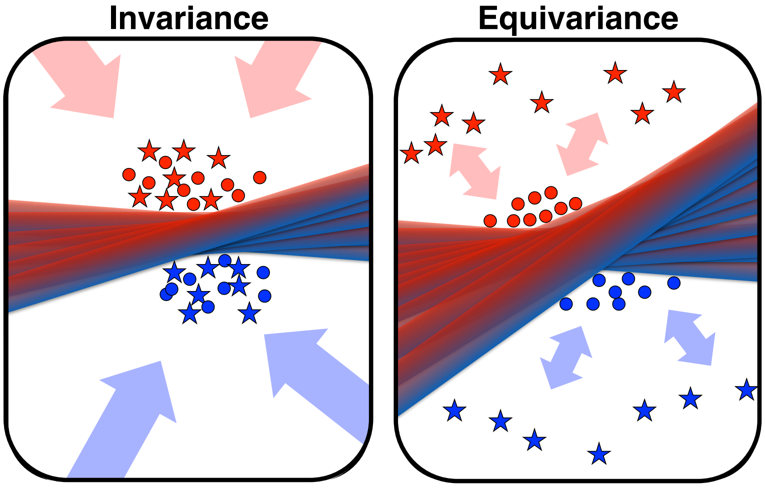


Figure 3.2: Binary classification task. Stars are features of strongly augmented images and circles are of weakly augmented images (please refer to Section 3.2.3 for details of strong and weak augmentation). While encouraging invariance by decreasing distance between features from differently augmented images gives good performance (left), encouraging equivariant representations by increasing the distance regularizes the feature space more, leading to even better generalization performance.

in low data regimes and at large imbalance ratios.

A preliminary version of this work has been published in [FKS21a]. In this work, we extend [FKS21a] in three aspects: (1) We extend the existing standard SSL settings by providing evaluations on wider range of the datasets and showing the benefit of the proposed technique on top of various SSL methods. In particular, combining with the recently published method FlexMatch [ZWH$^+$21], we can push the state-of-the-art even further under the standard settings. Moreover, we evaluate our method on ImageNet to verify that the method scales to larger datasets as well. (2) We evaluate our methods under a more realistic and challenging setting: imbalanced SSL, where the training data is not only partially annotated but also exhibits long-tailed class distribution. We achieve new state-of-the-art results on multiple imbalanced SSL benchmarks across a wide range of settings. (3) To give more in-depth insight into our method, we provide pseudo-code and more analysis of the method, especially the robustness against important hyper-parameters.

3.2 CR-Match

Consistency regularization is highly-successful and widely-adopted technique in SSL [BAP14a, LA17, SJT16a, SBL$^+$20, XDH$^+$19, KMHK20]. In this work, we aim to leverage and improve it by even further regularizing the feature space. To this end, we present a simple yet effective technique FeatDistLoss to explicitly regularize representation learning and classifier learning at the same time. We describe our SSL method, called CR-Match, which shows improved performance across many different settings, especially in scenarios with few labels. In this section, we first describe our technique FeatDistLoss and then present CR-Match that combines FeatDistLoss with other regularization techniques inspired from the literature.

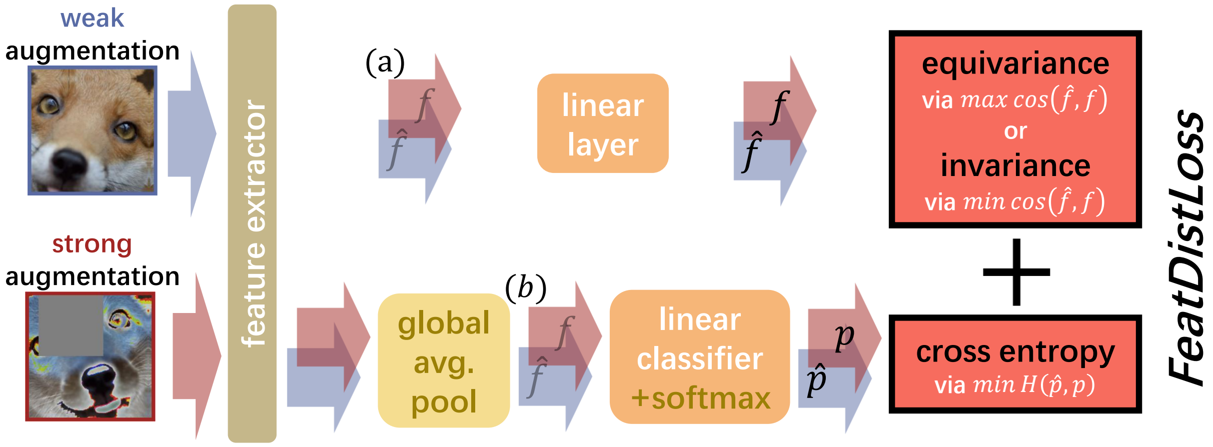


Figure 3.3: The proposed FeatDistLoss utilizes unlabeled images in two ways: On the classifier level, different versions of the same image should generate the same class label, whereas on the feature level, representations are encouraged to become either more equivariant (pushing away) or invariant (pulling together). f and $\hat{f}$ denote strong and weak features; p and $\hat{p}$ are predicted class distributions from strong and weak features; a) and b) denote features before and after the global average pooling layer. Our final model takes features from a) and encourages equivariance to differently augmented versions of the same image. An ablation study of other choices is in section 3.4.3.

3.2.1 Feature Distance Loss

Background: The idea of consistency regularization [BAP14a, LA17, SJT16a] is to encourage the model predictions to be invariant to input perturbations. Given a batch of n unlabeled images $\mathbf{u}_i, i \in (1, ..., n)$, consistency regularization can be formulated as the following loss function:

$$\frac{1}{n}\sum_{i=1}^{n} \|f(\mathcal{A}(\mathbf{u}_i)) - f(\alpha(\mathbf{u}_i))\|_2^2 \tag{3.1}$$

where f is an encoder network that maps an input image to a d-dimensional feature space; $\mathcal{A}$ and α are two stochastic functions which are, in our case, strong and weak augmentations, respectively (details in Section 3.2.3). By minimizing the L_2 distance between perturbed images, the representation is therefore encouraged to become more invariant with respect to different augmentations, which helps generalization. The intuition behind this is that a good model should be robust to data augmentations of the images.

FeatDistLoss: As shown in Figure 3.3, we extend the above consistency regularization idea by introducing consistency on the classifier level and invariance or equivariance on the feature level. FeatDistLoss thus allows to apply different types of control for these levels. In particular, when encouraging to reduce the feature distance, it becomes similar to classic consistency regularization, and encourages invariance between differently augmented images. As argued above, making the model predictions invariant to input perturbations gives good generalization performance. Instead, in this work we find it is more beneficial to treat images from different augmentations differently because some distorted images are largely different from their original images as demonstrated visually in Figure 3.1. Therefore, the final model (CR-Match) uses FeatDistLoss to increase the distance between image features from augmentations of different intensities while at the same time enforcing the same semantic label for them. Note that in Section 3.4.3, we conduct an ablation study on the choice of distance function, where we denote CR-Match as CR-Equiv, and the model that encourages invariance as CR-Inv.

The final objective for the FeatDistLoss consists of two terms: $\mathcal{L}_{Dist}$ (on the feature level), that explicitly regularizes feature distances between embeddings, and a standard cross-entropy loss $\mathcal{L}_{PseudoLabel}$ (on the classifier level) based on pseudo-labeling.

With $\mathcal{L}_{Dist}$ we either decrease or increase the feature distance between weakly and strongly augmented versions of the same image in a low-dimensional space projected from the original feature space to overcome the curse of dimensionality [Bel66]. Let $d(\cdot,\cdot)$ be a distance metric and z be a linear layer that maps the high-dimensional feature into a low-dimensional space. Given an unlabeled image $\mathbf{u}_i$, we first extract features with strong and weak augmentations by $f(\mathcal{A}(\mathbf{u}_i))$ and $f(\alpha(\mathbf{u}_i))$ as shown in Figure 3.3 (a), and then FeatDistLoss is computed as:

$$\mathcal{L}_{Dist}(\mathbf{u}_i) = d(z(f(\mathcal{A}(\mathbf{u}_i))), z(f(\alpha(\mathbf{u}_i)))) \tag{3.2}$$

Different choices of performing $\mathcal{L}_{Dist}$ are studied in Section 3.4.3, where we find empirically that applying $\mathcal{L}_{Dist}$ at (a) using cosine distance in Figure 3.3 gives the best performance. The use of the projection head z does not only reduce the computation burden as the original feature space is high-dimensional, but also brings additional performance improvements as shown in [CKNH20, CKS+20a].

At the same time, images from strong and weak augmentations should have the same class label because they are essentially generated from the same original image. Inspired by [SBL+20], given an unlabeled image $\mathbf{u}_i$, a pseudo-label distribution is first generated from the weakly augmented image by $\hat{\mathbf{p}}_i = g(f(\alpha(\mathbf{u}_i)))$, and then a cross-entropy loss is computed between the pseudo-label and the prediction for the corresponding strongly augmented version as:

$$\mathcal{L}_{PseudoLabel}(\mathbf{u}_i) = \ell_{CE}(\hat{\mathbf{p}}_i, g(f(\mathcal{A}(\mathbf{u}_i)))) \tag{3.3}$$

where ℓ_{CE} is the cross-entropy, g is a linear classifier that maps a feature representation to a class distribution, and $\mathcal{A}(\mathbf{u}_i)$ denotes the operator for strong augmentations.

Putting it all together, FeatDistLoss processes a batch of unlabeled data $\mathbf{u}_i, i \in (1, ..., B_u)$ with the following loss:

$$\mathcal{L}_U = \frac{1}{B_u} \sum_{i=1}^{B_u} \mathbb{1}\{c_i > \tau\}(\mathcal{L}_{Dist}(\mathbf{u}_i) + \mathcal{L}_{PseudoLabel}(\mathbf{u}_i)) \tag{3.4}$$

where $c_i = max\ \hat{\mathbf{p}}_i$ is the confidence score, and $\mathbb{1}\{\cdot\}$ is the indicator function which outputs 1 when the confidence score is above a threshold. This confidence thresholding mechanism ensures that the loss is only computed for unlabeled images for which the model generates a high-confidence prediction. Therefore, it controls the trade-off between the quality and the quantity of contributing unlabeled samples. As is shown in Section 3.4.2, a higher threshold τ is normally preferred because it alleviates the instability early in the training by eliminating less confident unlabeled samples. As training progresses, the model produces more confident predictions and more samples will contribute to the final loss, which also provides a natural curriculum to balance labeled and unlabeled losses [SBL+20]. Moreover, the thresholding mechanism is applied for both the feature level consistency and the classifier level consistency so that the two losses are well-synchronized.

As mentioned before, depending on the function d, FeatDistLoss can decrease the distance between features from different data augmentation schemes (when d is a distance function, thus pulling the representations together), or increase it (when d is a similarity function, thus pushing the representations apart). As shown in Table 3.6, we find that both cases results in an improved performance. However, increasing the distance between weakly and strongly augmented examples consistently results in better generalization performance. We conjecture that the reason lies in the fact that FeatDistLoss by increasing the feature distance explores equivariance properties (differently augmented versions of the same image having distinct features but the same label) of the representations. It encourages the model to have more distinct weakly and strongly augmented images while still imposing the same label, which leads to both more expressive representation and more powerful classifier. As we will show in Section 3.4.3, information like object location or orientation is more predictable from models trained with FeatDistLoss that pushes the representations apart. Additional ablation studies of other design choices such as the distance function and the linear projection z are also provided in Section 3.4.3.

3.2.2 Overall CR-Match

Algorithm 1 CR-Match Algorithm.

```
Require: Labeled batch $\mathcal{X} = \{(\mathbf{x}_i, \mathbf{p}_i) : i \in (1, \ldots, B_s)\}$, unlabeled batch $\mathcal{U} = \{\mathbf{u}_i : i \in (1, \ldots, B_u)\}$, confidence threshold $\tau$, FeatDistLoss weight $\lambda_u$, rotation prediction loss weight $\lambda_r$, classifier $g$, distance metric $d$, FeatDistLoss head $z$, rotation prediction head $h$.
▷ Cross-entropy loss for labeled data
$\mathcal{L}_S = \frac{1}{B_s} \sum_{i=1}^{B_s} \ell_{CE}(\mathbf{p}_i, g(\alpha(\mathbf{x}_i)))$
for $i = 1$ to $B_u$ do
    ▷ Extract representation from weak data augmentation
    $\mathbf{u}_i^w = f(\alpha(\mathbf{u}_i))$
    ▷ Extract representation from strong data augmentation
    $\mathbf{u}_i^s = f(\mathcal{A}(\mathbf{u}_i))$
    ▷ Compute confidence score from the weakly augmented image
    $c_i = max\ g(\mathbf{u}_i^w)$
end for
▷ Cross-entropy loss with pseudo-label for unlabeled data
$\mathcal{L}_{Pseudo} = \frac{1}{B_u} \sum_{i=1}^{B_u} \mathbb{1}\{c_i > \tau\}\ \ell_{CE}(g(\mathbf{u}_i^w), \mathbf{u}_i^s)$
▷ Increase the feature distance for unlabeled data
$\mathcal{L}_{Dist} = \frac{1}{B_u} \sum_{i=1}^{B_u} \mathbb{1}\{c_i > \tau\}\ - d(z(\mathbf{u}_i^w), z(\mathbf{u}_i^s))$
▷ rotation prediction loss
$\mathcal{L}_{Rot} = \frac{1}{4B_u} \sum_{i=1}^{B_u} \sum_{r \in \mathbb{R}} \ell_{CE}(r, h(R(\mathbf{u}_i^w, r)))$
return $\mathcal{L}_S + \lambda_u(\mathcal{L}_{Pseudo} + \mathcal{L}_{Dist}) + \lambda_r \mathcal{L}_{Rot}$
```

Now we describe our SSL method called CR-Match leveraging the above FeatDistLoss. Pseudo-code for processing a batch of labeled and unlabeled examples is shown in algorithm 1.

Given a batch of labeled images with their labels as $\mathcal{X} = \{(\mathbf{x}_i, \mathbf{p}_i) : i \in (1, ..., B_s)\}$ and a batch of unlabeled images as $\mathcal{U} = \{\mathbf{u}_i : i \in (1, ..., B_u)\}$. [1] CR-Match minimizes the following learning objective:

$$\mathcal{L}_S(\mathcal{X}) + \lambda_u \mathcal{L}_U(\mathcal{U}) + \lambda_r \mathcal{L}_{Rot}(\mathcal{U}) \tag{3.5}$$

where $\mathcal{L}_S$ is the supervised cross-entropy loss for labeled images with weak data augmentation regularization; $\mathcal{L}_U$ is our novel feature distance loss for unlabeled images which explicitly regularizes the distance between weakly and strongly augmented images in the feature space;

[1] In practice, unlabeled data includes all labeled data without labels.

and $\mathcal{L}_{Rot}$ is a self-supervised loss for unlabeled images and stands for rotation prediction from [GSK18] to provide an additional supervisory and regularizing signal.

Fully supervised loss for labeled data: We use cross-entropy loss with weak data augmentation regularization for labeled data:

$$\mathcal{L}_S = \frac{1}{B_s} \sum_{i=1}^{B_s} \ell_{CE}(\mathbf{p}_i, g(f(\alpha(\mathbf{x}_i)))) \tag{3.6}$$

where ℓ_{CE} is the cross-entropy loss, $\alpha(\mathbf{x}_i)$ is the extracted feature from a weakly augmented image $\mathbf{x}_i$, g is the same linear classifier as in equation 3.2, and $\mathbf{p}_i$ is the corresponding label for $\mathbf{x}_i$.

Self-supervised loss for unlabeled data: Rotation prediction [GSK18] (RotNet) is one of the most successful self-supervised learning methods, and has been shown to be complementary to SSL methods [ZOKB19, BCG+19, REH+20]. Here, we create four rotated images by 0°, 90°, 180°, and 270° for each unlabeled image $\mathbf{u}_i$ for $i \in (1, ..., \mu B)$. Then, classification loss is applied to train the model predicting the rotation as a four-class classification task:

$$\mathcal{L}_{Rot} = \frac{1}{4B_u} \sum_{i=1}^{B_u} \sum_{r \in \mathbb{R}} \ell_{CE}(r, h(\alpha(R(\mathbf{u}_i, r)))) \tag{3.7}$$

where $\mathbb{R}$ is $\{0°, 90°, 180°, 270°\}$ and r refers to one of the four rotations, h denotes a three-layer MLP with its hidden dimension the same as the input dimension. Using a predictor head is shown to be beneficial for such an auxiliary loss [CKNH20, CKS+20a]. Note that rotation prediction, though commonly used, might also have adverse effects. For example, numbers six and nine in most print fonts are centrosymmetric, rotating one upside down gives the other.

3.2.3 Implementation Details

Data augmentation: As mentioned above, CR-Match adopts two types of data augmentations: weak augmentation and strong augmentation from [SBL+20]. Specifically, the weak augmentation α corresponds to a standard random cropping and random mirroring with probability 0.5, and the strong augmentation $\mathcal{A}$ is a combination of RandAugment [CZSL20] and CutOut [DT17]. At each training step, we uniformly sample two operations for the strong augmentation from a collection of transformations and apply them with a randomly sampled magnitude from a predefined range. The complete table of transformation operations for the strong augmentation is provided in the supplementary material.

Other implementation details: For our results in Section 3.4 and Section 3.5, we minimize the cosine similarity in FeatDistLoss, and use a fully-connected layer for the projection layer z, which maps the feature from the original un-flattened 8192-dimension space into a 128-dimension space, the same dimension as the feature dimension for classification. The dimension of the original feature space and the patch size are fixed and depend on the architecture, which is chosen following the previous conventions [OOR+18, BCG+19, BCC+20, SBL+20]. In our case, $8192 = 8 \times 8 \times 128$, where the patch size is 8×8, and there are 128 feature maps. The predictor head h in rotation prediction loss consists of two fully-connected layers and a ReLU as non-linearity. We use the same $\lambda_u = \lambda_r = 1$ in all experiments since CR-Match shows good robustness within a range of loss weights in our preliminary experiments. We train our model for 512 epochs on CIFAR-10, CIFAR-100, and SVHN. On STL-10 and Mini-ImageNet, we train the model for 300 epochs. Other hyper-parameters are from [SBL+20] for the compatibility.

Specifically, the confidence thresholds τ for pseudo-label selection is 0.95. We use SGD with momentum 0.9 and cosine learning rate schedule from [SBL$^+$20] starting from 0.03, batch size B_s is 64 for labeled data, and B_u is $7 \times B_s$. The final performance is reported using an exponential moving average of model parameters as recommended by [TV17a]. As a common practice, we repeat each experiment with five different data splits and report the mean and the standard deviation of the error rate.

3.3 Related Work

SSL is a broad field aiming to exploit both labeled and unlabeled data. Consistency regularization is a powerful method for SSL [RBH$^+$15a, SJT16a, BAP14a]. The idea is that the model should output consistent predictions for perturbed versions of the same input. Many works explored different ways to generate such perturbations. For example, [TV17a] uses an exponential moving average of the trained model to produce another input; [SJT16a, LA17] use random max-pooling and Dropout [SHK$^+$14]; [XDH$^+$19, BCC$^+$20, SBL$^+$20, KMHK20] use advanced data augmentation; [BCG$^+$19, VLK$^+$19, BCC$^+$20] use MixUp regularization [ZCDLP18], which encourages convex behavior "between" examples; [GWL21] enforces label consistency with alpha-divergence. Another spectrum of popular approaches is pseudo-labeling [Scu65, Nes83, Lee13], where the model is trained with artificial labels. [AOA$^+$20a] trained the model with "soft" pseudo-labels from network predictions; [PXDL20] proposed a meta learning method that deploys a teacher model to adjust the pseudo-label alongside the training of the student; [SBL$^+$20, Lee13] learn from "hard" pseudo-labels and only retain a pseudo-label if the largest class probability is above a predefined threshold; [ZWH$^+$21] further refines the thresholding mechanism by adaptively adjusting thresholds for different classes according to the learning effect of each class. Furthermore, there are many excellent works around generative models [KMRW14, Ode16a, DGF16a] and graph-based methods [LZL$^+$18, LWHL19, BDLR06, Joa03]. We refer to [CSZ09, Zhu05a, ZG09a] for a more comprehensive introduction of SSL methods.

Noise injection plays a crucial role in consistency regularization [XDH$^+$19]. Thus advanced data augmentation, especially combined with weak data augmentation, introduces stronger noise to unlabeled data and brings substantial improvements [BCC$^+$20, SBL$^+$20]. [SBL$^+$20] proposes to integrate pseudo-labeling into the pipeline by computing pseudo-labels from weakly augmented images, and then uses the cross-entropy loss between the pseudo-labels and strongly augmented images. Besides the classifier level consistency, our model also introduces consistency on the feature level, which explicitly regularizes representation learning and shows improved generalization performance. Moreover, self-supervised learning is known to be beneficial in the context of SSL. In [HFW$^+$20, CKNH20, CKS$^+$20a, REH$^+$20], self-supervised pre-training is used to initialize SSL. However, these methods normally have several training phases, where many hyper-parameters are involved. We follow the trend of [ZOKB19, BCC$^+$20] to incorporate an auxiliary self-supervised loss alongside training. Specifically, we optimizes a rotation prediction loss [GSK18].

Another paradigm of SSL is to first perform self-supervised pre-training on unlabeled data and then fine-tune the pre-trained model with labeled data. In particular, contrastive learning based methods are gaining popularity and achieve good performance recently [CKS$^+$20a, HFW$^+$20, CMM$^+$20, GSA$^+$20, BPL22, CH21]. The goal of contrastive representation learning is to learn an embedding space in which different versions of the same image stay close to each other while features of different images are far apart. Different to this stream of works, our FeatDistLoss with equivariance pushes apart features from different augmentations of the same image while enforcing the same semantic label, which leads to both more expressive representation and more powerful classifier. Moreover, FeatDistLoss does not have the collapse problem [CH21] due to the availability of labeled data.

Equivariant representations are recently explored by capsule networks [SFH17, HSF18]. They replaced max-pooling layers with convolutional strides and dynamic routing to preserve more information about the input, allowing for preservation of part-whole relationships in the data. It has been shown, that the input can be reconstructed from the output capsule vectors. Another stream of work on group equivariant networks [CW16a, WC19, CW16b] explores various equivariant architectures that produce transform in a predictable linear manner under transformations of the input. Different from previous work, our work explores equivariant representations in the sense that differently augmented versions of the same image are represented by different points in the feature space despite the same semantic label. As we will show in section 3.4.3, information like object location or orientation is more predictable from our model when features are pushed apart from each other.

Imbalanced semi-supervised learning. While SSL has been extensively studied, the setting of class-imbalanced semi-supervised is rather under-explored. Most successful methods from standard SSL do not generalize well to this more realistic scenario without addressing the data imbalance explicitly. Hyun et al. [HJK20] proposed a suppressed consistency loss to suppress the loss on minority classes. Kim et al. [KHP+20b] proposed Distribution Aligning Refinery (DARP) to refine raw pseudo-labels via convex optimization. Wei et al. [WSM+21] found that the raw SSL methods usually have high recall and low precision for head classes while the reverse is true for the tail classes and further proposed a reverse sampling method for unlabeled data based on that. BiS [HKY+21] implements a novel sampler which is helpful for the encoder in the beginning but classifier in the end. DASO [OKK21a] refines pseudo-labels by two complementary classifiers. ABC [LSK21] introduces an auxiliary classifier which is trained in a balanced way to help the model while sharing the same backbone. As is shown in Section 3.5, we examine the effectiveness of our method on top of state-of-the-art imbalanced SSL frameworks and show improved results.

3.4 Experimental Results

Following protocols from previous work [BCG+19, SBL+20], we conduct experiments on several commonly used SSL image classification benchmarks to test the efficacy of CR-Match. We show our main results in Section 3.4.1, where we achieve state-of-the-art error rates across all settings on SVHN [NWC+11b], CIFAR-10 [KH+09a], CIFAR-100 [KH+09a], STL-10 [CNL11a], and mini-ImageNet [RL17]. In our ablation study in Section 3.4.2 we analyze the effect of FeatDistLoss and RotNet across different settings. Finally, in Section 3.4.3 we extensively analyse various design choices for our FeatDistLoss.

3.4.1 Main Results

In the following, each dataset subsection includes two paragraphs. The first provides technical details and the second discusses experimental results.

CIFAR-10, CIFAR-100, and SVHN. We follow prior work [SBL+20] and use 4, 25, and 100 labels per class on CIFAR-100 and SVHN without extra data. For CIFAR-10, we experiment with settings of 4, 25, and 400 labels per class. We create labeled data by random sampling, and the remaining images are regarded as unlabeled by discarding their labels. Following [BCG+19,

	CIFAR-10			CIFAR-100		
Per class labels	4 labels	25 labels	400 labels	4 labels	25 labels	100 labels
Mean Teacher [TV17a]	-	32.32±2.30*	9.19±0.19*	-	53.91±0.57*	35.83±0.24*
MixMatch [BCG^{+}19]	47.54±11.50*	11.08±0.87	6.24±0.06	67.61±1.32*	39.94±0.37*	25.88±0.30
UDA [XDH^{+}19]	29.05±5.93*	5.43±0.96	4.32±0.08*	59.28±0.88*	33.13±0.22*	24.50±0.25*
ReMixMatch [BCC^{+}20]	19.10±9.64*	6.27±0.34	5.14±0.04	44.28±2.06*	27.43±0.31*	23.03±0.56*
FixMatch (RA) [SBL^{+}20]	13.81±3.37	*5.07*±0.65	*4.26*±0.05	48.85±1.75	28.29±0.11	22.60±0.12
FixMatch (CTA) [SBL^{+}20]	11.39±3.35	*5.07*±0.33	4.31±0.15	49.95±3.01	28.64±0.24	23.18±0.11
FeatMatch [KMHK20]	-	6.00±0.41	4.64±0.11	-	-	-
FlexMatch [ZWH^{+}21]	**5.19**±0.05	5.33±0.12	4.47±0.09	45.91±1.76	28.11±0.20	23.04±0.28
CR-Match	10.70±2.91	**5.05**±0.12	**3.96**±0.16	*39.45*±1.69	*25.43*±0.14	**20.40**±0.08
CR-Match§	*5.52*±0.32	5.21±0.06	*4.26*±0.19	**35.72**±0.50	**24.61**±0.37	*20.91*±0.24

Table 3.1: Error rates on CIFAR-10, and CIFAR-100. A Wide ResNet-28-2 [ZK16a] is used for CIFAR-10 and a Wide ResNet-28-8 with 135 filters per layer [BCG^{+}19] is used for CIFAR-100. We use the same code base as [SBL^{+}20] (i.e., same network architecture and training protocol) to make the results directly comparable. The best number is in bold and the second best number is in italic. *Numbers are generated by [SBL^{+}20]. CR-Match§ refers to CR-Match combined with CPL [ZWH^{+}21] from FlexMatch.

	STL-10	SVHN		
Per class labels	100 labels	4 labels	25 labels	100 labels
Mean Teacher [TV17a]	21.34±2.39*	-	3.57±0.11*	3.42±0.07*
MixMatch [BCG^{+}19]	10.18±1.46	42.55±14.53*	3.78±0.26	3.27±0.31
UDA [XDH^{+}19]	7.66±0.56*	52.63±20.51*	2.72±0.40	*2.23*±0.07
ReMixMatch [BCC^{+}20]	6.18±1.24	*3.34*±0.20*	3.10±0.50	2.83±0.30
FixMatch (RA) [SBL^{+}20]	7.98±1.50	3.96±2.17	*2.48*±0.38	2.28±0.11
FixMatch (CTA) [SBL^{+}20]	*5.17*±0.63	7.65±7.65	2.64±0.64	2.36±0.19
FeatMatch [KMHK20]	-	-	3.34±0.19†	3.10±0.06†
FlexMatch [ZWH^{+}21]	6.15±0.25	20.81±5.26	17.32±2.07	12.90±2.68
CR-Match	**4.89**±0.17	**2.79**±0.93	**2.35**±0.29	**2.08**±0.07

Table 3.2: Error rates on Mini-ImageNet with 40 labels and 100 labels per class. All methods are evaluated on the same ResNet-18 architecture. *Numbers are generated by [SBL^{+}20]. †Numbers are produced without CutOut. The best number is in bold and the second best number is in italic.

	Mini-ImageNet	
Per class labels	40 labels	100 labels
Mean Teacher [TV17a]	72.51±0.22	57.55±1.11
Label Propagation [ITAC19a]	70.29±0.81	57.58±1.47
PLCB [AOA^{+}20a]	56.49±0.51	46.08±0.11
FeatMatch [KMHK20]	*39.05*±0.06	*34.79*±0.22
CR-Match	**34.87**±0.99	**32.58**±1.60

Table 3.3: Error rates on Mini-ImageNet with 40 labels and 100 labels per class. All methods are evaluated on the same ResNet-18 architecture. *Numbers are generated by [SBL^{+}20]. †Numbers are produced without CutOut. The best number is in bold and the second best number is in italic.

SBL$^+$20, BCC$^+$20], we use a Wide ResNet-28-2 [ZK16a] with 1.5M parameters on CIFAR-10 and SVHN, and a Wide ResNet-28-8 with 135 filters per layer (26M parameters) on CIFAR-100.

As shown in Table 3.1, 3.2, and 3.3, our method improves over previous methods across all settings, and defines a new state-of-the-art. Most importantly, we improve error rates in low data regimes by a large margin (e.g., with 4 labeled examples per class on CIFAR-100, we outperform FlexMatch and the second best method by 10.19% and 8.56% in absolute value respectively). Prior works [SBL$^+$20, BCG$^+$19, BCC$^+$20] have reported results using a larger network architecture on CIFAR-100 to obtain better performance. On the contrary, we additionally evaluate our method on the small network used in CIFAR-10 and find that our method is more than 17 times ($17 \approx 26/1.5$) parameter-efficient than FixMatch. We reach 46.05% error rate on CIFAR-100 with 4 labels per class using the small model, which is still slightly better than the result of FixMatch using a larger model.

STL-10. STL-10 contains 5,000 labeled images of size 96-by-96 from 10 classes and 100,000 unlabeled images. The dataset pre-defines ten folds of 1,000 labeled examples from the training data, and we evaluate our method on five of these ten folds as in [SBL$^+$20, BCC$^+$20]. Following [BCG$^+$19], we use the same Wide ResNet-37-2 model (comprising 5.9M parameters), and report error rates in Table 3.2.

Our method achieves state-of-the-art performance with 4.89% error rate. Note that FixMatch with error rate 5.17% used the more advanced CTAugment [BCC$^+$20], which learns augmentation policies alongside model training. When evaluated with the same data augmentation (RandAugment) as we use in CR-Match, our result surpasses FixMatch by 3.09% (3.09%=7.98%-4.89%), which indicates that CR-Match itself induces a strong regularization effect.

Mini-ImageNet. We follow [ITAC19a, AOA$^+$20a, KMHK20] to construct the mini-ImageNet training set. Specifically, 50,000 training examples and 10,000 test examples are randomly selected for a predefined list of 100 classes [RL17] from ILSVRC [DDS$^+$09]. Following [KMHK20], we use a ResNet-18 network [HZRS16a] as our model and experiment with settings of 40 labels per class and 100 labels per class.

As shown in Table 3.3, our method consistently improves over previous methods and achieves a new state-of-the-art in both the 40-label and 100-label settings. Especially in the 40-label case, CR-Match achieves an error rate of 34.87% which is 4.18% higher than the second best result. Note that our method is 2 times more data efficient than the second best method FeatMatch [KMHK20] (FeatMatch, using 100 labels per class, reaches a similar error rate as our method with 40 labeled examples per class).

ImageNet. To verify the effectiveness of our method on large scale datasets, we conduct experiments on ImageNet-1k. Following [ZWH$^+$21], we take ~10% (100,000) training images as the labeled set and construct unlabeled set using the rest of the images. The validation setting remains the same. We train a ResNet-50 [HZRS16a] with the same hyper-parameters from [ZWH$^+$21]. Note that FixMatch and FlexMatch use different protocols on ImageNet, and we follow the setup from FlexMatch therefore the numbers are directly comparable.

Table 3.4 shows the error rate comparison after running 2^{20} iterations. Our method outperforms the previous state-of-the-art by 1.04% absolute top-5 error rate, which demonstrates the efficacy of the proposed method at large scale dataset.

Method	Top-1	Top-5
FixMatch [SBL$^+$20]	43.66*	21.80*
FlexMatch [ZWH$^+$21]	41.85*	19.48*
CR-Match§	**40.69**	**18.44**

Table 3.4: Error rates on ImageNet after 2^{20} iterations. CR-Match§ refers to CR-Match combined with CPL [ZWH$^+$21] from FlexMatch. *Numbers are from [ZWH$^+$21].

RotNet	FeatDistLoss	MiniImageNet@40	CIFAR10@4	CIFAR100@4	SVHN@4
		35.13	11.86	46.22	2.42
	✓	34.14	**10.33**	43.48	2.34
✓		34.64	11.27	41.48	2.21
✓	✓	**33.82**	10.92	**39.22**	**2.09**

Table 3.5: Ablation studies across different settings. Error rates are reported for a single split.

3.4.2 Ablation Study

In this section, we analyze how FeatDistLoss and RotNet influence the performance across different settings, particularly when there are few labeled samples. We conduct experiments on a single split on CIFAR-10, CIFAR-100, and SVHN with 4 labeled examples per class, and on MiniImageNet with 40 labels per class. Specifically, we remove the $\mathcal{L}_{Dist}$ from equation 3.4 and train the model again using the same training scheme for each setting. We do not ablate $\mathcal{L}_{Pseudo}$ and $\mathcal{L}_S$ due to the fact that removing one of them leads to a divergence of training.

We report final test error rates in Table 3.5. We see that both RotNet and FeatDistLoss contribute to the final performance while their proportions can be different depending on the setting and dataset. For MiniImageNet, CIFAR-100 and SVHN, the combination of both outperforms the individual losses. For CIFAR-10, FeatDistLoss even outperforms the combination of both. This suggests that RotNet and FeatDistLoss are both important components for CR-Match to achieve the state-of-the-art performance. Note that RotNet can be replaced by other types of self-supervision as well. We opt RotNet due to its superior performance in our initial experiments. On CIFAR-100 with 4 labels per class, CRMatch with SimCLR achieves an error rate of 42.50% compared to that of 39.22% from CRMatch with RotNet. More details of the experiment are provided in the supplementary material.

Figure 3.4 shows a more detailed analysis of the training process on CIFAR-100 with 4 labels per class for CR-Match and CR-Match without FeatDistLoss. The confidence threshold in CR-Match filters out unconfident predictions during training. Therefore, at each training step only images with confidence scores above the threshold contribute to the loss. We observe that CR-Match improves pseudo-labels for the unlabeled data, as it achieves a lower error rate of all unlabeled images as well as contributing unlabeled images during the training while maintaining the percentage of contributing images.

The increasing of the pseudo-label error rate in Figure 3.4 middle is due to the increasing of the percentage of contributing pseudo-labels and the prediction confidence. At the beginning of the training, the contributing pseudo-labels are mostly correct as only a small number of samples are highly confident and, thus, selected. However, during the course of the training, the overall prediction confidence increases, resulting in more unlabeled data being used, which

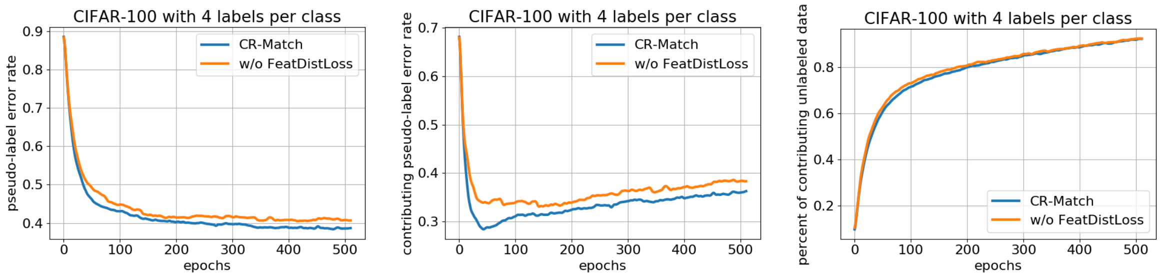


Figure 3.4: Ablation study of our best model on CIFAR-100 with 4 labels per class. **Left:** CR-Match has a lower pseudo-label error rate. **Middle:** If only the confident predictions are taken into account, CR-Match outperforms the other with an even larger margin in terms of pseudo-label error rate. **Right:** In spite of a better pseudo-label error rate on contributing unlabeled images, the percentage of contributing unlabeled images is maintained the same for CR-Match.

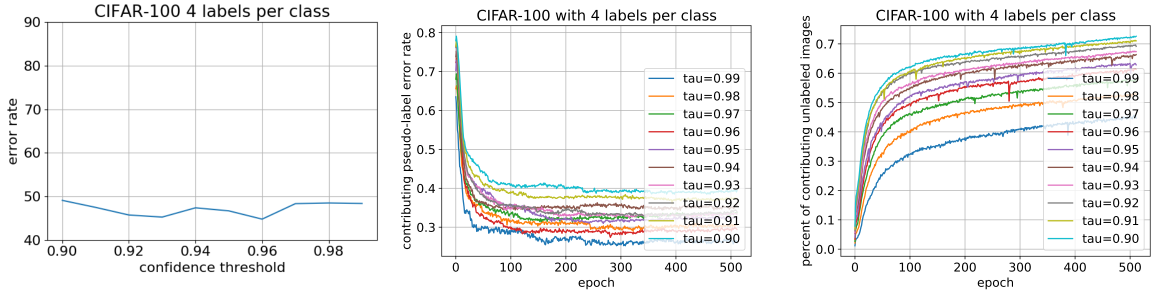


Figure 3.5: **Left:** Effect of different confidence thresholds on error rate. We run experiments on a single split of CIFAR-100 with 4 labels per class. The model is a Wide-ResNet-28-2. Our model shows good robustness against small changes in the confidence threshold. **Middle:** Effect of different confidence thresholds on pseudo-label error rate during the training. **Right:** Effect of different confidence thresholds on the number of unlabeled training samples.

introduces more errors in pseudo-labels.

Effect of different confidence thresholds. For the main results in Section 3.4.1, we use a confidence threshold of 0.95 following [SBL$^+$20]. We now study the model robustness against different confidence thresholds. Experiments are conducted on a single split with 4 labeled examples from CIFAR-100 on a Wide ResNet-28-2. Figure 3.5 shows the error rate of CR-Match when using a confidence threshold from 0.90 to 0.99. In general, the thresholding mechanism provides the model a relatively smooth transition between learning from labeled data and learning from unlabeled data. A low percentage of the contributing unlabeled data at the beginning of the training can alleviate the potential error introduced by the low-quality pseudo-labels. This suggests that the quality of pseudo-labels is more important than the quantity for reaching a high accuracy at the early stage. As the model learns from the labeled data, the error rate of the pseudo-label decreases, and the model becomes more confident about its predictions. Then, the number of unlabeled data that contribute to the final loss gradually increases, which allows the model to continue learning from unlabeled data. Figure 3.5 left also implies that our model is quite robust against small changes in the confidence threshold.

3.4.3 Influence of Feature Distance Loss

In this section, we analyze different design choices for FeatDistLoss to provide additional insights of how it helps generalization. We focus on a single split with 4 labeled examples from CIFAR-100 and report results for a Wide ResNet-28-2 [ZK16a]. For fair comparison, the same 4 random labeled examples for each class are used across all experiments in this section.

Different distance metrics for FeatDistLoss. Here we discuss the effect of different metric functions d for FeatDistLoss. Specifically, we compare two groups of functions in Table 3.6: metrics that increase the distance between features, including cosine similarity, negative JS divergence, and L2 similarity (i.e. normalized negative L2 distance); metrics that decrease the distance between features, including cosine distance, JS divergence, and L2 distance. We find that both increasing and decreasing distance between features of different augmentations give reasonable performance. However, increasing the distance always performs better than the counterpart (e.g., cosine similarity is better than cosine distance). We conjecture that decreasing the feature distance corresponds to an increase of the invariance to data augmentation and leads to ignorance of information like rotation or translation of the object. In contrast, increasing the feature distance while still imposing the same label makes the representation equivariant to these augmentations, resulting in more descriptive and expressive representation with respect to augmentation. Moreover, a classifier has to cover a broader space in the feature space to recognize rather dissimilar images from the same class, which leads to improved generalization. In summary, we found that both increasing and decreasing feature distance improve over the model which only applies consistency on the classifier level, whereas increasing distances shows better performance by making representations more equivariant.

Please refer to the supplementary material for experiments of combining both invariant and equivariant loss in FeatDistLoss.

Metric		Error rate
Impose equivariance	cosine similarity	**45.52**
	L_2 similarity	46.22
	negative JS div.	46.46
Impose invariance	cosine distance	46.98
	L_2 distance	48.74
	JS divergence	47.48
CR-Match w/o FeatDistLoss		48.89

Table 3.6: Effect of different distance functions for FeatDistLoss. The same split on CIFAR-100 with 4 labels per class and a Wide ResNet-28-2 is used for all experiments. Metrics that pull features together performs worse than those that push features apart. The error rate of CR-Match without FeatDistLoss is shown at the bottom.

Invariance and equivariance. Here we provide an additional analysis to demonstrate that increasing the feature distance provides equivariant features while the other provides invariant features. Based on the intuition that specific transformations of the input image should be more predictable from equivariant representations, we quantify the equivariance by how accurate a linear classifier can distinguish between features from augmented and original images. Specifically, we compare two models from Table 3.6: the model trained with cosine similarity denoted as *CR-Equiv* and the model trained with cosine distance denoted as *CR-Inv*.

Transformations	Feature extractor	
	CR-Equiv	CR-Inv
Translation	33.22±0.28	36.80±0.30
Scaling	11.09±0.66	14.87±0.40
Rotation	15.05±0.33	21.92±0.32
ColorJittering	31.04±0.50	35.99±0.27

Table 3.7: Error rates of binary classification (whether a specific augmentation is applied) on the features from CR-Equiv (increasing the cosine distance) and CR-Inv (decreasing the cosine distance). We evaluate translation, scaling, rotation, and color jittering. Lower error rate indicates more equivariant features. Results are averaged over 10 runs.

We train a linear SVM to predict whether a certain transformation is applied for the input image. 1000 test images from CIFAR-100 are used for training and the rest (9000) for validation. The binary classifier is trained by an SGD optimizer with an initial learning rate of 0.001 for 50 epochs, and the feature extractor is fixed during training. We evaluate translation, scaling, rotation, and color jittering in Table 3.7. All augmentations are from the standard PyTorch library. The SVM has a better error rate across all augmentations when trained on CR-Equiv features, which means information like object location or orientation is more predictable from CR-Equiv features, suggesting that CR-Equiv produces more equivariant features than CR-Inv. Furthermore, if the SVM is trained to classify strongly and weakly augmented image features, CR-Equiv achieves a 0.27% test error while CR-Inv is 46.18%.

Regularization on the classifier level. As we described in Section 3.2, FeatDistLoss contains two levels of regularization: On the feature level, representations are encouraged to become more equivariant. On the classifier level, the same class label is imposed on different versions of the same image via pseudo-labeling. Here we provide more insights into the regularization on the classifier level in Table 3.8. Specifically, we conduct experiments on replacing or complementing the CE loss with Jensen-Shannon divergence. First, we can see that removing the classifier loss and using only the equivariant loss on the feature level leads to a significant drop on performance (from 45.52% to 91.53%). This is because $\mathcal{L}_{Dist}$ alone will just make the model aware of the difference between augmentations but does not help the classifier to distinguish between classes of unlabeled data, making the classifier unable to benefit from the usage of unlabeled data. Thus, the performance is on par with the model trained on labeled data only (91.28% error rate) Second, complementing the cross-entropy loss on the classifier level with Jensen-Shannon divergence, improves the performance (45.01%) while replacing it leads to inferior performance (76.83%).

Different data augmentations for FeatDistLoss. In our main results in Section 3.4.1, FeatDistLoss is computed between features generated by weak augmentation and strong augmentation. Here we investigate the impact of FeatDistLoss with respect to different types of data augmentations. Specifically, we evaluate the error rate of CR-Inv and CR-Equiv under three augmentation strategies: weak-weak pair indicates that FeatDistLoss uses two weakly augmented images, weak-strong pair indicates that FeatDistLoss uses a weak augmentation and a strong augmentation, and strong-strong pair indicates that FeatDistLoss uses two strongly augmented images.

As shown in Table 3.9, using either CR-Inv or CR-Equiv using weak-strong pairs conistently outperforms the other augmentation settings (weak-weak and strong-strong). Additionally, CR-Equiv consistently achieves better generalization performance across all three settings. In

Classifier level	Feature level	Error rate
None	None	91.28
None	Equiv.	91.53
CE	Equiv.	**45.52**
JSD	Equiv.	76.83
CE + JSD	Equiv.	**45.01**

Table 3.8: Effect of different regularization techniques on the classifier level. CE denotes cross-entropy loss. JSD denotes Jensen-Shannon divergence. Equiv. denotes the equivariance version of $\mathcal{L}_{Dist}$. Note that the chance level is 99%. None + None represents the model trained with labeled data only. The same split on CIFAR-100 with 4 labels per class and a Wide ResNet-28-2 is used for all experiments.

Error rate	CR-Inv	CR-Equiv
Weak-Weak	48.88	48.51
Weak-Strong	**46.98**	**45.52**
Strong-Strong	48.57	48.05

Table 3.9: Effect of combinations of weak and strong augmentation in FeatDistLoss on a Wide ResNet-28-2 for CR-Inv and CR-Equiv.

particular, in the case advocated in this chapter, namely using weak-strong pairs, CR-Equiv outperforms CR-Inv by 1.46%. Even in the other two settings, CR-Equiv leads to improved performance even though only by a small margin. This suggests that, on the one hand, that it is important to use different types of augmentations for our FeatDistLoss. And on the other hand, maximizing distances between images that are inherently different while still imposing the same class label makes the model more robust against changes in the feature space and thus gives better generalization performance.

Linear projection and confidence threshold in FeatDistLoss. As mentioned in Section 3.2, we apply $\mathcal{L}_{Dist}$ at (a) in Figure 3.3 with a linear layer mapping the feature from the encoder to a low-dimensional space before computing the loss, to alleviate the curse of dimensionality. Also, the loss only takes effect when the model's prediction has a confidence score above a predefined threshold τ. Here we study the effect of other design choices in Table 3.10. While features after the global average pooling (i.e. (b)) gives a better result than the ones directly from the feature extractor, (b) performs worse than (a) when additional projection heads are added. Thus, we use features from the feature extractor in CR-Match. The error rate increases

Features taken from Fig. 3.3 at	feature	feature + linear	feature + MLP
(a)	48.37	**45.52**	47.52
(b)	47.37	46.10	47.15

Table 3.10: Effect of the projection head z, and the place to apply $\mathcal{L}_{Dist}$. (a) denotes un-flattened features taken from the feature extractor directly. (b) denotes features after the global average pooling. MLP has 2 FC layers and a ReLU. Removing the linear projection head harms the test error, and a non-linear projection head does not improve the performance further.

from 45.52% to 48.37% and 47.52% when removing the linear layer and replacing the linear layer by a MLP (two fully-connected layers and a ReLU activation function), respectively. This suggests that a lower dimensional space serves better for comparing distances, but a non-linear mapping does not give further improvement. Moreover, when we apply FeatDistLoss for all pairs of input images by removing the confidence threshold, the test error increases from 45.52% to 46.94%, which suggests that regularization should be only performed on features that are actually used to update the model parameters, and ignoring those that are also ignored by the model.

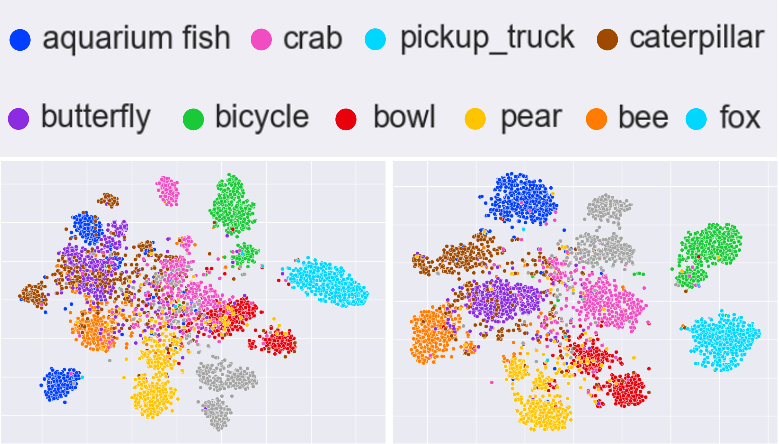


Figure 3.6: We plot t-SNE of input image features extracted by a CR-Match model trained without FeatDistLoss (left) and a CR-Match model with it (right). The better separation from CR-Match suggests that FeatDistLoss improves decision boundaries.

FeatDistLoss improves decision boundaries.

As suggested by Figure 3.2, models trained with FeatDistLoss tend to have improved decision boundaries. Here we take two models from section 3.4.2, CR-Match (39.22% error rate) and CR-Match without FeatDistLoss (41.48% error rate), and plot t-SNE plots of features extracted from unlabeled images. As shown in Figure 3.6, CR-Match with FeatDistLoss produces better separation between classes. For example, CR-Match forms two clearer clusters for caterpillar and butterfly, while CR-Match without FeatDistLoss mostly mixes them up. Another example is that the overlap between crab, bowl, and pear is much less for CR-Match compared to CR-Match without FeatDistLoss. Moreover, the improved decision boundaries also lead to better per-class error rate. The standard deviation of per-class error rates for CR-Match is 4.34% lower than that from CR-Match without FeatDistLoss (30.83% v.s. 26.49%).

Additional analysis on FeatDistLoss.

To further verify the importance of FeatDistLoss, we show in Figure 3.7 the contribution of FeatDistLoss compared to other losses. The model is CR-Equiv. trained on CIFAR-100 with 4 labels per class. We can see that during the training, the two components of FeatDistLoss, $\mathcal{L}_{Dist}$ and $\mathcal{L}_{PseudoLabel}$, account for a large portion of the overall loss, thus, the gradient. Note that $\mathcal{L}_{Dist}$ is the negative cosine distance, thus, ranging from 1 to -1.

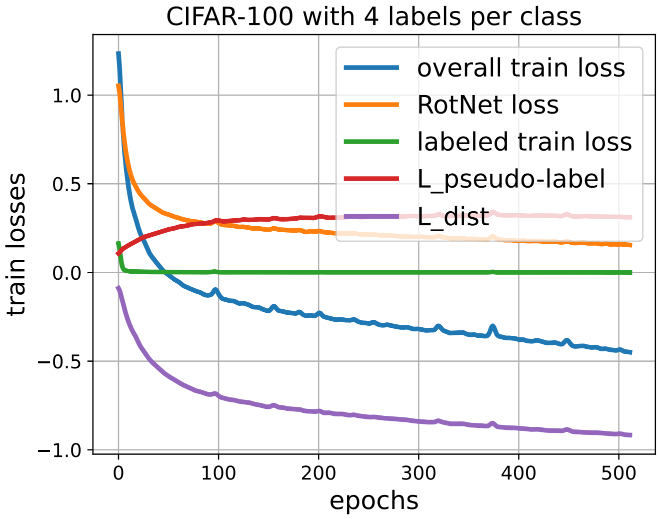


Figure 3.7: The amount of the contribution of the regularization term in the loss. The model is CR-Equiv. trained on CIFAR-100 with 4 labels per class.

3.5 Experiments on Imbalanced SSL

In this section, we go beyond the standard setting and evaluate the efficacy of our method under imbalanced SSL settings where both labeled and unlabeled data follow class imbalanced distributions. We first present the problem setup of imbalanced SSL. Then, we introduce the construction of the datasets before showing the final evaluation results.

Problem setup and notations. For a K-class classification problem, there is a labeled set $\mathcal{X} = \{(\mathbf{x}_n, y_n) : n \in (1, ..., N)\}$ and an unlabeled set $\mathcal{U} = \{\mathbf{u}_m : m \in (1, ..., M)\}$, where $\mathbf{x}_n, \mathbf{u}_m \in \mathbb{R}^d$ are training examples and $y_n \in \{1, ..., K\}$ are class labels for labeled examples. N_k and M_k denote the numbers of labeled and unlabeled examples in class k, respectively, i.e., $\sum_{k=1}^{K} N_k = N$ and $\sum_{k=1}^{K} M_k = M$. Without loss of generality, we assume the classes are sorted by the number of training samples in descending order, i.e., $N_1 \geq N_2 \geq ... \geq N_k$. The goal is to train a classifier $f : \mathbb{R}^d \to \{1, ..., K\}$ on $\mathcal{X} \cup \mathcal{U}$ that generalizes well on a class-balanced test set.

Datasets. We consider three common datasets in the field to evaluate the efficacy of CRMatch for imbalanced SSL: CIFAR10-LT [KH+09a], CIFAR100-LT [KH+09a], and Semi-Aves [SM21a].

For CIFAR-10-LT and CIFAR100-LT, we follow the convention [KHP+20b, WSM+21] and randomly select some training images for each class determined by a pre-defined imbalance ratio γ as the labeled and the unlabeled set. Specifically, we set $N_k = N_1 \cdot \gamma^{-\frac{k-1}{K-1}}$ for labeled data and $M_k = M_1 \cdot \gamma^{-\frac{k-1}{K-1}}$ for unlabeled data. We use $N_1 = 1500$; $M_1 = 3000$ for CIFAR-10 and $N_1 = 150$; $M_1 = 300$ for CIFAR-100, respectively. Following [KHP+20b, WSM+21], we report results with imbalance ratio $\gamma = 50$, 100 and 150 for CIFAR10-LT and $\gamma = 20$, 50 and 100 for CIFAR100-LT. Therefore, the number of labeled samples for the least class is 10 and 1 for CIFAR-10 with $\gamma = 150$ and CIFAR-100 with $\gamma = 100$, respectively.

Semi-Aves is a subset of bird species from the Aves kingdom of the iNaturalist 2018 dataset. There are 200 in-class and 800 out-of-class categories. The dataset consists of a labeled set L_{in} with 3,959 labeled images, an in-class unlabeled set U_{in} with 26,640 images, an out-of-class unlabeled set U_{out} with 122,208 images, a validation set L_{val} of 2,000 images, and 8,000 test images. The training data in L_{in}, U_{in}, and U_{out} has imbalanced distributions, specifically L_{in}

has 5 to 43 images and U_{in} has 16 to 229 images per class. The validation data and test data have a uniform distribution with 40 and 10 images per class, repectively. In our experiments, we use L_{in} or $L_{in} \cup L_{val}$ as the labeled set and U_{in} as the unlabeled set. We do not use unlabeled images from U_{out} since out-of-class unlabeled images are found empirically harmful to the final performance [OOR+18] and making good use of out-of-class unlabeled images is out of the scope of this chapter. More details on the class distribution can be found in [SM21a].

Class index		1	2	3	4	5	6	7	8	9	10	Avg.
Recall	FixMatch + CReST+	**98.6**	99.3	85.8	77.4	84.4	63.4	77.2	55.5	37.4	34.6	71.3
	CR-Match + CReST+	**98.6**	**99.6**	**88.8**	**82.5**	**86.7**	**67.8**	**78.7**	**57.0**	**42.7**	**40.6**	**74.3**
Precision	FixMatch + CReST+	53.3	61.4	71.4	57.9	77.0	82.4	93.0	**97.1**	97.6	**98.0**	78.9
	CR-Match + CReST+	**56.1**	**62.9**	**75.1**	**63.7**	**78.6**	**83.3**	**94.5**	**97.1**	**98.1**	97.1	**80.7**

Table 3.11: Class-wise precision and recall (%) on the balanced test set of CIFAR-10-LT. Models are trained with imbalance ratio $\gamma = 150$.

Implementation details. Due to the performance superiority of $\mathcal{L}_{U-equiv}$ over $\mathcal{L}_{U-inv}$, we use CR-Equiv throughout this section. For all experiments in this section, we use the same hyper-parameters and design choices from the CIFAR experiments in Section 3.4.1. We deploy FixMatch [SBL+20] as the base SSL method due to its superiority under the standard SSL settings. A Wide ResNet-28-2 [ZK16a] is used as the backbone as recommended by [OOR+18]. We base our implementation on the public codebases of each methods. Therefore, method-specific hyper-parameters follow the same as in their original papers [KHP+20b, WSM+21]. For example, all experiments on CIFAR-LT are trained with batch size 64 using Adam optimizer [KB15] with a constant learning rate of 0.002 without any decay. We train the models for 500 epochs, each of which has 500 steps, resulting in a total number of 2.5×10^5 training iterations. On Semi-Aves, we follow the hyper-parameters from [OKK21a]. For example, the models are trained for 90 epochs with a batch size of 256, and the optimizer is SGD with a learning rate of 0.04. For all experiments, we report the average test accuracy of the last 20 epochs following [OOR+18].

Results on CIFAR-10 and CIFAR-100. Table 3.12 and Table 3.13 compare our method with various SSL algorithms and long-tailed recognition algorithms on CIFAR-10-LT and CIFAR-100-LT with various imbalance ratios γ. Adding our method shows improved performance in most of settings. Our method combining with CoSSL [FDKS22] achieves the best or comparable performance across all settings. In particular, CRMatch + CoSSL outperforms others at large imbalance ratios (82.29% v.s. the second best 81.28% on CIFAR-10 at imbalance ratio $\gamma = 150$), which indicates the superiority of our method in handling severe dataset imbalance.

To analyze how the improvement is obtained, we compare the class-wise precision and recall of CReST+ and CReST+ with our method in Table 3.11. Both models are trained with imbalance ratio $\gamma = 150$ on CIFAR-10-LT using the same data split. The class indices are sorted according to the number of samples in descending order, i.e., class 1 has the largest number of data. For CReST+, the head classes tend to have higher precision but lower recall while the tail classes have lower precision but higher recall. By adding our method, the recall on the tail classes can be significantly improved without sacrificing much precision, which leads to the overall better performance. Similarly, the precision of the head classes is improved while the recall remains at the same level.

Results on Semi-Aves. As Semi-Aves is naturally imbalanced ($\gamma \approx 9$ and 4 for L_{train} and $L_{in} \cup L_{val}$, respectively), we compare CRMatch with other methods using different numbers of

	CIFAR-10-LT		
	γ=50	γ=100	γ=150
vanilla	65.2±0.05*	58.8±0.13*	55.6±0.43*
Long-tailed recognition methods			
Re-sampling [Jap00]	64.3±0.48*	55.8±0.47*	52.2±0.05*
LDAM-DRW [CWG$^+$19]	68.9±0.07*	62.8±0.17*	57.9±0.20*
cRT [KXR$^+$20]	67.8±0.13*	63.2±0.45*	59.3±0.10*
SSL methods			
FixMatch [SBL$^+$20]	81.58±0.34	74.74±1.35	70.04±0.77
ReMixMatch [BCC$^+$20]	82.79±0.17	76.81±0.23	72.53±1.16
FlexMatch [ZWH$^+$21]	81.89±0.25	74.94±0.96	70.09±0.42
CR-Match	82.87±0.04	76.54±0.87	72.14±0.76
FixMatch + DARP [KHP$^+$20b]	82.46±0.30	76.51±0.50	71.88±1.02
ReMixMatch + DARP [KHP$^+$20b]	82.88±0.23	76.77±0.29	72.90±0.95
FlexMatch + DARP [KHP$^+$20b]	81.93±0.22	74.84±0.66	70.46±0.58
CR-Match + DARP	83.22±0.27	77.32±0.29	73.44±0.06
FixMatch + CReST+ [WSM$^+$21]	82.25±0.08	76.31±0.23	71.70±0.83
ReMixMatch + CReST+ [WSM$^+$21]	83.71±0.17	79.13±0.19	75.17±0.31
FlexMatch + CReST+ [WSM$^+$21]	82.75±0.25	77.23±0.35	72.21±0.11
CR-Match + CReST+	84.11±0.32	78.55±0.55	74.21±0.11
FixMatch + CoSSL [FDKS22]	86.63±0.24	83.10±0.48	80.15±0.59
ReMixMatch + CoSSL [FDKS22]	*87.55±0.06*	*84.15±0.65*	*81.28±0.95*
FlexMatch + CoSSL [FDKS22]	86.30±0.30	81.61±0.74	78.80±0.73
CR-Match + CoSSL [FDKS22]	**88.11±0.17**	**84.80±0.54**	**82.29±0.33**

Table 3.12: Classification accuracy (%) on CIFAR-10-LT using a Wide ResNet-28-2 under the uniform test distribution of three different class-imbalance ratios γ. The numbers are averaged over 5 different folds. We use the same code base as [KHP$^+$20b] for fair comparison following [OOR$^+$18]. Numbers with * are taken from the original papers. The best number is in bold and the second best number is in italic.

labeled data. We report the raw performance of backbone algorithms as well as the performance with CoSSL [FDKS22] considering its superior performance on CIFAR-10-LT and CIFAR-100-LT. From Table 3.14, we can see that CRMatch outperforms other backbone algorithms by a large margin in both settings. While CoSSL leads to improvement in all methods, CRMatch still achieves the best performance, which demonstrates the effectiveness of our method in realistic settings.

	CIFAR-100-LT		
	γ=20	γ=50	γ=100
FixMatch [SBL$^+$20]	49.58±0.90	42.10±0.38	37.46±0.48
ReMixMatch [BCC$^+$20]	51.46±0.51	44.37±0.62	39.29±0.59
FlexMatch [ZWH$^+$21]	51.00±0.75	42.86±0.42	37.20±0.51
CR-Match	52.03±0.42	44.37±0.57	39.32±0.31
FixMatch + DARP [KHP$^+$20b]	50.89±0.86	43.12±0.61	38.19±0.47
ReMixMatch + DARP [KHP$^+$20b]	51.95±0.40	45.24±0.46	39.50±0.58
FlexMatch + DARP [KHP$^+$20b]	50.78±0.71	42.81±0.36	36.99±0.66
CR-Match + DARP	49.33±0.32	44.13±0.38	39.18±0.80
FixMatch + CReST+ [WSM$^+$21]	51.87±0.11	45.25±0.06	40.41±0.35
ReMixMatch + CReST+ [WSM$^+$21]	51.22±0.38	45.91±0.33	41.24±0.79
FlexMatch + CReST+ [WSM$^+$21]	51.16±0.63	43.12±0.57	38.09±0.58
CR-Match + CReST+	53.77±0.36	46.44±0.58	40.94±0.43
FixMatch + CoSSL [FDKS22]	53.99±0.87	47.78±0.53	42.87±0.61
ReMixMatch + CoSSL [FDKS22]	**55.92±0.69**	**49.10±0.59**	44.10±0.68
FlexMatch + CoSSL [FDKS22]	53.46±0.79	46.83±0.80	41.42±0.58
CR-Match + CoSSL [FDKS22]	*55.34±0.43*	*48.83±0.87*	**44.21±0.61**

Table 3.13: Classification accuracy (%) on CIFAR-100-LT under the uniform test distribution of three different class-imbalance ratios γ. The numbers are averaged over 5 different folds. We reproduce all numbers using the same codebase from [KHP$^+$20b] for a fair comparison. The best number is in bold and the second best number is in italic.

	Semi-Aves	
	$\mathcal{X} = L_{in} \cup L_{val}$	$\mathcal{X} = L_{in}$
FixMatch [SBL$^+$20]	53.15	42.46
ReMixMatch [BCC$^+$20]	51.28	40.10
FlexMatch [ZWH$^+$21]	52.78	43.50
CRMatch	*54.53*	44.42
FixMatch + CoSSL [FDKS22]	54.15	*44.58*
ReMixMatch + CoSSL [FDKS22]	54.13	43.97
FlexMatch + CoSSL [FDKS22]	53.98	44.09
CRMatch + CoSSL [FDKS22]	**54.90**	**45.81**

Table 3.14: Classification accuracy (%) on Semi-Aves under the uniform test distribution. L_{train} and $L_{in} \cup L_{val}$ have imbalance ratio $\gamma \approx 9$ and $\gamma \approx 4$, respectively. The best number is in bold and the second best number is in italic.

3.6 Conclusion

The idea of consistency regularization gives rise to many successful works for SSL [BAP14a, LA17, SJT16a, SBL$^+$20, XDH$^+$19, KMHK20]. While making the model invariant against input perturbations induced by data augmentation gives improved performance, the scheme tends to be suboptimal when augmentations of different intensities are used. In this work, we propose a simple yet effective improvement, called FeatDistLoss. It introduces consistency regularization on both the classifier level, where the same class label is imposed for versions of the same image, and the feature level, where distances between features from augmentations of different intensities is increased. By encouraging the representation to distinguish between weakly and strongly augmented images, FeatDistLoss encourages more equivariant representations, leading to improved classification boundaries, and a more robust model.

Through extensive experiments we show the superiority of our training framework, and define a new state-of-the-art on both standard and imbalanced semi-supervised learning benchmarks. Particularly, our method outperforms previous methods in low data regimes by significant margins, e.g., on CIFAR-100 with 4 annotated examples per class, our error rate (39.45%) is 4.83% better than the second best (44.28%). In future work, we are interested in integrating more prior knowledge and stronger regularization into SSL to further push the performance in low data regimes.

4 Self-adaptive Thresholding for Semi-supervised Learning

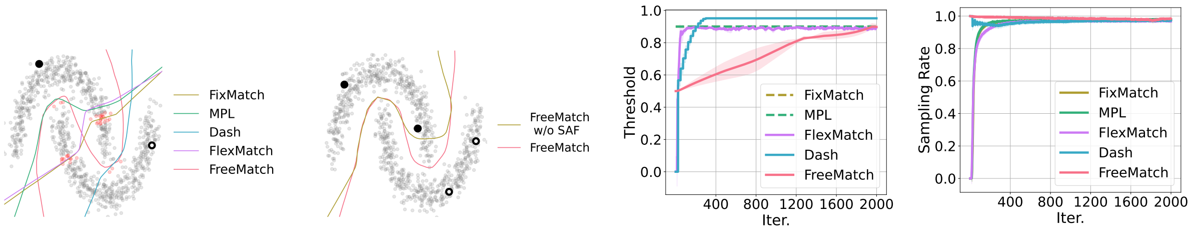
Contents

In this chapter, we explore pseudo-labeling, another widely adopted methodology in semi-supervised learning (SSL). While confidence threshold-based pseudo-labeling has significantly advanced SSL, we argue that existing methods might fail to utilize the unlabeled data more effectively since they either use a pre-defined / fixed threshold or an ad-hoc threshold adjusting scheme. We first analyze a motivating example to obtain intuitions on the relationship between the desirable threshold and model's learning status. Based on the analysis, we hence propose *FreeMatch* to adjust the confidence threshold in a self-adaptive manner according to the model's learning status. We further introduce a self-adaptive class fairness regularization penalty to encourage the model for diverse predictions during the early training stage. Extensive experiments indicate the superiority of FreeMatch especially when the labeled data are extremely rare. FreeMatch achieves **5.78**%, **13.59**%, and **1.28**% error rate reduction over the latest state-of-the-art method FlexMatch on CIFAR-10 with 1 label per class, STL-10 with 4 labels per class, and ImageNet with 100 labels per class, respectively. Moreover, FreeMatch can also boost the performance of imbalanced SSL.

This chapter is based on [WCH^{+}23]. As one of the co-authors, Yue Fan was involved in the weekly and more detailed discussions and contributed to the writing of the paper and the imbalanced SSL experiments.

4.1 Introduction

The superior performance of deep learning heavily relies on supervised training with sufficient labeled data [HZRS16b, VSP^{+}17, DXX18]. However, it remains laborious and expensive to obtain massive labeled data. To alleviate such reliance, semi-supervised learning (SSL) [Zhu05b, ZG09b, SBL^{+}20, RHS05a, GTM^{+}16, KDGBA19, DYY^{+}17] is developed to improve the model's generalization performance by exploiting a large volume of unlabeled data. Pseudo labeling [Lee13, XLHL20b, McL75, RDRS20] and consistency regularization [BAP14b, ST17, SJT16b] are



(a) Decision boundary (b) Self-adaptive fairness (c) Confi. threshold (d) Sampling rate

Figure 4.1: Demonstration of how FreeMatch works on the "two-moon" dataset. (a) Decision boundary of FreeMatch and other SSL methods. (b) Decision boundary improvement of self-adaptive fairness (SAF) on two labeled samples per class. (c) Class-average confidence threshold. (d) Class-average sampling rate of FreeMatch during training.

two popular paradigms designed for modern SSL. Recently, their combinations have shown promising results [XDH+20b, SBL+20, PDXL21, XSY+21, ZWH+21]. The key idea is that the model should produce similar predictions or the same pseudo labels for the same unlabeled data under different perturbations following the smoothness and low-density assumptions in SSL [CSZ06].

A potential limitation of these threshold-based methods is that they either need a *fixed threshold* [XDH+20b, SBL+20, ZWH+21, GL22] or an *ad-hoc threshold adjusting scheme* [XSY+21] to compute the loss with only confident unlabeled samples. Specifically, UDA [XDH+20b] and FixMatch [SBL+20] retain a fixed high threshold to ensure the quality of pseudo labels. However, a fixed high threshold (0.95) could lead to low data utilization in the early training stages and ignore the different learning difficulties of different classes. Dash [XSY+21] and AdaMatch [BRS+22] propose to gradually grow the fixed *global* (dataset-specific) threshold as the training progresses. Although the utilization of unlabeled data is improved, their ad-hoc threshold adjusting scheme is arbitrarily controlled by hyper-parameters and thus disconnected from model's learning process. FlexMatch [ZWH+21] demonstrates that different classes should have different *local* (class-specific) thresholds. While the local thresholds take into account the learning difficulties of different classes, they are still mapped from a *pre-defined fixed* global threshold. Adsh [GL22] obtains adaptive thresholds from a pre-defined threshold for imbalanced Semi-supervised Learning by optimizing the the number of pseudo labels for each class. In a nutshell, these methods might be incapable or insufficient in terms of adjusting thresholds according to model's learning progress, thus impeding the training process especially when labeled data is too scarce to provide adequate supervision.

For example, as shown in Fig. 4.1 (a), on the "two-moon" dataset with only 1 labeled sample for each class, the decision boundaries obtained by previous methods fail in the low-density assumption. Then, two questions naturally arise: *1) Is it necessary to determine the threshold based on the model learning status?* and *2) How to adaptively adjust the threshold for best training efficiency?*

In this chapter, we first leverage a motivating example to demonstrate that different datasets and classes should determine their global (dataset-specific) and local (class-specific) thresholds based on the model's learning status. Intuitively, we need a low global threshold to utilize more unlabeled data and speed up convergence at early training stages. As the prediction confidence increases, a higher global threshold is necessary to filter out wrong pseudo labels to alleviate the confirmation bias [AOA+20b]. Besides, a local threshold should be defined on each class based on the model's confidence about its predictions. The "two-moon" example in Fig. 4.1 (a) shows that the decision boundary is more reasonable when adjusting the thresholds based on the model's learning status.

We then propose *FreeMatch* to adjust the thresholds in a *self-adaptive* manner according to learning status of each class [GPSW17]. Specifically, FreeMatch uses the self-adaptive thresholding (SAT) technique to estimate both the global (dataset-specific) and local thresholds (class-specific) via the exponential moving average (EMA) of the unlabeled data confidence. To handle barely supervised settings [SBL$^+$20] more effectively, we further propose a class fairness objective to encourage the model to produce fair (i.e., diverse) predictions among all classes (as shown in Fig. 4.1 (b)). The overall training objective of FreeMatch maximizes the mutual information between model's input and output [JB91], producing confident and diverse predictions on unlabeled data. Benchmark results validate its effectiveness. To conclude, our contributions are:

- Using a motivating example, we discuss *why* thresholds should reflect the model's learning status and provide some intuitions for designing a threshold-adjusting scheme.

- We propose a novel approach, FreeMatch, which consists of Self-Adaptive Thresholding (SAT) and Self-Adaptive class Fairness regularization (SAF). SAT is a threshold-adjusting scheme that is *free* of setting thresholds manually and SAF encourages diverse predictions.

- Extensive results demonstrate the superior performance of FreeMatch on various SSL benchmarks, especially when the number of labels is very limited (e.g, an error reduction of **5.78**% on CIFAR-10 with 1 labeled sample per class).

4.2 A Motivating Example

In this section, we introduce a binary classification example to motivate our threshold-adjusting scheme. Despite the simplification of the actual model and training process, the analysis leads to some interesting implications and provides insight into how the thresholds should be set.

We aim to demonstrate the necessity of the self-adaptability and increased granularity in confidence thresholding for SSL. Inspired by [YX20], we consider a binary classification problem where the true distribution is an even mixture of two Gaussians (i.e., the label Y is equally likely to be positive ($+1$) or negative (-1)). The input X has the following conditional distribution:

$$X \mid Y = -1 \sim \mathcal{N}(\mu_1, \sigma_1^2), X \mid Y = +1 \sim \mathcal{N}(\mu_2, \sigma_2^2). \tag{4.1}$$

We assume $\mu_2 > \mu_1$ without loss of generality. Suppose that our classifier outputs confidence score $s(x) = 1/[1 + \exp(-\beta(x - \frac{\mu_1+\mu_2}{2}))]$, where β is a positive parameter that reflects the model learning status and it is expected to gradually grow during training as the model becomes more confident. Note that $\frac{\mu_1+\mu_2}{2}$ is in fact the Bayes' optimal linear decision boundary. We consider the scenario where a fixed threshold $\tau \in (\frac{1}{2}, 1)$ is used to generate pseudo labels. A sample x is assigned pseudo label $+1$ if $s(x) > \tau$ and -1 if $s(x) < 1 - \tau$. The pseudo label is 0 (masked) if $1 - \tau \leq s(x) \leq \tau$.

We then derive the following theorem to show the necessity of self-adaptive threshold:

Theorem 4.2.1. *For a binary classification problem as mentioned above, the pseudo label Y_p has the* *following probability distribution:*

$$\begin{aligned}
P(Y_p = 1) &= \frac{1}{2}\Phi(\frac{\frac{\mu_2-\mu_1}{2} - \frac{1}{\beta}\log(\frac{\tau}{1-\tau})}{\sigma_2}) + \frac{1}{2}\Phi(\frac{\frac{\mu_1-\mu_2}{2} - \frac{1}{\beta}\log(\frac{\tau}{1-\tau})}{\sigma_1}),\\
P(Y_p = -1) &= \frac{1}{2}\Phi(\frac{\frac{\mu_2-\mu_1}{2} - \frac{1}{\beta}\log(\frac{\tau}{1-\tau})}{\sigma_1}) + \frac{1}{2}\Phi(\frac{\frac{\mu_1-\mu_2}{2} - \frac{1}{\beta}\log(\frac{\tau}{1-\tau})}{\sigma_2}),\\
P(Y_p = 0) &= 1 - P(Y_p = 1) - P(Y_p = -1),
\end{aligned} \tag{4.2}$$

where Φ is the cumulative distribution function of a standard normal distribution. Moreover, $P(Y_p = 0)$ increases as $\mu_2 - \mu_1$ gets smaller.

Theorem 4.2.1 has the following implications or interpretations:

(i) Trivially, unlabeled data utilization (sampling rate) $1 - P(Y_p = 0)$ is directly controlled by threshold τ. As the confidence threshold τ gets larger, the unlabeled data utilization gets lower. At early training stages, adopting a high threshold may lead to low sampling rate and slow convergence since β is still small.

(ii) More interestingly, $P(Y_p = 1) \neq P(Y_p = -1)$ if $\sigma_1 \neq \sigma_2$. In fact, the larger τ is, the more imbalanced the pseudo labels are. This is potentially undesirable in the sense that we aim to tackle a balanced classification problem. Imbalanced pseudo labels may distort the decision boundary and lead to the so-called pseudo label bias. An easy remedy for this is to use class-specific thresholds τ_2 and $1 - \tau_1$ to assign pseudo labels.

(iii) The sampling rate $1 - P(Y_p = 0)$ decreases as $\mu_2 - \mu_1$ gets smaller. In other words, the more similar the two classes are, the more likely an unlabeled sample will be masked. As the two classes get more similar, there would be more samples mixed in feature space where the model is less confident about its predictions, thus a moderate threshold is needed to balance the sampling rate. Otherwise we may not have enough samples to train the model to classify the already difficult-to-classify classes.

The intuitions provided by Theorem 4.2.1 is that at the early training stages, τ should be low to encourage diverse pseudo labels, improve unlabeled data utilization and fasten convergence. However, as training continues and β grows larger, a consistently low threshold will lead to unacceptable confirmation bias. Ideally, the threshold τ should increase along with β to maintain a stable sampling rate throughout. Since different classes have different levels of intra-class diversity (different σ) and some classes are harder to classify than others ($\mu_2 - \mu_1$ being small), a fine-grained *class-specific* threshold is desirable to encourage fair assignment of pseudo labels to different classes. The challenge is how to design a threshold adjusting scheme that takes all implications into account, which is the main contribution of this chapter. We demonstrate our algorithm by plotting the average threshold trend and marginal pseudo label probability (i.e. sampling rate) during training in Fig. 4.1 (c) and 4.1 (d). To sum up, we should determine global (dataset-specific) and local (class-specific) thresholds by estimating the learning status via predictions from the model. Then, we detail FreeMatch.

4.3 Preliminaries

In SSL, the training data consists of labeled and unlabeled data. Let $\mathcal{D}_L = \{(x_b, y_b) : b \in [N_L]\}$ and $\mathcal{D}_U = \{u_b : b \in [N_U]\}$[1] be the labeled and unlabeled data, where N_L and N_U is their

[1] $[N] := \{1, 2, \ldots, N\}$.

number of samples, respectively. The supervised loss for labeled data is:

$$\mathcal{L}_s = \frac{1}{B}\sum_{b=1}^{B} \mathcal{H}(y_b, p_m(y|\omega(x_b))), \tag{4.3}$$

where B is the batch size, $\mathcal{H}(\cdot,\cdot)$ refers to cross-entropy loss, $\omega(\cdot)$ means the stochastic data augmentation function, and $p_m(\cdot)$ is the output probability from the model.

For unlabeled data, we focus on pseudo labeling using cross-entropy loss with confidence threshold for entropy minimization. We also adopt the "Weak and Strong Augmentation" strategy introduced by UDA [XDH$^+$20b]. Formally, the unsupervised training objective for unlabeled data is:

$$\mathcal{L}_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) > \tau) \cdot \mathcal{H}(\hat{q}_b, Q_b). \tag{4.4}$$

We use q_b and Q_b to denote abbreviation of $p_m(y|\omega(u_b))$ and $p_m(y|\Omega(u_b))$, respectively. $\hat{q}_b$ is the hard "one-hot" label converted from q_b, μ is the ratio of unlabeled data batch size to labeled data batch size, and $\mathbb{1}(\cdot > \tau)$ is the indicator function for confidence-based thresholding with τ being the threshold. The weak augmentation (i.e., random crop and flip) and strong augmentation (i.e., RandAugment [CZSL20]) is represented by $\omega(\cdot)$ and $\Omega(\cdot)$ respectively.

Besides, a fairness objective $\mathcal{L}_f$ is usually introduced to encourage the model to predict each class at the same frequency, which usually has the form of $\mathcal{L}_f = \mathbf{U} \log \mathbb{E}_{\mu B}[q_b]$ [AK10], where $\mathbf{U}$ is a uniform prior distribution. One may notice that using a uniform prior not only prevents the generalization to non-uniform data distribution but also ignores the fact that the underlying pseudo label distribution for a mini-batch may be imbalanced due to the sampling mechanism. The uniformity across a batch is essential for fair utilization of samples with per-class threshold, especially for early-training stages.

4.4 FreeMatch

4.4.1 Self-Adaptive Thresholding

We advocate that the key to determining thresholds for SSL is that thresholds should reflect the learning status. The learning effect can be estimated by the prediction confidence of a well-calibrated model [GPSW17]. Hence, we propose *self-adaptive thresholding* (SAT) that automatically defines and adaptively adjusts the confidence threshold for each class by leveraging the model predictions during training. SAT first estimates a global threshold as the EMA of the confidence from the model. Then, SAT modulates the global threshold via the local class-specific thresholds estimated as the EMA of the probability for each class from the model. When training starts, the threshold is low to accept more possibly correct samples into training. As the model becomes more confident, the threshold adaptively increases to filter out possibly incorrect samples to reduce the confirmation bias. Thus, as shown in Fig. 4.2, we define SAT as $\tau_t(c)$ indicating the threshold for class c at the t-th iteration.

Self-adaptive Global Threshold We design the global threshold based on the following two principles. First, the global threshold in SAT should be related to the model's confidence on unlabeled data, reflecting the overall learning status. Moreover, the global threshold should stably increase during training to ensure incorrect pseudo labels are discarded. We set the global threshold τ_t as average confidence from the model on unlabeled data, where t represents

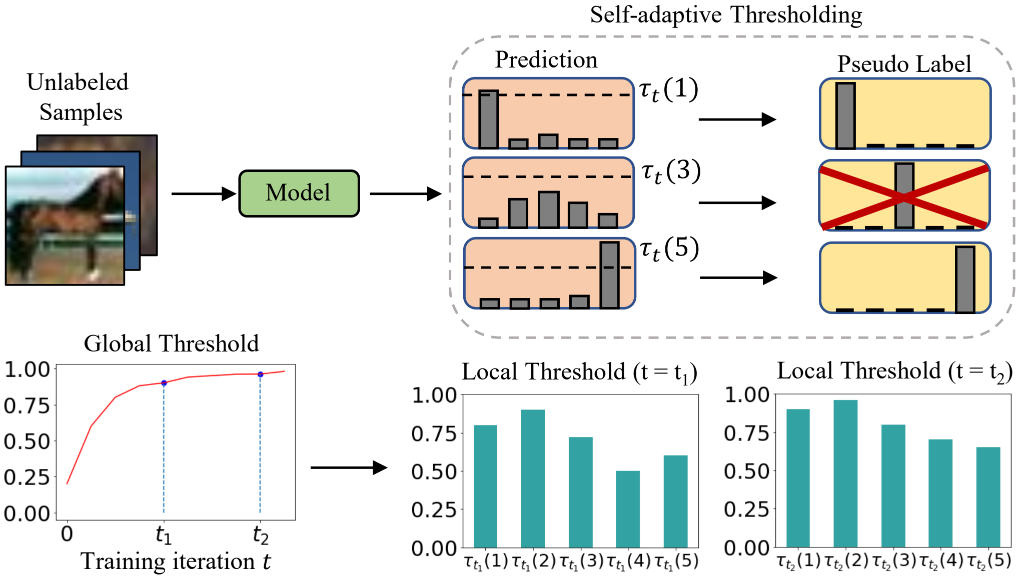

Self-adaptive Thresholding
Unlabeled Samples
Model
Prediction
Pseudo Label
$\tau_t(1)$
$\tau_t(3)$
$\tau_t(5)$
Global Threshold
1.00
0.75
0.50
0.25
0.00
0
t_1
t_2
Training iteration t
Local Threshold (t = t$_1$)
$\tau_{t_1}(1)$ $\tau_{t_1}(2)$ $\tau_{t_1}(3)$ $\tau_{t_1}(4)$ $\tau_{t_1}(5)$
Local Threshold (t = t$_2$)
$\tau_{t_2}(1)$ $\tau_{t_2}(2)$ $\tau_{t_2}(3)$ $\tau_{t_2}(4)$ $\tau_{t_2}(5)$

Figure 4.2: Illustration of Self-Adaptive Thresholding (SAT). FreeMatch adopts both global and local self-adaptive thresholds computed from the EMA of prediction statistics from unlabeled samples. Filtered (masked) samples are marked with red X.

the t-th time step (iteration). However, it would be time-consuming to compute the confidence for all unlabeled data at every time step or even every training epoch due to its large volume. Instead, we estimate the global confidence as the exponential moving average (EMA) of the confidence at each training time step. We initialize τ_t as $\frac{1}{C}$ where C indicates the number of classes. The global threshold τ_t is defined and adjusted as:

$$\tau_t = \begin{cases} \frac{1}{C}, & \text{if } t = 0, \\ \lambda \tau_{t-1} + (1-\lambda) \frac{1}{\mu B} \sum_{b=1}^{\mu B} \max(q_b), & \text{otherwise,} \end{cases} \tag{4.5}$$

where $\lambda \in (0, 1)$ is the momentum decay of EMA.

Self-adaptive Local Threshold The local threshold aims to modulate the global threshold in a class-specific fashion to account for the intra-class diversity and the possible class adjacency. We compute the expectation of the model's predictions on each class c to estimate the class-specific learning status:

$$\tilde{p}_t(c) = \begin{cases} \frac{1}{C}, & \text{if } t = 0, \\ \lambda \tilde{p}_{t-1}(c) + (1-\lambda) \frac{1}{\mu B} \sum_{b=1}^{\mu B} q_b(c), & \text{otherwise,} \end{cases} \tag{4.6}$$

where $\tilde{p}_t = [\tilde{p}_t(1), \tilde{p}_t(2), \ldots, \tilde{p}_t(C)]$ is the list containing all $\tilde{p}_t(c)$. Integrating the global and local thresholds, we obtain the final self-adaptive threshold $\tau_t(c)$ as:

$$\tau_t(c) = \text{MaxNorm}(\tilde{p}_t(c)) \cdot \tau_t = \frac{\tilde{p}_t(c)}{\max\{\tilde{p}_t(c) : c \in [C]\}} \cdot \tau_t, \tag{4.7}$$

where MaxNorm is the Maximum Normalization (i.e., $x' = \frac{x}{\max(x)}$). Finally, the unsupervised training objective $\mathcal{L}_u$ at the t-th iteration is:

$$\mathcal{L}_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) > \tau_t(\arg\max(q_b)) \cdot \mathcal{H}(\hat{q}_b, Q_b). \tag{4.8}$$

4.4.2 Self-Adaptive Fairness

We include the class fairness objective as mentioned in Section 4.3 into FreeMatch to encourage the model to make diverse predictions for each class and thus produce a meaningful self-adaptive threshold, especially under the settings where labeled data are rare. Instead of using a uniform prior as in [AOA+20b], we use the EMA of model predictions $\tilde{p}_t$ from Eq. 4.6 as an estimate of the expectation of prediction distribution over unlabeled data. We optimize the cross-entropy of $\tilde{p}_t$ and $\overline{p} = \mathbb{E}_{\mu B}[p_m(y|\Omega(u_b))]$ over mini-batch as an estimate of $H(\mathbb{E}_u\left[p_m(y|u)\right])$. Considering that the underlying pseudo label distribution may not be uniform, we propose to modulate the fairness objective in a self-adaptive way, i.e., normalizing the expectation of probability by the histogram distribution of pseudo labels to counter the negative effect of imbalance as:

$$\begin{aligned} \overline{p} &= \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(q_b\right) \geq \tau_t(\arg\max\left(q_b\right)\right) Q_b, \\ \overline{h} &= \operatorname{Hist}_{\mu B}\left(\mathbb{1}\left(\max\left(q_b\right) \geq \tau_t(\arg\max\left(q_b\right)\right) \hat{Q}_b\right). \end{aligned} \tag{4.9}$$

Similar to $\tilde{p}_t$, we compute $\tilde{h}_t$ as:

$$\tilde{h}_t = \lambda \tilde{h}_{t-1} + (1-\lambda)\operatorname{Hist}_{\mu B}\left(\hat{q}_b\right). \tag{4.10}$$

The self-adaptive fairness (SAF) L_f at the t-th iteration is formulated as:

$$\mathcal{L}_f = -\mathcal{H}\left(\operatorname{SumNorm}\left(\frac{\tilde{p}_t}{\tilde{h}_t}\right), \operatorname{SumNorm}\left(\frac{\overline{p}}{\overline{h}}\right)\right), \tag{4.11}$$

where SumNorm $= (\cdot)/\sum(\cdot)$. SAF encourages the expectation of the output probability for each mini-batch to be close to a marginal class distribution of the model, after normalized by histogram distribution. It helps the model produce diverse predictions especially for barely supervised settings [SBL+20], thus converges faster and generalizes better. This is also shown in Fig. 4.1 (b).

The overall objective for FreeMatch at t-th iteration is:

$$\mathcal{L} = \mathcal{L}_s + w_u\mathcal{L}_u + w_f\mathcal{L}_f, \tag{4.12}$$

where w_u and w_f represents the loss weight for $\mathcal{L}_u$ and $\mathcal{L}_f$ respectively. With $\mathcal{L}_u$ and $\mathcal{L}_f$, FreeMatch maximizes the mutual information between its outputs and inputs. We present the procedure of FreeMatch in Algorithm 2.

Algorithm 2 FreeMatch algorithm at t-th iteration.

```
Input: Number of classes $C$, labeled batch $\mathcal{X} = \{(x_b, y_b) : b \in (1,2,\dots,B)\}$, unlabeled batch $\mathcal{U} = \{u_b : b \in (1,2,\dots,\mu B)\}$, unsupervised loss weight $w_u$, fairness loss weight $w_f$, and EMA decay $\lambda$.
Compute $\mathcal{L}_s$ for labeled data
    $\mathcal{L}_s = \frac{1}{B}\sum_{b=1}^{B} \mathcal{H}(y_b, p_m(y|\omega(x_b)))$
Update the global threshold
    $\tau_t = \lambda \tau_{t-1} + (1-\lambda)\frac{1}{\mu B}\sum_{b=1}^{\mu B} max(q_b)$    // $q_b$ is $p_m(y|\omega(u_b))$, shape of $\tau_t$: $[1]$
Update the local threshold
    $\tilde{p}_t = \lambda \tilde{p}_{t-1} + (1-\lambda)\frac{1}{\mu B}\sum_{b=1}^{\mu B} q_b$    // Shape of $\tilde{p}_t$: $[C]$
Update histogram for $\tilde{p}_t$
    $\tilde{h}_t = \lambda \tilde{h}_{t-1} + (1-\lambda)\,\mathrm{Hist}_{\mu B}\left(\hat{q}_b\right)$    // Shape of $\tilde{h}_t$: $[C]$
for $c = 1$ to $C$ do
    $\tau_t(c) = \mathrm{MaxNorm}(\tilde{p}_t(c)) \cdot \tau_t$    // Calculate SAT
end for
Compute $\mathcal{L}_u$ on unlabeled data
    $\mathcal{L}_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(q_b\right) \geq \tau_t(\arg\max\left(q_b\right))\right) \cdot \mathcal{H}(\hat{q}_b, Q_b)$
Compute expectation of probability on unlabeled data
    $\overline{p} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(q_b\right) \geq \tau_t(\arg\max\left(q_b\right)\right) Q_b$    // $Q_b$ is an abbr. of $p_m(y|\Omega(u_b))$, shape of $\overline{p}$: $[C]$
Compute histogram for $\overline{p}$
    $\overline{h} = \mathrm{Hist}_{\mu B}\left(\mathbb{1}\left(\max\left(q_b\right) \geq \tau_t(\arg\max\left(q_b\right)\right) \hat{Q}_b\right)$    // Shape of $\overline{h}$: $[C]$
Compute $\mathcal{L}_f$ on unlabeled data
    $\mathcal{L}_f = -\mathcal{H}\left(\mathrm{SumNorm}(\frac{\tilde{p}_t}{\tilde{h}_t}), \mathrm{SumNorm}(\frac{\overline{p}}{\overline{h}})\right)$
Return: $\mathcal{L}_s + w_u \cdot \mathcal{L}_u + w_f \cdot \mathcal{L}_f$
```

4.5 Experiments

4.5.1 Setup

We evaluate FreeMatch on common benchmarks: CIFAR-10/100 [KH+09b], SVHN [NWC+11a], STL-10 [CNL11b] and ImageNet [DDS+09]. Following previous work [SBL+20, XSY+21, ZWH+21, OOR+18], we conduct experiments with varying amounts of labeled data. In addition to the commonly-chosen labeled amounts, following [SBL+20], we further include the most challenging case of CIFAR-10: each class has only *one* labeled sample.

For fair comparison, we train and evaluate all methods using the unified codebase TorchSSL [ZWH+21] with the same backbones and hyperparameters. Concretely, we use Wide ResNet-28-2 [ZK16b] for CIFAR-10, Wide ResNet-28-8 for CIFAR-100, Wide ResNet-37-2 [ZWB20] for STL-10, and ResNet-50 [HZRS16b] for ImageNet. We use SGD with a momentum of 0.9 as optimizer. The initial learning rate is 0.03 with a cosine learning rate decay schedule as $\eta = \eta_0 \cos(\frac{7\pi k}{16K})$, where η_0 is the initial learning rate, $k(K)$ is the current (total) training step and we set $K = 2^{20}$ for all datasets. At the testing phase, we use an exponential moving average with the momentum of 0.999 of the training model to conduct inference for all algorithms. The batch size of labeled data is 64 except for ImageNet where we set 128. We use the same weight decay value, pre-defined threshold τ, unlabeled batch ratio μ and loss

Dataset	CIFAR-10				CIFAR-100			SVHN			STL-10	
# Label	10	40	250	4000	400	2500	10000	40	250	1000	40	1000
Π Model	79.18±1.11	74.34±1.76	46.24±1.29	13.13±0.59	86.96±0.80	58.80±0.66	36.65±0.00	67.48±0.95	13.30±1.12	7.16±0.11	74.31±0.85	32.78±0.40
Pseudo Label	80.21±0.55	74.61±0.26	46.49±2.20	15.08±0.19	87.45±0.85	57.74±0.28	36.55±0.24	64.61±5.6	15.59±0.95	9.40±0.32	74.68±0.99	32.64±0.71
VAT	79.81±1.17	74.66±2.12	41.03±1.79	10.51±0.12	85.20±1.40	46.84±0.79	32.14±0.19	74.75±3.38	4.33±0.12	4.11±0.20	74.74±0.38	37.95±1.12
MeanTeacher	76.37±0.44	70.09±1.60	37.46±3.30	8.10±0.21	81.11±1.44	45.17±1.06	31.75±0.23	36.09±3.98	3.45±0.03	3.27±0.05	71.72±1.45	33.90±1.37
MixMatch	65.76±7.06	36.19±6.48	13.63±0.59	6.66±0.26	67.59±0.66	39.76±0.48	27.78±0.29	30.60±8.39	4.56±0.32	3.69±0.37	54.93±0.96	21.70±0.68
ReMixMatch	20.77±7.48	9.88±1.03	6.30±0.05	4.84±0.01	42.75±1.05	**26.03**±0.35	**20.02**±0.27	24.04±9.13	6.36±0.22	5.16±0.31	32.12±6.24	6.74±0.14
UDA	34.53±10.69	10.62±3.75	5.16±0.06	4.29±0.07	46.39±1.59	27.73±0.21	22.49±0.23	5.12±4.27	**1.92**±0.05	**1.89**±0.01	37.42±8.44	6.64±0.17
FixMatch	24.79±7.65	7.47±0.28	**4.86**±0.05	4.21±0.08	46.42±0.82	28.03±0.16	22.20±0.12	3.81±1.18	2.02±0.02	1.96±0.03	35.97±4.14	6.25±0.33
Dash	27.28±14.09	8.93±3.11	5.16±0.23	4.36±0.11	44.82±0.96	27.15±0.22	21.88±0.07	2.19±0.18	2.04±0.02	1.97±0.01	34.52±4.30	6.39±0.56
MPL	23.55±6.01	6.62±0.91	5.76±0.24	4.55±0.04	46.26±1.84	27.71±0.19	21.74±0.09	9.33±8.02	2.29±0.04	2.28±0.02	35.76±4.83	6.66±0.00
FlexMatch	13.85±12.04	4.97±0.06	4.98±0.09	4.19±0.01	39.94±1.62	26.49±0.20	21.90±0.15	8.19±3.20	6.59±2.29	6.72±0.30	29.15±4.16	5.77±0.18
FreeMatch	**8.07**±4.24	**4.90**±0.04	4.88±0.18	**4.10**±0.02	**37.98**±0.42	26.47±0.20	21.68±0.03	**1.97**±0.02	1.97±0.01	1.96±0.03	**15.56**±0.55	**5.63**±0.15

Table 4.1: Error rates on CIFAR-10/100, SVHN, and STL-10 datasets. The fully-supervised results of STL-10 are unavailable since we do not have label information for its unlabeled data. **Bold** indicates the best result and underline indicates the second-best result.

weights introduced for Pseudo-Label [Lee13], Π model [RBH+15b], Mean Teacher [TV17b], VAT [MMKI18a], MixMatch [BCG+19], ReMixMatch [BCC+20], UDA [XDH+20b], FixMatch [SBL+20], and FlexMatch [ZWH+21].

We implement MPL based on UDA as in [PDXL21], where we set temperature as 0.8 and w_u as 10. We do not fine-tune MPL on labeled data as in [PDXL21] since we find fine-tuning will make the model overfit the labeled data especially with very few of them. For Dash, we use the same parameters as in [XSY+21] except we warm-up on labeled data for 2 epochs since too much warm-up will lead to the overfitting (i.e. 2,048 training iterations). For FreeMatch, we set $w_u = 1$ for all experiments. Besides, we set $w_f = 0.01$ for CIFAR-10 with 10 labels, CIFAR-100 with 400 labels, STL-10 with 40 labels, ImageNet with 100k labels, and all experiments for SVHN. For other settings, we use $w_f = 0.05$. For SVHN, we find that using a low threshold at early training stage impedes the model to cluster the unlabeled data, thus we adopt two training techniques for SVHN: (1) warm-up the model on only labeled data for 2 epochs as Dash; and (2) restrict the SAT within the range $[0.9, 0.95]$. We train each algorithm 3 times using different random seeds and report the best error rates of all checkpoints [ZWH+21].

4.5.2 Quantitative Results

The Top-1 classification error rates of CIFAR-10/100, SVHN, and STL-10 are reported in Table 4.1. The results on ImageNet with 100 labels per class are in Table 4.2. These quantitative results demonstrate that FreeMatch achieves the best performance on CIFAR-10, STL-10, and ImageNet datasets, and it produces very close results on SVHN to the best competitor. On CIFAR-100, FreeMatch is better than ReMixMatch when there are 400 labels. The good performances of ReMixMatch on CIFAR-100 (2500) and CIFAR-100 (10000) are probably brought by the mix up [ZCDLP18] technique and the self-supervised learning part. On ImageNet with 100k labels, FreeMatch significantly outperforms the latest counterpart FlexMatch by **1.28**%[2]. We also notice that FreeMatch exhibits fast computation in ImageNet from Table 4.2. Note that FlexMatch is much slower than FixMatch and FreeMatch because it needs to maintain a list that records whether each sample is clean, which needs heavy indexing computation budget on large datasets.

[2] Following [ZWH+21], we train ImageNet for 2^{20} iterations like other datasets for a fair comparison. We use 4 Tesla V100 GPUs on ImageNet.

Noteworthy is that, FreeMatch consistently outperforms other methods by a large margin on settings with *extremely limited labeled data*: **5.78**% on CIFAR-10 with 10 labels, **1.96**% on CIFAR-100 with 400 labels, and surprisingly **13.59**% on STL-10 with 40 labels. STL-10 is a more realistic and challenging dataset compared to others, which consists of a large unlabeled set of 100k images. The significant improvements demonstrate the capability and potential of FreeMatch to be deployed in real-world applications.

	Top-1	Top-5	Runtime (sec./iter.)
FixMatch	43.66	21.80	**0.4**
FlexMatch	41.85	19.48	0.6
FreeMatch	**40.57**	**18.77**	**0.4**

Table 4.2: Error rates and runtime on ImageNet with 100 labels per class.

4.5.3 Qualitative Analysis

We present some qualitative analysis: Why and how does FreeMatch work? What other benefits does it bring? We evaluate the class average threshold and average sampling rate on STL-10 (40) (i.e., 40 labeled samples on STL-10) of FreeMatch to demonstrate how it works aligning with our theoretical analysis. We record the threshold and compute the sampling rate for each batch during training. The sampling rate is calculated on unlabeled data as $\frac{\sum_b^{\mu B} \mathbb{1}(\max(q_b)>\tau_t(\arg\max(q_b)))}{\mu B}$. We also plot the convergence speed in terms of accuracy and the confusion matrix to show the proposed component in FreeMatch helps improve performance. From Fig. 4.4 (a) and 4.4 (b), one can observe that the threshold and sampling rate change of FreeMatch is mostly consistent with our theoretical analysis. That is, at the early stage of training, the threshold of FreeMatch is relatively lower, compared to FlexMatch and FixMatch, resulting in higher unlabeled data utilization (sampling rate), which fastens the convergence. As the model learns better and becomes more confident, the threshold of FreeMatch increases to a high value to alleviate the confirmation bias, leading to stably high sampling rate. Correspondingly, the accuracy of FreeMatch increases vastly (as shown in Fig. 4.4 (c)) and resulting better class-wise accuracy (as shown in Fig. 4.4 (c)). Note that Dash fails to learn properly due to the employment of the high sampling rate until 100k iterations.

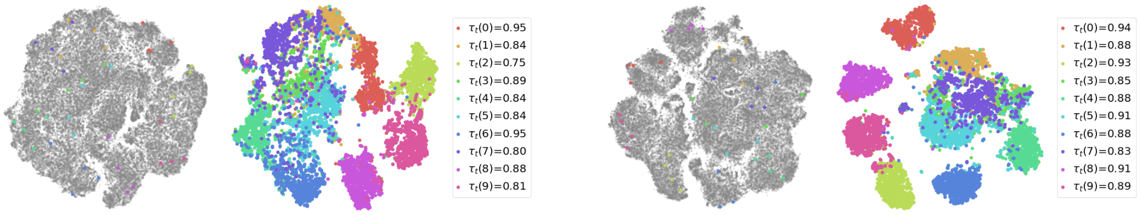
To further demonstrate the effectiveness of the class-specific threshold in FreeMatch, we present the t-SNE [VdMH08] visualization of features of FlexMatch and FreeMatch on STL-10 (40) in Fig. 4.3. We exhibit the corresponding local threshold for each class. Interestingly, FlexMatch has a high threshold, i.e., pre-defined 0.95, for class 0 and class 6, yet their feature variances are very large and are confused with other classes. This means the class-wise thresholds in FlexMatch cannot accurately reflect the learning status. In contrast, FreeMatch clusters most classes better. Besides, for the similar classes $1, 3, 5, 7$ that are confused with each other, FreeMatch retains a higher average threshold 0.87 than 0.84 of FlexMatch, enabling to mask more wrong pseudo labels.

4.5.4 Ablation Study

Self-adaptive Threshold

We conduct experiments on the components of SAT in FreeMatch and compare to the components in FlexMatch [ZWH+21], FixMatch [SBL+20], Class-Balanced Self-Training (CBST) [ZYKW18], and Relative Threshold (RT) in AdaMatch [BRS+22]. The ablation is conducted on CIFAR-10 (40 labels).

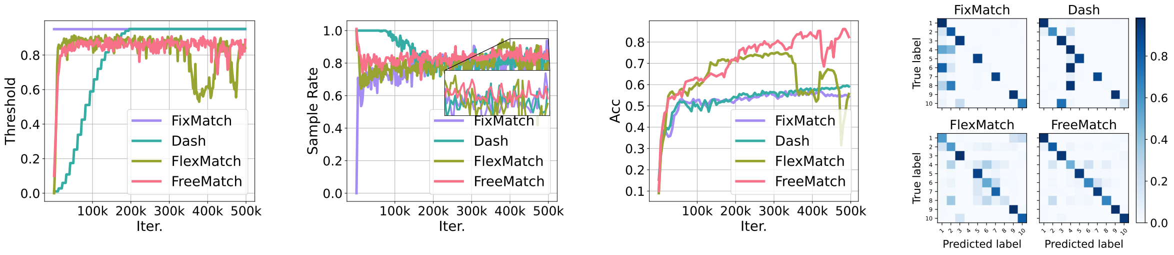
As shown in Table 4.3, SAT achieves the best performance among all the threshold schemes.



(a) FlexMatch (train, test)

(b) FreeMatch (train, test)

Figure 4.3: T-SNE visualization of FlexMatch and FreeMatch features on STL-10 (40). Unlabeled data is indicated by gray color. Local threshold $\tau_t(c)$ for each class is shown on the legend.



(a) Confidence threshold

(b) Sampling rate

(c) Accuracy

(d) Confusion matrix

Figure 4.4: How FreeMatch works in STL-10 with 40 labels, compared to others. (a) Class-average confidence threshold; (b) class-average sampling rate; (c) convergence speed in terms of accuracy; (d) confusion matrix, where fading colors of diagonal elements refer to the disparity of accuracy.

Self-adaptive global threshold τ_t and local threshold $\text{MaxNorm}(\tilde{p}_t(c))$ themselves also achieve comparable results, compared to the fixed threshold τ, demonstrating both local and global threshold proposed are good learning effect estimators. When using CPL $\mathcal{M}(\beta(c))$ to adjust τ_t, the result is worse than the fixed threshold and exhibits larger variance, indicating potential instability of CPL. AdaMatch [BRS$^+$22] uses the RT, which can be viewed as a global threshold at t-th iteration computed on the predictions of labeled data without EMA, whereas FreeMatch conducts computation of τ_t with EMA on unlabeled data that can better reflect the overall data distribution. For class-wise threshold, CBST [ZYKW18] maintains a pre-defined sampling rate, which could be the reason for its bad performance since the sampling rate should be changed during training as we analyzed in Sec. 4.2. Note that we did not include L_f in this ablation for a fair comparison.

Self-adaptive Fairness As illustrated in Table 4.4, we also empirically study the effect of SAF on CIFAR-10 (10 labels). We study the original version of fairness objective as in [AOA$^+$20b]. Based on that, we study the operation of normalization probability by histograms and show that countering the effect of imbalanced underlying distribution indeed helps the model to learn and diverse better. One may notice that adding original fairness regularization alone already helps improve the performance. Whereas adding normalization operation in the log operation hurts the performance, suggesting the underlying batch data are indeed not uniformly distributed. We also evaluate Distribution Alignment (DA) for class fairness in ReMixMatch [BCC$^+$20] and AdaMatch [BRS$^+$22], showing inferior results than SAF. A possible reason for the worse performance of DA (AdaMatch) is that it only uses labeled batch prediction as the target distribution which cannot reflect the true data distribution especially when labeled data is

Threshold	CIFAR-10 (40)
τ (FixMatch)	7.47±0.28
$\tau * \mathcal{M}(\beta(c))$ (FlexMatch)	4.97±0.06
$\tau * \text{MaxNorm}(\tilde{p}_t(c))$	5.13±0.03
τ_t (Global)	6.06±0.65
$\tau_t * \mathcal{M}(\beta(c))$	8.40±2.49
CBST	16.65±2.90
RT (AdaMatch)	6.09±0.65
SAT (Global and Local)	**4.92**±0.04

Table 4.3: Comparison of different thresholding schemes.

Fairness	CIFAR-10 (10)
w/o fairness	10.37±7.70
$U \log \overline{p}$	9.57±6.67
$U \log \text{SumNorm}(\frac{\overline{p}}{\overline{h}})$	12.07±5.23
DA (AdaMatch)	32.94±1.83
DA (ReMixMatch)	11.06±8.21
SAF	**8.07**±4.24

Table 4.4: Comparison of different class fairness items.

scarce and changing the target distribution to the ground truth uniform, i.e., DA (ReMixMatch), is better for the case with extremely limited labels.

4.6 Related Work

To reduce confirmation bias [AOA+20b] in pseudo labeling, confidence-based thresholding techniques are proposed to ensure the quality of pseudo labels [XDH+20b, SBL+20, ZWH+21, XSY+21], where only the unlabeled data whose confidences are higher than the threshold are retained. UDA [XDH+20b] and FixMatch [SBL+20] keep the fixed pre-defined threshold during training. FlexMatch [ZWH+21] adjusts the pre-defined threshold in a class-specific fashion according to the per-class learning status estimated by the number of confident unlabeled data. A co-current work Adsh [GL22] explicitly optimizes the number of pseudo labels for each class in the SSL objective to obtain adaptive thresholds for imbalanced Semi-supervised Learning. However, it still needs a user-predefined threshold. Dash [XSY+21] defines a threshold according to the loss on labeled data and adjusts the threshold according to a fixed mechanism. A more recent work, AdaMatch [BRS+22], aims to unify SSL and domain adaptation using a pre-defined threshold multiplying the average confidence of the labeled data batch to mask noisy pseudo labels. It needs a pre-defined threshold and ignores the unlabeled data distribution especially when labeled data is too rare to reflect the unlabeled data distribution. Besides, distribution alignment [BCC+20, BRS+22] is also utilized in Adamatch to encourage fair predictions on unlabeled data. Previous methods might fail to choose meaningful thresholds due to ignorance of the relationship between the model learning status and thresholds. [CWKM20, KML20] try to understand self-training / thresholding from the theoretical perspective. We use a motivating example to derive some implications and further adjust meaningful thresholds according to the learning status satisfying the derived implications.

Except consistency regularization, entropy-based regularization is also used in SSL. Entropy minimization [GB04] encourages the model to make confident predictions for all samples disregarding the actual class predicted. Maximization of expectation of entropy [AK10, AOA$^+$20b] over all samples is also proposed to induce fairness to the model, enforcing the model to predict each class at the same frequency. But previous ones assume a uniform prior for underlying data distribution and also ignore the batch data distribution. Distribution alignment [BCC$^+$20] adjusts the pseudo labels according to labeled data distribution and the EMA of model predictions.

4.7 Conclusion

We proposed FreeMatch that utilizes self-adaptive thresholding and class-fairness regularization for SSL. FreeMatch outperforms strong competitors across a variety of SSL benchmarks, especially in the barely-supervised setting. We believe that confidence thresholding has more potential in SSL. A potential limitation is that the adaptiveness still originates from the heuristics of the model prediction, and we hope the efficacy of FreeMatch inspires more research for optimal thresholding.

5 Addressing the Quantity-Quality Trade-off in Semi-supervised Learning

Contents

Following a similar theme to the previous chapter, this chapter further explores threshold-based pseudo-labeling for semi-supervised learning (SSL) from a different perspective. We first revisit the popular pseudo-labeling methods via a unified sample weighting formulation and demonstrate the inherent quantity-quality trade-off problem of pseudo-labeling with thresholding, which may prohibit learning. To this end, we propose SoftMatch to overcome the trade-off by maintaining both high quantity and high quality of pseudo-labels during training, effectively exploiting the unlabeled data. We derive a truncated Gaussian function to weight samples based on their confidence, which can be viewed as a soft version of the confidence threshold. We further enhance the utilization of weakly-learned classes by proposing a uniform alignment approach. In experiments, SoftMatch shows substantial improvements across a wide variety of benchmarks, including image, text, and imbalanced classification.

This chapter is based on [CTF+23]. As one of the co-authors, Yue Fan was involved in the weekly and more detailed discussions and contributed to the writing of the paper and the imbalanced SSL experiments.

5.1 Introduction

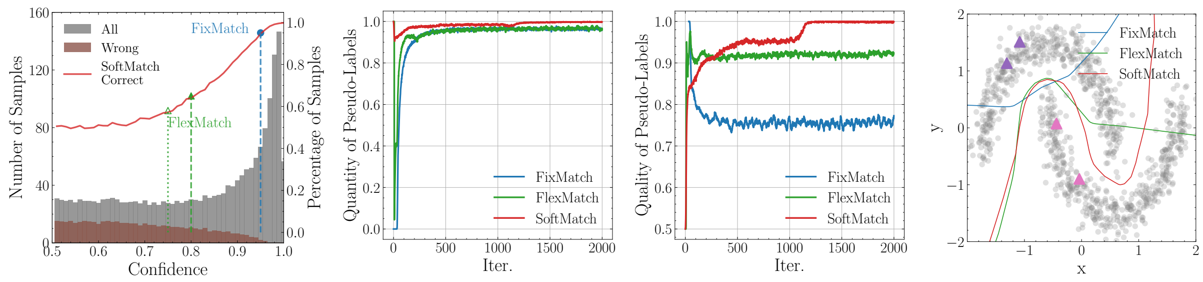
Semi-Supervised Learning (SSL), concerned with learning from a few labeled data and a large amount of unlabeled data, has shown great potential in practical applications for significantly reduced requirements on laborious annotations [FKS21b, XDH+20b, SBL+20, PDXL21, ZWH+21, XSY+21, XZH+21, CYZW21, OOR+18]. The main challenge of SSL lies in how to effectively exploit the information of unlabeled data to improve the model's generalization performance [CSZ06]. Among the efforts, pseudo-labeling [Lee13, AOA+20b] with confidence thresholding [XDH+20b, SBL+20, XSY+21, ZWH+21] is highly-successful and widely-adopted.

The core idea of threshold-based pseudo-labeling [XDH+20b, SBL+20, XSY+21, ZWH+21] is to train the model with pseudo-label whose prediction confidence is above a hard threshold, with the others being simply ignored. However, such a mechanism inherently exhibits the *quantity-quality trade-off*, which undermines the learning process. On the one hand, a high confidence threshold as exploited in FixMatch [SBL+20] ensures the quality of the pseudo-labels. However, it discards a considerable number of unconfident yet correct pseudo-labels. As an example shown in Fig. 5.1 (a), **around 71% correct pseudo-labels are *excluded* from the training.** On the other hand, dynamically growing threshold [XSY+21, BRS+22], or class-wise threshold [ZWH+21] encourages the utilization of more pseudo-labels but inevitably fully enrolls erroneous pseudo-labels that may mislead training. As an example shown by FlexMatch [ZWH+21] in Fig. 5.1 (b), **about 16% of the utilized pseudo-labels are *incorrect*.** In summary, the quantity-quality trade-off with a confidence threshold limits the unlabeled data utilization, which may hinder the model's generalization performance.

In this work, we formally define the quantity and quality of pseudo-labels in SSL and summarize the inherent trade-off present in previous methods from a perspective of unified sample weighting formulation. We first identify the fundamental reason behind the quantity-quality trade-off is the lack of sophisticated assumption imposed by the weighting function on the distribution of pseudo-labels. Especially, confidence thresholding can be regarded as a step function assigning binary weights according to samples' confidence, which assumes pseudo-labels with confidence above the threshold are equally correct while others are wrong. Based on the analysis, we propose SoftMatch to overcome the trade-off by maintaining high quantity and high quality of pseudo-labels during training. A truncated Gaussian function is derived from our assumption on the marginal distribution to fit the confidence distribution, which assigns lower weights to possibly correct pseudo-labels according to the deviation of their confidence from the mean of Gaussian. The parameters of the Gaussian function are estimated using the historical predictions from the model during training. Furthermore, we propose Uniform Alignment to resolve the imbalance issue in pseudo-labels, resulting from different learning difficulties of different classes. It further consolidates the quantity of pseudo-labels while maintaining their quality. On the two-moon example, as shown in Fig. 5.1 (c) and Fig. 5.1 (b), SoftMatch achieves a distinctively better accuracy of pseudo-labels while retaining a consistently higher utilization ratio of them during training, therefore, leading to a better-learned decision boundary as shown in Fig. 5.1 (d). We demonstrate that SoftMatch achieves a new state-of-the-art on a wide range of image and text classification tasks. We further validate the robustness of SoftMatch against long-tailed distribution by evaluating imbalanced classification tasks.

Our contributions can be summarized as:

- We demonstrate the importance of the unified weighting function by formally defining the quantity and quality of pseudo-labels, and the trade-off between them. We identify that the inherent trade-off in previous methods mainly stems from the lack of careful design on the distribution of pseudo-labels, which is imposed directly by the weighting function.
- We propose SoftMatch to effectively leverage the unconfident yet correct pseudo-labels, fitting a truncated Gaussian function the distribution of confidence, which overcomes the trade-off. We further propose Uniform Alignment to resolve the imbalance issue of pseudo-labels while maintaining their high quantity and quality.
- We demonstrate that SoftMatch outperforms previous methods on various image and text evaluation settings. We also empirically verify the importance of maintaining the



(a) Confi. Dist. (b) Quantity (c) Quality (d) Decision Bound.

Figure 5.1: Illustration on Two-Moon Dataset with only 4 labeled samples (triangle purple/pink points) with others as unlabeled samples in training a 3-layer MLP classifier. (a) Confidence distribution, including all predictions and wrong predictions. The red line denotes the correct percentage of samples used by SoftMatch. The part of the line above scatter points denotes the correct percentage for FixMatch (blue) and FlexMatch (green). (b) Quantity of pseudo-labels; (c) Quality of pseudo-labels; (d) Decision boundary. SoftMatch exploits almost all samples during training with lowest error rate and best decision boundary.

high accuracy of pseudo-labels while pursuing better unlabeled data utilization in SSL.

5.2 Revisit Quantity-Quality Trade-off of SSL

In this section, we formulate the quantity and quality of pseudo-labels from a unified sample weighting perspective, by demonstrating the connection between sample weighting function and the quantity/quality of pseudo-labels. SoftMatch is naturally inspired by revisiting the inherent limitation in quantity-quality trade-off of the existing methods.

5.2.1 Problem Statement

We first formulate the framework of SSL in a C-class classification problem. Denote the labeled and unlabeled datasets as $\mathcal{D}_L = \{\mathbf{x}_i^l, \mathbf{y}_i^l\}_{i=1}^{N_L}$ and $\mathcal{D}_U = \{\mathbf{x}_i^u\}_{i=1}^{N_U}$, respectively, where $\mathbf{x}_i^l, \mathbf{x}_i^u \in \mathbb{R}^d$ is the d-dimensional labeled and unlabeled training sample, and $\mathbf{y}_i^l$ is the one-hot ground-truth label for labeled data. We use N_L and N_U to represent the number of training samples in $\mathcal{D}_L$ and $\mathcal{D}_U$, respectively. Let $\mathbf{p}(\mathbf{y}|\mathbf{x}) \in \mathbb{R}^C$ denote the model's prediction. During training, given a batch of labeled data and unlabeled data, the model is optimized using a joint objective $\mathcal{L} = \mathcal{L}_s + \mathcal{L}_u$, where $\mathcal{L}_s$ is the supervised objective of the cross-entropy loss ($\mathcal{H}$) on the B_L-sized labeled batch:

$$\mathcal{L}_s = \frac{1}{B_L} \sum_{i=1}^{B_L} \mathcal{H}(\mathbf{y}_i, \mathbf{p}(\mathbf{y}|\mathbf{x}_i^l)). \tag{5.1}$$

For the unsupervised loss, most existing methods with pseudo-labeling [Lee13, AOA+20b, XDH+20b, SBL+20, XSY+21, ZWH+21] exploit a confidence thresholding mechanism to mask out the unconfident and possibly incorrect pseudo-labels from training. In this chapter, we take a step further and present a *unified* formulation of the confidence thresholding scheme (and other schemes) from the sample weighting perspective. Specifically, we formulate the unsupervised loss $\mathcal{L}_u$ as the **weighted cross-entropy** between the model's prediction of the strongly-augmented data $\Omega(\mathbf{x}^u)$ and pseudo-labels from the weakly-augmented data $\omega(\mathbf{x}^u)$:

$$\mathcal{L}_u = \frac{1}{B_U} \sum_{i=1}^{B_U} \lambda(\mathbf{p}_i) \mathcal{H}(\hat{\mathbf{p}}_i, \mathbf{p}(\mathbf{y}|\Omega(\mathbf{x}_i^u))), \tag{5.2}$$

where $\mathbf{p}$ is the abbreviation of $\mathbf{p}(\mathbf{y}|\omega(\mathbf{x}^u))$, and $\hat{\mathbf{p}}$ is the one-hot pseudo-label $\operatorname{argmax}(\mathbf{p})$; $\lambda(\mathbf{p})$ is the sample weighting function with range $[0, \lambda_{\max}]$; and B_U is the batch size for unlabeled data.

5.2.2 Quantity-Quality Trade-off from Sample Weighting Perspective

In this section, we demonstrate the importance of the unified weighting function $\lambda(\mathbf{p})$, by showing its different instantiations in previous methods and its essential connection with model predictions. We start by formulating the *quantity* and *quality* of pseudo-labels.

Definition 5.2.1 (Quantity of pseudo-labels)**.** The quantity $f(\mathbf{p})$ of pseudo-labels enrolled in training is defined as the expectation of the sample weight $\lambda(\mathbf{p})$ over the unlabeled data:

$$f(\mathbf{p}) = \mathbb{E}_{\mathcal{D}_U}[\lambda(\mathbf{p})] \in [0, \lambda_{\max}]. \tag{5.3}$$

Definition 5.2.2 (Quality of pseudo-labels)**.** The quality $g(\mathbf{p})$ is the expectation of the weighted 0/1 error of pseudo-labels, assuming the label $\mathbf{y}^u$ is given for $\mathbf{x}^u$ for only theoretical analysis purpose:

$$g(\mathbf{p}) = \sum_{i}^{N_U} \mathbb{1}(\hat{\mathbf{p}}_i = \mathbf{y}_i^u) \frac{\lambda(\mathbf{p}_i)}{\sum_j^{N_U} \lambda(\mathbf{p}_j)} = \mathbb{E}_{\bar{\lambda}(\mathbf{p})}[\mathbb{1}(\hat{\mathbf{p}} = \mathbf{y}^u)] \in [0, 1], \tag{5.4}$$

where $\bar{\lambda}(\mathbf{p}) = \lambda(\mathbf{p}) / \sum \lambda(\mathbf{p})$ is the probability mass function (PMF) of $\mathbf{p}$ being close to $\mathbf{y}^u$.

Based on the definitions of quality and quantity, we present the *quantity-quality trade-off* of SSL.

Definition 5.2.3 (The quantity-quality trade-off)**.** Due to the implicit assumptions of PMF $\bar{\lambda}(\mathbf{p})$ on the marginal distribution of model predictions, the lack of sophisticated design on it usually results in a trade-off in quantity and quality - when one of them increases, the other must decrease. Ideally, a well-defined $\lambda(\mathbf{p})$ should reflect the true distribution and lead to both high quantity and quality.

Despite its importance, $\lambda(\mathbf{p})$ has hardly been defined explicitly or properly in previous methods. In this chapter, we first summarize $\lambda(\mathbf{p})$, $\bar{\lambda}(\mathbf{p})$, $f(\mathbf{p})$, and $g(\mathbf{p})$ of relevant methods, as shown in Table 5.1. For example, naive pseudo-labeling [Lee13] and loss weight ramp-up scheme [ST17, TV17b, BCG+19, BCC+20] exploit the fixed sample weight to fully enroll all pseudo-labels into training. It is equivalent to set $\lambda = \lambda_{\max}$ and $\bar{\lambda} = 1/N_U$, regardless of $\mathbf{p}$, which means each pseudo-label is assumed equally correct. We can verify the quantity of pseudo-labels is maximized to $\lambda_{\max}$. However, maximizing quantity also fully involves the erroneous pseudo-labels, resulting in deficient quality, especially in early training. This failure trade-off is due to the implicit uniform assumption on PMF $\bar{\lambda}(\mathbf{p})$ that is far from the realistic situation.

In confidence thresholding [AOA+20b, SBL+20, XDH+20b], we can view the sample weights as being computed from a step function with confidence $\max(\mathbf{p})$ as the input and a pre-defined threshold τ as the breakpoint. It sets $\lambda(\mathbf{p})$ to $\lambda_{\max}$ when the confidence is

Scheme	Pseudo-Label	FixMatch	SoftMatch
$\lambda(\mathbf{p})$	$\lambda_{\max}$	$\begin{cases}\lambda_{\max}, & \text{if } \max(\mathbf{p}) \geq \tau, \\ 0.0, & \text{otherwise.}\end{cases}$	$\begin{cases}\lambda_{\max} \exp\left(-\frac{(\max(\mathbf{p})-\mu_t)^2}{2\sigma_t^2}\right), & \text{if } \max(\mathbf{p}) < \mu_t, \\ \lambda_{\max}, & \text{otherwise.}\end{cases}$
$\bar{\lambda}(\mathbf{p})$	$1/N_U$	$\begin{cases}1/\hat{N}_U^\tau, & \text{if } \max(\mathbf{p}) \geq \tau, \\ 0.0, & \text{otherwise.}\end{cases}$	$\begin{cases}\frac{\exp(-\frac{(\max(\mathbf{p}_i)-\hat{\mu}_t)^2}{2\hat{\sigma}_t{}^2})}{\frac{N_U}{2}+\sum_i^{\frac{N_U}{2}} \exp(-\frac{(\max(\mathbf{p}_i)-\hat{\mu}_t)^2}{2\hat{\sigma}_t{}^2})}, & \max(\mathbf{p}) < \mu_t \\ \frac{1}{\frac{N_U}{2}+\sum_i^{\frac{N_U}{2}} \exp(-\frac{(\max(\mathbf{p}_i)-\hat{\mu}_t)^2}{2\hat{\sigma}_t{}^2})}, & \max(\mathbf{p}) \geq \mu_t\end{cases}$
$f(\mathbf{p})$	$\lambda_{\max}$	$\lambda_{\max}\hat{N}_U^\tau/N_U$	$\lambda_{\max}/2 + \lambda_{\max}/N_U \sum_i^{\frac{N_U}{2}} \exp(-\frac{(\max(\mathbf{p}_i)-\hat{\mu}_t)^2}{2\hat{\sigma}_t{}^2})$
$g(\mathbf{p})$	$\sum_i^{N_U} \frac{\mathbb{1}(\hat{\mathbf{p}}=\mathbf{y}^u)}{N_U}$	$\sum_i^{\hat{N}_U} \mathbb{1}(\hat{\mathbf{p}} = \mathbf{y}^u)/\hat{N}_U^\tau$	$\sum_j^{\hat{N}_U^{\mu_t}} \mathbb{1}(\hat{\mathbf{p}}_j = \mathbf{y}_j^u)/2\hat{N}_U +$ $\sum_i^{N_U-\hat{N}_U^{\mu_t}} \mathbb{1}(\hat{\mathbf{p}}_i = \mathbf{y}_i^u) \exp(-\frac{(\max(\mathbf{p}_i)-\mu_t)^2}{\sigma_t^2})/2(N_U - \hat{N}_U^{\mu_t})$
Note	High Quantity Low Quality	Low Quantity High Quality	High Quantity High Quality

Table 5.1: Summary of different sample weighting function $\lambda(\mathbf{p})$, probability density function $\bar{\lambda}(\mathbf{p})$ of $\mathbf{p}$, quantity $f(\mathbf{p})$ and quality $g(\mathbf{p})$ of pseudo-labels used in previous methods and SoftMatch.

above τ and otherwise 0. Denoting $\hat{N}_U^\tau = \sum_i^{N_U} \mathbb{1}(\max(\mathbf{p}) \geq \tau)$ as the total number of samples whose predicted confidence are above the threshold, $\bar{\lambda}$ is set to a uniform PMF with a total mass of $\hat{N}_U^\tau$ within a fixed range $[\tau, 1]$. This is equal to constrain the unlabeled data as $\hat{\mathcal{D}}_U^\tau = \{\mathbf{x}^u; \max(\mathbf{p}(\mathbf{y}|\mathbf{x}^u)) \geq \tau\}$, with others simply being discarded. We can derive the quantity and the quality as shown in Table 5.1. A trade-off exists between the quality and quantity of pseudo-labels in confidence thresholding controlled by τ. On the one hand, while a high threshold ensures quality, it limits the quantity of enrolled samples. On the other hand, a low threshold sacrifices quality by fully involving more but possibly erroneous pseudo-labels in training. The trade-off still results from the over-simplification of the PMF from actual cases. Adaptive confidence thresholding [ZWH+21, XSY+21] adopts the dynamic and class-wise threshold, which alleviates the trade-off by evolving the (class-wise) threshold during learning. They impose a further relaxation on the assumption of distribution, but the uniform nature of the assumed PMF remains unchanged.

While some methods indeed consider the definition of $\lambda(\mathbf{p})$ [RYS20, HWH+21, KMK+22], interestingly, they all neglect the assumption induced on the PMF. The lack of sophisticated modeling of $\bar{\lambda}(\mathbf{p})$ usually leads to a quantity-quality trade-off in the unsupervised loss of SSL, which motivates us to propose SoftMatch to overcome this challenge.

5.3 SoftMatch

5.3.1 Gaussian Function for Sample Weighting

Inherently different from previous methods, we generally assume the underlying PMF $\bar{\lambda}(\mathbf{p})$ of marginal distribution follows a *dynamic and truncated Gaussian* distribution of mean μ_t and variance σ_t at t-th training iteration. We choose Gaussian for its maximum entropy property and empirically verified better generalization. Note that this is equivalent to treat the deviation of confidence $\max(\mathbf{p})$ from the mean μ_t of Gaussian as a proxy measure of the correctness of the model's prediction, where samples with higher confidence are less prone to be erroneous than that with lower confidence, consistent to the observation as shown in Fig. 5.1 (a). To this end, we can derive $\lambda(\mathbf{p})$ as:

$$\lambda(\mathbf{p}) = \begin{cases} \lambda_{\max} \exp\left(-\frac{(\max(\mathbf{p})-\mu_t)^2}{2\sigma_t^2}\right), & \text{if } \max(\mathbf{p}) < \mu_t, \\ \lambda_{\max}, & \text{otherwise.} \end{cases} \tag{5.5}$$

which is also a truncated Gaussian function within the range $[0, \lambda_{\max}]$, on the confidence $\max(\mathbf{p})$.

However, the underlying true Gaussian parameters μ_t and σ_t are still unknown. Although we can set the parameters to fixed values as in FixMatch [SBL+20] or linearly interpolate them within some pre-defined range as in Ramp-up [TV17b], this might again over-simplify the PMF assumption as discussed before. Recall that the PMF $\bar{\lambda}(\mathbf{p})$ is defined over $\max(\mathbf{p})$, we can instead *fit* the truncated Gaussian function directly to the confidence distribution for better generalization. Specifically, we can estimate μ and σ^2 from the historical predictions of the model. At t-th iteration, we compute the empirical mean and the variance as:

$$\begin{aligned} \hat{\mu}_b &= \hat{\mathbb{E}}_{B_U}[\max(\mathbf{p})] = \frac{1}{B_U}\sum_{i=1}^{B_U} \max(\mathbf{p}_i), \\ \hat{\sigma}_b^2 &= \hat{\text{Var}}_{B_U}[\max(\mathbf{p})] = \frac{1}{B_U}\sum_{i=1}^{B_U} \left(\max(\mathbf{p}_i) - \hat{\mu}_b\right)^2. \end{aligned} \tag{5.6}$$

We then aggregate the batch statistics for a more stable estimation, using Exponential Moving Average (EMA) with a momentum m over previous batches:

$$\begin{aligned} \hat{\mu}_t &= m\hat{\mu}_{t-1} + (1-m)\hat{\mu}_b, \\ \hat{\sigma}_t^2 &= m\hat{\sigma}_{t-1}^2 + (1-m)\frac{B_U}{B_U - 1}\hat{\sigma}_b^2, \end{aligned} \tag{5.7}$$

where we use unbiased variance for EMA and initialize $\hat{\mu}_0$ as $\frac{1}{C}$ and $\hat{\sigma}_0^2$ as 1.0. The estimated mean $\hat{\mu}_t$ and variance $\hat{\sigma}_t^2$ are plugged back into Eq. 5.5 to compute sample weights.

Estimating the Gaussian parameters adaptively from the confidence distribution during training not only improves the generalization but also better resolves the quantity-quality trade-off. We can verify this by computing the quantity and quality of pseudo-labels as shown in Table 5.1. The derived quantity $f(\mathbf{p})$ is bounded by $[\frac{\lambda_{\max}}{2}(1 + \exp(-\frac{(\frac{1}{C}-\hat{\mu}_t)^2}{2\hat{\sigma}_t{}^2})), \lambda_{\max}]$, indicating SoftMatch guarantees at least $\lambda_{\max}/2$ of quantity during training. As the model learns better and becomes more confident, i.e., $\hat{\mu}_t$ increases and $\hat{\sigma}_t$ decreases, the lower tail of the quantity becomes much tighter. While quantity maintains high, the quality of pseudo-labels also improves. As the tail of the Gaussian exponentially grows tighter during training, the erroneous pseudo-labels where the model is highly unconfident are assigned with lower weights, and those whose confidence are around $\hat{\mu}_t$ are more efficiently utilized. The truncated Gaussian weighting function generally behaves as **a soft and adaptive version of confidence thresholding**, thus we term the proposed method as SoftMatch.

5.3.2 Uniform Alignment for Fair Quantity

As different classes exhibit different learning difficulties, generated pseudo-labels can have potentially *imbalanced* distribution, which may limit the generalization of the PMF assumption

[OOR+18, ZWH+21]. To overcome this problem, we propose Uniform Alignment (UA), encouraging more uniform pseudo-labels of different classes. Specifically, we define the distribution in pseudo-labels as the expectation of the model predictions on unlabeled data: $\mathbb{E}_{\mathcal{D}_U}[\mathbf{p}(\mathbf{y}|\mathbf{x}^u)]$. During training, it is estimated as $\hat{\mathbb{E}}_{B_U}[\mathbf{p}(\mathbf{y}|\mathbf{x}^u)]$ using the EMA of batch predictions on unlabeled data. We use the ratio between a uniform distribution $\mathbf{u}(C) \in \mathbb{R}^C$ and $\hat{\mathbb{E}}_{B_U}[\mathbf{p}(\mathbf{y}|\mathbf{x}^u)]$ to normalize the each prediction $\mathbf{p}$ on unlabeled data and use the normalized probability to calculate the per-sample loss weight. We formulate the UA operation as:

$$\mathrm{UA}(\mathbf{p}) = \mathrm{Normalize}\left(\mathbf{p} \cdot \frac{\mathbf{u}(C)}{\hat{\mathbb{E}}_{B_U}[\mathbf{p}]}\right), \tag{5.8}$$

where the $\mathrm{Normalize}(\cdot) = (\cdot)/\sum(\cdot)$, ensuring the normalized probability sums to 1.0. With UA plugged in, the final sample weighting function in SoftMatch becomes:

$$\lambda(\mathbf{p}) = \begin{cases} \lambda_{\max} \exp\left(-\frac{(\max(\mathrm{UA}(\mathbf{p})) - \hat{\mu}_t)^2}{2\hat{\sigma}_t^2}\right), & \text{if } \max(\mathrm{UA}(\mathbf{p})) < \hat{\mu}_t, \\ \lambda_{\max}, & \text{otherwise.} \end{cases} \tag{5.9}$$

When computing the sample weights, UA encourages larger weights to be assigned to less-predicted pseudo-labels and smaller weights to more-predicted pseudo-labels, alleviating the imbalance issue.

An essential difference between UA and Distribution Alignment (DA) [BCC+20] proposed earlier lies in the computation of unsupervised loss. The normalization operation makes the predicted probability biased towards the less-predicted classes. In DA, this might not be an issue, as the normalized prediction is used as *soft target* in the cross-entropy loss. However, with pseudo-labeling, more erroneous pseudo-labels are probably created after normalization, which damages the quality. UA avoids this issue by exploiting original predictions to compute pseudo-labels and normalized predictions to compute sample weights, maintaining both the quantity and quality of pseudo-labels in SoftMatch. The complete training algorithm is shown in Alg. 3.

Algorithm 3 SoftMatch algorithm.

```
Input: Number of classes $C$, labeled batch $\{\mathbf{x}_i, \mathbf{y}_i\}_{i \in [B_L]}$, unlabeled batch $\{\mathbf{u}_i\}_{i \in [B_U]}$, and EMA momentum $m$.
Define: $\mathbf{p}_i = \mathbf{p}(\mathbf{y}|\omega(\mathbf{u}_i))$
$\mathcal{L}_s = \frac{1}{B_L} \sum_{i=1}^{B_L} \mathcal{H}(\mathbf{y}_i, \mathbf{p}(\mathbf{y}|\omega(\mathbf{x}_i)))$   // Compute $\mathcal{L}_s$ on labeled batch
$\hat{\mu}_b = \frac{1}{B_U} \sum_{i=1}^{B_U} \max(\mathbf{p}_i)$   // Compute the mean of confidence
$\hat{\sigma}^2 = \frac{1}{B_U} \sum_{i=1}^{B_U} (\max(\mathbf{p}_i) - \hat{\mu}_b)^2$   // Compute the variance of confidence
$\hat{\mu}_t = m\hat{\mu}_{t-1} + (1-m)\hat{\mu}_b$   // Update EMA of mean
$\hat{\sigma}_t^2 = m\hat{\sigma}_{t-1}^2 + (1-m)\frac{B_U}{B_U - 1}\hat{\sigma}_b^2$   // Update EMA of variance
for $i$ = 1 to $B_U$ do
    $\lambda(\mathbf{p}_i) = \begin{cases} \exp\left(-\frac{(\max(\mathrm{UA}(\mathbf{p}_i)) - \hat{\mu}_t)^2}{2\hat{\sigma}_t^2}\right), & \text{if } \max(\mathrm{UA}(\mathbf{p}_i)) < \hat{\mu}_t, \\ 1.0, & \text{otherwise.} \end{cases}$   // Compute loss weight
end for
$\mathcal{L}_u = \frac{1}{B_U} \sum_{i=1}^{B_U} \lambda(\mathbf{p}_i)\mathcal{H}(\hat{\mathbf{p}}_i, \mathbf{p}(\mathbf{y}|\Omega(\mathbf{u}_i)))$   // Compute $\mathcal{L}_u$ on unlabeled batch
Return: $\mathcal{L}_s + \mathcal{L}_u$
```

5.4 Experiments

While most SSL literature performs evaluation on image tasks, we extensively evaluate SoftMatch on various datasets including image and text datasets with classic and long-tailed settings. Moreover, We provide ablation study and qualitative comparison to analyze the effectiveness of SoftMatch. [1]

5.4.1 Classic Image Classification

Setup. For the classic image classification setting, we evaluate on CIFAR-10/100 [KH+09b], SVHN[NWC+11a], STL-10 [CNL11b] and ImageNet [DDS+09], with various numbers of labeled data, where class distribution of the labeled data is balanced. We use the WRN-28-2 [ZK16b] for CIFAR-10 and SVHN, WRN-28-8 for CIFAR-100, WRN-37-2 [ZWB20] for STL-10, and ResNet-50 [HZRS16b] for ImageNet. For all experiments, we use SGD optimizer with a momentum of 0.9, where the initial learning rate η_0 is set to 0.03. We adopt the cosine learning rate annealing scheme to adjust the learning rate with a total training step of 2^{20}. The labeled batch size B_L is set to 64 and the unlabeled batch size B_U is set to 7 times of B_L for all datasets. We set m to 0.999 and divide the estimated variance $\hat{\sigma}_t$ by 4 for 2σ of the Gaussian function. We record the EMA of model parameters for evaluation with a momentum of 0.999. Each experiment is run with three random seeds on labeled data, where we report the top-1 error rate.

Results. SoftMatch obtains the state-of-the-art results on almost all settings in Table 5.2 and Table 5.3, except CIFAR-100 with 2,500 and 10,000 labels and SVHN with 1,000 labels, where the results of SoftMatch are comparable to previous methods. Notably, FlexMatch exhibits a performance drop compared to FixMatch on SVHN, since it enrolls too many erroneous pseudo-labels at the beginning of the training that prohibits learning afterward. In contrast, SoftMatch surpasses FixMatch by 1.48% on SVHN with 40 labels, demonstrating its superiority for better utilization of the pseudo-labels. On more realistic datasets, CIFAR-100 with 400 labels, STL-10 with 40 labels, and ImageNet with 10% labels, SoftMatch exceeds FlexMatch by a margin of 7.73%, 2.84%, and 1.33%, respectively. SoftMatch shows the comparable results to FlexMatch on CIFAR-100 with 2,500 and 10,000 labels, whereas ReMixMatch [BCC+20] demonstrates the best results due to the Mixup [ZCDLP18] and Rotation loss.

5.4.2 Long-Tailed Image Classification

Setup. We evaluate SoftMatch on a more realistic and challenging setting of imbalanced SSL [KHP+20a, WSM+21, LSK21, FDS22], where both the labeled and the unlabeled data exhibit long-tailed distributions. Following [FDS22], the imbalance ratio γ ranges from 50 to 150 and 20 to 100 for CIFAR-10-LT and CIFAR-100-LT, respectively. Here, γ is used to exponentially decrease the number of samples from class 0 to class C [FDS22]. We compare SoftMatch with two strong baselines: FixMatch [SBL+20] and FlexMatch [ZWH+21]. All experiments use the same WRN-28-2 [ZK16b] as the backbone and the same set of common hyper-parameters. Each experiment is repeated five times with different data splits, and we report the average test accuracy and the standard deviation.

[1] All experiments in Section 5.4.1, Section 5.4.2, and Section 5.4.5 are conducted with TorchSSL [ZWH+21] and Section 5.4.3 are conducted with USB [WCF+22a] since it only supports NLP tasks back then. More recent results of SoftMatch are included in USB along its updates, refer https://github.com/Hhhhhhao/SoftMatch for details.

Dataset	CIFAR-10			CIFAR-100			SVHN		STL-10	
# Label	40	250	4,000	400	2,500	10,000	40	1,000	40	1,000
PseudoLabel	74.61±0.26	46.49±2.20	15.08±0.19	87.45±0.85	57.74±0.28	36.55±0.24	64.61±5.60	9.40±0.32	74.68±0.99	32.64±0.71
MeanTeacher	70.09±1.60	37.46±3.30	8.10±0.21	81.11±1.44	45.17±1.06	31.75±0.23	36.09±3.98	3.27±0.05	71.72±1.45	33.90±1.37
MixMatch	36.19±6.48	13.63±0.59	6.66±0.26	67.59±0.66	39.76±0.48	27.78±0.29	30.60±8.39	3.69±0.37	54.93±0.96	21.70±0.68
ReMixMatch	9.88±1.03	6.30±0.05	4.84±0.01	42.75±1.05	**26.03±0.35**	**20.02±0.27**	24.04±9.13	5.16±0.31	32.12±6.24	6.74±0.14
UDA	10.62±3.75	5.16±0.06	4.29±0.07	46.39±1.59	27.73±0.21	22.49±0.23	5.12±4.27	**1.89±0.01**	37.42±8.44	6.64±0.17
FixMatch	7.47±0.28	4.86±0.05	4.21±0.08	46.42±0.82	28.03±0.16	22.20±0.12	3.81±1.18	1.96±0.03	35.97±4.14	6.25±0.33
Influence	-	5.05±0.12*	4.35±0.06*	-	-	-	2.63±0.23*	2.34±0.15*	-	-
FlexMatch	4.97±0.06	4.98±0.09	4.19±0.01	39.94±1.62	26.49±0.20	21.90±0.15	8.19±3.20	6.72±0.30	29.15±4.16	5.77±0.18
SoftMatch	**4.91±0.12**	**4.82±0.09**	**4.04±0.02**	**37.10±0.77**	26.66±0.25	22.03±0.03	**2.33±0.25**	2.01±0.01	**21.42±3.48**	**5.73±0.24**

Table 5.2: Top-1 error rate (%) on CIFAR-10, CIFAR-100, STL-10, and SVHN of 3 different random seeds. Numbers with $*$ are taken from the original papers. The best number is in bold.

# Label	100k	400k
FixMatch	43.66	32.28
FlexMatch	41.85	31.31
SoftMatch	**40.52**	**29.49**

Table 5.3: Top1 error rate (%) on ImageNet. The best number is in bold.

Dataset	CIFAR-10-LT			CIFAR-100-LT		
Imbalance γ	50	100	150	20	50	100
FixMatch	18.46±0.30	25.11±1.20	29.62±0.88	50.42±0.78	57.89±0.33	62.40±0.48
FlexMatch	18.13±0.19	25.51±0.92	29.80±0.36	49.11±0.60	57.20±0.39	62.70±0.47
SoftMatch	**16.55±0.29**	**22.93±0.37**	**27.40±0.46**	**48.09±0.55**	**56.24±0.51**	**61.08±0.81**

Table 5.4: Top1 error rate (%) on CIFAR-10-LT and CIFAR-100-LT of 5 different random seeds. The best number is in bold.

Results. As is shown in Table 5.4, SoftMatch achieves the best test error rate across all long-tailed settings. The performance improvement over the previous state-of-the-art is still significant even at large imbalance ratios. For example, SoftMatch outperforms the second-best by 2.4% at $\gamma = 150$ on CIFAR-10-LT, which suggests the superior robustness of our method against data imbalance.

Discussion. Here we study the design choice of uniform alignment as it plays a key role in SoftMatch's performance on imbalanced SSL. We conduct experiments with different target distributions for alignment. Specifically, the default uniform target distribution $\mathbf{u}(C)$ can be replaced by ground-truth class distribution or the empirical class distribution estimated by seen labeled data during training. The results in Section 5.4.5 show a clear advantage of using uniform distribution. Uniform target distribution enforces the class marginal to become uniform, which has a strong regularization effect of balancing the head and tail classes in imbalanced classification settings.

5.4.3 Text Classification

Setup. In addition to image classification tasks, we further evaluate SoftMatch on text topic classification tasks of AG News and DBpedia, and sentiment tasks of IMDb, Amazon-5, and Yelp-5 [MDP+11, ZZL15]. We split a validation set from the training data to evaluate the algorithms. For Amazon-5 and Yelp-5, we randomly sample 50,000 samples per class from the training data to reduce the training time. We fine-tune the pre-trained BERT-Base [DCLT18a] model for all datasets using UDA [XDH+20b], FixMatch [SBL+20], FlexMatch [ZWH+21], and SoftMatch. We use AdamW [KB15, LH17] optimizer with an initial learning rate of $1e-5$ and the same cosine scheduler as image classification tasks. All algorithms are trained for a total iteration of 2^{18}. The fine-tuned model is directly used for evaluation rather than the EMA version. To reduce the GPU memory usage, we set both B_L and B_U to 16. Other algorithmic

Datasets	AG News		DBpedia		IMDb	Amazon-5	Yelp-5
# Labels	40	200	70	280	100	1000	1000
UDA	16.83±1.68	14.34±1.9	4.11±1.44	6.93±3.85	18.33±0.61	50.29±4.6	47.49±6.83
FixMatch	17.10±3.13	11.24±1.43	2.18±0.92	1.42±0.18	7.59±0.28	42.70±0.53	39.56±0.7
FlexMatch	15.49±1.97	10.95±0.56	2.69±0.34	1.69±0.02	7.80±0.23	42.34±0.62	**39.01±0.17**
SoftMatch	**12.68±0.23**	**10.41±0.13**	**1.68±0.34**	**1.27±0.1**	**7.48±0.12**	**42.14±0.92**	39.31±0.45

Table 5.5: Top1 error rate (%) on text datasets of 3 different random seeds. Best numbers are in bold.

hyper-parameters stay the same as image classification tasks.

Results. The results on text datasets are shown in Section 5.4.3. SoftMatch consistently outperforms other methods, especially on the topic classifications tasks. For instance, SoftMatch achieves an error rate of 12.68% on AG news with only 40 labels and 1.68% on DBpedia with 70 labels, surpassing the second best by a margin of 2.81% and 0.5% respectively. On sentiment tasks, SoftMatch also shows the best results on Amazon-5 and IMDb, and comparable results to its counterpart on Yelp-5.

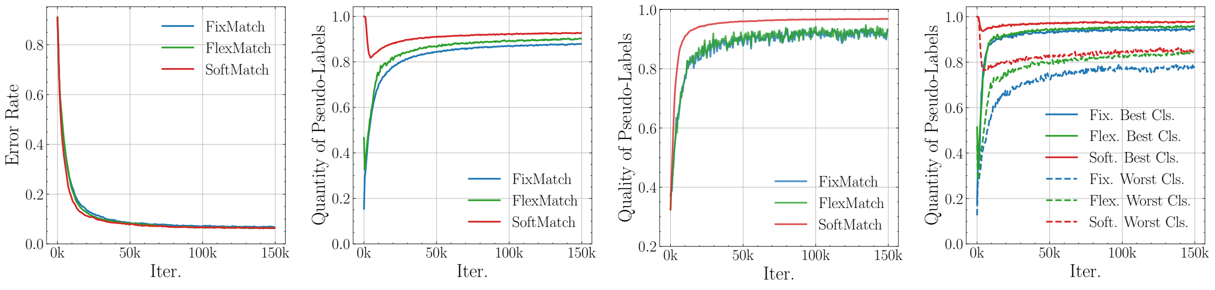
5.4.4 Qualitative Analysis

In this section, we provide a qualitative comparison on CIFAR-10 with 250 labels of FixMatch [SBL+20], FlexMatch [ZWH+21], and SoftMatch from different aspects, as shown in Fig. 5.2. We compute the error rate, the quantity, and the quality of pseudo-labels to analyze the proposed method, using the ground truth of unlabeled data that is unseen during training.

SoftMatch utilizes the unlabeled data better. From Section 5.4.4 and Section 5.4.4, one can observe that SoftMatch obtains highest quantity and quality of pseudo-labels across the training. Larger error with more fluctuation is present in quality of FixMatch and FlexMatch due to the nature of confidence thresholding, where significantly more wrong pseudo-labels are enrolled into training, leading to larger variance in quality and thus unstable training. While attaining a high quality, SoftMatch also substantially improves the unlabeled data utilization ratio, i.e., the quantity, as shown in Section 5.4.4, demonstrating the design of truncated Gaussian function could address the quantity-quality trade-off of the pseudo-labels. We also present the quality of the best and worst learned classes, as shown in Section 5.4.4, where both retain the highest along training in SoftMatch. The well-solved quantity-quality trade-off allows **SoftMatch achieves better performance on convergence and error rate**, especially for the first 50k iterations, as in Section 5.4.4.

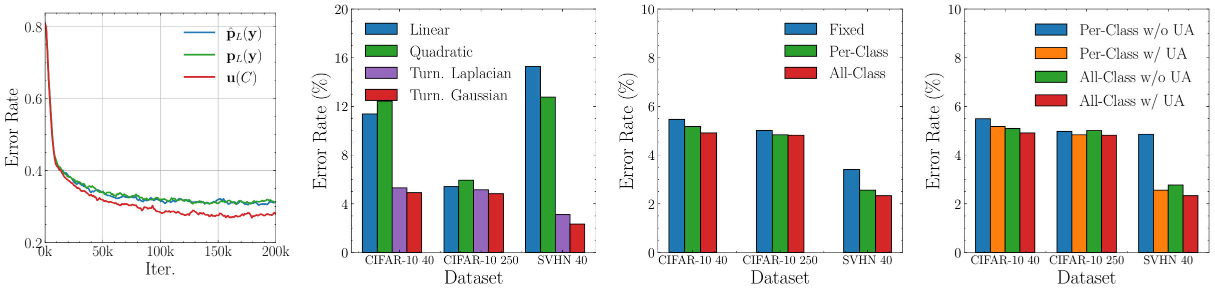
5.4.5 Ablation Study

Sample Weighting Functions. We validate different instantiations of $\lambda(\mathbf{p})$ to verify the effectiveness of the truncated Gaussian assumption on PMF $\lambda(\bar{\mathbf{p}})$, as shown in Section 5.4.5. Both linear function and Quadratic function fail to generalize and present large performance gap between Gaussian due to the naive assumption on PMF as discussed before. Truncated Laplacian assumption also works well on different settings, but truncated Gaussian demonstrates the most robust performance.



(a) Eval. Error (b) Quantity (c) Quality (d) Cls. Quality

Figure 5.2: Qualitative analysis of FixMatch, FlexMatch, and SoftMatch on CIFAR-10 with 250 labels. (a) Evaluation error; (b) Quantity of Pseudo-Labels; (c) Quality of Pseudo-Labels; (d) Quality of Pseudo-Labels from the best and worst learned class. Quality is computed according to the underlying ground truth labels. SoftMatch achieves significantly better performance.



(a) L.T. UA (b) Weight. Func. (c) Gau. Param. (d) UA

Figure 5.3: Ablation study of SoftMatch. (a) Target distributions for Uniform Alignment (UA) on long-tailed setting; (b) Error rate of different sample functions; (c) Error rate of different Gaussian parameter estimation, with UA enabled; (d) Ablation on UA with Gaussian parameter estimation;

Gaussian Parameter Estimation. SoftMatch estimates the Gaussian parameters μ and σ^2 directly from the confidence generated from all unlabeled data along the training. Here we compare it (*All-Class*) with two alternatives: (1) *Fixed*: which uses pre-defined μ and σ^2 of 0.95 and 0.01. (2) *Per-Class*: where a Gaussian for each class instead of a global Gaussian weighting function. As shown in Section 5.4.5, the inferior performance of *Fixed* justifies the importance of adaptive weight adjustment in SoftMatch. Moreover, *Per-Class* achieves comparable performance with SoftMatch at 250 labels, but significantly higher error rate at 40 labels. This is because an accurate parameter estimation requires many predictions for each class, which is not available for *Per-Class*.

Uniform Alignment on Gaussian. To verify the impact of UA, we compare the performance of SoftMatch with and without UA, denoted as all-class with UA and all-class without UA in Section 5.4.5. Since the per-class estimation standalone can also be viewed as a way to achieve fair class utilization [ZWH$^+$21], we also include it in comparison. Removing UA from SoftMatch has a slight performance drop. Besides, per-class estimation produces significantly inferior results on SVHN.

5.5 Related Work

Pseudo-labeling [Lee13] generates artificial labels for unlabeled data and trains the model in a self-training manner. Consistency regularization [ST17] is proposed to achieve the goal of producing consistent predictions for similar data points. A variety of works focus on improving the pseudo-labeling and consistency regularization from different aspects, such as loss weighting [ST17, TV17b, ITAC19b, RYS20], data augmentation [GB04, SJT16b, MMKI18a, BCG+19, BCC+20, XDH+20b, CZSL20, SJT16b], label allocation [TBV21], feature consistency [LXH21, ZYH+22, FKS21b], and confidence thresholding [SBL+20, ZWH+21, XSY+21].

Loss weight ramp-up strategy is proposed to balance the learning on labeled and unlabeled data. [ST17, TV17b, BCG+19, BCC+20]. By progressively increasing the loss weight for the unlabeled data, which prevents the model involving too much ambiguous unlabeled data at the early stage of training, the model therefore learns in a curriculum fashion. Per-sample loss weight is utilized to better exploit the unlabeled data [ITAC19b, RYS20]. The previous work "Influence" shares a similar goal with us, which aims to calculate the loss weight for each sample but for the motivation that not all unlabeled data are equal [RYS20]. SAW [LWG+22] utilizes effective weights [CJL+19] to overcome the class-imbalanced issues in SSL. Modeling of loss weight has also been explored in semi-supervised segmentation [HWH+21]. De-biased self-training [CJW+22, WWLY22] study the data bias and training bias brought by involving pseudo-labels into training, which is similar exploration of quantity and quality in SoftMatch. [KMK+22] proposed to use a small network to predict the loss weight, which is orthogonal to our work.

Confidence thresholding methods [SBL+20, XDH+20b, ZWH+21, XSY+21] adopt a threshold to enroll the unlabeled samples with high confidence into training. FixMatch [SBL+20] uses a fixed threshold to select pseudo-labels with high quality, which limits the data utilization ratio and leads to imbalanced pseudo-label distribution. Dash [XSY+21] gradually increases the threshold during training to improve the utilization of unlabeled data. FlexMatch [ZWH+21] designs class-wise thresholds and lowers the thresholds for classes that are more difficult to learn, which alleviates class imbalance.

5.6 Conclusion

In this chapter, we revisit the quantity-quality trade-off of pseudo-labeling and identify the core reason behind this trade-off from a unified sample weighting. We propose SoftMatch with truncated Gaussian weighting function and Uniform Alignment that overcomes the trade-off, yielding both high quantity and quality of pseudo-labels during training. Extensive experiments demonstrate the effectiveness of our method on various tasks. We hope more works can be inspired in this direction, such as designing better weighting functions that can discriminate correct pseudo-labels better.

II

Realistic Semi-Supervised Learning

The previous part of the thesis focuses on standard semi-supervised learning (SSL), where both labeled and unlabeled data are clean and well-balanced. In this part, we explore realistic SSL and aim to bridge the gap between conventional SSL methodologies and their applicability in real-world scenarios by addressing both data and evaluation realism. Specifically,

in Chapter 6, we examine imbalanced SSL, where both labeled and unlabeled data follow a long-tailed distribution. To address this, we introduce a novel co-learning framework that decouples representation learning from classifier learning, while coupling them effectively through a shared encoder and pseudo-label generation. Additionally, we propose a Tail-class Feature Enhancement (TFE) method that enriches tail-class data diversity by leveraging unlabeled data, resulting in more robust classifiers. Together, these innovations set a new state-of-the-art on multiple imbalanced SSL benchmarks across a range of evaluation settings.

In Chapter 7, we study another realistic setting of open-set SSL, where the unlabeled dataset includes outliers that belong to classes not present in the labeled dataset. Here, the objective is to accurately classify inliers while distinguishing outliers. To achieve this, we develop a simple but strong baseline that utilizes outliers to enhance inlier classification without compromising outlier detection. Despite its simplicity, this approach achieves significant improvements in both inlier classification and out-of-distribution (OOD) detection.

In Chapter 8, we tackle the limitations of current SSL benchmarks by constructing a Unified SSL Benchmark (USB) for classification. Our benchmark encompasses a diverse range of tasks across domains and evaluates numerous SSL approaches under the pretraining-finetune paradigm, which is more practical than traditional training from scratch. Furthermore, we release an open-source, modular, and extensible codebase to support the continued advancement of SSL research.

6 Co-Learning of Representation and Classifier for Imbalanced Semi-Supervised Learning

Contents

In this chapter, we deviate from the standard semi-supervised learning (SSL) setting where unlabeled data are well-balanced. We study the more realistic setting of class-imbalanced data, called imbalanced SSL, which is largely underexplored, and standard SSL tends to underperform. We propose a novel co-learning framework (CoSSL), which decouples representation and classifier learning while coupling them closely. To handle the data imbalance, we devise Tail-class Feature Enhancement (TFE) for classifier learning. Furthermore, the current evaluation protocol for imbalanced SSL focuses only on balanced test sets, which has limited practicality in real-world scenarios. Therefore, we further conduct a comprehensive evaluation under various shifted test distributions. In experiments, we show that our approach outperforms other methods over a large range of shifted distributions, achieving state-of-the-art performance on benchmark datasets ranging from CIFAR-10, CIFAR-100, ImageNet, to Food-101.

This chapter is based on [FDS22]. Yue Fan was the lead author of this paper and conducted all the experiments and wrote most parts of the paper.

6.1 Introduction

Imbalanced data distributions are ubiquitous, and pose great challenges for standard deep learning methods. Many approaches have been proposed for long-tailed recognition, where the number of (labeled) examples exhibits a long-tailed distribution with heavy class imbalance [LMZ$^+$19, GZH$^+$16, EVGW$^+$10, LMB$^+$14, KZG$^+$17, VHP17]. While semi-supervised learning (SSL) in the class-balanced setting has shown great promise, in this chapter we are interested in the challenging and realistic setting of *imbalanced SSL* where both the labeled and the unlabeled data are class-imbalanced, as shown in Fig. 6.1.

Despite a few pioneer works [KHP$^+$20b, WSM$^+$21], existing solutions from long-tailed recognition and SSL do not generalize well to this setting. On the one hand, long-tailed recognition [CBHK02, HM13, HG09, HLLT16, BMM18] is not designed to utilize unlabeled

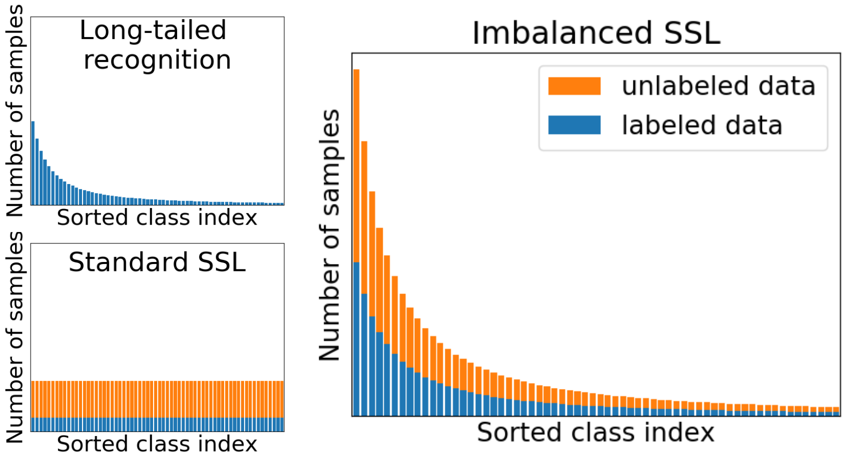


Figure 6.1: Conventional recognition tasks focus on constrained settings: long-tailed recognition does not involve unlabeled data; semi-supervised learning (SSL) assumes class-balanced distributions for both labeled and unlabeled data. In this work, we aim at *imbalanced SSL*, where the training data is partially annotated, and both labeled and unlabeled data are not manually balanced. This setting is more general and poses great challenges to existing algorithms. A robust learning algorithm should still be able to learn a good classifier under this setting.

data despite being good at handling data imbalance. Semi-supervised learning (SSL) [RBH$^+$15a, SJT16a, BAP14a, Scu65, Nes83, Lee13, BCG$^+$19, BCC$^+$20, SBL$^+$20], on the other hand, can effectively leverage unlabeled data but can not address data imbalance. In certain cases, standard SSL methods trained with imbalanced unlabeled datasets can lead to even worse results than a simple re-balancing method without using any unlabeled data [KHP$^+$20b], which counters the promise of SSL.

In this chapter, we address the imbalanced SSL problem by leveraging strong SSL algorithms [BCG$^+$19, BCC$^+$20, SBL$^+$20, XDH$^+$19] and recent success of decoupling representation and classifier learning from long-tailed recognition [KXR$^+$20]. To this end, we propose CoSSL, a novel co-learning framework for imbalanced SSL, which closely couples representation and classifier while the training of them is decoupled. As shown in Fig. 6.2, CoSSL consists of three modules: semi-supervised representation learning, classifier learning, and pseudo-label generation. In our co-learning framework, the representation learning module and the classifier learning module are trained separately without the gradient exchange. Nonetheless, the two modules in CoSSL are still connected via a shared encoder and pseudo-label generation. It can then bootstrap itself by exchanging information between the two modules: 1) a shared encoder from the representation learning is passed to classifier training for feature extraction; and 2) the enhanced classifier is used to generate better pseudo-labels for the representation learning. We show the superiority of our co-learning framework empirically, outperforming previous state-of-the-art methods by a large margin, especially in the case of severe imbalance. Moreover, we propose Tail-class Feature Enhancement (TFE) for improved classifier learning for imbalanced SSL, which utilizes unlabeled data as a source of augmentation to enhance the data diversity of tail classes, leading to a more robust classifier.

Furthermore, the standard evaluation protocol of long-tailed recognition and SSL normally assumes that the test data are from a uniform class distribution [BCG$^+$19, SBL$^+$20, BCC$^+$20, CWG$^+$19, KXR$^+$20, MJR$^+$21, THZ20, WLK$^+$20, LWK$^+$20]. However, this is insufficient to reflect the diversity of real-world applications, where users may have different needs. It is strongly desired that the trained model can perform well over a large range of varying distributions, including those that are radically different from the training distribution. Therefore, in this chapter, we adopt the shifted evaluation from [HHC+21], where the test data are from variously shifted class distributions. We further distinguish between unknown shifted evaluation and known shifted evaluation, depending on whether test distribution is known a *priori* during training. This evaluation protocol can be used for long-tailed recognition as well.

Our contributions are: (1) We propose a novel co-learning framework CoSSL for imbalanced SSL, which decouples representation and classifier learning while coupling them closely via a shared encoder and pseudo-label generation. (2) We devise a novel Tail-class Feature Enhancement (TFE) method to increase the data diversity of tail classes by utilizing unlabeled data, leading to more robust classifiers. (3) We propose new evaluation criteria for imbalanced SSL and conduct a comprehensive evaluation. CoSSL achieves new state-of-the-art results on multiple imbalanced SSL benchmarks across a wide range of evaluation settings.

6.2 Related work

Semi-supervised learning. Many efforts have been made in various directions in SSL. For example, many recent powerful methods [RBH+15a, SJT16a, BAP14a] are based on consistency regularization, where the idea is that the model should output consistent predictions for perturbed versions of the same input. Another spectrum of popular approaches is pseudo-labeling [Scu65, Nes83, Lee13] or self-training [RHS05b], where the model is trained with artificial labels. Furthermore, there are many excellent works around generative models [KMRW14, Ode16a, DGF16a] and graph-based methods [LZL+18, LWHL19, BDLR06, Joa03]. A more comprehensive introduction of SSL methods is available in [CSZ09, Zhu05a, ZG09a]. However, none of the aforementioned works have studied SSL in the class-imbalanced setting, in which the standard SSL methods fail to generalize well.

Long-tailed recognition. Research on class-imbalanced supervised learning has attracted increasing attention. In particular, many recent efforts have been made to improve the performance under imbalanced data by decoupling the learning of representation and classifier head [KXR+20, MJR+21, THZ20, WLK+20, LWK+20]. In the two-stage framework from [KXR+20], an instance-balanced sampling scheme was first used for representation learning. In the second stage, the classifier head is simply retrained by a class-balanced sampling. We found that this scheme is also very competitive for imbalanced SSL in our preliminary experiments. In contrast to this line of works, our co-learning framework focuses on imbalanced SSL and largely simplifies the training pipeline compared to the two-stage framework [KXR+20]. The joint training enables interaction between representation learning and classifier learning, which brings additional benefits to the final performance. In contrast to BBN [ZCWC20], which has a single loss for two branches and learns the classifier and the representation jointly, CoSSL independently learns classifier and representation with different losses while still connecting them via EMA and pseudo-labeling. Evaluation under shifted distributions was also proposed by [HHC+21], however, we take a step further and consider settings where the test-time distribution is given or not as prior knowledge during the training.

Imbalanced semi-supervised learning. While SSL has been extensively studied, the setting of class-imbalanced semi-supervised is rather under-explored. Most successful methods from standard SSL do not generalize well to this more realistic scenario without addressing the data imbalance explicitly. Hyun et al. [HJK20] proposed a suppressed consistency loss to suppress the loss on minority classes. Kim et al. [KHP+20b] proposed Distribution Aligning Refinery (DARP) to refine raw pseudo-labels via convex optimization. Wei et al. [WSM+21] found that the raw SSL methods usually have high recall and low precision for head classes while the reverse is true for the tail classes and further proposed a reverse sampling method for unlabeled data based on that. BiS [HKY+21] implements a novel sampler which is helpful for the encoder in the beginning but classifier in the end, however, CoSSL trains the encoder and the classifier independently with different samplers and losses. In contrast to DASO [OKK21a], where pseudo-labels are refined by two complementary classifiers, CoSSL uses a balanced classifier, which is trained by TFE with unlabeled data, to generate pseudo-labels. Another concurrent work ABC [LSK21] introduces an auxiliary classifier which is trained in a balanced way to help the model while sharing the same backbone. CoSSL differs from ABC [LSK21] in: (1) the training of representation and classifier is decoupled; (2) the classifier and the encoder are actively connected to help each other via pseudo-label generation; (3) enhancing tail classes with unlabeled data.

6.3 CoSSL: Co-learning for imbalanced SSL

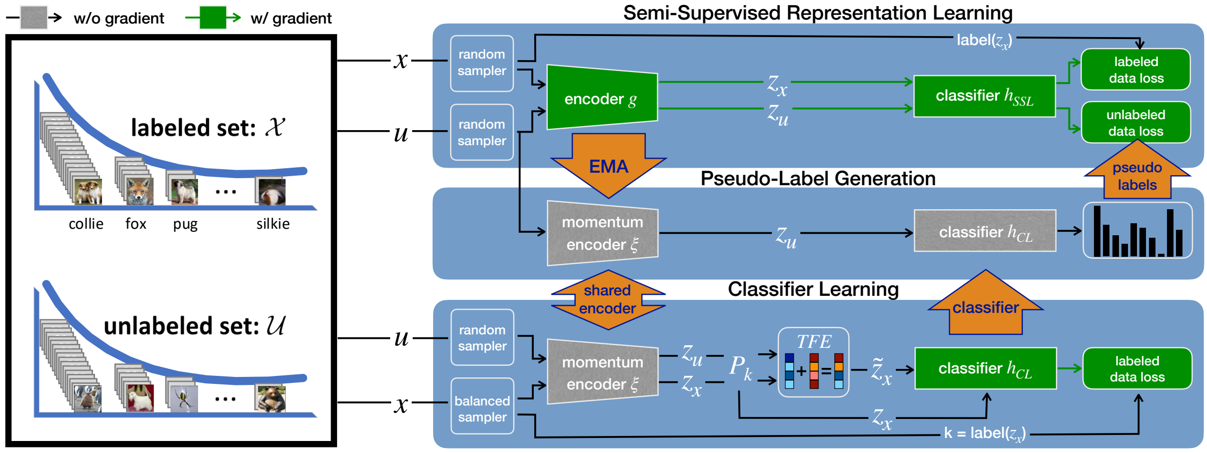


Figure 6.2: Our co-learning framework CoSSL decouples the training of representation and classifier while coupling them in a non-gradient manner. CoSSL consists of three modules: a semi-supervised representation learning module, a balanced classifier learning module, and a carefully designed pseudo-label generation module. The representation module provides a momentum encoder for feature extraction in the other two modules, and the classifier module produces a balanced classifier using our novel Tail-class Feature Enhancement (TFE). Then, pseudo-label module generates pseudo-labels for the representation module using the momentum encoder and the balanced classifier. The interplay between these modules enhances each other, leading to both a more powerful representation and a more balanced classifier. Additionally, our framework is flexible as it can accommodate any standard SSL methods and classifier learning methods.

In this section, we first present the problem setup of imbalanced semi-supervised learning (SSL). Based on this, we introduce CoSSL, a flexible co-learning framework for imbalanced SSL in Section 6.3.1.

Problem setup and notations: For a K-class classification problem, there is a labeled set $\mathcal{X} = \{(\mathbf{x}_n, y_n) : n \in (1, ..., N)\}$ and an unlabeled set $\mathcal{U} = \{\mathbf{u}_m : m \in (1, ..., M)\}$, where

$\mathbf{x}_n, \mathbf{u}_m \in \mathbb{R}^d$ are training examples and $y_n \in \{1, ..., K\}$ are class labels for labeled examples. N_k and M_k denote the numbers of labeled and unlabeled examples in class k, respectively, i.e., $\sum_{k=1}^{K} N_k = N$ and $\sum_{k=1}^{K} M_k = M$. Without loss of generality, we assume the classes are sorted by the number of training samples in descending order, i.e., $N_1 \geq N_2 \geq ... \geq N_k$. The goal of imbalanced SSL is to train a classifier $f : \mathbb{R}^d \rightarrow \{1, ..., K\}$ that generalizes well over a large range of varying test data distributions.

6.3.1 Co-learning representation and classifier

The two-stage framework [KXR+20, MJR+21, THZ20, WLK+20, LWK+20] from long-tailed recognition is quite successful for supervised learning with imbalanced data. It decouples representation and classifier by retraining a classifier after the representation learning. While classifier re-training (cRT) [KXR+20] is out-of-the-box a strong baseline, as we will see in the experimental section 6.4.1, the method has its own limitations when applied to imbalanced SSL: (1) unlabeled data is not utilized during cRT; (2) the two-stage training scheme makes it impossible to refine the pseudo-labels, which in turn limits the quality of feature representation learning.

This motivates us to propose CoSSL, a co-learning framework for imbalanced SSL with a mutual interplay between representations and classifier learning. While decoupling the training of the representations and the classifier, we couple them *without* gradient propagation, so that the final model leverages from the interactions between all the co-modules in our framework. As illustrated in Fig. 6.2, CoSSL consists of three modules: a semi-supervised representation learning module, a classifier learning module, and a pseudo-label generation module. The feature encoder from the representation learning module is shared with the classifier module to learn a better classifier, and the improved classifier is used to generate better pseudo-labels for the representation learning module to further improve the feature encoder. This joint framework largely simplifies the training pipeline compared to the two-stage framework and enables interaction between the representation learning and the classifier learning, which brings additional benefits to the final performance (see Section 6.4.5 for ablation).

Semi-supervised representation learning: The goal of the semi-supervised representation learning module is to obtain a strong feature encoder by exploring unlabeled data. Thanks to the flexibility of our framework, we can use and evaluate a variety of SSL methods [BCG+19, BCC+20, SBL+20]. Given a batch of unlabeled data sampled from the random sampler, we first pass the unlabeled data to the pseudo-label generation module. Then, the unlabeled data loss is computed using the generated pseudo-labels. Meanwhile, a batch of labeled data is sampled by the random sampler, and the labeled data loss is computed. The resulting encoder is accumulated into a momentum encoder and further passed to the classifier module for feature extraction to enhance the classifier training as shown in Fig. 6.2.

Classifier learning with Tail-class Feature Enhancement: Inspired by the success of cRT, we train a separate classifier in the classifier learning module and aim to further improve it by using unlabeled data. To this end, we propose Tail-class Feature Enhancement (TFE) that exploits unlabeled data by blending unlabeled data features with labeled data features while preserving the label of the labeled sample. Specifically, at each training step, we train the classifier using blended features between labeled and unlabeled data with labels from labeled data. We deploy a class-balanced sampler and a random sampler to sample a labeled example $(\mathbf{x}_i, y_i)$ and an unlabeled example $\mathbf{u}_j$. Then the new fused feature for classifier training is generated by:

$$\tilde{\mathbf{z}} = \lambda \xi(\mathbf{x}_i) + (1 - \lambda)\xi(\mathbf{u}_j) \quad \text{and} \quad \tilde{y} = y_i \tag{6.1}$$

Algorithm 4 Classifier training with **T**ail-class **F**eature **E**nhancement

```
Input: Labeled set $\mathcal{X}$, unlabeled set $\mathcal{U}$, feature encoder $\xi$, parameter $\mu$, and batch size $B$
for $b = 1$ to $B$ do
    // Sample labeled and unlabeled examples
    $\mathbf{x}_i, y_i \sim$ Class-balanced sampler($\mathcal{X}$)
    $\mathbf{u}_j \sim$ Random sampler($\mathcal{U}$)
    $P_{y_i} = \frac{N_1 - N_{y_i}}{N_1}$    // Compute the blend probability
    if Uniform$(0, 1) \leq P_{y_i}$ then
        // Generate features by feature blending
        $\lambda \sim$ Uniform$(\mu, 1)$
        $\tilde{\mathbf{z}}_b = \lambda\xi(\mathbf{x}_i) + (1 - \lambda)\xi(\mathbf{u}_j)$
        $\tilde{y}_b = y_i$
    else
        // Use features of labeled data directly
        $\tilde{\mathbf{z}}_b = \xi(\mathbf{x}_i)$
        $\tilde{y}_b = y_i$
    end if
end for
return $\{\tilde{\mathbf{z}}, \tilde{y}\}$    // Features for classifier training
```

where ξ is the momentum encoder from the representation learning module and the fusion factor λ is sampled from a uniform distribution over the interval $[\mu, 1]$. We consider samples of λ with a value of at least μ to ensure the validity of the label y_i for the synthesized sample.

To enhance the data diversity of tail classes, we train the classifier using different portions of fused examples in a stochastic way. The feature blending is applied with a blend probability that depends on the number of data for each class so that the more labeled data a class has, the less fused data is synthesized for classifier learning. Formally, given a labeled example from class k, we apply feature blending with probability P_k defined as:

$$P_k = \frac{N_1 - N_k}{N_1} \tag{6.2}$$

where N_k is the number of examples from the k-th class, and N_1 is the number of examples of the first class (with the most labeled data). Such a class-dependent blend probability encourages more augmented data from feature blending for tail classes, thus, improving the data diversity of tail classes. For instance, there is no fused data for the first class, which has the most labeled data, since $P_1 = 0$. For a tail class with only 5% samples of the first class, the blend probability will be as high as 95%. Note, that since fused data share the same label with the labeled data, the class distribution is uniform during cRT as the labeled set is sampled using a class-balanced sampler. Pseudo-code for processing a batch of labeled and unlabeled examples can be found in Alg. 4.

Pseudo-label generation: As standard SSL methods suffer from biased pseudo-labels under data imbalance [KHP⁺20b, WSM⁺21], we devise a pseudo-label generation module to generate high-quality pseudo-labels by combining the strengths of the representation learning module and the classifier learning module. Given a batch of unlabeled data, it first uses the momentum encoder ξ from the representation learning to extract features since the representations learned from instance-balanced sampling from SSL is the most generalizable [KXR⁺20]. Then the pseudo-labels are predicted using the classifier trained from TFE leveraging its robustness against data imbalance. Our pseudo-label generation module replaces the original pseudo-labeling part of the SSL algorithm in the representation learning module and enables the trained classifier to enhance representation learning. Note, that no gradient updates happen at this step.

Overall co-learning framework: The three aforementioned modules, while being decoupled, are closely coupled with each other in a non-gradient manner. CoSSL can then bootstrap itself by exchanging information between them: the representation learning module provides a momentum encoder for better feature extraction for training classifiers and pseudo-labeling. And the improved classifier, in turn, generates high-quality pseudo-labels to further enhance representation learning. Specifically, denote the overall network by f, which consists of a feature extractor network $g(\cdot)$ and a classifier head $h(\cdot)$. At training iteration t, the three modules operate successively as shown in Fig. 6.2. (1) For the classifier module, a batch of labeled data and unlabeled data from $\mathcal{X}$ and $\mathcal{U}$ are sampled using a class-balanced sampler and a random sampler, respectively. Then, the features are extracted by a momentum encoder $\xi(\cdot)$ of $g(\cdot)$, which is provided by the representation learning module. We update ξ by $\xi_t = m\xi_{t-1} + (1-m)g_t$ where $\xi_0 = g_0$ and $m \in [0, 1)$ is a momentum coefficient. Then, the classifier h is trained using our TFE with standard cross-entropy loss. (2) For the pseudo-label generation module, it encodes a new batch of unlabeled data with the same momentum encoder ξ_t and predicts the pseudo-labels using the classifier h from the classifier module. (3) The generated pseudo-labels are then fed into the representation module to compute the unlabeled data loss. Meanwhile, a new batch of labeled data is used in the representation module.

CoSSL fits particularly well for imbalanced SSL as the representation module and the classifier module, despite being decoupled, can enhance each other via pseudo-labeling and the momentum encoder, leading to both a more powerful representation and a more balanced classifier. We find empirically that coupling representation and classifier without explicit gradient propagation leads to a better performance than variants with it. (see Section 6.4.5). Moreover, our co-learning framework is very flexible as it can accommodate any standard SSL algorithm and classifier learning method, which makes it possible to benefit from the most advanced approaches.

6.4 Experimental evaluation

In this section, we conduct extensive experiments to evaluate the efficacy of our framework. In Section 6.4.1, 6.4.2, and 6.4.3, we compare our method with existing works and show that we achieve state-of-the-art performance for the commonly used uniform test evaluation. Section 6.4.4 evaluates different methods over a large range of imbalance settings, and we distinguish between two cases: the distributions are unknown or known a priori during training. A detailed analysis of our framework can be found in Section 6.4.5.

6.4.1 Main results on CIFAR-10 and CIFAR-100

Datasets. Following common practice [CJL+19, CWG+19], we employ CIFAR10-LT and CIFAR100-LT for imbalanced SSL by randomly selecting some training images for each class determined by a pre-defined imbalance ratio γ as the labeled and the unlabeled set. Specifically, we set $N_k = N_1 \cdot \gamma^{-\frac{k-1}{K-1}}$ for labeled data and $M_k = M_1 \cdot \gamma^{-\frac{k-1}{K-1}}$ for unlabeled data. For results in the main chapter, we use $N_1 = 1500$; $M_1 = 3000$ for CIFAR-10 and $N_1 = 150$; $M_1 = 300$ for CIFAR-100, respectively. Following [KHP+20b, WSM+21], we report results with imbalance ratio $\gamma = 50$, 100 and 150 for CIFAR10-LT and $\gamma = 20$, 50 and 100 for CIFAR100-LT. Therefore, the number of labeled samples for the least class is 10 and 1 for CIFAR-10 with $\gamma = 150$ and CIFAR-100 with $\gamma = 100$, respectively.

	CIFAR-10-LT		
	γ=50	γ=100	γ=150
vanilla	$65.2^*_{\pm 0.05}$	$58.8^*_{\pm 0.13}$	$55.6^*_{\pm 0.43}$
Long-tailed recognition methods			
w/ Re-sampling [Japoo]	$64.3^*_{\pm 0.48}$	$55.8^*_{\pm 0.47}$	$52.2^*_{\pm 0.05}$
w/ LDAM-DRW [CWG+19]	$68.9^*_{\pm 0.07}$	$62.8^*_{\pm 0.17}$	$57.9^*_{\pm 0.20}$
w/ cRT [KXR+20]	$67.8^*_{\pm 0.13}$	$63.2^*_{\pm 0.45}$	$59.3^*_{\pm 0.10}$
SSL methods			
MixMatch [BCG+19]	$73.2^*_{\pm 0.56}$	$64.8^*_{\pm 0.28}$	$62.5^*_{\pm 0.31}$
w/ DARP [KHP+20b]	$75.2^*_{\pm 0.47}$	$67.9^*_{\pm 0.14}$	$65.8^*_{\pm 0.52}$
w/ CReST+ [WSM+21]	$79.0^*_{\pm 0.26}$	$71.9^*_{\pm 0.33}$	$68.3^*_{\pm 0.57}$
w/ CoSSL	**80.31**$_{\pm 0.31}$	**76.4**$_{\pm 1.14}$	**73.5**$_{\pm 1.25}$
ReMixMatch [BCC+20]	$81.5^*_{\pm 0.26}$	$73.8^*_{\pm 0.38}$	$69.9^*_{\pm 0.47}$
w/ Re-sampling [Japoo]	$83.6_{\pm 0.54}$	$76.7_{\pm 0.24}$	$71.5_{\pm 0.64}$
w/ LDAM-DRW [CWG+19]	$85.9_{\pm 0.23}$	$80.5_{\pm 0.71}$	$76.1_{\pm 0.53}$
w/ DARP [KHP+20b]	$82.1^*_{\pm 0.14}$	$75.8^*_{\pm 0.09}$	$71.0^*_{\pm 0.27}$
w/ DARP + cRT [KHP+20b]	$87.3^*_{\pm 0.16}$	$83.5^*_{\pm 0.07}$	$79.7^*_{\pm 0.54}$
w/ CReST+ [WSM+21]	$83.7_{\pm 0.15}$	$78.8_{\pm 0.54}$	$75.2_{\pm 0.30}$
w/ CReST+ + LA [WSM+21]	$84.2_{\pm 0.11}$	$81.3_{\pm 0.34}$	$79.2_{\pm 0.31}$
w/ CoSSL	**87.7**$_{\pm 0.21}$	**84.1**$_{\pm 0.56}$	**81.3**$_{\pm 0.83}$
FixMatch [SBL+20]	$79.2^*_{\pm 0.33}$	$71.5^*_{\pm 0.72}$	$68.4^*_{\pm 0.15}$
w/ Re-sampling [Japoo]	$84.8_{\pm 0.21}$	$78.9_{\pm 0.63}$	$75.2_{\pm 0.45}$
w/ LDAM-DRW [CWG+19]	$80.0_{\pm 0.60}$	$73.1_{\pm 0.81}$	$69.1_{\pm 0.51}$
w/ DARP [KHP+20b]	$81.8^*_{\pm 0.24}$	$75.5^*_{\pm 0.04}$	$70.4^*_{\pm 0.25}$
w/ DARP + cRT[KHP+20b]	$85.8_{\pm 0.43}$	$82.4_{\pm 0.26}$	$79.6_{\pm 0.42}$
w/ CReST+ [WSM+21]	$83.9^*_{\pm 0.14}$	$77.4^*_{\pm 0.36}$	$72.8^*_{\pm 0.58}$
w/ CReST+ + LA [WSM+21]	$84.9_{\pm 0.02}$	$80.8_{\pm 0.20}$	$77.5_{\pm 0.74}$
w/ CoSSL	**86.8**$_{\pm 0.30}$	**83.2**$_{\pm 0.49}$	**80.3**$_{\pm 0.55}$

Table 6.1: Classification accuracy (%) on CIFAR-10-LT using a Wide ResNet-28-2 under the uniform test distribution of three different class-imbalance ratios γ. The numbers are averaged over 5 different folds. We use the same code base as [KHP+20b] for fair comparison following [OOR+18]. Numbers with * are taken from the original papers. The best number is in bold.

Setup. Following [KHP$^+$20b, CMGS10], we evaluate our method with MixMatch [BCG$^+$19], ReMixMatch [BCC$^+$20], and FixMatch [SBL$^+$20] under the same implementation (as recommended by [OOR$^+$18]) using Wide ResNet-28-2 [ZK16a] as the backbone. The hyper-parameter μ in Alg. 4 is set to 0.6 based on the ablation study in Section 6.4.5. We apply TFE module in the last 20% of iterations for faster training and better accuracy. As our implementation is based on the public codebase from [KHP$^+$20b], we use the same hyper-parameters as theirs. For example, all experiments are trained with batch size 64 using Adam optimizer [KB15] with a constant learning rate of 0.002 without any decay. We train all models for 500 epochs, each of which has 500 steps, resulting in a total number of 2.5×10^5 training iterations. For all experiments, we report the average test accuracy of the last 20 epochs following [OOR$^+$18]. For CReST+, we use the official TensorFlow implementation. As for data augmentation for TFE, we use the strong augmentation from [SBL$^+$20], which consists of RandAugment [CZSL20] and CutOut [DT17].

Results. Table 6.1 and Table 6.2 compare our method with various SSL algorithms and long-tailed recognition algorithms on CIFAR-10-LT and CIFAR-100-LT with various imbalance ratios γ. Our method achieves the best performance across all settings with significant margins over the previous state-of-the-art. Noticeably, our method is particularly good at larger imbalance ratios. For example, we outperform the second-best by an absolute accuracy of 7.5% on CIFAR-10-LT at imbalance ratio $\gamma = 150$ with FixMatch, which underlines the superiority of our method. Replacing MixMatch with ReMixMatch or FixMatch as the representation learning module can increase test accuracy on CIFAR-10-LT at imbalance ratio $\gamma = 150$ by 7.8% and 6.8%, respectively. On CIFAR-100-LT, we evaluate our method on top of ReMixMatch and FixMatch as they give the best performance on CIFAR-10-LT. Besides the best performance across settings, our method also improves performance for small imbalance ratios as well (4.5% higher than the second-best at imbalance ratio $\gamma = 20$ with ReMixMatch).

6.4.2 Main results on Small-ImageNet-127

Dataset. ImageNet127 is originally introduced in [HAE16] and used by [WSM$^+$21] for imbalanced SSL. It is a naturally imbalanced dataset with imbalance ratio $\gamma \approx 286$ by grouping the 1000 classes of ImageNet [DDS$^+$09] into 127 classes based on the WordNet hierarchy. Due to limited resources, we are not able to conduct experiments on ImageNet127 with the full resolution[2]. Instead, we propose a down-sampled version of ImageNet127 to test the effectiveness of our method on a large-scale dataset. Inspired by [CLH17], we down-sample the original images from ImageNet127 to smaller images of 32×32 or 64×64 pixels using the box method from Pillow library (different down-sampling techniques yield very similar performance as pointed out by [CLH17]). Following [WSM$^+$21], we randomly select 10% training samples as the labeled set. The test set is unchanged, and averaged class recall is used to achieve a balanced metric.

Setup and results. We evaluate our method using FixMatch [SBL$^+$20] with ResNet-50 [HZRS16a] due to its good performance on CIFAR. For all experiments, we train for a total number of 500 epochs. For CReST+, we train for 5 generations with 100 epoch per generation. The rest of hyper-parameters are the same as used in CIFAR-LT. As for data augmentation of TFE, we use random crop and horizontal flipping. Table 6.3 summarizes the results on

[0] Note that the results from [KHP$^+$20b] with $\gamma = 20$ are not used here because they were produced by $N_1 = 300, M_1 = 150$: https://github.com/bbuing9/DARP/blob/master/run.sh

[2] One run of vanilla FixMatch on ImageNet127 on a single NVIDIA Tesla V100 takes 10676.5 hours which is about 444 days.

	CIFAR-100-LT		
	γ=20	γ=50	γ=100
ReMixMatch [BCC+20]	51.6$_{\pm 0.43}$	44.2$_{\pm 0.59}$	39.3$_{\pm 0.43}$
w/ Re-sampling [Japoo]	50.0$_{\pm 0.56}$	42.9$_{\pm 0.95}$	37.8$_{\pm 0.46}$
w/ LDAM-DRW [CWG+19]	54.5$_{\pm 0.95}$	47.5$_{\pm 0.79}$	42.3$_{\pm 0.35}$
w/ DARP [KHP+20b]	51.9$_{\pm 0.35}$	44.7$_{\pm 0.66}$	39.8$_{\pm 0.53}$
w/ DARP + cRT [KHP+20b]	54.5$_{\pm 0.42}$	48.5$_{\pm 0.91}$	43.7$_{\pm 0.81}$
w/ CReST+ [WSM+21]	51.3$_{\pm 0.34}$	45.5$_{\pm 0.76}$	41.0$_{\pm 0.78}$
w/ CReST+ + LA [WSM+21]	51.9$_{\pm 0.60}$	46.6$_{\pm 1.14}$	41.7$_{\pm 0.69}$
w/ CoSSL	**55.8**$_{\pm 0.62}$	**48.9**$_{\pm 0.61}$	**44.1**$_{\pm 0.59}$
FixMatch [SBL+20]	49.6$_{\pm 0.78}$	42.1$_{\pm 0.33}$	37.6$_{\pm 0.48}$
w/ Re-sampling [Japoo]	49.9$_{\pm 0.76}$	43.2$_{\pm 0.54}$	38.2$_{\pm 0.60}$
w/ LDAM-DRW [CWG+19]	51.1$_{\pm 0.45}$	40.4$_{\pm 0.46}$	34.7$_{\pm 0.22}$
w/ DARP [KHP+20b]	50.8$_{\pm 0.77}$	43.1$_{\pm 0.54}$	38.3$_{\pm 0.47}$
w/ DARP + cRT [KHP+20b]	51.4$_{\pm 0.68}$	44.9$_{\pm 0.54}$	40.4$_{\pm 0.78}$
w/ CReST+ [WSM+21]	51.8$_{\pm 0.12}$	44.9$_{\pm 0.50}$	40.1$_{\pm 0.65}$
w/ CReST+ + LA [WSM+21]	52.9$_{\pm 0.07}$	47.3$_{\pm 0.17}$	42.7$_{\pm 0.70}$
w/ CoSSL	**53.9**$_{\pm 0.78}$	**47.6**$_{\pm 0.57}$	**43.0**$_{\pm 0.61}$

Table 6.2: Classification accuracy (%) on CIFAR-100-LT under the uniform test distribution of three different class-imbalance ratios γ. The numbers are averaged over 5 different folds. We reproduce all numbers using the same codebase from [KHP+20b] for a fair comparison[1]. The best number is in bold.

Small-ImageNet-127. CoSSL achieves the best and the second-best performance for image sizes 32 and 64, respectively.

	Small-ImageNet-127		Food-101-LT	
	32×32	64×64	$\gamma = 50$	$\gamma = 100$
FixMatch	29.7	42.3	42.6	35.3
w/ DARP [KHP+20b]	30.5	42.5	42.0	34.2
w/ DARP + cRT [KHP+20b]	39.7	51.0	41.5	34.4
w/ CReST+ [WSM+21]	32.5	44.7	43.8	31.2
w/ CReST+ + LA [WSM+21]	40.9	**55.9**	47.7	36.1
w/ CoSSL	**43.7**	53.8	**49.0**	**40.4**

Table 6.3: Averaged class recall (%) on Small-ImageNet-127 and Food-101. We test image size 32×32 and 64×64 for Small-ImageNet-127 and $\gamma = 50$ and $\gamma = 100$ for Food-101.

6.4.3 Main results on Food-101

Dataset. To evaluate the effectiveness of our method on high-resolution images, we use the fine-grained image classification dataset Food-101 [BGVG14]. The original dataset consists of 101 food categories, with 101,000 images. For each class, 250 manually reviewed test images are provided as well as 750 training images. All images were rescaled to have a maximum side length of 512 pixels. We construct Food-101-LT for imbalanced SSL using the same way as CIFAR-10-LT with imbalance ratio $\gamma = 50$ and 100.

Setup. We consider FixMatch [SBL+20] as the SSL algorithm due to its good performance. We train a ResNet-50 [HZRS16a] for 1,000 epochs of unlabeled dataset using a SGD optimizer with momentum 0.9. The learning rate is set to 0.04 without decay, with a linear warm-up for the first 5 epochs. We set the labeled batch size as 256 and the unlabeled batch size as 512. The EMA decay rate is 0.999. We use random crop and horizontal flipping for TFE.

Results. Table 6.3 shows the results on Food-101-LT. Compared to other methods, which give marginal improvements or, in some cases, even worse performance over the baseline, our method consistently improves the accuracy. We outperform the second-best by 1.3% and 4.3% at imbalance ratio $\gamma = 50$ and 100, respectively.

6.4.4 Evaluation at unknown and known shifted test distributions

As mentioned above, the standard evaluation under uniform test distribution is often limited in reflecting real-world scenarios. To this end, we conduct a more realistic evaluation by assessing different methods at shifted test distributions. Moreover, we argue that the test distribution can be given as prior knowledge in real-world applications in some cases. Thus, we distinguish two types of shifted evaluation: known test distributions in which the test distribution is given during training, and unknown test distributions in which this information is unknown. When the test distribution is known, the imbalanced SSL method should be able to accommodate the information for further improvement.

Test imbalance ratio	512	256	**150**	128	64	32	16	8	4	2	1	-2	-4	-8	-16	-32	-64	-128	-256	-512	Mean
	Unknown test-time imbalance ratio																				
Fix	94.83	93.95	93.13	92.87	91.24	89.11	86.62	82.90	78.92	73.58	67.83	61.83	55.41	49.50	44.46	40.37	36.88	33.89	30.95	29.04	66.36
Fix + PC	94.63	93.95	93.30	92.95	91.54	89.89	87.87	84.89	82.05	77.97	73.49	68.86	63.88	59.45	55.70	52.76	50.24	47.90	45.77	44.23	72.57
Fix + vanilla cRT	94.78	93.90	93.17	92.83	91.24	89.24	86.87	83.75	80.29	75.54	70.40	65.10	59.47	54.36	49.86	46.35	43.39	40.81	38.34	36.61	69.31
Fix + DARP	**95.14**	**94.46**	**93.73**	**93.50**	**92.18**	**90.12**	87.70	84.39	81.03	76.26	71.15	66.12	60.99	56.10	52.28	48.84	45.75	43.25	40.79	39.17	70.65
Fix + CReST+	94.18	93.39	92.74	92.45	91.05	89.04	86.70	83.52	80.20	76.05	71.75	67.28	62.76	58.73	55.68	52.89	50.47	48.49	46.61	45.54	71.98
Fix + CoSSL	91.73	91.13	90.90	90.60	89.85	89.07	**87.95**	**86.24**	**84.60**	**82.61**	**80.40**	**78.39**	**76.03**	**74.19**	**73.21**	**72.49**	**71.43**	**70.64**	**70.02**	**69.71**	**81.06**
	Known test-time imbalance ratio																				
Fix + PC	94.98	94.00	93.13	92.83	91.16	89.24	87.03	84.00	81.03	77.31	73.49	70.10	66.79	64.21	62.69	61.89	62.41	63.26	64.80	66.50	77.04
Fix + vanilla cRT	95.14	94.32	93.39	93.25	91.35	89.24	86.73	83.45	79.85	75.04	70.40	65.76	60.65	56.67	53.81	52.04	51.07	51.09	49.98	51.60	72.24
Fix + DARP + PC	**95.19**	**94.46**	**93.73**	**93.54**	**92.32**	**90.32**	**88.17**	**85.53**	83.00	79.96	76.82	74.33	72.05	70.88	70.37	70.53	70.98	71.39	72.19	73.07	80.94
Fix + CReST+ + PC	94.48	93.44	92.74	92.49	91.09	89.17	87.20	84.75	82.60	79.86	77.74	76.09	74.41	74.03	74.40	75.40	76.38	77.22	78.66	80.29	82.62
Fix + CoSSL + PC	92.83	91.59	90.90	90.31	89.22	87.93	86.42	85.01	**84.00**	**82.57**	**82.00**	**81.70**	**81.72**	**81.66**	**82.94**	**84.66**	**85.77**	**86.83**	**87.58**	**88.31**	**86.20**

Table 6.4: Classification accuracy (%) on CIFAR-10-LT with imbalance ratio $\gamma = 150$. We test different methods on top of FixMatch [SBL+20] for known and unknown shifted distributions. Post-compensation (PC) [HHC+21] is deployed to utilize the information of the known test distribution.

Inspired by [HHC+21], we construct shifted test sets with a wide range of imbalance ratios. When $\gamma > 0$, the number of test examples of class k is defined as $N_k = N_1 \cdot \gamma^{-\frac{k-1}{K-1}}$, where class 1 has the most test data. Similarly, $N_k = N_1 \cdot |\gamma|^{\frac{k-1}{K-1}}$ when $\gamma < 0$, where class 1 has the least test data, and, thus, test set is weighted in favor of tail classes. For unknown distributions, we train different methods and evaluate them directly over a family of shifted distributions. The mean accuracy is also reported. When the distribution is known during training, we deploy post-compensation [HHC+21] as a post-processing method to utilize this information for all methods. For all experiments, we use FixMatch and train on CIFAR-10-LT with imbalance ratio $\gamma = 150$. Then, we evaluate different methods at unknown and known shifted test distributions varying from imbalance ratio $\gamma = 512$ to -512. All experiments are run with the same data split and the training protocol from Section 6.4.1.

Table 6.4 summarizes the results. Compared to other methods, our approach has higher mean accuracy for both known and unknown distributions, which is mainly due to the good performance at the negative test imbalance ratios. For example, while being lower at positive ratios, our method is 24.17% and 8.02% better than the second-best at imbalance ratio $\gamma = -512$ in known and unknown cases, respectively. Our method also shows good robustness against the change of test imbalance ratios. For known test distribution, as the information of test distributions is utilized during the training in our method, we achieve a more balanced performance under various imbalance ratios. For example, the performance gap between $\gamma = 512$ and $\gamma = -512$ is 4.52% for our method compared to 14.19% for CReST+ and 22.12% for DARP. Despite the improved performance from our method, the relatively lower results at the negative ratios also indicate that none of the existing methods, including ours, can achieve a real balanced performance. Note that our protocol can be applied for imbalanced supervised learning as well.

Benefits of decoupling		CIFAR-10	CIFAR-100
Fix.	-	69.16	36.79
	two-stage	73.52	39.54
	CoSSL	**80.24**	**42.09**
ReMix.	-	70.89	38.48
	two-stage	80.30	41.87
	CoSSL	**81.94**	**43.14**

Table 6.5: Both decoupled approaches (two-stage, CoSSL) show better results over the joint training. Particularly, our co-learning achieves the best performance across settings.

		CIFAR-10	CIFAR-100
Fix.	no sg	76.46	39.82
	CoSSL	**80.24**	**42.09**
ReMix.	no sg	68.12	42.65
	CoSSL	**81.94**	**43.14**

Table 6.6: Performance degrades if representation is updated with gradients from the classifier module.

Pseudo-label generation		CIFAR-10	CIFAR-100
Fix.	h_{SSL}	78.23	40.43
	h_{CL}	**80.24**	**42.09**
ReMix.	h_{SSL}	80.22	42.09
	h_{CL}	**81.94**	**43.14**

Table 6.7: Benefits of using classifier learning module to generate pseudo-labels. h_{SSL} is the classifier from the representation learning module, h_{CL} is the classifier from the classifier learning module.

	Use $\mathcal{U}$	Enhancement	CIFAR-10	CIFAR-100
Fix.		-	77.22	41.33
	✓	mixUp[ZCDLP18]	77.36	41.24
	✓	MFW[YZC21]	77.91	41.61
	✓	TFE	**80.24**	**42.09**
ReMix.		-	79.53	42.60
	✓	mixUp[ZCDLP18]	81.75	42.27
	✓	MFW[YZC21]	81.48	42.51
	✓	TFE	**81.94**	**43.14**

Table 6.8: Test accuracy of different classifier learning methods in CoSSL. cRT+ denotes classifier re-training with mixUp.

6.4.5 Ablation study

In this section, we first analyze different design choices for CoSSL to provide additional insights into how it helps generalization. Then, we provide detailed ablation studies on TFE. We use CIFAR-10-LT with $\gamma = 150$ as our main ablation settings. We focus on a single split and report results for a Wide ResNet-28-2 [ZK16a] with FixMatch [SBL$^+$20] backbone. For fair comparison, the same data split is used for all experiments in this section.

Benefits of the co-learning framework. We attribute the success of CoSSL to four aspects. (1) Decoupling representation and classifier is crucial for imbalanced SSL, and our co-learning framework which further couples them closely is superior to the standard two-stage approach. As is shown in Table 6.5, both decoupled training schemes (co-learning and two-stage) show significant performance improvement over the joint training method. In particular, our co-learning approach CoSSL shows preferred test accuracy to the two-stage approach across settings, which suggests the importance of coupling representation and classifier while being decoupled. (2) It is more beneficial to not update the representation directly with the gradient from the classifier learning module. In Table 6.6, test accuracy shows 3.78% drop when representation is updated with gradients from the classifier learning module. (3) Instead, it is advantageous to use the balanced classifier h_{CL} for pseudo-label generation due to its robustness against data imbalance, as is shown in Table 6.7. (4) Last but not least, it is important to utilize unlabeled data for classifier learning, and modifications we proposed in TFE are important for the final performance. Table 6.8 compares performance of different classifier learning strategies for CoSSL. Methods that leverage unlabeled data (MFW[YZC21] and TFE) outperform the ones that do not (cRT and cRT+) in most cases. In particular, TFE achieves the best accuracy across different settings, which justifies its importance to CoSSL.

Design choices in TFE. Dedicated to imbalanced SSL, TFE differs from existing feature mixing approaches in three important aspects. First, we utilize class-dependent blend probability P_k to encourage more augmentation for the tail classes, thus, improving the final performance as is shown in Table 6.9. Removing the mechanism of P_k decreases the performance by 2.34%. Second, the fusion factor λ is sampled from a uniform distribution between μ and 1. This strategy shows better empirical results than the commonly used beta distribution and other variants of uniform distribution. Thirdly, TFE does not apply label blending. Table 6.9 shows a performance drop of 7.00% when labels are mixed with pseudo-labels from unlabeled data. TFE does not only show the best performance in our joint framework but also shows the best performance in the two-stage framework.

Test Acc.	CIFAR-10
blend labels with pseudo-labels	73.24
image-level enhancement	78.92
remove blend probability	77.90
TFE	**80.24**

Table 6.9: Design choices in TFE.

6.5 Conclusion and limitations

In this work, we study imbalanced SSL, which is a more general setting as both labeled and unlabeled data from imbalanced distributions. We propose CoSSL, a flexible co-learning framework for imbalanced SSL, which decouples the representation learning and classifier learning while connecting them by sharing learned features and generated pseudo-labels. We also design Tail-class Feature Enhancement for learning the classifier with unlabeled data and enhancing the performance at tail classes. Integrating TFE and strong SSL methods into our CoSSL framework, we achieve new state-of-the-art results across a variety of imbalanced SSL benchmarks, especially when the imbalance ratio is large. At the evaluation, we address the limitation of the conventional uniform protocol by evaluating methods at shifted distributions and considering known and unknown test distribution during training. Such a comprehensive evaluation provides more insights into the existing methods and uncovers limitations.

This work, however, is also subject to several limitations. First, this chapter focuses on the object recognition problem under class-imbalanced distribution. Therefore, caution must be taken when generalizing to other vision tasks. Second, our method only considers in-class unlabeled data whose potential class labels are covered by the labeled set. However, there are often a large number of out-of-class unlabeled data available in real-world applications. And they are often mixed with in-class unlabeled data, which can be detrimental if not properly handled. Our method, at the current stage, is not able to handle such a case and effectively leverage out-of-class unlabeled data, which we leave for future work. Thirdly, as we have seen from Section 6.4.4, all of the existing methods, including ours, can not achieve a real balanced performance across test distributions. The performance at distributions that are radically different from the training distribution is relatively lower.

7 Boosting Performance of Open-Set Semi-Supervised Learning

Contents

Another unrealistic assumption of standard semi-supervised learning (SSL) is that it assumes unlabeled data are always from the same classes of labeled data. Therefore, SSL models often underperform in open-set scenarios, where unlabeled data contain outliers from novel categories that do not appear in the labeled set. In this chapter, we study the challenging and realistic open-set SSL setting, where the goal is to both correctly classify inliers and to detect outliers. Intuitively, the inlier classifier should be trained on inlier data only. However, we find that inlier classification performance can be largely improved by incorporating high-confidence pseudo-labeled data, regardless of whether they are inliers or outliers. Also, we propose to utilize non-linear transformations to separate the features used for inlier classification and outlier detection in the multi-task learning framework, preventing adverse effects between them. Additionally, we introduce pseudo-negative mining, which further boosts outlier detection performance. The three ingredients lead to what we call **S**imple but **S**trong **B**aseline (SSB) for open-set SSL. In experiments, SSB greatly improves both inlier classification and outlier detection performance, outperforming existing methods by a large margin.

This chapter is based on [FKDS23]. Yue Fan was the lead author of this paper and conducted all the experiments and wrote most part of the paper.

7.1 Introduction

Semi-supervised learning (SSL) has achieved great success in improving model performance by leveraging unlabeled data [Lee13, LA17, TV17a, MMKI18b, BCG+19, BCC+20, SBL+20, XDH+19, FKS21b, ZWH+21, WCF+22a]. However, standard SSL assumes that the unlabeled samples come from the same set of categories as the labeled samples, which makes them struggle in open-set settings [OOR+18], where unlabeled data contain out-of-distribution (OOD) samples from novel classes that do not appear in the labeled set (see Fig. 7.1). In this chapter, we study this more realistic setting called *open-set semi-supervised learning*, where the goal is to learn both a good closed-set classifier to classify inliers and to detect outliers as

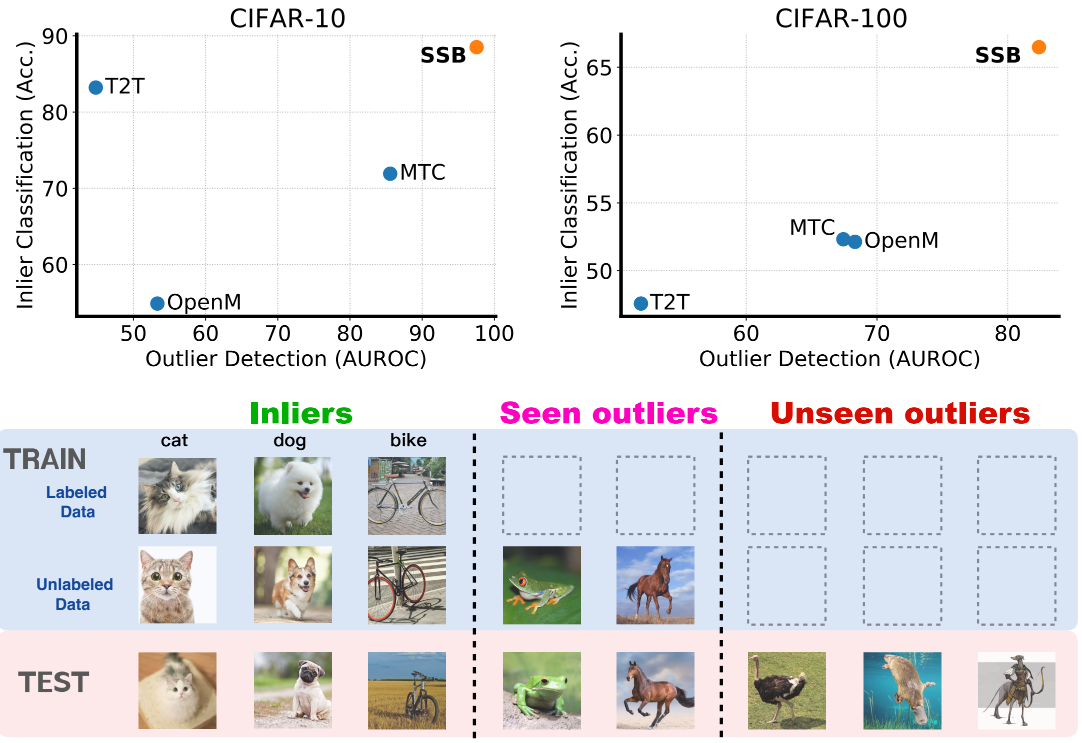


Figure 7.1: Open-set semi-supervised learning considers a realistic and challenging setting, where unlabeled data contains samples from novel classes (**seen outliers**) that do not appear in the labeled data. At test time, the model should correctly classify **inliers**, while identifying outliers seen during the training and, most importantly, **unseen outliers** that do not appear in the training set. We measure test accuracy for the inlier classification performance and AUROC for the outlier detection performance. Our method (SSB) achieves superior performance in both tasks.

shown in Fig. 7.1.

Recent works on open-set SSL [HFC^{+}21, CZLG20, SKS21, YIIA20a, GZJ^{+}20a, HHLY22, HHYY22, HYG22] have achieved strong performance [WBH19, MR05, HA04, AY01] through a multi-task learning framework, which consists of an inlier classifier, an outlier detector, and a shared feature encoder, as shown in Figure 7.2. The outlier detector is trained to filter out OOD data from the unlabeled data so that the classifier is only trained on inliers. However, this framework has two major drawbacks. First, detector-based filtering often removes many inliers along with OOD data, leading to suboptimal classification performance due to the low utilization ratio of unlabeled data. Second, the inlier classifier which shares the same feature encoder with the outlier detector can have an adverse effect on the detection performance as shown in Table 7.1.

To this end, we contribute a **S**imple but **S**trong **B**aseline, **SSB**, for open-set SSL with three ingredients to address the above issues. (1) In contrast to detector-based filtering aiming to remove OOD data, we propose to incorporate pseudo-labels with high inlier classifier confidence into the training, *irrespective of whether a sample is an inlier or OOD*. This not only effectively improves the unlabeled data utilization ratio but also includes many useful OOD data that can be seen as natural data augmentations of inliers (see Fig. 7.5). (2) Instead of directly sharing features between the classifier and detector, we add non-linear transformations for the task-specific heads and find that this effectively reduces mutual interference between them, resulting in more specialized features and improved performance for both tasks. (3) In addition, we propose pseudo-negative mining to further improve outlier detector training by enhancing the data diversity of OOD data with pseudo-outliers. Despite its simplicity, SSB achieves significant improvements in both inlier classification and OOD detection. As shown in Fig. 7.1, existing methods either struggle in detecting outliers or have difficulties with inlier classification while SSB obtains good performance for both tasks.

7.2 Related Work

Semi-supervised learning. Semi-supervised learning (SSL) aims to improve model performance by exploiting both labeled and unlabeled data. As one of the most widely used techniques, pseudo-labeling [Lee13] is adopted by many strong SSL methods [SBL+20, BCG+19, BCG+19, XDH+19, ZWH+21, AOA+20a, PXDL20, XLHL20a, BRS+22, LXH21]. The idea is to generate artificial labels for unlabeled data to improve model training. [BCG+19, BCC+20] compute soft pseudo-labels and then apply MixUp [ZCDLP18] with labeled data to improve the performance; [SBL+20, XDH+19, ZWH+21] achieves good performance by combining pseudo-labeling with consistency regularization [LA17, MMKI18b, TV17a]; [PXDL20] proposes a meta learning approach that uses a teacher model to refine the pseudo-labels based on the training of a student model; [XLHL20a] leverages the idea of self-training which generates pseudo labels in an iterative way and inject noise to each training stage. In this chapter, we also adopt a simple confidence-based pseudo-labeling [SBL+20] for classifier training, which is an effective way of leveraging unlabeled data to improve the model performance. Compared to standard SSL, SSB has an additional outlier detector, which enables the model to reject samples that do not belong to any of the inlier classes.

Open-set SSL & Class-mismatched SSL. First shown by [OOR+18], standard SSL methods suffer from performance degradation when there are out-of-distribution (OOD) samples in unlabeled data. Since then, various approaches have been proposed to address this challenge [CZLG20, GZJ+20a, YIIA20a, SKS21, HFC+21, PYJS22, HHLY22, HHYY22, HYG22]. Existing methods seek to alleviate the effect of OOD data by filtering them out in different ways so that the classification model is trained with inliers only. For example, [CZLG20] uses model ensemble [Sch90] to compute soft pseudo-labels and performs filtering with a confidence threshold; [GZJ+20a] proposes a bi-level optimization to weaken the loss weights for OOD data; [YIIA20a] assigns an OOD score to each unlabeled data and refines it during the training; [SKS21] leverages one-vs-all (OVA) classifiers [SS21] for OOD detection and propose a consistency loss to train them; [HFC+21] proposes a cross-modal matching module to detector outliers. [HYG22] employs adversarial domain adaptation to filter unlabeled data and find recyclable OOD data to improve the performance; [HHLY22] uses energy-discrepancy to identify inliers and outliers. In contrast, we show that if the representations of the inlier classifier and the outlier detector are well-separated, OOD data turns out to be a powerful source to improve the inlier classification without degrading the detection performance. So, instead of filtering OOD data, we use a simple confidence-based pseudo-labeling to incorporate them into the training.

Open-world SSL. Open-set SSL is similar to open-world SSL [CBL21, RKK+22, RKS22] but bears several important differences. While both have unlabeled data of novel classes during the training, the goal of open-world SSL is to classify inliers and discover new classes from OOD data instead of rejecting them. Another important difference is that open-world SSL is often a transductive learning setting while open-set SSL requires generalization beyond the current distribution. Namely, the model should be able to detect OOD data from novel classes that present in the training set as well as OOD data from classes that are never seen during training.

7.3 SSB: Simple but Strong Baseline for Open-Set Semi-Supervised Learning

In this section, we first present the problem setup of open-set semi-supervised learning (SSL). Then, we give an overview of our method SSB in Section 7.3.1 before presenting details of the three simple yet effective ingredients used in our method in Section 7.3.2, 7.3.3, and 7.3.4.

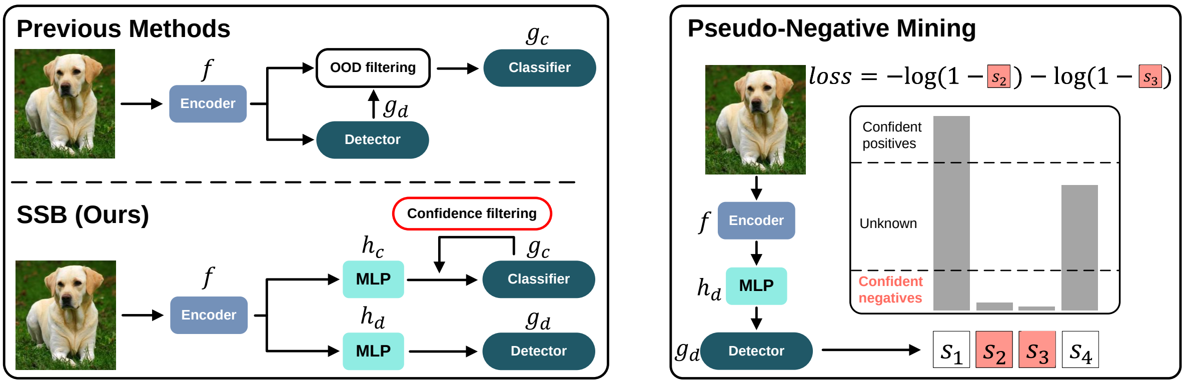


Figure 7.2: **Left:** Our baseline for open-set SSL consists of an inlier classifier g_c, an outlier detector g_d, and a shared feature encoder f whose features are separated from the task-specific heads by two projection heads h_c and h_d. Unlike the detector-based filtering, we adopt confidence-based pseudo-labeling by the inlier classifier to leverage useful OOD data for classifier training. For detector training, we train one-vs-all (OVA) classifiers as in OpenMatch [SKS21]. **Right:** Given the inlier scores (s_1 to s_4), pseudo-negative mining selects confident negatives (s_2 and s_3 in the figure), whose inlier scores are lower than a pre-defined threshold, as pseudo-outliers to help the outlier detector training.

Problem setup and notations: As shown in Fig. 7.1, open-set SSL generalizes the settings of standard SSL and out-of-distribution (OOD) detection. It considers three disjoint sets of classes: $\mathcal{C}$ corresponds to the inlier classes that are partially annotated, $\mathcal{U}_\mathcal{S}$ contains the outlier classes seen during training but without annotations, and lastly, $\mathcal{U}_\mathcal{U}$ is composed of the classes that are not seen during training (only seen at test time). The training data contains a small labeled set $\mathcal{D}_{\text{labeled}} = \{(\mathbf{x}_i^l, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{C}$ and a large unlabeled set $\mathcal{D}_{\text{unlabeled}} = \{(\mathbf{x}_i^u)\}_{i=1}^M \subset \mathcal{X}$, where $\mathcal{X}$ is the input space. While the labeled set only consists of samples of inlier classes, the unlabeled set contains both samples from $\mathcal{C}$ and $\mathcal{U}_\mathcal{S}$. Thus, the the ground-truth label of $\mathbf{x}^u$ is from $\mathcal{C} \cup \mathcal{U}_\mathcal{S}$ with $\mathcal{C} \cap \mathcal{U}_\mathcal{S} = \emptyset$.

The goal of open-set SSL is to train a model that can perform good inlier classification as well as detecting both seen and unseen outliers. Without loss of generality, consider a test set $\mathcal{D}_{\text{test}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times (\mathcal{C} \cup \mathcal{U}_\mathcal{S} \cup \mathcal{U}_\mathcal{U})$, where $\mathcal{C} \cap \mathcal{U}_\mathcal{U} = \emptyset$ and $\mathcal{U}_\mathcal{S} \cap \mathcal{U}_\mathcal{U} = \emptyset$. The learned model should be able to correctly classify inliers $\{(\mathbf{x}_i | y_i \in \mathcal{C})\}$ and detect outliers from $\{(\mathbf{x}_i | y_i \in \mathcal{U}_\mathcal{S})\}$ as well as $\{(\mathbf{x}_i | y_i \in \mathcal{U}_\mathcal{U})\}$, which is crucial for practical applications.

7.3.1 Method Overview

Following [HFC+21, CZLG20, SKS21, YIIA20a, GZJ+20a, HHLY22, HHYY22, HYG22], we adopt a multi-task learning framework for open-set SSL, which performs inlier classification and outlier detection. As shown in Fig. 7.2, SSB comprises four components: (1) An inlier classifier g_c, (2) an outlier detector g_d, (3) a shared feature encoder f, and (4) importantly, two projection heads h_c and h_d. Inspired by [SKS21], the outlier detector g_d consists of $|\mathcal{C}|$ one-vs-all (OVA) binary classifiers, each of which is trained to distinguish inliers from outliers for each single class. Given a batch of labeled data $\mathbf{X}^l = \{(\mathbf{x}^l_i, y_i)\}_{i=1}^{B_l}$ and unlabeled data $\mathbf{X}^u = \{(\mathbf{x}^u_i)\}_{i=1}^{B_u}$, the total loss for training the model is:

$$L_{total} = L_{cls}(\mathbf{X}^l, \mathbf{X}^u; f, h_c, g_c) + L_{det}(\mathbf{X}^l, \mathbf{X}^u; f, h_d, g_d) \tag{7.1}$$

where L_{cls} and L_{det} are the classification and detection losses, respectively. For the sake of brevity, we will drop the dependencies of the loss function on f, h_c, g_c, h_d, and g_d in the following. The complete algorithm of SSB is summarized by Alg. 5.

During inference, the test image is first fed to the inlier classifier to compute the class prediction. Then, the corresponding detector is used to decide whether it is an inlier of the predicted class or an outlier. We explain the details of SSB in the following three sections.

7.3.2 Boosting Inlier Classification with Classifier Pseudo-Labeling

Existing methods for open-set SSL [HFC+21, CZLG20, SKS21, YIIA20a, GZJ+20a] aim to eliminate OOD data from the classifier training. This is typically accomplished by training outlier detectors that can filter out OOD data from unlabeled data, as shown in Fig. 7.2. However, as we will see in Table 7.3, detector-based filtering often removes many inliers along with OOD data, which leads to a low utilization ratio of unlabeled data and hinders inlier classification performance.

In this work, instead of using detector-based filtering, we propose to incorporate unlabeled data with confident pseudo-labels (as generated by the *inlier classifier*) into the training, *irrespective of whether it is inlier or OOD data*. This not only effectively improves the unlabeled data utilization ratio but also includes many useful OOD data as natural data augmentations of inliers into the training (see Fig. 7.5). Inspired by [SBL+20], we train the model with pseudo-labels from the inlier classifier whose confidence scores are above a pre-defined threshold. Specifically, for each unlabeled sample $\mathbf{x}^u_i$, we first predict the pseudo-label distribution as $\hat{p}^u_i = \text{softmax}(h_c(g_c(f(\mathbf{x}^u_i))))$. Then, the confidence score of the pseudo-label is computed as $\max \hat{p}^u_i$. Finally, the cross-entropy loss is calculated for samples whose pseudo-labels have confidence scores greater than a pre-defined threshold τ as:

$$L^u_{cls}(\mathbf{X}^u) = \frac{1}{B_u} \sum_{i=1}^{B_u} \mathbb{1}(\max \hat{p}^u_i \geq \tau) H(\hat{p}^u_i, \hat{y}^u_i) \tag{7.2}$$

where $H(\cdot, \cdot)$ denotes the cross-entropy, $\hat{y}^u_i = \text{argmax} \hat{p}^u_i$, and $\mathbb{1}(\cdot)$ is the indicator function which outputs 1 when the confidence score is above the threshold τ.

The total classification loss is computed as the summation of a labeled data loss and the unlabeled data loss as:

$$L_{cls}(\mathbf{X}^l, \mathbf{X}^u) = L^l_{cls}(\mathbf{X}^l) + L^u_{cls}(\mathbf{X}^u) \tag{7.3}$$

where L^l_{cls} is a standard cross-entropy loss for labeled data.

Algorithm 5 SSB algorithm for Closed-Set Classification and OOD Detection.

```
Input: Labeled set $\mathcal{D}_{\text{labeled}} = \{(\mathbf{x}_i^l, y_i)\}_{i=1}^N$, unlabeled set $\mathcal{D}_{\text{unlabeled}} = \{(\mathbf{x}_i^u)\}_{i=1}^M$, feature encoder $f$, inlier classifier $g_c$, outlier detector $g_d$, two MLP projection heads $h_c$ and $h_d$, thresholds $\tau$ and $\theta$, batch size $B_l$ and $B_u$, loss weights $\lambda_{det}^u$, $\lambda_{OC}^u$, and $\lambda_{em}^u$, warm-up iterations $T_0$, total number of training iterations $T$
Initialize the parameters of $f$, $g_c$, $g_d$, $h_c$, and $h_d$ randomly
for $t = 1$ to $T$ do
    // Sample labeled and unlabeled data
    $\{\mathbf{x}_i^l, y_i\}_{i=1}^{B_l} \sim$ Random sampler($\mathcal{D}_{\text{labeled}}$)
    $\{\mathbf{x}_i^u\}_{i=1}^{B_u} \sim$ Random sampler($\mathcal{D}_{\text{unlabeled}}$)
    // Compute classification losses
    $\hat{p}_i^u = \text{softmax}(g_c(h_c(f(\mathbf{x}_i^u)))), i = 1, ..., B_u$    // Compute pseudo-label distributions
    $\hat{y}_i^u = \text{argmax} \hat{p}_i^u, i = 1, ..., B_u$    // Compute pseudo-labels
    $L_{cls}^l = \frac{1}{B_l} \sum_{i=1}^{B_l} H(g_c(h_c(f(\mathbf{x}_i^l)), y_i)$    // Labeled data loss
    $L_{cls}^u = \frac{1}{B_u} \sum_{i=1}^{B_u} \mathbb{1}(\max \hat{p}_i^u \geq \tau) H(\hat{p}_i^u, \hat{y}_i^u)$    // Unlabeled data loss as in Equation (2)
    $L_{cls} = L_{cls}^l + L_{cls}^u$
    // Compute detection losses
    $L_{det}^l = -\frac{1}{B_l} \sum_{i=1}^{B_l} log(p_{y_i}(\mathbf{x}_i^l)) + \frac{1}{|\mathcal{C}|-1} \sum_{j \neq y_i} log(1 - p_j(\mathbf{x}_i^l))$    // Detection loss for labeled data as in Equation (4)
    $L_{det}^u = -\frac{1}{B_u} \sum_{i=1}^{B_u} \frac{1}{\sum_c \mathbb{1}(p_c > \theta)} \sum_{c=1}^{|\mathcal{C}|} \mathbb{1}(p_c > \theta) log(1 - p_c(\mathbf{x}_i^u))$    // Pseudo-negative mining as in Equation (5)
    $L_{em}^u = \frac{1}{B_u} \sum_{i=1}^{B_u} entropy(g_d(h_d(f(\mathbf{x}_i^u))))$    // Entropy minimization loss as in [GB05]
    $L_{OC}^u = \frac{1}{B_u} \sum_{i=1}^{B_u} ||g_d(h_d(f(\mathcal{T}_1(\mathbf{x}_i^u)))) - g_d(h_d(f(\mathcal{T}_2(\mathbf{x}_i^u))))||^2$    // Open-set consistency loss as in [SKS21]
    $L_{det} = L_{det}^l + \lambda_{det}^u L_{det}^u + \lambda_{OC}^u L_{OC}^u + \lambda_{em}^u L_{em}^u$
    // Total loss
    $L_{total} = L_{cls} + \mathbb{1}(t > T_0) L_{det}$
    Update parameters in $f$, $g_c$, $g_d$, $h_c$, and $h_d$ with SGD
end for
return $f$, $g_c$, $g_d$, $h_c$, and $h_d$
```

Despite its simplicity, we obtain a substantial performance improvement in inlier classification through classifier confidence-based pseudo-labeling as shown in Table 7.1. Our method is conceptually different from previous methods as we aim to leverage OOD data rather than remove them. On the one hand, our method effectively improves the unlabeled data utilization ratio as shown in Table 7.3, which leads to great inlier classification performance improvement. On the other hand, our method provides an effective way of leveraging useful OOD data for classifier training. In fact, many OOD data are natural data augmentations of inliers and are beneficial for classification performance if used carefully. As shown in Fig. 7.5, the selected OOD data present large visual similarities with samples of inlier classes, and, thus, significantly enhance the data diversity, leading to improved generalization performance.

7.3.3 Non-Linear Feature Boosting

In previous methods, simply including OOD samples into the classifier training harms detection performance since the inlier classifier and the outlier detector use the same feature representation [SKS21, YIIA20a, HFC+21]. On the one hand, the classifier uses OOD data as pseudo-inliers, thus mixing their representations in the feature space. On the other hand, the outlier detector is trained to distinguish inliers and outliers, which leads to separated representations in the feature space. As a result, the contradiction between the classifier and the outlier detector ultimately adversely affects each other, which limits the overall performance, as shown in Table 7.1.

In this work, we find empirically that simply adding non-linear transformations between the task-specific heads and the shared feature encoder can effectively mitigate the adverse effect. Given a sample $\mathbf{x}_i$, two multi-layer perceptron (MLP) projection heads h_c and h_d are used to transform the features from the encoder. The output of the network is thus $h_c(g_c(f(\mathbf{x}_i)))$ for the classifier and $h_d(g_d(f(\mathbf{x}_i)))$ for the outlier detector. Compared to the previous methods, the non-linear transformations effectively prevent mutual interference between the classifier and detector, resulting in more specialized features and improved performance in both tasks. In Table 7.1, while the OOD detection performance degenerates when adding OOD data for classifier training for the model without the projection heads, SSB, in contrast, still exhibits excellent performance in detecting outliers with the help of the projection heads. Moreover, the efficacy of the non-linear projection head also generalizes to other frameworks. We show in the experiment section that it is compatible with various SSL backbones and open-set SSL methods and leads to performance improvement.

7.3.4 Outlier Detection with Pseudo-Negative Mining

In this section, we first describe the outlier detector used in SSB and then introduce a simple yet effective technique called pseudo-negative mining to improve the outlier detector training.

Following [SKS21], we adopt $|\mathcal{C}|$ one-vs-all (OVA) binary classifiers for OOD detection, where each OVA classifier is trained to distinguish between inliers and outliers for each individual inlier class. Given a labeled sample $\mathbf{x}_i^l$ from class y_i, it is regarded as an inlier for class y_i and an outlier for class $k, k \neq y_i$. Therefore, the OVA classifiers can be trained using binary cross-entropy loss on the positive-negative pairs constructed from the labeled set as:

$$L_{det}^l(\mathbf{X}^l) = -\frac{1}{B_l}\sum_{i=1}^{B_l} log(p_{y_i}(\mathbf{x}_i^l)) + \frac{1}{K}\sum_{k \neq y_i} log(1 - p_k(\mathbf{x}_i^l)) \tag{7.4}$$

where $p_k(\mathbf{x}_i^l)$ is the inlier score of $\mathbf{x}_i^l$ for class k computed by the k-th detector and $K = |\mathcal{C}| - 1$.

However, due to data scarcity, it is difficult to learn good representations for outliers with labeled data only. To this end, we propose pseudo-negative mining to further improve the outlier detector training by leveraging confident negatives as pseudo-outliers to enhance the data diversity of OOD data. As shown in Fig. 7.2, given an unlabeled sample $\mathbf{x}_i^u$, we consider it as a pseudo-outlier for class k if the inlier score for class k is lower than a pre-defined threshold. Then, $\mathbf{x}_i^u$ is used as a negative sample to calculate the cross-entropy loss of class k. The final loss for $\mathbf{x}_i^u$ is the summation over all classes using it as the negative sample:

$$L_{det}^u(\mathbf{x}_i^u) = -\frac{1}{\sum_k \mathbb{1}(p_k < \theta)}\sum_{k=1}^{|\mathcal{C}|} \mathbb{1}(p_k < \theta) log(1 - p_k(\mathbf{x}_i^u)) \tag{7.5}$$

where p_k is the inlier score from the k-th detector and $\mathbb{1}(\cdot)$ is the indicator function which outputs 1 when the confidence score is less than the threshold θ. This increases the data diversity of outliers and improves generalization performance as shown in Table 7.5. Compared to standard pseudo-labels, pseudo-outliers have much higher precision because we specify which classes the sample does not belong to rather than which class it belongs to. The latter is a more difficult task than the former. Therefore, pseudo-negative mining is less susceptible to inaccurate predictions while increasing data utilization.

Our final loss for detector training also includes Open-set Consistency (OC) loss [SKS21] and entropy minimization (EM) [GB05] because they can lead to further improvement. The overall loss for training the detector is as follows:

$$L_{det}(\mathbf{X}^l, \mathbf{X}^u) = L^l_{det}(\mathbf{X}^l) + \lambda^u_{det} L^u_{det}(\mathbf{X}^u) + \lambda^u_{OC} L^u_{OC}(\mathbf{X}^u) + \lambda^u_{em} L^u_{em}(\mathbf{X}^u) \tag{7.6}$$

where λ^u_{det}, λ^u_{OC}, and λ^u_{em} are loss weights; L^u_{OC} is the soft open-set consistency regularization loss, which enhances the smoothness of the OVA classifier with respect to input transformations; L^u_{em} is the entropy minimization loss, which encourages more confident predictions.

7.4 Experiments

In this section, we first compare SSB with existing methods in Section 7.4.1, and then provide an ablation study and further analysis in Section 7.4.2.

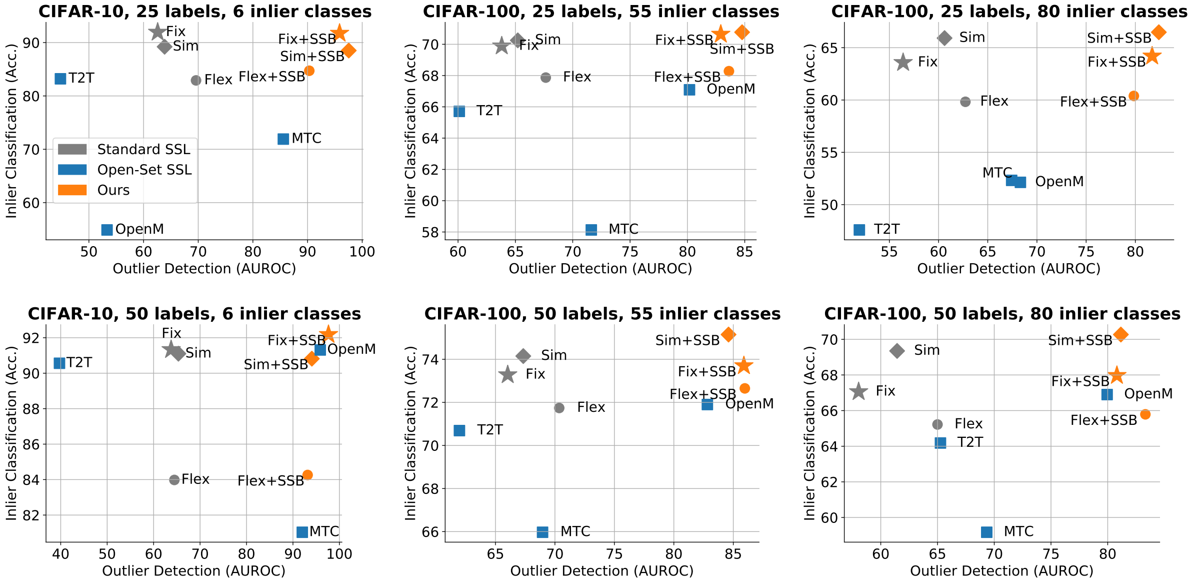


Figure 7.3: **Classification and detection performance on CIFAR-10 and CIFAR-100 with varying numbers of inlier classes and labeled data.** We measure test accuracy for the inliers classification performance and AUROC for the outlier detection performance. While standard SSL methods suffer in outlier detection and open-set SSL methods suffer in inlier classification, SSB achieves good performance in both tasks. Noted that the reported outlier detection performance is the *average AUROC in detecting both seen and unseen outliers*.

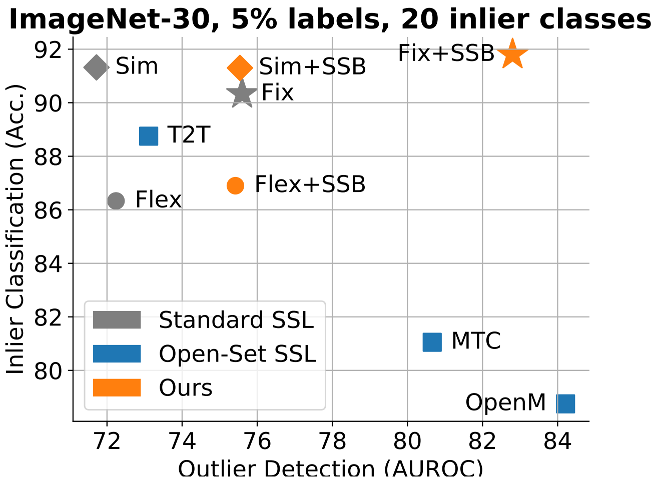


Figure 7.4: **Classification performance versus the outlier detection performance on ImageNet-30.** SSB achieves good performance in both inlier classification and OOD detection.

7.4.1 Main Results

Datasets & Evaluation. As mentioned in Section 7.3, the goal of open-set SSL is to train a good inlier classifier as well as an outlier detector that can identify both seen and unseen outliers. Therefore, we need to construct three class spaces: inlier classes $\mathcal{C}$, seen outlier classes $\mathcal{U}_\mathcal{S}$, and unseen outlier classes $\mathcal{U}_\mathcal{U}$. For each setting: the labeled set contains samples from $\mathcal{C}$ only; the unlabeled set contains samples from $\mathcal{C}$ and $\mathcal{U}_\mathcal{S}$; the test set contains samples from $\mathcal{C}, \mathcal{U}_\mathcal{S}$, and $\mathcal{U}_\mathcal{U}$. The inlier classification performance is evaluated on $\mathcal{C}$ using test accuracy as in standard supervised learning. The OOD detection performance is measured by AUROC following [SKS21] and we report the **average performance in detecting seen outliers and unseen outliers**.

Following [SKS21], we evaluate SSB on CIFAR-10 [KH+09a], CIFAR-100 [KH+09a], and ImageNet [DDS+09] with different numbers of labeled data. For CIFAR-10, the 6 animal classes are used as inlier classes, and the rest 4 are used as seen outlier classes during the training. Additionally, test sets from SVHN [NWC+11b], CIFAR-100, LSUN [YSZ+15], and ImageNet are considered as unseen outliers, and used to evaluate the detection performance on unseen outliers. For CIFAR-100, the inlier-outlier split is performed on super classes, and two settings are considered: 80 inlier classes (20 outlier classes) and 55 inlier classes (45 outlier classes). Similar to CIFAR-10, test sets from SVHN, CIFAR-10, LSUN, and ImageNet are used to evaluate the detection performance on unseen outliers. For ImageNet, we follow [SKS21] to use ImageNet-30[HG16], which is a subset of ImageNet containing 30 distinctive classes. The first 20 classes are used as inlier classes while the rest 10 are used as outlier classes. Stanford Dogs [KJYFF11], CUB-200 [CZSL20], Flowers102 [NZ08], Caltech-256 [GHP+07], Describable Textures Dataset [CMK+14], LSUN are used as unseen outlier classes at test time.

Implementation details. We use Wide ResNet-28-2 [ZK16a] as the backbone for CIFAR experiments and ResNet-18 [HZRS16a] for ImageNet experiments. As standard SSL models do not have the notion of OOD detection, we adopt the method in [HG16], where the OOD score of an input image $\mathbf{x}$ is computed as $1 - \max \text{softmax}(f(\mathbf{x}))$ and f denotes the model. Thus, the input image is considered as an outlier if the OOD score is higher than a pre-defined threshold. For other open-set SSL methods, we directly employ the authors' implementations and follow their default hyper-parameters.

For SSB, we use two two-layer MLPs with ReLU [NH10] non-linearity to separate representations for all settings. The hidden dimension is 1024 for CIFAR settings and 4096 for ImageNet settings. For classifier training, we follow [SBL+20] and set the threshold τ as 0.95. For outlier detector training, we set λ_{det}^{u} as 1 for all settings and follow [SKS21] for the weights of OC loss and entropy minimization. The threshold θ is 0.01 for all experiments. Following [SKS21], we train our model for 512 epochs with SGD [KW52] optimizer. The learning rate is set as 0.03 with a cosine decay. The batch size is 64. Additionally, we defer the training of the outlier detector until epoch 475 to reduce the computational cost as we find empirically the deferred training does not comprise the model performance.

When combined with standard SSL methods (e.g. SSB + FlexMatch), we replace the classifier training losses in Equation 1 with the corresponding losses of different methods while keeping the outlier detector the same. When combined with open-set SSL methods (e.g. MTC + SSB), we make three modifications. First, we separate the outlier detector branch from the classifier branch using the proposed MLP projection head. Second, we replace the outlier detector training losses with our loss from Equation 6. Third, we do not filter unlabeled data with the outlier detector for classifier training.

Results. We compare SSB with both standard SSL and open-set SSL methods. Fig. 7.3 and 7.4 summarize the inlier test accuracy and outlier AUROC for CIFAR datasets and ImageNet, respectively. Considering the goal of open-set SSL is to achieve *both good inlier classification accuracy and outlier detection*, SSB greatly outperforms standard SSL methods in outlier detection, and open-set SSL methods in inlier classification. For example, on CIFAR-10 with 25 labels, the AUROC of our best method is 11.97% higher than the best method excluding ours. Moreover, when combined with standard SSL algorithms, our method demonstrates consistent improvement in OOD detection, and in most cases, better test accuracy for inlier classification. This suggests the flexibility of our method, which makes it possible to benefit from the most advanced approaches. Note that the performance improvement of SSB can not be simply explained by the increased number of parameters introduced in the projection heads. Please see Fig. 7.7 for a comparison between SSB and other methods + MLP heads.

Additionally, SSB is more robust to the number of labeled data than others. We achieve reasonable performance given a small number of labeled data while other methods fail to generalize. For example, on CIFAR-10 with 6 inlier classes, OpenMatch has similar inlier accuracy as ours at 50 labels. When the number of labeled data is halved, their performance decreases to 54.88% while our method still has a test accuracy of 91.74%.

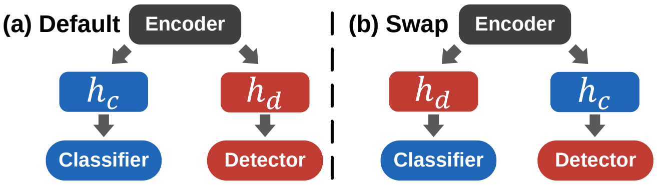
7.4.2 Ablation Study

In this section, we analyze the design choices of SSB and show their importance through ablation experiments. If not specified, we use CIFAR-10 with 25 labeled data as our default setting for ablation. The same data split is used for fair comparison.

Importance of non-linear projection heads. As mentioned in Section 7.3, we use 2-layer MLPs to mitigate the adverse effect between the inlier classifier and outlier detector. Here we study the effect of the projection heads in Table 7.1. As we can see, incorporating confidence filtering yields a significant improvement in inlier classification performance (resulting in a 12.23% to 13.18% increase). However, the OOD detection performance experiences a substantial decline when the projection heads are missing (AUROC from 89.67% to 63.46%). This is because the classifier tends to mix the features of inliers and outliers with the same pseudo-labels in a shared feature space, which contradicts the goal of the outlier detector. The addition of the projection heads not only restores the OOD detection performance but also achieves superior results when combined with confidence filtering. Adding the projection heads in combination with confidence filtering not only restores the OOD detection performance but achieves even better performance, which indicates the importance of representation separation. Note that it is important to have two independent projection heads for the inlier classifier and outlier detector. A shared projection head does not restore the OOD detection performance as shown in Table 7.1. Moreover, we show in Table 7.2 that both classification and detection performance degrade when swapping the task-specific features of a pre-trained model with the fixed encoder. In particular, when re-training the detector (just a fully-connected layer) on top of classification features, seen AUROC drops from 89.18% to 53.99%, which suggests our model learns more task-specific features. Therefore, the utilization of the projection heads separates concerns between the classifier and detector, which eases the difficulties of the task and allows them to be trained jointly without adversely affecting each other.

Proj. head	Conf. filter	Inlier Cls. (Acc.)	Outlier Det. (AUROC)
		78.05	89.67
shared		76.75	91.92
separate		78.47	90.92
	✓	90.28	63.46
shared	✓	90.93	63.87
separate	✓	**91.65**	**94.76**

Table 7.1: **Effect of the projection head and confidence-based pseudo-labeling for classifier training.** We use a 2-layer MLP as the projection head. All models are trained with pseudo-negative mining on the same data split.



Nearest Neighbor	Inlier Cls. (Acc.)	Outlier Det. (AUROC)
default (a)	**55.04**	**99.43**
swap cls. & det. features (b)	53.70	77.89

Table 7.2: **Classification and detection performance using features of different heads.** We fix the encoder and MLP heads and evaluate the classification and detection performance using nearest neighbors on labeled set. Our model learns specialized features since swapping h_c and h_d leads to inferior performance in both tasks.

Improving data utilization with confidence-based pseudo-labeling. Here we study the effect of different classifier training strategies. We compare three unlabeled data filtering methods for classifier training: (1) *det.* selects pseudo-inliers with the outlier detector as in [SKS21]; (2)

det. (tuned), where we choose the selection threshold in detector-based filtering so that the recall of actual inlier samples matches ours; (3) *conf.* uses unlabeled data whose confidence is higher than a pre-defined threshold, which is our method. As shown in Table 7.3, although *det.* successfully removes many OOD data, it also eliminates many inliers, resulting in a low utilization ratio of unlabeled data (0.29% unlabeled data are used in training). In contrast, our method includes pseudo-labels with high classifier confidence into the training, irrespective of whether a sample is out-of-distribution, which leads to a high utilization ratio of unlabeled data (94.22%), thus, outperforming *det.* with a large margin. Moreover, our method also outperforms *det (tuned)* whose data selection threshold is tuned for better performance. This is because we incorporate a significant amount of OOD data in the training process (40.16% v.s. 16.90%). In fact, many OOD data are natural data augmentation of inliers, which can substantially improve closed-set classification if used carefully. When removing pseudo-labeled OOD data using an oracle during the training. The inlier classification accuracy decreases by 3.37% on CIFAR-10 with 25 labels (from 91.65% to 88.28%), which suggests pseudo-labeled OOD data are helpful for inlier classification. In Fig. 7.5, we visualize top-5 confident OOD samples predicted for three inlier classes from *conf.* on CIFAR-100. We can see that the selected samples are related to the inlier classes and contain the corresponding semantics despite being outliers. For example, OOD data selected for *sea* are images with sea background

	Filter method	det.	det (tuned)	conf. (ours)
Test	Inlier Clf. (Acc.)	47.20	86.53	**91.65**
	Outlier Det. (AUROC)	57.72	87.87	**94.76**
Train	Utilization ratio of:			
	- Unlabeled	0.29	58.09	94.22
	- OOD data	0.04	16.90	40.16
	Prec. of pseudo-inliers	95.17	86.53	58.30
	Recall of inliers	0.47	93.86	92.14

Table 7.3: **Effect of different OOD filtering methods for classifier training.** We compare three filtering methods: *conf.* denotes the confidence-based pseudo-labeling; *det.* uses the outlier detector to select pseudo-inliers for classifier training; *det. (tuned)* is a tuned version of *det.* that matches the recall of inliers with our method. We compare the performance as well as the data utilization ratio, precision, and recall of the inliers from unlabeled data during training. All models are trained with pseudo-negative mining and the projection head on the same data split.

Effect of pseudo-negative mining. Table 7.5 shows the effect of pseudo-negative mining. We compare our pseudo-negative mining with standard pseudo-labeling which predicts artificial labels for unlabeled data and uses confident predictions with labeled data loss. While standard pseudo-labeling does not help the OOD detection performance further, pseudo-negative mining improves the seen AUROC by 4.73% over the model without pseudo-negative mining. Compared to standard pseudo-labeling, pseudo-negative mining not only includes more unlabeled data into the training, but also presents high precision for the selected pseudo-outliers as shown in Fig. 7.6.

As mentioned in Section 7.3.4, we utilize unlabeled data with low inlier scores as pseudo-outliers to enhance the data diversity of outlier classes. An unlabeled sample is used as a

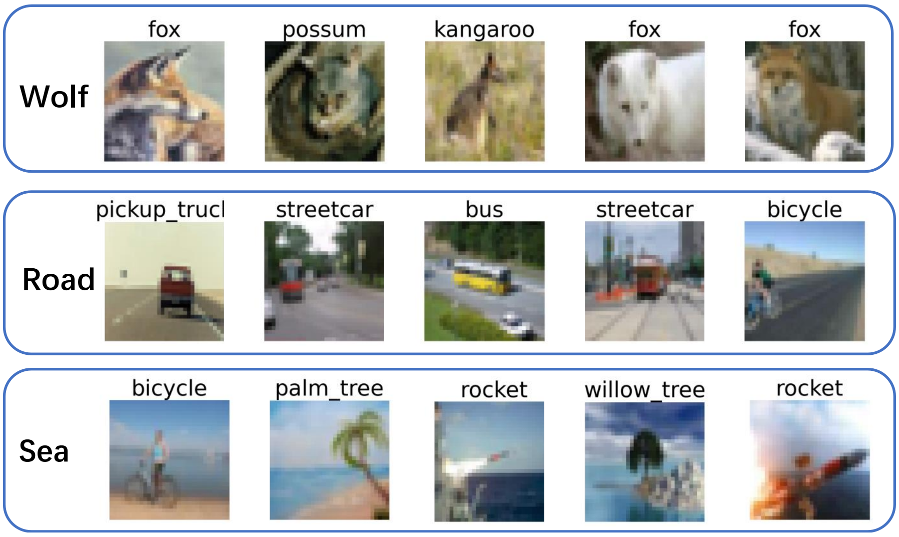


Figure 7.5: **OOD samples can be used as data augmentation to improve the generalization performance.** The figure shows three semantic classes from labeled data (wolf, road, and sea), and top-5 confident OOD samples predicted for those classes. The ground-truth semantic class of the OOD sample is on the top of each image. We can see that OOD data with high confidence present large visual similarities to the corresponding semantic classes.

pseudo-outlier only if its confidence score is less than a pre-defined threshold θ. Table 7.4 compares the results of different thresholds. We can see that our method achieves similar performance as long as θ takes a relatively small value, which suggests the good robustness of our method against this hyper-parameter.

Threshold θ	Inlier Cls. (Acc.)	Outlier Det. (seen AUROC)
0.2	91.87	92.96
0.1	**92.03**	93.16
0.05	91.97	94.21
0.01	91.65	**94.76**
0.005	91.52	94.75
0.001	91.70	94.15

Table 7.4: **Effect of different thresholds θ for pseudo-negative mining.** Our method shows good robustness against a wide range of thresholds. We use CIFAR-10 with 25 labeled data here.

Ablation on outlier detectors. Here, we compare the performance of different outlier detection methods. Specifically, we choose three schemes from recent works, including the binary classifier from MTC [YIIA20a], cross-modal matching from T2T [HFC$^+$21], and OVA classifiers from OpenMatch [SKS21]. As shown in Table 7.6. While all methods show reasonable performance, OVA classifiers exhibit the best performance in both inlier classification and OOD detection. Hence, we use OVA classifiers as the outlier detector in our final model.

Compatibility with other open-set SSL methods. We evaluated the compatibility of our

Pseudo-labeling	Inlier Cls. (Acc.)	Outlier Det. (AUROC)
None	91.52	90.03
Standard	91.63	89.69
Pseudo-neg.	**91.65**	**94.76**

Table 7.5: **Effect of pseudo-negative mining for OOD detection.** All models are trained with confidence-based pseudo-labeling and a 2-layer MLP projection head on the same data split.

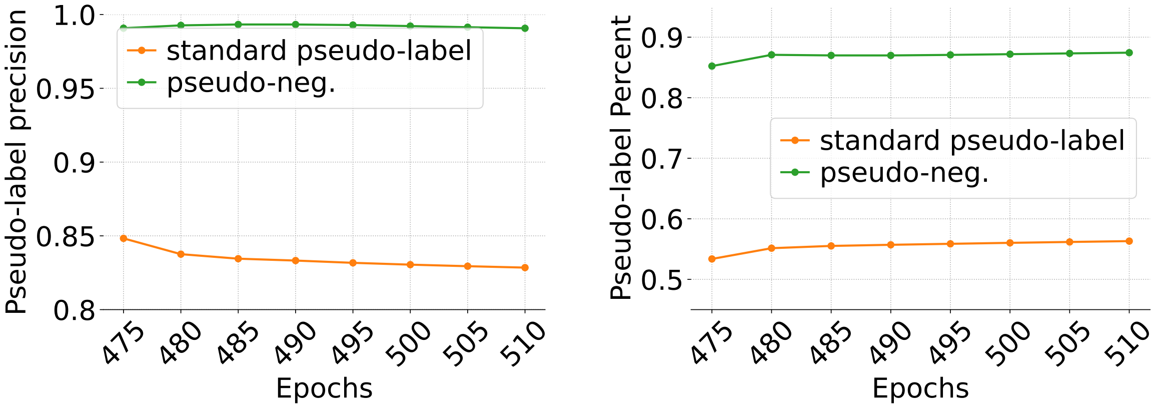


Figure 7.6: Compared to standard pseudo-labeling, pseudo-negative mining has not only higher prediction precision, but also higher data utilization rate.

method with other open-set SSL techniques in Table 7.7. Our results indicate that our method is highly compatible, as all existing methods showed improved performance in both inlier classification and outlier detection when combined with our approach. This demonstrates the flexibility of our method and suggests that it can be easily integrated into existing frameworks as a plug-and-play solution.

Equal-parameter comparison. As mentioned in Section 7.4.1, the performance improvement of SSB can not be simply explained by the increased number of parameters introduced in the projection heads. Here we compare SSB with other methods + MLP heads so that they have the same number of parameters as SSB. As shown in Fig. 7.7, adding MLP heads improves the performance of other methods, but SSB still greatly outperforms all of them, indicating that the performance improvement of our method can not be merely explained by the increase of the model capacity.

OOD Detector	Inlier Cls. (Acc.)	Outlier Det. (AUROC)
binary classifier [YIIA20a]	70.93	76.12
cross-modal matching [HFC+21]	69.27	75.99
OVA classifiers [SKS21]	**71.00**	**82.62**

Table 7.6: **Comparison between different outlier detectors.** The experiment is conducted on CIFAR-100 with 55 inlier classes and 25 labels per class.

	Inlier Cls. (Acc.)	Outlier Det. (AUROC)
MTC	60.24	69.88
MTC + Ours	**60.42**	**74.98**
T2T	64.78	52.93
T2T + Ours	**66.98**	**69.50**
OpenMatch	68.53	80.00
OpenM. + Ours	**71.00**	**82.62**

Table 7.7: **Integrating our method with other open-set SSL methods improves performance.** The setting is CIFAR-100 with 55 inlier classes and 25 labels per class.

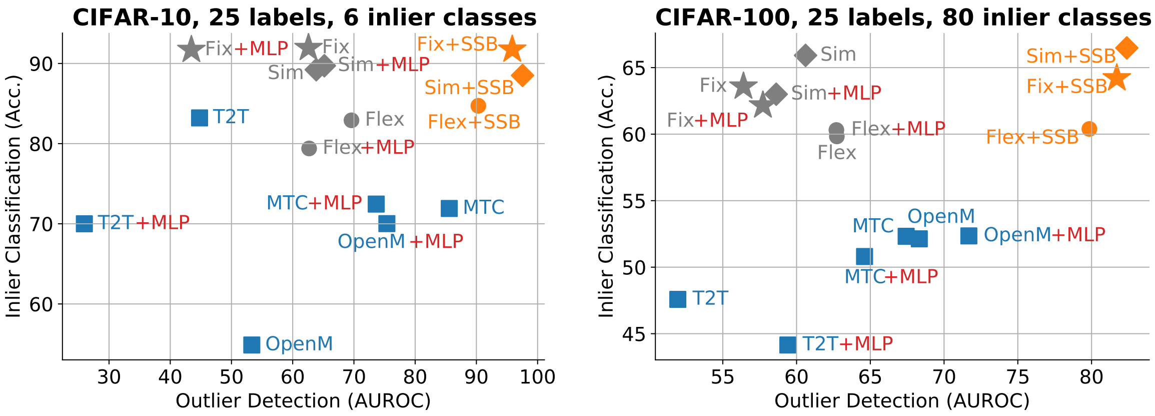


Figure 7.7: **Comparison between SSB and other methods with the same model parameters.** The performance improvement of SSB can not be simply explained by the increased number of parameters.

7.5 Conclusion and Limitations

In this chapter, we study a realistic and challenging setting, open-set SSL, where unlabeled data contains outliers from categories that do not appear in the labeled data. We first demonstrate that classifier-confidence-based pseudo-labeling can effectively improve the unlabeled data utilization ratio and leverage useful OOD data, which largely improves the classification performance. We find that adding non-linear transformations between the task-specific head and the shared features provides sufficient decoupling of the two heads, which prevents mutual interference and improves performance in both tasks. Additionally, we propose pseudo-negative mining to improve OOD detection. It uses pseudo-outliers to enhance the representation learning of OOD data, which further improves the model's ability to distinguish between inliers and OOD samples. Overall, we achieve state-of-the-art performance on several benchmark datasets, demonstrating the effectiveness of the proposed method.

Nonetheless, SSB has potential limitations. Despite the improved overall performance, the outlier detector suffers from overfitting as the performance gap between detecting seen outliers and unseen outliers is still very large. Therefore, in the future, more regularizations need to be considered to improve the generalization. Another drawback is that our method is not able to deal with long-tail distributions, which is also very realistic in practice. Presumably, our method will have difficulty distinguishing inliers of tail classes and OOD data due to the data scarcity at tail.

A Unified Semi-supervised Learning Benchmark for Classification 8

Contents

While the previous chapters focus on the data perspective of realistic semi-supervised learning (SSL), this chapter approaches it from the evaluation standpoint. Currently, popular SSL evaluation protocols are often constrained to computer vision (CV) tasks. In addition, previous work typically trains deep neural networks from scratch, which is time-consuming and environmentally unfriendly. To address the above issues, we construct a Unified SSL Benchmark (USB) for classification by selecting 15 diverse, challenging, and comprehensive tasks from CV, natural language processing (NLP), and audio processing (Audio), on which we systematically evaluate the dominant SSL methods, and also open-source a modular and extensible codebase for fair evaluation of these SSL methods. We further provide the pre-trained versions of the state-of-the-art neural models for CV tasks to make the cost affordable for further tuning. USB enables the evaluation of a single SSL algorithm on more tasks from multiple domains but with less cost. Specifically, on a single NVIDIA V100, only 39 GPU days are required to evaluate FixMatch on 15 tasks in USB while 335 GPU days (279 GPU days on 4 CV datasets except for ImageNet) are needed on 5 CV tasks with TorchSSL.

This chapter is based on [WCF+22a]. Yue Fan, as the co-first author, was involved in the idea proposal, weekly and more detailed discussions, and contributed to the codebase implementation and the final paper writing.

8.1 Introduction

Neural models give competitive results when trained using supervised learning on sufficient high-quality labeled data [HZRS16b, RDGF16, HS97, VSP+17, YD16, GQC+20, VQK+21]. However, it can be laborious and expensive to obtain abundant annotations for model training [RDS+15b, WSM+19]. To address this issue, **semi-supervised learning (SSL)** emerges as

Domain & Backbone	Dataset	Classification Task	Hours × Settings × Seeds	Total GPU Hours	Total GPU Hours w/o ImageNet
CV, ResNets	CIFAR-10	Natural Image	110 × 3 × 3	8031 GPU Hours (335 GPU Days)	6687 GPU Hours (279 GPU Days)
	CIFAR-100	Natural Image	300 × 3 × 3		
	SVNH	Digital	108 × 3 × 3		
	STL-10	Natural Image	225 × 3 × 3		
	ImageNet	Natural Image	336 hours × 4 GPUs		

(a) TorchSSL [ZWH+21]

Domain & Backbone	Dataset	Classification Task	Hours × Settings × Seeds	Total GPU Hours
CV, ViTs	CIFAR-100	Natural Image	11 × 2 × 3	924 GPU Hours (39 GPU Days)
	STL-10	Natural Image	18 × 2 × 3	
	EuroSAT	Satellite Image	10 × 2 × 3	
	TissueMNIST	Medical Image	8 × 2 × 3	
	Semi-Aves	Fine-grained, Long-tailed Natural Image	13 × 1 × 3	
NLP, Bert	IMDB	Movie Review Sentiment	8 × 2 × 3	
	AG News	News Topic	6 × 2 × 3	
	Amazon Review	Product Review Sentiment	8 × 2 × 3	
	Yahoo! Answer	QA Topic	7 × 2 × 3	
	Yelp Review	Restaurant Review Sentiment	8 × 2 × 3	
Audio, Wave2Vec 2.0 and HuBert	GTZAN	Music Genre	12 × 2 × 3	
	UrtraSound8k	Urban Sound Event	15 × 2 × 3	
	FSDnoisy18k	Sound Event	17 × 1 × 3	
	Keyword Spotting	Keyword	10 × 2 × 3	
	ESC-50	Environmental Sound Event	18 × 2 × 3	

(b) USB

Table 8.1: A summary of datasets and training cost used in (a) the existing popular protocol and (b) USB. USB largely reduces the training cost while providing a diverse, challenging, and comprehensive benchmark covering a wide range of datasets from various domains. Training cost is estimated by using FixMatch [SBL+20] on a single NVIDIA V100 GPU from Microsoft Azure Machine Learning platform, except for ImageNet where 4 V100s are used. Experiments in (a) follow the settings in [ZWH+21].

an effective paradigm to improve model generalization with limited labeled data and massive unlabeled data [RVR18, Zhu05b, ZG09b, VEH20, OHT20, QL20].

SSL has made remarkable progress in recent years [ZOKB19, LSH20a, CKS+20b, PDXL21, SBL+20, ZWH+21], yet there are still several limitations with the popular evaluation protocol in the literature [OOR+18, SBL+20, ZWH+21]. First, existing benchmarks are mostly constrained to plain computer vision (CV) tasks (i.e., CIFAR-10/100, SVHN, STL-10, and ImageNet classification [OOR+18, BCC+20, SBL+20, XSY+21, ZWH+21], as summarized in TorchSSL [ZWH+21]), precluding consistent and diverse evaluation over tasks in natural language processing (NLP), audio processing (Audio), etc., where the lack of labeled data is a general issue and SSL has gained increasing research attention recently [CYY20, BWA+19, CLP22]. Second, the existing protocol (e.g., TorchSSL [ZWH+21]) can be mainly time-consuming and environmentally unfriendly because it typically trains deep neural models from scratch [BCG+19, BCC+20, XDH+20b, SBL+20, XSY+21, ZWH+21]. Specifically, as shown in Table 8.1a, it takes about 335 GPU days (279 GPU days without ImageNet) to evaluate FixMatch [SBL+20] with TorchSSL [ZWH+21]. Such a high cost can make it unaffordable for research labs (particularly in academia) to conduct SSL research. Recently, the pre-training and fine-tuning paradigm [DCLT18b, LOG+19, HBT+21, HCX+21] achieves promising results. Compared with training from scratch, pre-training has much reduced cost in SSL. However, there are relatively few benchmarks that offer a fair test bed for SSL with the pre-trained versions of neural models.

To address the above issues and facilitate general SSL research, we propose **USB: a Unified**

SSL Benchmark for classification [1]. USB offers a *diverse* and *challenging* benchmark across five CV datasets, five NLP datasets, and five Audio datasets (Table 8.1b), enabling consistent evaluation over multiple tasks from different domains. Moreover, USB provides comprehensive evaluations of SSL algorithms with even fewer labeled data compared with TorchSSL, as the performance gap between SSL algorithms diminishes when the amount of labeled samples becomes large. Benefiting from the rapidly developed neural architectures, we introduce pre-trained Transformers [VSP$^+$17] into SSL instead of training ResNets [HZRS16b] from scratch to reduce the training cost for CV tasks. Specifically, we find that using pre-trained Vision Transformers (ViT) [DBK$^+$20] can largely reduce the number of training iterations (e.g., by 80% from 1,000k to 200k on CV tasks) without hurting the performance, and most SSL algorithms achieve even better performance with less training iterations.

As illustrated in Table 8.1b, using USB, we spend only **39 GPU days** to evaluate the performance of an SSL algorithm (i.e., FixMatch) on a single NVIDIA V100 over these **15 datasets**, in contrast to TorchSSL, which costs about **335 GPU days** on only **5 CV datasets** (279 GPU days on 4 CV datasets except for ImageNet). To further facilitate SSL research, we open-source the codebase and pre-trained models [2] for unified and consistent evaluation of SSL methods. In addition, we also provide config files that contain all the hyper-parameters to easily reproduce our results reported in this work. We obtain some interesting findings by evaluating 14 SSL algorithms (Section 8.5.4): (1) introducing diverse tasks from diverse domains can be beneficial to comprehensive evaluation of an SSL algorithm; (2) pre-training is more efficient and can improve the generalization; (3) unlabeled data do not consistently improve the performance especially when labeled data is scarce.

To conclude, our contributions are three-fold:

- We propose USB: a unified and challenging semi-supervised learning benchmark for classification with 15 tasks on CV, NLP, and Audio for fair and consistent evaluations. To our humble knowledge, we are the first to discuss whether current SSL methods that work well on CV tasks generalize to NLP and Audio tasks.
- We provide an environmentally friendly and low-cost evaluation protocol with pre-training & fine-tuning paradigm, reducing the cost of SSL experiments. The advantages of USB as compared to other related benchmarks are shown in Table 8.2.
- We implement 14 SSL algorithms and open-source a modular codebase and config files for easy reproduction of the reported results in this work. we also provide documents and tutorials for easy modification. Our codebase is extensible and open for continued development through community effort, where we expect new algorithms, models, config files and results are constantly added.

8.2 Related Work

Deep semi-supervised learning originates from Π model [RBH$^+$15b], where it solves the task of image classification by using consistency regularization that forces the model to output similar predictions when fed two augmented versions of the same unlabeled data. Subsequent methods can be classified as the variants of Π model, where the difference lies in enforcing the

[1] The word 'unified' means the unification of different algorithms on various application domains.

[2] https://github.com/microsoft/Semi-supervised-learning. We also provide the training logs of the experiments in this chapter. Note that the results and training logs will be continuously updated/provided if we reorganize the codes for better use or add more algorithms and datasets. Microsoft Research Asia (MSRA) will provide both the support and resources for future updates.

Benchmark	# SSL algorithms	Domian	# Tasks	Pre-trained	Training hours using FixMatch
Realistic SSL evaluation [OOR+18]	4	CV	3	✗	-
TorchSSL [ZWH+21]	9	CV	5	✗	6687
USB	14	CV, NLP, Audio	15	✓	924

Table 8.2: The comparison between USB and other related benchmarks.

consistency between model perturbation [TV17b], data perturbation [MMKI18a, XDH+20b], and exploiting unlabeled data [SBL+20, ZWH+21]. Since the best results in both CV and NLP are given by such algorithms, we choose them as typical representative methods in USB. While most SSL methods have seen their use in CV tasks, NLP has witnessed recent growth in SSL solutions [XDH+20b, CYY20]. However, only some of the popular methods [XDH+20b] in CV have been used in the NLP literature, probably because other methods give lower results or have not been investigated. This gives us motivation for evaluation of SSL methods on various domains in USB.

As shown in Table 2, related benchmarks include Realistic SSL evaluation [OOR+18] and TorchSSL [ZWH+21]. Realistic SSL evaluation [OOR+18] has 4 SSL algorithms and 3 CV classification tasks and TorchSSL has 9 SSL algorithms and 5 CV classification tasks. Both of them are no longer maintained/updated. Thus it is of significance to build an SSL community that can continuously update SSL algorithms and neural models to boost the development of SSL. Besides, previous benchmarks mainly train the models from scratch, which is computation expensive and time consuming, since SSL algorithms are known to be difficult to converge [AFIW18]. In USB, we consider using pre-trained models to boost the performance while being more efficient and friendly to researchers.

In the following, we will first introduce the tasks, datasets, algorithms, and benchmark results of USB. Then, the codebase structure of USB will be presented in Section 8.6.

8.3 Tasks and Datasets

USB consists of 15 datasets from CV, NLP, and Audio domains. Every dataset in USB is under a permissive license that allows usage for research purposes. The datasets are chosen based on the following considerations: (1) the tasks should be diverse and cover multiple domains; (2) the tasks should be challenging, leaving room for improvement; (3) the training is reasonably environmentally friendly and affordable to research labs (in both the industry and academia).

8.3.1 CV Tasks

The details of the CV datasets are shown in Table 8.3. We include CIFAR-100 [KH+09b] and STL-10 [CNL11b] from TorchSSL since they are still challenging. The TissueMNIST [YSN21, YSW+21], EuroSAT [HBDB19, HBDB18], and Semi-Aves [SM21b] are datasets in the domains of medical images, satellite images, and fine-grained natural images. CIFAR-10 [KH+09b] and SVHN [NWC+11a] in TorchSSL are not included in USB because the state-of-the-art SSL algorithms [XDH+20b, SBL+20, XSY+21] have achieved similar performance on these datasets to fully-supervised training with abundant fully labeled training data [3]. SSL algorithms have a relatively large room for improvement on all chosen CV datasets in USB.

[3] We highly recommend reporting ImageNet [RDS+15b] results since it is a reasonable dataset for hill-climbing [SBL+20, ZYH+22, ZWH+21].

Domain	Dataset	#Label per class	#Training data	#Validation data	#Test data	#Class
CV	CIFAR-100	2 / 4	50,000	-	10,000	100
	STL-10	4 / 10	5,000 / 100,000	-	8,000	10
	EuroSat	2 / 4	16,200	-	5,400	10
	TissueMNIST	10 / 50	165,466	-	47,280	8
	Semi-Aves	15-53	5,959 / 26,640	-	4,000	200
NLP	IMDB	10 / 50	23,000	2,000	25,000	2
	Amazon Review	50 / 200	250,000	25,000	65,000	5
	Yelp Review	50 / 200	250,000	25,000	50,000	5
	AG News	10 / 50	100,000	10,000	7,600	4
	Yahoo! Answer	50 / 200	500,000	50,000	60,000	10
Audio	Keyword Spotting	5 / 20	18,538	2,577	2,567	10
	ESC-50	5 / 10	1,200	400	400	50
	UrbanSound8k	10 / 40	7,079	816	837	10
	FSDnoisy18k	52-171	1,772 / 15,813	-	947	20
	GTZAN	10 / 40	7,000	1,500	1,500	10

Table 8.3: Details of the datasets in USB. Two *#Label per class* settings are chosen for each dataset except Semi-Aves and FSDnoisy18k, which have long-tailed distributed data. Labeled data are sampled from the training data for each dataset except STL-10, Semi-Aves, and FSDNoisy18k, where the split of labeled and unlabeled data is pre-defined (e.g. 5,959 labeled images and 26,640 unlabeled images in Semi-Aves). Following [SBL$^+$20, ZWH$^+$21], validation data are not provided for CV datasets. The NLP validation data are sampled from the original training datasets. All test sets are kept unchanged.

8.3.2 NLP Tasks

The detailed dataset statistics of NLP tasks in USB are described in Table 8.3. We mostly followed previous work in the NLP literature, and thus the existing datasets in USB cover most test sets used in the existing work [CYY20, LLO21, XDH$^+$20b]. We include widely used IMDB [MDP$^+$11], AG News [ZZL15], and Yahoo! Answer [CRRS08] from the previous protocol [CYY20, LLO21, XDH$^+$20b], which are still challenging for SSL. Since IMDB is a binary sentiment classification task, we further add Amazon Review [ML13] and Yelp Review [Asg16] to evaluate SSL algorithms on more fine-grained sentiment classification tasks. DBpedia is removed from the previous protocol [CYY20, LLO21, XDH$^+$20b] because we find that the state-of-the-art SSL algorithms have achieved similar performance on it when compared with fully-supervised training. For all tasks in NLP, we obtain the labeled datasets, unlabeled datasets, and validation sets by randomly sampling from their original training datasets while keeping the original test datasets unchanged, mainly following previous work [CYY20, LLO21].

8.3.3 Audio Tasks

USB includes five audio classification datasets as shown in Table 8.3. We choose the tasks to cover different domains such as urban sound (UrbanSound8k [SJB14], ESC-50 [Pic15], and FSDNoisy18k [FPE$^+$19]), human sound (Keyword Spotting [YCC$^+$21]), and music (GTZAN) [Stu13]. All chosen datasets are challenging even for state-of-the-art SSL algorithms. For example, FSDNoisy18k is a realistic dataset containing a small labeled set and a large unlabeled set. To the best of our knowledge, we are the first to systematically evaluate SSL algorithms on Audio tasks. Although there is a concurrent work [CLP22], our study includes more algorithms and more datasets than [CLP22].

Algorithm	PL	CR Loss	Thresholding	Dist. Align.	Self-supervised	Mixup	W-S Aug.
Π-Model		MSE					
Pseudo Labeling	✓	CE					
Mean Teacher		MSE					
VAT		CE					
MixMatch		MSE				✓	
ReMixMatch		CE		✓	Rotation	✓	✓
UDA		CE	✓				✓
FixMatch	✓	CE	✓				✓
Dash	✓	CE	✓				✓
CoMatch	✓	CE	✓	✓	Contrastive		✓
CRMatch	✓	CE	✓		Rotation		✓
FlexMatch	✓	CE	✓				✓
AdaMatch	✓	CE	✓	✓			✓
SimMatch	✓	CE	✓	✓	Contrastive		✓

Table 8.4: Essential components used in 14 SSL algorithms supported in USB. PL, CR, Dist. Align., and W-S Aug., MSE, CE are the abbreviations for Pseudo Labeling, Consistency Regularization, Distribution Alignment, Weak-Strong Augmentation, Mean Squared Error, and Cross-Entropy, respectively. PL denotes hard 'one-hot' labels adopted in CR Loss.

8.4 SSL Algorithms

We implement 14 SSL algorithms in the codebase for USB, including Π model [RBH$^+$15b], Pseudo Labeling [Lee13], Mean Teacher [TV17b], VAT [MMKI18a], MixMatch [BCG$^+$19], ReMixMatch [BCC$^+$20], UDA [XDH$^+$20b], FixMatch [SBL$^+$20], Dash [XSY$^+$21], CoMatch [LXH21], CRMatch [FKS21b], FlexMatch [ZWH$^+$21], AdaMatch [BRS$^+$22], and SimMatch [ZYH$^+$22], all of which exploit unlabeled data by encouraging invariant predictions to input perturbations [VEH20, OHT20, YSKX21, Spr15, DGF16b, DYY$^+$17, KSF17]. Such consistency regularization methods give the strongest performance in SSL since the model is robust to different perturbed versions of unlabeled data, satisfying the smoothness and low-density assumptions in SSL [CSZ09].

The above SSL algorithms use Cross-Entropy (CE) loss on labeled data but differ in the way on unlabeled data. As shown in Table 8.4, Pseudo Labeling [Lee13] turns the predictions of the unlabeled data into hard 'one-hot' labels and treats the 'one-hot' pseudo-labels as the supervision signals. Thresholding reduces the noisy pseudo labels by masking out the unlabeled samples whose maximum probabilities are smaller than the pre-defined threshold. Distribution Alignment aims to correct the output distribution to make it more in line with the target distribution (e.g., uniform distribution). Self-supervised learning, Mixup, and Stronger augmentations techniques also can help learn better representation. We summarize the key components exploited in the implemented consistency regularization based algorithms in Table 8.4.

8.5 Benchmark Results

[4] For CV tasks, we follow [ZWH$^+$21] to report the best number of all checkpoints to avoid unfair comparisons caused by different convergence speeds. For NLP and Audio tasks, we choose the best model using the validation datasets and then evaluate it on the test datasets.

[4] **Note that all experimental results and training logs will be continuously updated in https://github.com/microsoft/Semi-supervised-learning. Please refer to the latest results for comparison.**

Dataset	CIFAR-100		STL-10		Euro-SAT		TissueMNIST		Semi-Aves	Friedman	Final	Mean
# Label	200	400	20	40	20	40	80	400	5,959	rank	rank	error rate
Fully-Supervised	8.44±0.07	8.44±0.07	-	-	0.94±0.07	0.89±0.05	29.15±0.13	29.10±0.02	-	-	-	-
Supervised	35.63±0.36	26.08±0.50	47.02±1.48	26.02±0.72	27.12±1.26	16.90±1.48	59.91±2.93	54.10±1.52	41.55±0.29	-	-	-
Π-model	36.24±0.27	26.49±0.64	44.38±1.59	25.76±2.37	24.51±1.02	11.58±1.32	56.79±5.91	**47.50±1.71**	39.23±0.36	10.11	11	34.72
Pseudo-Labeling	33.16±1.20	25.29±0.67	45.13±4.08	26.20±1.53	23.64±0.90	15.61±2.51	56.22±4.01	50.36±1.62	40.13±0.09	9.89	10	35.08
Mean Teacher	35.61±0.38	25.97±0.37	39.94±1.99	20.16±1.25	26.51±1.15	17.05±2.07	61.40±2.48	55.22±2.06	38.52±0.27	10.89	14	35.60
VAT	31.61±1.37	21.29±0.32	52.03±0.48	23.10±0.72	24.77±1.94	9.30±1.23	58.50±6.41	51.31±1.66	39.00±0.30	10.11	12	34.55
MixMatch	37.43±0.58	26.17±0.24	48.98±1.41	25.56±3.00	29.86±2.89	16.39±3.17	55.73±2.29	49.08±1.06	37.22±0.15	10.11	12	36.27
ReMixMatch	**20.85±1.42**	**16.80±0.59**	30.61±3.47	18.33±1.98	**4.53±1.60**	**4.10±0.37**	59.29±5.16	52.92±3.93	**30.40±0.33**	4.00	1	26.43
UDA	30.75±1.03	19.94±0.32	39.22±2.87	23.59±2.97	11.15±1.20	5.99±0.75	55.88±3.26	51.42±2.05	32.55±0.26	6.89	7	30.05
FixMatch	30.45±0.65	19.48±0.93	42.06±3.94	24.05±1.79	12.48±2.57	6.41±1.64	55.95±4.06	50.93±1.23	31.74±0.33	6.56	6	30.39
Dash	30.19±1.34	18.90±0.42	43.34±1.46	25.90±0.35	9.44±0.75	7.00±1.39	57.00±2.81	50.93±1.54	32.56±0.39	7.44	9	30.58
CoMatch	35.68±0.54	26.10±0.09	**29.70±1.17**	21.46±1.34	5.25±0.49	4.89±0.86	57.15±3.46	51.83±0.71	41.39±0.16	7.22	8	30.38
CRMatch	29.43±1.11	18.50±0.26	30.55±2.01	**17.43±1.96**	14.52±1.34	7.00±0.69	**54.84±3.05**	51.10±1.59	31.97±0.10	4.67	2	28.37
FlexMatch	27.08±0.90	17.67±0.66	37.58±2.97	23.40±1.50	7.07±2.32	5.58±0.57	57.23±2.50	52.06±1.78	33.09±0.16	6.44	5	28.97
AdaMatch	21.27±1.04	17.01±0.55	36.25±1.89	23.30±0.73	5.70±0.37	4.92±0.87	57.87±4.47	52.28±0.79	31.54±0.10	5.22	3	27.79
SimMatch	23.26±1.25	16.82±0.40	34.12±1.63	22.97±2.04	6.88±1.77	5.86±1.07	57.91±4.60	51.14±1.83	34.14±0.30	5.44	4	28.12

Table 8.5: Error rate (%) and Rank with CV tasks in USB. For Semi-Aves and STL10, as they have unlabeled sets, we do not report the fully-supervised results. We follow [SBL+20, ZWH+21, XDH+20b] to show error rates as default.

In addition to mean error rate over the tasks, we use Friedman rank [Fri37, Fri40] to fairly compare the performance of different algorithms in various settings:

$$\text{rank}_F = \frac{1}{m}\sum_{i=1}^{m}\text{rank}_i,$$

where m is the number of evaluation settings (i.e., how many experimental settings we use, e.g., $m = 9$ in Table 8.5), and rank_i is the rank of an SSL algorithm in the i-th setting. We re-rank all algorithms to give final ranks based on their Friedman rankings. Note that all ranks are in ascending order because the lower error rate corresponds to a better performance. Note that 'supervised' denotes training with the partially chosen labeled data while 'fully-supervised' refers to training using all data with full annotations in our reported results.

The results for the 14 SSL algorithms on the datasets from CV, NLP, and Audio are shown in Table 8.5, Table 8.6, and Table 8.7, respectively. We adopt the pre-trained Vision Transformers (ViT) [VSP+17, DBK+20, DCLT18b, BZMA20] instead of training ResNets [HZRS16b] from scratch for CV tasks. For NLP, we adopt Bert [DCLT18b]. Wav2Vec 2.0 [BZMA20] and HuBert [HBT+21] are used for Audio.

8.5.1 CV Results

The results are illustrated in Table 8.5. Thanks to the good initialization of representation on unlabeled data given by the pre-trained ViT, SSL algorithms, even without using thresholding techniques, often achieve much better performance than the previous performance shown in TorchSSL [ZWH+21]. Among all the SSL algorithms, ReMixMatch [BCC+20] ranks at the first and outperforms other SSL algorithms, due to the usage of Mixup, Distribution Alignment, and rotation self-supervised loss. Its superiority is especially demonstrated in the evaluation of Semi-Aves, a long-tailed and fine-grained CV dataset that is more realistic. Notice that SSL algorithms with self-supervised feature loss generally perform well than other SSL algorithms, e.g., CRMatch [FKS21b] and SimMatch [ZYH+22] rank second and fourth respectively. Adaptive thresholding algorithms also demonstrate their effectiveness, e.g., AdaMatch [BRS+22] and FlexMatch [ZWH+21] rank at third and fifth respectively. While better

Dataset	IMDB		AG News		Amazon Review		Yahoo! Answer		Yelp Review		Friedman	Final	Mean
# Label	20	100	40	200	250	1000	500	2000	250	1000	rank	rank	error rate
Fully-Supervised	5.87$_{\pm0.01}$	5.84$_{\pm0.12}$	5.74$_{\pm0.30}$	5.64$_{\pm0.05}$	36.81$_{\pm0.05}$	36.88$_{\pm0.19}$	26.25$_{\pm1.07}$	25.55$_{\pm0.43}$	31.74$_{\pm0.23}$	32.70$_{\pm0.58}$	-	-	-
Supervised	20.63$_{\pm3.13}$	13.47$_{\pm0.55}$	15.01$_{\pm1.21}$	13.00$_{\pm1.00}$	51.74$_{\pm0.63}$	47.34$_{\pm0.66}$	37.10$_{\pm1.22}$	33.56$_{\pm0.08}$	50.27$_{\pm0.51}$	46.96$_{\pm0.42}$	-	-	-
Π-Model	49.02$_{\pm1.37}$	27.57$_{\pm15.85}$	46.84$_{\pm6.20}$	13.44$_{\pm0.76}$	73.53$_{\pm6.92}$	48.27$_{\pm0.48}$	41.37$_{\pm2.15}$	32.96$_{\pm0.16}$	73.35$_{\pm2.31}$	52.02$_{\pm1.48}$	11.80	12	45.84
Pseudo-Labeling	26.38$_{\pm4.04}$	21.38$_{\pm1.34}$	23.86$_{\pm7.63}$	12.29$_{\pm0.40}$	53.00$_{\pm1.48}$	46.49$_{\pm0.45}$	38.60$_{\pm1.09}$	33.44$_{\pm0.24}$	55.70$_{\pm0.95}$	47.72$_{\pm0.37}$	10.60	11	35.89
Mean Teacher	21.27$_{\pm3.72}$	14.11$_{\pm1.77}$	14.98$_{\pm1.10}$	13.23$_{\pm1.12}$	51.67$_{\pm0.45}$	47.51$_{\pm0.24}$	36.97$_{\pm1.02}$	33.43$_{\pm0.22}$	51.07$_{\pm1.44}$	46.61$_{\pm0.34}$	9.30	10	33.09
VAT	32.59$_{\pm4.69}$	14.42$_{\pm2.53}$	15.00$_{\pm1.12}$	11.59$_{\pm0.94}$	50.38$_{\pm0.83}$	46.04$_{\pm0.28}$	35.16$_{\pm0.74}$	31.53$_{\pm0.41}$	52.76$_{\pm0.87}$	45.53$_{\pm0.13}$	8.40	8	33.50
UDA	9.36$_{\pm1.26}$	8.33$_{\pm0.61}$	18.73$_{\pm2.68}$	12.34$_{\pm1.90}$	52.48$_{\pm1.20}$	45.51$_{\pm0.61}$	35.31$_{\pm0.43}$	32.01$_{\pm0.68}$	58.22$_{\pm0.40}$	42.18$_{\pm0.68}$	8.70	9	31.45
FixMatch	8.20$_{\pm0.29}$	**7.36**$_{\pm0.07}$	22.80$_{\pm5.18}$	11.43$_{\pm0.65}$	47.85$_{\pm1.22}$	43.73$_{\pm0.45}$	34.15$_{\pm0.94}$	30.76$_{\pm0.53}$	50.34$_{\pm0.40}$	41.99$_{\pm0.58}$	5.60	7	29.86
Dash	8.93$_{\pm1.27}$	7.97$_{\pm0.53}$	19.30$_{\pm6.73}$	11.20$_{\pm1.12}$	47.79$_{\pm1.03}$	43.52$_{\pm0.07}$	35.10$_{\pm1.36}$	30.51$_{\pm0.47}$	47.99$_{\pm1.05}$	41.59$_{\pm0.61}$	5.10	6	29.39
CoMatch	7.36$_{\pm0.26}$	7.41$_{\pm0.20}$	13.25$_{\pm1.31}$	11.61$_{\pm0.42}$	48.98$_{\pm1.20}$	44.37$_{\pm0.25}$	33.48$_{\pm0.67}$	**30.19**$_{\pm0.22}$	46.49$_{\pm1.42}$	41.11$_{\pm0.53}$	3.80	3	28.43
CRMatch	7.88$_{\pm0.24}$	7.68$_{\pm0.35}$	13.35$_{\pm1.06}$	11.36$_{\pm1.04}$	46.23$_{\pm0.85}$	43.69$_{\pm0.48}$	33.07$_{\pm0.68}$	30.62$_{\pm0.47}$	46.61$_{\pm1.02}$	41.80$_{\pm0.77}$	3.70	2	28.23
FlexMatch	7.35$_{\pm0.10}$	7.80$_{\pm0.24}$	16.90$_{\pm6.76}$	11.43$_{\pm0.91}$	**45.75**$_{\pm1.21}$	43.14$_{\pm0.82}$	35.81$_{\pm1.09}$	31.42$_{\pm0.41}$	46.37$_{\pm0.74}$	**40.86**$_{\pm0.74}$	4.10	5	28.68
AdaMatch	9.62$_{\pm1.26}$	7.81$_{\pm0.46}$	**12.92**$_{\pm1.53}$	**11.03**$_{\pm0.62}$	46.75$_{\pm1.23}$	43.50$_{\pm0.67}$	**32.97**$_{\pm0.43}$	30.82$_{\pm0.29}$	48.16$_{\pm0.80}$	41.71$_{\pm1.08}$	4.00	4	28.53
SimMatch	**7.24**$_{\pm0.02}$	7.44$_{\pm0.20}$	14.80$_{\pm0.57}$	11.12$_{\pm0.15}$	47.27$_{\pm1.73}$	**43.09**$_{\pm0.50}$	34.15$_{\pm0.91}$	30.64$_{\pm0.42}$	**46.40**$_{\pm1.71}$	41.24$_{\pm0.17}$	2.90	1	28.34

Table 8.6: Error rate (%) and Rank with NLP tasks in USB.

results of the evaluated SSL algorithms are obtained on CIFAR-100, Euro-SAT, and Semi-Aves, we also observe that the performance is relatively lower on STL-10 and TissueMNIST. The reason for lower performance on STL-10 might result from the usage of the self-supervised pre-trained model [HCX$^+$21], rather than the supervised pre-trained model is used in other settings. Since TissueMNIST is a medial-related dataset, the biased pseudo-labels might produce a destructive effect that impedes training and leads to bad performance. The debiasing of pseudo-labels and safe semi-supervised learning would be interesting topics in future work, especially for medical applications of SSL algorithms.

8.5.2 NLP Results

The results of NLP tasks are demonstrated in Table 8.6. The overall ranking of SSL algorithms in NLP is similar to that in CV. However, the SSL algorithm that works well in NLP does not always guarantee good performance in CV, which shows that the performance of SSL algorithms will be affected largely by data domains. For example, SimMatch which ranks first in NLP does not have the best performance in CV tasks (ranks fourth). The ranking of CoMatch is also increased in NLP, compared to that in CV. A possible reason is the different pre-training in backbones. For BERT, a masked language modeling objective is used during pre-training [DCLT18b], thus the self-supervised feature loss might further improve the representation during fine-tuning with SSL algorithms. We observe that adaptive thresholding methods, such as FlexMatch and AdaMatch, consistently achieve good performance on both CV and NLP, even without self-supervised loss. Note that we do not evaluate MixMatch and ReMixMatch on NLP and Audio tasks because we find that mixing sentences with different lengths harms the model's performance.

8.5.3 Audio Results

The results of Audio tasks are shown in Table 8.7. AdaMatch outperforms other algorithms in Audio tasks, while SimMatch demonstrates a similar performance to AdaMatch. An interesting finding is that CRMatch performs well on CV and NLP tasks, but badly in Audio tasks. We hypothesize that this is partially due to the noisy nature of the raw data in audio tasks. Except for Keyword Spotting, the gap between the performance of fully-supervised learning and that of SSL algorithms in Audio tasks is larger than in CV and NLP tasks. The reason behind this is probably that we exploit models that take waveform as input, rather than Mel

Dataset	GTZAN		UrbanSound8k		Keyword Spotting		ESC-50		FSDnoisy	Friedman	Final	Mean
# Label	100	400	100	400	50	100	250	500	1,772	rank	rank	error rate
Fully-Supervised	5.98$_{\pm0.32}$	5.98$_{\pm0.32}$	16.65$_{\pm1.71}$	16.61$_{\pm1.71}$	2.12$_{\pm0.11}$	2.25$_{\pm0.02}$	26.00$_{\pm2.13}$	26.00$_{\pm2.13}$	-	-	-	-
Supervised	52.16$_{\pm1.83}$	31.53$_{\pm0.52}$	40.42$_{\pm1.00}$	28.55$_{\pm1.90}$	6.80$_{\pm1.16}$	5.25$_{\pm0.56}$	51.58$_{\pm1.12}$	35.67$_{\pm0.42}$	35.20$_{\pm1.50}$	-	-	-
Π-Model	74.07$_{\pm0.62}$	33.18$_{\pm3.64}$	54.24$_{\pm6.01}$	25.89$_{\pm1.51}$	64.39$_{\pm4.10}$	25.48$_{\pm4.94}$	47.25$_{\pm1.14}$	36.00$_{\pm1.62}$	35.73$_{\pm0.87}$	10.67	12	44.03
Pseudo-Labeling	57.29$_{\pm2.80}$	33.93$_{\pm0.69}$	42.09$_{\pm2.41}$	27.00$_{\pm1.34}$	7.82$_{\pm1.64}$	5.16$_{\pm0.14}$	49.33$_{\pm2.52}$	35.58$_{\pm1.05}$	35.34$_{\pm1.60}$	10.00	10	32.62
Mean Teacher	51.40$_{\pm3.48}$	31.60$_{\pm1.46}$	41.70$_{\pm3.39}$	28.91$_{\pm0.93}$	5.95$_{\pm0.44}$	5.39$_{\pm0.42}$	50.25$_{\pm1.95}$	37.33$_{\pm1.20}$	35.83$_{\pm1.22}$	10.33	11	32.04
VAT	79.51$_{\pm1.99}$	35.38$_{\pm7.80}$	49.62$_{\pm2.42}$	27.68$_{\pm1.39}$	**2.18$_{\pm0.08}$**	**2.23$_{\pm0.08}$**	46.42$_{\pm1.90}$	36.92$_{\pm2.25}$	32.07$_{\pm1.05}$	8.33	9	34.67
UDA	46.56$_{\pm8.69}$	23.62$_{\pm0.63}$	37.28$_{\pm3.17}$	20.27$_{\pm1.58}$	2.52$_{\pm0.15}$	2.62$_{\pm0.10}$	42.75$_{\pm0.89}$	33.50$_{\pm1.95}$	30.80$_{\pm0.47}$	6.33	7	26.66
FixMatch	36.04$_{\pm4.57}$	22.09$_{\pm0.65}$	36.12$_{\pm4.26}$	21.43$_{\pm2.88}$	4.84$_{\pm3.57}$	2.38$_{\pm0.03}$	**37.75$_{\pm3.19}$**	30.67$_{\pm1.05}$	30.31$_{\pm1.08}$	4.00	3	24.63
Dash	47.00$_{\pm3.65}$	23.42$_{\pm0.83}$	42.02$_{\pm5.02}$	22.26$_{\pm0.89}$	5.70$_{\pm4.40}$	2.52$_{\pm0.16}$	48.17$_{\pm1.16}$	32.75$_{\pm2.27}$	33.19$_{\pm0.95}$	7.56	8	28.56
CoMatch	36.93$_{\pm1.23}$	22.20$_{\pm1.39}$	**30.59$_{\pm2.45}$**	21.35$_{\pm1.49}$	11.39$_{\pm0.85}$	9.44$_{\pm1.52}$	40.17$_{\pm2.08}$	**29.83$_{\pm1.31}$**	27.63$_{\pm1.35}$	5.11	6	25.50
CRMatch	40.58$_{\pm3.97}$	22.64$_{\pm1.22}$	39.47$_{\pm4.66}$	20.11$_{\pm2.63}$	2.40$_{\pm0.13}$	2.49$_{\pm0.08}$	42.67$_{\pm0.51}$	33.58$_{\pm1.93}$	30.45$_{\pm1.52}$	5.00	5	26.04
FlexMatch	34.60$_{\pm4.07}$	21.82$_{\pm1.17}$	40.18$_{\pm2.73}$	22.82$_{\pm3.10}$	2.42$_{\pm0.08}$	2.57$_{\pm0.25}$	39.58$_{\pm0.59}$	29.92$_{\pm1.85}$	**26.36$_{\pm0.55}$**	4.11	4	24.47
AdaMatch	**31.38$_{\pm0.41}$**	**20.73$_{\pm0.67}$**	35.76$_{\pm6.39}$	21.15$_{\pm1.22}$	2.49$_{\pm0.08}$	2.49$_{\pm0.10}$	39.17$_{\pm1.74}$	31.33$_{\pm1.23}$	27.95$_{\pm0.74}$	2.89	1	23.61
SimMatch	32.42$_{\pm2.18}$	20.80$_{\pm0.77}$	31.70$_{\pm6.05}$	**19.55$_{\pm1.89}$**	2.57$_{\pm0.08}$	2.53$_{\pm0.22}$	39.92$_{\pm2.35}$	32.83$_{\pm1.43}$	28.16$_{\pm0.87}$	3.67	2	23.39

Table 8.7: Error rate (%) and Rank with Audio tasks in USB. Fully-supervised result is not reported for FSDNoisy18k due to the unknown labels of its unlabeled set.

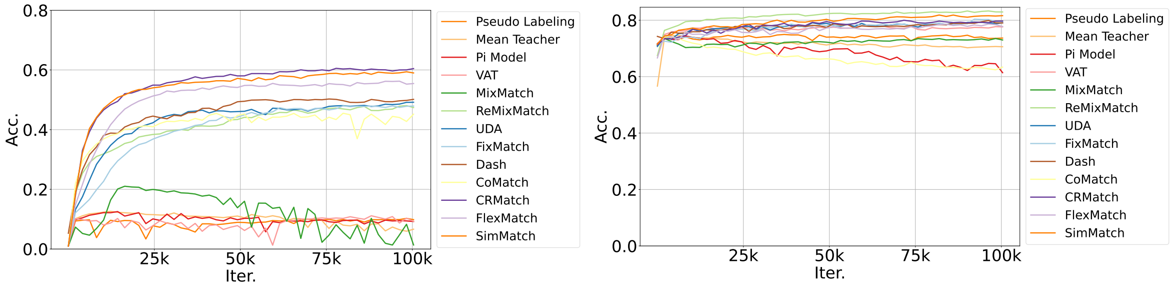
spectrogram. Raw waveform might contain more noisy information that would be harmful to semi-supervised training. We identify exploring audio models based on Mel spectrogram as one of the future directions of USB.

8.5.4 Discussion

The evaluation results of SSL algorithms using USB are generally consistent with the results reported by previous work [OOR$^+$18, BCG$^+$19, BCC$^+$20, XDH$^+$20b, SBL$^+$20, ZWH$^+$21]. However, using USB, we still provide some distinct quantitative and qualitative analysis to inspire the community. This section aims to answer the following questions: (1) Why should we evaluate an SSL algorithm on diverse tasks across domains? (2) Which option is better in the SSL scenario, training from scratch or using pre-training? (3) Does SSL consistently guarantee the performance improvement when using the state-of-the-art neural models as the backbones?

Performance Comparisons Table 8.8 shows the performance comparison of SSL algorithms in CV, NLP and Audio tasks. Although the ranking of each SSL algorithm in each domain is roughly close, the differences between ranks of SSL algorithms in different domains cannot be ignored. For example, FixMatch, CoMatch and CrMatch show large difference ($Rank_{max} - Rank_{min} \geq 4$) on the ranks across domains, which indicates that NLP and Audio tasks may have different characteristics compared with CV tasks that are more amenable to certain types of SSL algorithms compared with others. From the task perspective, it is important to consider such characteristics for guiding the choice of SSL methods. From the benchmarking perspective, it is useful to introduce diverse tasks from multiple domains when evaluating an SSL algorithm.

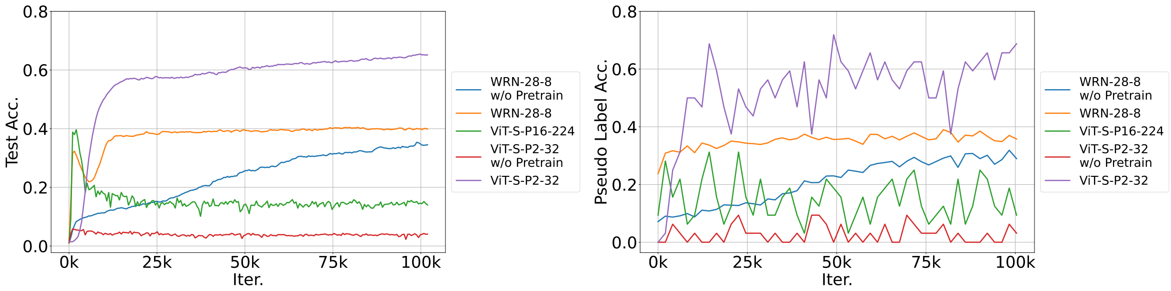
Effectiveness of Pre-training As shown in Figure 8.1 (a) and Figure 8.1 (b), benefiting from the pre-trained ViT, the training becomes more efficient, and most SSL algorithms achieve higher optimal performance. Note that Pseudo Labeling, Mean Teacher, Π model, VAT, and MixMatch barely converge if training WRN-28-8 from scratch. A possible reason is that the scarce labeled data cannot provide enough supervision for unlabeled data to form correct clusters. However, these methods can achieve sufficiently reasonable results when using pre-trained ViT. As illustrated in Figure 8.2, using ViT without pre-training performs the worst among different backbones. The reason can be that ViT is data hungry if trained from scratch [DBK$^+$20, HWC$^+$22, TCD$^+$21]. However, after appropriate pre-training, ViT performs the best among all the backbones. In addition, we provide the T-SNE visualization of the features in Figure 8.3, where the pretrained ViT model demonstrates the most separable feature



(a) WRN-28-8 from scratch.

(b) Pre-trained ViT-S-P2-32.

Figure 8.1: Comparison of test accuracy of SSL algorithms on CIFAR-100 with 400 labels. (a) Existing protocol which trains WRN-28-8 from scratch; (b) USB CV protocol which trains ImageNet-1K pre-trained ViT-S-P2-32, where S denotes small, P denotes patch size, and 32 is input image size.



(a) Test accuracy.

(b) Pseudo-label accuracy.

Figure 8.2: Pre-training ablation on CIFAR-400 with 400 labels. Test and pseudo-label accuracy are compared with WRN-28-8 without pre-training, pre-trained WRN-28-8, pre-trained ViT-S-P16-224, ViT-S-P2-32 without pre-training, and pre-trained ViT-S-P2-32.

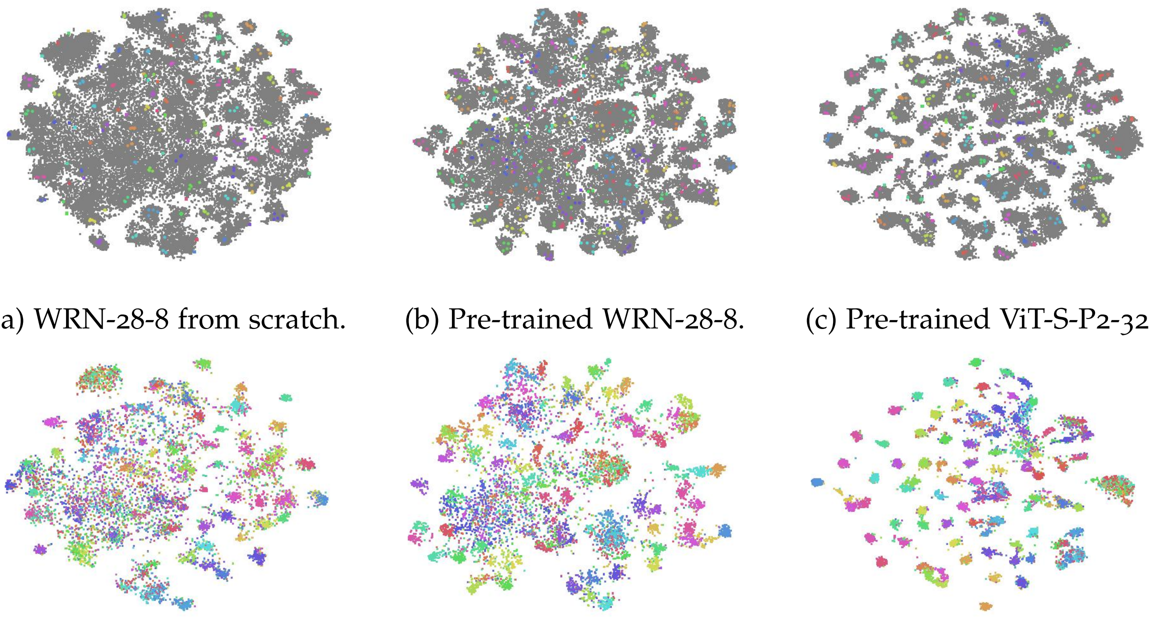
	Π-Model	Pseudo-Labeling	Mean Teacher	VAT	UDA	FixMatch	Dash	CoMatch	CRMatch	FlexMatch	AdaMatch	SimMatch
CV	10	9	12	11	6	5	8	7	1	4	2	3
NLP	12	11	10	8	9	7	6	3	2	5	4	1
Audio	12	10	11	9	7	3	8	6	5	4	1	2
$\text{Rank}_{max} - \text{Rank}_{min}$	2	2	2	3	3	4	2	4	4	1	3	2

Table 8.8: Final ranks of SSL algorithms. Note that the rank for CV tasks here is different from the ones in Table 8.5 because we ignore MixMatch and ReMixMatch here to remove the effects of their missing ranks in NLP and Audio.

	Π-Model	Pseudo-Labeling	Mean Teacher	VAT	MixMatch	ReMixMatch	UDA	FixMatch	Dash	CoMatch	CRMatch	FlexMatch	AdaMatch	SimMatch
CV	2	1	3	1	4	0	0	0	0	2	0	0	0	0
NLP	9	7	5	3	-	-	2	1	1	0	0	1	0	0
Audio	7	5	6	4	-	-	0	0	1	2	0	0	0	0

Table 8.9: This table shows how many times an SSL algorithm is worse than supervised training, where the numbers of total settings are 9, 10, and 9 for CV, NLP, and Audio respectively.

space after training. In a word, pre-trained ViT makes the training more efficient and improves the generalization performance of SSL algorithms. For NLP tasks, we observe similar results, yet the improvement can be relatively less significant since pre-training is the de-facto fashion in the field.



(a) WRN-28-8 from scratch.
(b) Pre-trained WRN-28-8.
(c) Pre-trained ViT-S-P2-32.
(d) WRN-28-8 from scratch.
(e) Pre-trained WRN-28-8.
(f) Pre-trained ViT-S-P2-32.

Figure 8.3: T-SNE visualization of FixMatch features on training data (first row) and testing data (second row) of CIFAR-100 (400 labels). Different colors refer to labeled data with different classes while unlabeled data is indicated by gray color.

Robustness SSL sometimes hurts the generalization performance due to the large differences between the number of labeled data and the number of unlabeled data as shown in Table 8.9. We refer to an SSL algorithm as a robust SSL algorithm if it is consistently better than the supervised training setting. SSL algorithms cannot always outperform supervised training especially when labeled data is scarce. We find that CRMatch, AdaMatch and SimMatch are relatively robust SSL algorithms in USB. Although previous work has done some research towards robust SSL when using support vector machine [LZ15, Nob06], we hope that our finding can serve as the motivation to delve into deep learning based robust SSL methods.

8.6 Codebase Structure of USB

In this section, we provide an overview of the codebase structure of USB, where four abstract layers are adopted. The layers include the core layer, algorithm layer, extension layer, and API layer in the bottom up direction as shown in Fig. 8.4.

Core Layer. In the core layer, we implement the commonly used core functions for training SSL algorithms. Besides, the code regarding datasets, data loaders, and models used in USB is also provided in the core layer. For flexible training, we implement common training hooks similar to MMCV [Con18], which can be modified and extended in the upper layers.

Algorithm Layer. In the algorithm layer, we first implement the base class for SSL algorithms, where we initialize the datasets, data loaders, and models from the core layer. Instead of implementing SSL algorithms independently as in TorchSSL [ZWH+21], we further abstract the SSL algorithms, enabling better code reuse and making it easier to implement new algorithms. Except for the standalone implementation of loss functions used in SSL algorithms

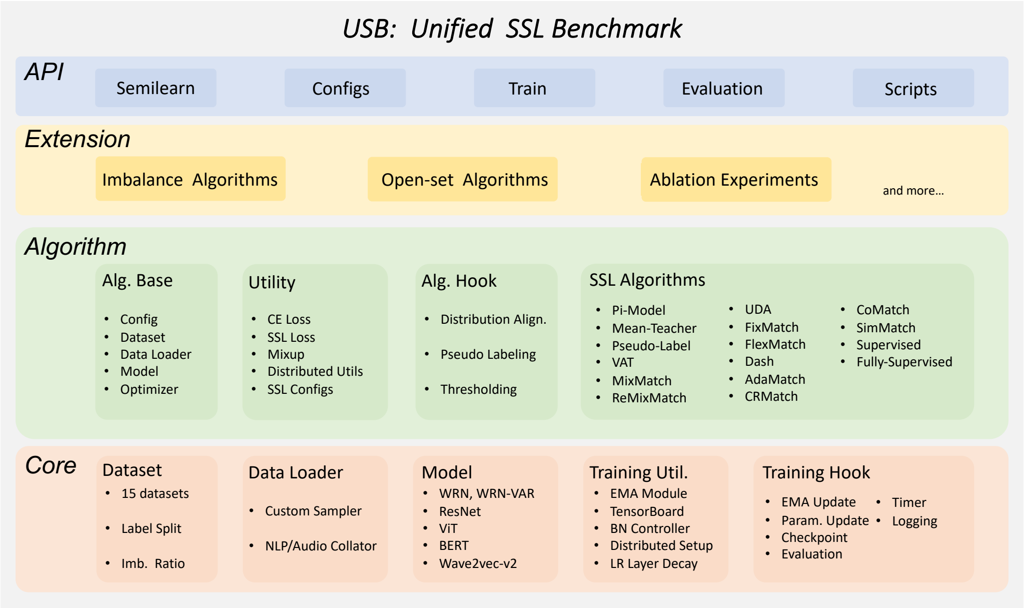


Figure 8.4: Structure of USB Codebase, consisting of 4 layers. The core layer provides the common functions, datasets, and models for SSL algorithms. The algorithm layer mainly implements the related SSL algorithms, with a high abstract level of algorithm components. Upon the algorithm layer, we use an extension layer for easy and flexible extension of core SSL algorithms. The top API layer supports a public python package SEMILEARN: pip install semilearn.

and algorithm-specific configurations, we further provide algorithm hooks according to the algorithm components summarized in Table 8.4. The algorithm hooks not only highlight the common part of different algorithms but also allows for a very easy and flexible combination of different components to resemble a new algorithm or conduct an ablation study. Based on this, we support 14 core SSL algorithms in USB, with two extra supervised learning variants. More algorithms are expected to be added through continued extension of USB.

Extension Layer. The extension layer is where we further extend the core SSL algorithms to different applications. Continuted effort are made on the extension of core SSL algorithms to imbalanced SSL algorithms [KHP$^+$20a, WZH$^+$22, LWZL11, HJK20, WSM$^+$21, YX20, FDS22, OKK21b] and open-set SSL algorithms [SKS21, GZJ$^+$20b, YIIA20b, LCG$^+$21, HXH$^+$21]. Systematic ablation study can also be conducted in the extension layer by inheriting either the core components and algorithms from the core layer or the algorithm layer.

API Layer. We wrap the core functions and algorithms in USB in the API layer as a public python package SEMILEARN. SEMILEARN is friendly for users from different backgrounds who want to employ SSL algorithms in new applications. Training and inference can be done in only a few lines of code with SEMILEARN. In addition, we provide the configuration files of all algorithms supported in USB with detailed parameter settings, which allows for reproduction of the results present in USB.

8.7 Limitation

Our primary focus is on semi-supervised classification in this chapter. However, there are other SSL tasks that the SSL community should not ignore. USB currently does not include SSL tasks such as imbalanced semi-supervised learning [KHP+20a, LWZL11, HJK20, WSM+21, YX20, FDS22, OKK21b], open-set semi-supervised learning [SKS21, GZJ+20b, YIIA20b, LCG+21, HXH+21], semi-supervised sequence modeling [CLML18, CRLL20, LM05, DL15, BWA+19, WWL+22], semi-supervised text generation [LWO22, HGSR19, CY21], semi-supervised regression [WL07, JXE18, KKKR18, ZL+05, LZZ17], semi-supervised object detection [TWG+16, TRW+21, XZH+21, TCLZ21, GWD+19, LMH+20], semi-supervised clustering [BBM02, Bai13, GCB04, BBM04], etc. In addition, we do not implement generative adversarial networks based SSL algorithms [KMJRW14, Spr15, Ode16b, DGF16b] and graph neural network based SSL algorithms [VQK+21, FZD+20, SZC19, GZQ+22, ZCHC20] in USB, which are also important to the SSL community. Moreover, it is of great importance to extend current SSL to distributional shift settings, such as domain adaptation [WFC+18, WCH+17] and out-of-distribution generalization [WLL+22], as well as time series anaysis [DWF+21]. We plan to evolve the benchmark in future iterations over time by extending it with more tasks.

8.8 Conclusion

We constructed USB, a unified SSL benchmark for classification that aims to enable consistent evaluation over multiple datasets from multiple domains and reduce the training cost to make the evaluation of SSL more affordable. With USB, we evaluate 14 SSL algorithms on 15 tasks across domains. We find that (1) although the performance of SSL algorithms is roughly close across domains, introducing diverse tasks from multiple domains is still necessary in the SSL scenario because the performance of SSL algorithms are not exactly steady across domains; (2) pre-training techniques can be helpful in the SSL scenario because it can not only accelerate the training but also improve the generalization performance; (3) unlabeled data sometimes hurts the performance especially when labeled data is extremely scarce. USB is a project for open extension and we plan to extend USB with more challenging tasks other than classification and introduce new algorithms.

III

Learning with foundation models

While the previous parts discuss the data perspective in AI systems, this section shifts to the model perspective in representation learning, aiming to develop versatile vision models capable of tackling a wide array of vision tasks. Specifically,

in Chapter 9, we introduce a diffusion-based vision generalist, unifying four distinct types of dense prediction tasks under a conditional image generation framework and repurposing pre-trained diffusion models to achieve this. Our investigation reveals a list of interesting findings and provides a recipe for fine-tuning pre-trained text-to-image diffusion models for dense vision tasks.

9 Toward a Diffusion-Based Generalist for Dense Vision Tasks

Contents

In this chapter, we discuss the model design in modern AI systems. In particular, we are interested in building generalized models that can solve many computer vision tasks simultaneously. Recent works have shown image itself can be used as a natural interface for general-purpose visual perception and demonstrated inspiring results. In this chapter, we explore diffusion-based vision generalists, where we unify different types of dense prediction tasks as conditional image generation and re-purpose pre-trained diffusion models for it. However, directly applying off-the-shelf latent diffusion models leads to a quantization issue. Thus, we propose to perform diffusion in pixel space and provide a recipe for finetuning pre-trained text-to-image diffusion models for dense vision tasks. In experiments, we evaluate our method on four different types of tasks and show competitive performance to the other vision generalists.

This chapter is based on [FXZ+24]. Yue Fan was the lead author of this paper and conducted all the experiments and wrote most parts of the paper.

9.1 Introduction

The field of artificial intelligence has made significant progress in building generalized model frameworks. In particular, autoregressive transformers [VSP+17] have become a prominent unified approach in Natural Language Processing (NLP), effectively addressing a wide range of tasks with a singular model architecture [TLI+23, DCLT18a, RWC+19, RSR+20]. However, in computer vision (CV), building a unified framework remains challenging due to the inherent diversity of the tasks and output formats. Consequently, state-of-the-art computer vision models still have many complex task-specific designs [CMS+20, CSK21, CMS+22, LWLJ22, WCB+22], making it difficult for feature sharing across tasks and, thus, limiting knowledge transfer.

The stark contrast between NLP and CV has given rise to a growing interest in developing unified approaches for vision tasks [LCZ+22, CSL+21, CSL+22, WWC+23, WZC+23, ZZL+22]. Recently, [WWC+23, WZC+23] have shown image itself can be used as a robust interface

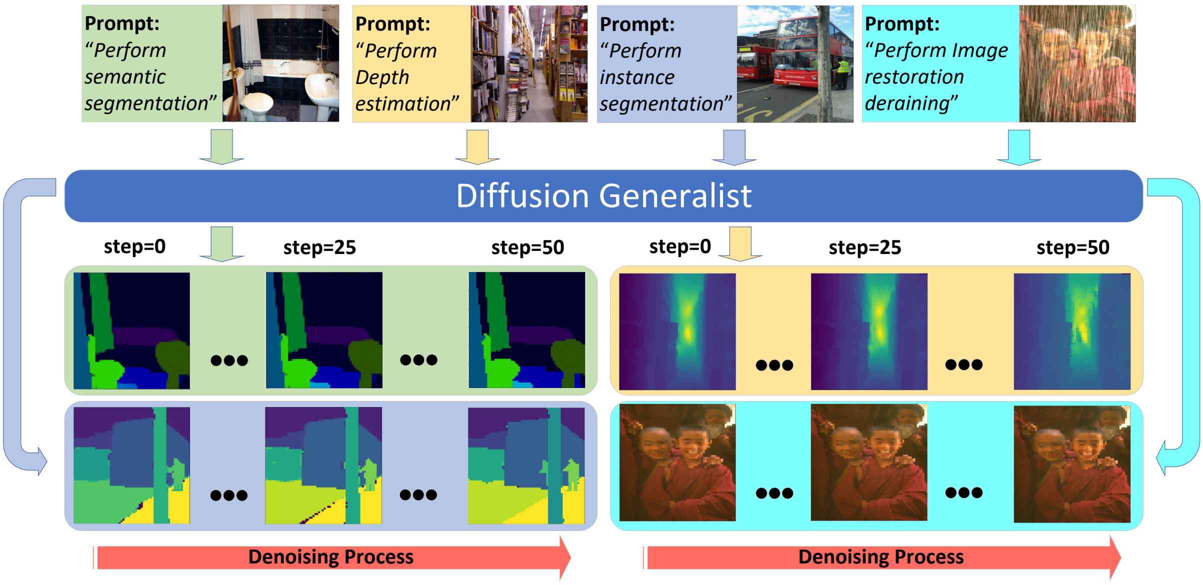

Prompt:
"Perform semantic segmentation"
Prompt:
"Perform Depth estimation"
Prompt:
"Perform instance segmentation"
Prompt:
"Perform Image restoration deraining"
Diffusion Generalist
step=0
step=25
step=50
step=0
step=25
step=50
Denoising Process
Denoising Process

Figure 9.1: We present a diffusion-based vision generalist for dense vision tasks. Given an input image, the model performs the corresponding task following the text instruction. We showcase the effectiveness of our model on depth estimation, semantic segmentation, panoptic segmentation, and three types of image restoration tasks. The images are the actual output of our model.

for unifying different vision tasks and demonstrated good performance. In this chapter, we propose a multi-task diffusion generalist for dense vision tasks by reformulating the dense prediction tasks as conditional image generation, and re-purpose pre-trained latent diffusion models for it. Fig. 9.1 visualizes the output of our model on semantic segmentation, panoptic segmentation, depth estimation, and image restoration. Based on text prompts, our model can perform different tasks with one set of parameters. However, directly finetuning the pre-trained latent diffusion models (e.g. Stable Diffusion [RBL+22b]) leads to quantization errors for segmentation tasks (see Table 9.2). To this end, we propose to do pixel-space diffusion which effectively improves the generation quality and does not suffer from quantization errors. Moreover, our exploration into training diffusion models as vision generalists reveals a list of interesting findings as follows:

- Diffusion-based generalists show superior performance over the non-diffusion-based generalists on tasks involving semantics or global understanding of the scene.
- We find conditioning on the image feature extracted from powerful pre-trained image encoders results in better performance than directly conditioning on the raw image.
- Pixel diffusion is better than latent diffusion as it does not have the quantization issue while upsampling.
- We observe that text-to-image generation pre-training stabilizes the training and leads to better performance.

In experiments, we demonstrate the model's versatility across six different dense prediction tasks on depth estimation, semantic segmentation, panoptic segmentation, image denoising, image draining, and light enhancement. Our method achieves competitive performance to the current state-of-the-art in many settings.

9.2 Related Work

Unified framework & Unified model: Efforts have been made to unify various vision tasks with a single model, resulting in several vision generalists [LCZ+22, CSL+21, CSL+22, WWC+23, WZC+23, KSPB+22]. Inspired by the success of sequence-to-sequence modeling in Natural Language Processing (NLP), Pix2Seq [CSL+21, CSL+22] leverages a plain autoregressive transformer and tackles many vision tasks with next-token prediction. For example, bounding boxes in object detection are cast as sequences of discrete tokens, and masks in semantic segmentation are encoded with coordinates of object polygons [CKUF17]. The idea was further developed in Unified-IO [LCZ+22], where dense prediction such as segmentation, depth map, and image restoration are also unified as tokens by using the corresponding image features from a vector quantization variational auto-encoder (VQ-VAE) [VDOV+17]. On the output side, the predicted image tokens are then decoded into masks and depth maps as the final prediction. Similarly, OFA [WYM+22] unified a diverse set of cross-modal and unimodal tasks in a simple sequence-to-sequence learning framework and achieved competitive performance pretrained with only 20M publicly available image-text pairs. Painter [WWC+23] and SegGPT [WZC+23], on the other hand, reformulate different vision tasks as an image inpainting problem, and perform in-context learning following [BGD+22]. Unlike the previous work, our method unifies different vision tasks under a conditional image generation framework and introduces a diffusion-based vision generalist for it.

Unified framework & Task-specific model: Besides the aforementioned literature, there is another line of related works that pursue unified architecture but task-specific models. UViM [KSPB+22] addressed the high-dimensionality output space of vision tasks via learned guiding code, where a short sequence modeled by an additional language model to encode task-specific information guides the prediction of the base model. Separate models are trained for different tasks as the guiding code is task-specific. XDecoder [ZDY+23] unified pixel-level image segmentation, image-level retrieval, and vision-language tasks with a generic decoding procedure, which predicts pixel-level masks and token-level semantics, and different combinations of the two outputs are used for different tasks. Despite their good performance, the task/modality-specific customization poses difficulty for knowledge sharing among different tasks and is also not friendly for supporting unseen tasks.

9.3 Toward a Diffusion-Based Generalist

9.3.1 Unification with Conditional Image Generation

As the output of most vision tasks can be always visualized as images, we redefine the output space of different vision tasks as RGB images and unify them as conditional image generation to tackle the inherent difference of output formats of different vision tasks. Given a input image x and the corresponding ground-truth y, we first transform y into RGB images and then pair it with a task descriptor in text. By doing so, training sets of different tasks are combined into a holistic training set. And training the model jointly on it enables the knowledge transfer between tasks. At test time, given a new image, the model can perform different tasks following the text instructions (examples in Fig. 9.1).

In this chapter, we consider four types of dense prediction tasks: depth estimation, semantic segmentation, panoptic segmentation, and image restoration.

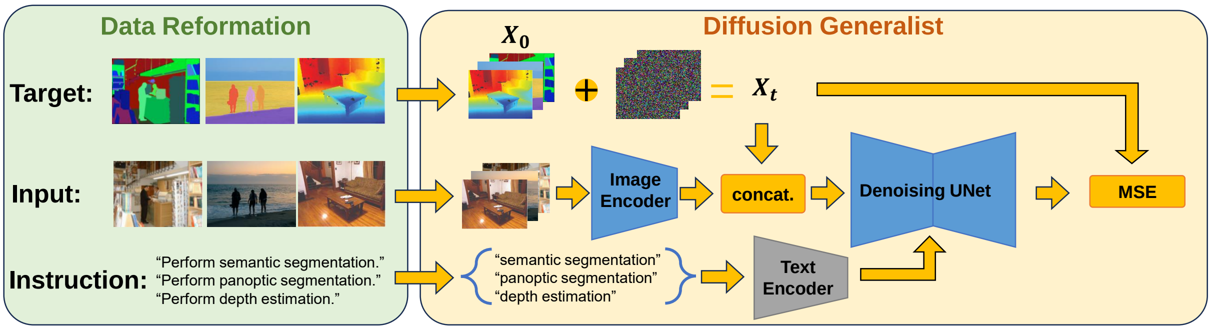


Figure 9.2: The training pipeline of the diffusion-based vision generalist consists of two parts: **Left:** Redefining the output space of different vision tasks as RGB images so that they can be unified under a conditional image generation framework. **Right:** We finetune a pre-trained diffusion model on the reformatted data from the first step. Diffusion is performed in the pixel space to mitigate the quantization error of the latent diffusion (see Table 9.2). The image and text conditionings are fed into the model via the corresponding encoders, where only the image encoder is tuned during the training.

Depth estimation outputs real number depth value for each pixel on x. Given the minimum and the maximum values, we map them into $[0, 255]$ linearly and discretize them into integers, which is then repeated and stacked along the channel to form the ground-truth RGB label.

Semantic segmentation predicts a class label for each pixel. We use a pre-defined injective class-to-color mapping to transform the segmentation mask into RGB images. Given a task with C categories, we define C colors which are evenly distributed in the 3-dimensional RGB space. Specifically, following [WWC⁺23], the class index is represented by a 3-digit number with b-base system, where $b = \lceil C^{\frac{1}{3}} \rceil$. Thus, the margin between two colors is defined as $\text{int}(\frac{256}{b})$. The color for the i-th class is then $[\text{int}(\frac{i}{b^2}) \times m, \text{int}(\frac{i}{b})\%b \times m, l\%b \times m]$. At test time, we find the nearest neighbor of the predicted color in the predefined class-to-color mapping and predict the corresponding category.

Panoptic segmentation is solved as a combination of semantic and instance segmentation. Semantic segmentation labels are constructed as stated above. For instance segmentation, we set N as the maximum number of instances a single training image can contain. Then, we define N colors which are evenly distributed in the 3-dimensional RGB space as in semantic segmentation. Finally, we assign colors to objects based on their spatial location to form the RGB ground-truth label. For example, the instance whose center is at the upper leftmost corner obtains the first color and the lower rightmost gets the last color. At test time, the model makes predictions twice with different text instructions and merge the results for panoptic segmentation.

Image restoration aims to predict the clean image from corrupted images. Thus, the output space is inherently RGB image and does not need further transformation to fit in the framework.

9.3.2 A Diffusion Multi-Task Generalist Framework

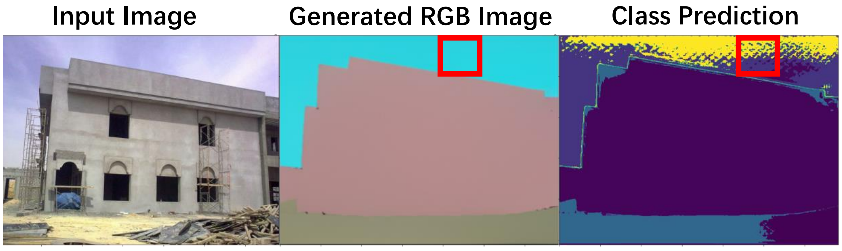
By reformating the output space of different vision tasks into images, it is natural to solve them together under a conditional image generation framework. To this end, we leverage the powerful diffusion models pre-trained for image generation and re-purpose them in our use case.

Fig. 9.2 shows the overall pipeline of the method, which is a conditional image generation framework with pixel-space diffusion. Given M tasks with datasets $\{\mathbf{I}^i, \mathbf{Y}^i\}_{i=1}^M$, where $\mathbf{I}^i$ are the input images of task i and $\mathbf{Y}^i$ are the corresponding ground-truth labels. We first transform the output into RGB image format $\mathbf{X}^i$ and augment each task with a text instruction T^i. At each training step, we randomly sample a subset of tasks and then sample data from each task. For each input data $\{I^i, X^i, T^i\}$, we first compute the multi-scale feature map of the original image I^i from the image encoder. Then, it is concatenated with the noised target image X_t^i before being fed into the UNet for the reconstruction loss. Note that the image feature can have a different spatial resolution than the target image X_t^i, in which case the concatenation will be performed on the interpolated image feature. In experiments, we find both the image feature resolution and the target resolution are important for the final performance but target resolution matters more. The text conditioning T^i is fed into the UNet via cross-attention [RBL+22b]. The whole pipeline is trained in an end-to-end manner except for the text encoder, which is frozen throughout the training. Compared to the standard diffusion model for conditional image generation, there are three main differences:

	Target	**Depth Estimation**	**Semantic Seg.**	**Panoptic Seg.**	**Denoising**	**Deraining**	**Light Enhance.**
	image	RMSE ↓	mIoU ↑	PQ ↑	SSIM ↑	SSIM ↑	SSIM ↑
	resolution	NYUv2	ADE-20K	COCO	SIDD	5 datasets	LoL
Generalist framework, task-specific models							
UViM [KSPB+22]	512 × 512	0.467	-	45.8%	-	-	-
Generalist models							
Unified-IO [LCZ+22]	256 × 256	0.385	25.7%	-	-	-	-
InstructCV [GPS+23]	256 × 256	0.297	47.2%	-	-	-	-
Painter [WWC+23]	448 × 448	**0.288**	**49.9%**	**43.4%**	**0.954**	**0.868**	**0.872**
Painter [WWC+23]	128 × 128	0.435†	28.4%†	22.6%†	0.922†	0.626†	0.773†
Ours	128 × 128	0.448	48.7%	40.3%	**0.954**	0.815	0.758

Table 9.1: Our method achieves competitive performance in most of the tasks while trained at a much smaller target resolution of 128 × 128. When compared at the same resolution, our method shows superior performance over the previous best method (Painter [WWC+23]), especially on semantic segmentation and panoptic segmentation. The best number is in bold and the second best number is underscored. †indicates numbers from our reproduction.

- We propose to directly perform diffusion in the pixel space. As shown in Table 9.2, when mapping from the latent space to the pixel space, visually uniform regions actually have pixels of many different RGB values. This variance can lead to inaccurate class mappings, and consequently, suboptimal performance for semantic and panoptic segmentation.
- The image conditioning is provided via a feature extractor (we use ConvNeXt [LMW+22]) and is concatenated to the target image X_0. Compared to the widely adopted method of directly concatenating the raw image as the condition, this brings significant performance improvement, especially for semantic and panoptic segmentation (see Table 9.3 for ablation).
- We remove the self-attention layers in the outermost layers of UNet. This is because the pixel space diffusion at large target image resolutions induces considerable memory costs. Removing them alleviates the issue without compromising the performance.



	Semantic Seg. mIoU ↑ ADE-20K	**Panoptic Seg.** PQ ↑ COCO
Latent Diffusion	17.1%	11.7%
Pixel Diffusion	48.0%	35.5%

Table 9.2: **Upper:** Semantic segmentation output of the latent diffusion model. The perceptually same colored regions have different pixel values and, therefore, are mapped to different class labels, leading to bad final performance. While the red box contains only one ground-truth class sky in generated RGB image, the final class prediction has four classes after the quantization. **Lower:** Latent diffusion suffers from the quantization issue while pixel diffusion achieves good performance.

9.4 Experimental Results

Here, we first explain experimental settings in Section 9.4.1. Then, we highlight important design choices in diffusion-based multi-task generalists in Section 9.4.2 before comparing our method with previous approaches in Section 9.4.3.

9.4.1 Datasets and Implementation Details

Datasets: We evaluate our method on six different dense prediction tasks with various output formats. For depth estimation, we use NYUv2 [SHKF12] and report the Root Mean Square Error (RMSE). For semantic segmentation, we evaluate on ADE20K [ZZP+17] and adopt the widely used metric of mean IoU (mIoU). For panoptic segmentation, we use MS-COCO [LMB+14] and report panoptic quality as the measure. During inference, the model is forwarded twice for each validation image with different instructions to obtain the results of semantic and instance segmentation respectively. The outputs are then merged together into the panoptic segmentation. Image restoration tasks are evaluated on several popular benchmarks, including SIDD [ALB18] for image denoising, LoL [WWYL18] for low-light image enhancement, and 5 merged datasets [ZAK+22] for deraining.

Implementation details. As mentioned above, we take the Stable Diffusion v1.4 [RBL+22b] checkpoint and finetune it jointly on six tasks. The image feature extractor is an ImageNet-21K [RDS+15c] pre-trained ConvNeXt-Large [LMW+22]. The text encoder is Open-CLIP [RKH+21b], which is used in Stable Diffusion [RBL+22b]. We adopt uniform sampling for each tasks except panoptic segmentation, whose weight is twice as much as the other tasks (as it is a combination of semantic and instance segmentation). Following [Che23], we also adjust the input scaling factor by a constant factor b in the forward noising processing of diffusion. We use AdamW optimizer [KB15] with constant learning rate of 0.0001, linearly warmed up in the first 20,000 iterations. The target image resolution is 128×128 while the conditioning image resolution is 512 × 512. We train our model for 180,000 steps in total with a batch size of 1024.

	Depth Estimation	**Semantic Seg.**	**Panoptic Seg.**	**Denoising**	**Deraining**	**Light Enhance.**
	RMSE ↓	mIoU ↑	PQ ↑	SSIM ↑	SSIM ↑	SSIM ↑
	NYUv2	ADE-20K	COCO	SIDD	5 datasets	LoL
Ours	0.511	**48.0%**	**35.5%**	0.949	0.772	**0.704**
Non-diffusion	**0.443**	42.4%	19.8%	**0.951**	**0.773**	0.703
Train from scratch	0.528	46.6%	33.6%	0.948	0.764	**0.704**
Direct concat.	0.476	37.6%	27.1%	0.941	0.772	0.687

Table 9.3: We analyze the important design choices of our method and aim to provide a recipe for training diffusion-based generalists: 1. diffusion models greatly outperform non-diffusion models on panoptic segmentation; 2. text-to-image generation pre-training leads to an overall better performance; 3. conditioning on image features extracted from an encoder gives significant improvement over the raw image.

9.4.2 Recipes for Diffusion-Based Generalists

In this section, we analyze the design choices of our method and show their importance through ablation experiments. Specifically, we show the importance of diffusion by training the same model as in Fig. 9.2 to directly generate target images without using diffusion (non-diffusion). We study the significance of image generation pre-training and image encoder by training models without them (train from scratch and direct concat.). If not specified, we train all models at a target resolution of 64 × 64 for 50,000 steps.

We attribute the success of our method to four aspects. (1) While having similar results on image restoration tasks, diffusion-based generalist achieves better performance than non-diffusion models on segmentation tasks which requires a global understanding of the scene and the semantics. For example, the diffusion model reaches 35.5% PQ for panoptic segmentation while the non-diffusion model has only 19.8% (Table 9.3 ours v.s. non-diffusion). (2) Image generation pre-training on large scale dataset transfers useful knowledge to the many downstream tasks. The model finetuned from Stable Diffusion v1.4 [RBL+22b] achieves better results than the one trained from scratch across the tasks (Table 9.3 ours vs train from scratch). (3) The image conditioning can take advantage of powerful pre-trained image encoders by conditioning on the image features rather than the raw image, which is in contrast to the standard practice for image generation tasks. On semantic segmentation and panoptic segmentation, extracting features gives 10.4% and 8.4% performance improvement, respectively (Table 9.3 ours v.s. direct concat.). (4) Pixel diffusion is better than latent diffusion as it does not suffer from the quantization issue while upsampling (see Table 9.2 for an example).

9.4.3 Comparisons with Prior Art

We compare our model with recent vision generalists in Table 9.1. With a much smaller target image resolution at 128 × 128, our method achieves competitive performance across the tasks. In particular, when compared with the previous best model Painter [WWC+23] at the same target resolution, our method has a significant margin over them, which highlights the potential of our method at a higher resolution.

9.4.4 Qualitative Results

In this section, we visualize the output of our method on six different tasks in Fig. 9.3. We use DDIM at inference time with 50 steps. Each figure shows the output of the denoising process at the 0-th, 25-th, and 50-th steps.

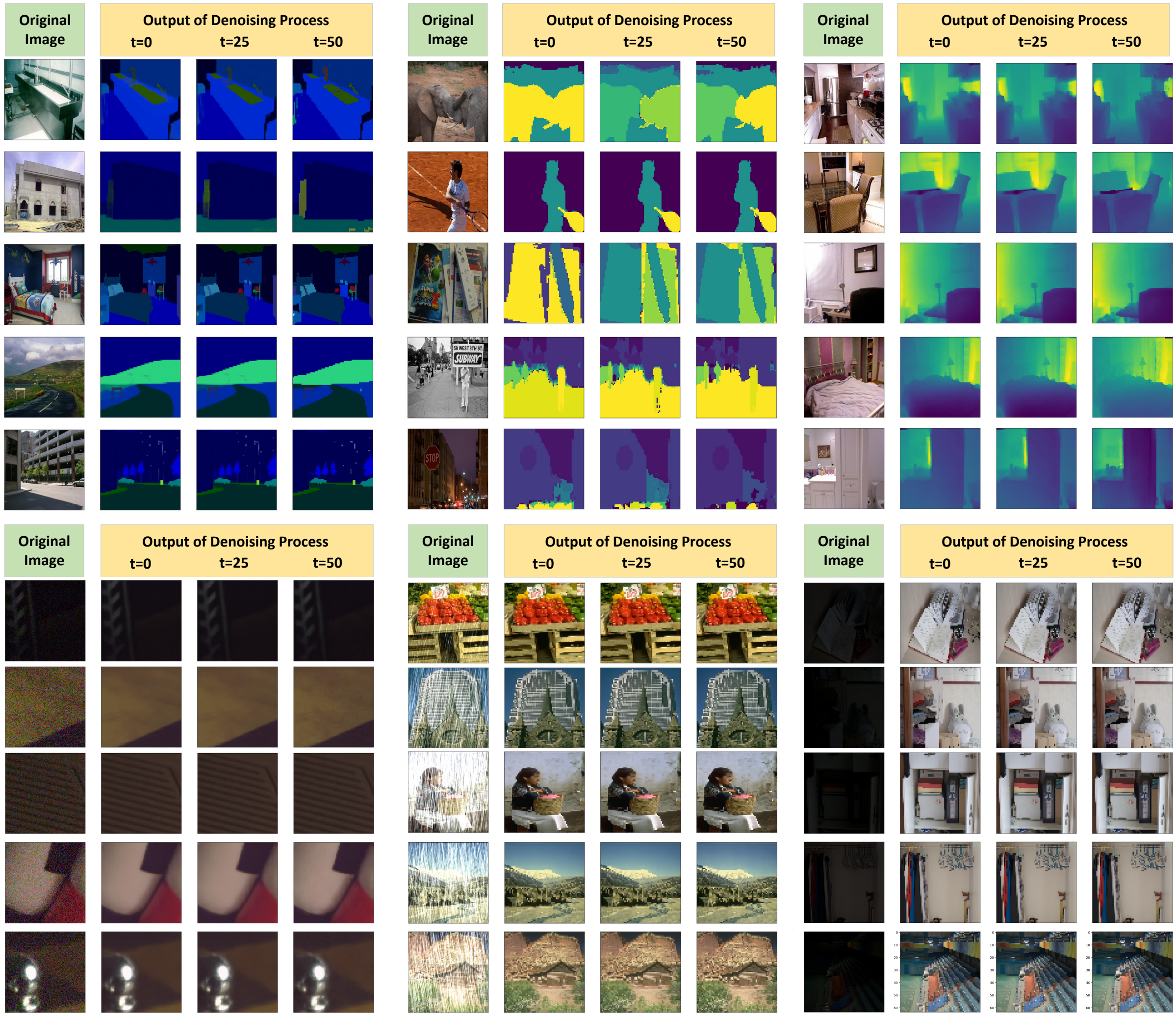

Original Image
Output of Denoising Process
t=0
t=25
t=50
Original Image
Output of Denoising Process
t=0
t=25
t=50
Original Image
Output of Denoising Process
t=0
t=25
t=50
Original Image
Output of Denoising Process
t=0
t=25
t=50
Original Image
Output of Denoising Process
t=0
t=25
t=50
Original Image
Output of Denoising Process
t=0
t=25
t=50

Figure 9.3: Qualitative results on images from the validation sets of ADE20K, MS-COCO, NYU-V2, SIDD, Deraining, and LOL. Following a raster scan order, the text prompts are "Performance semantic segmentation", "Performance instance segmentation", "Performance depth estimation", "Performance image restoration denoising", "Performance image restoration deraining", and "Performance image restoration light enhancement", respectively. The images are not cherry-picked.

9.4.5 Ablation Study

In this section, we analyze the effect of other important hyper-parameters of our method, such as batch size, target image resolution, and noise-signal ratio. Similar to Section 9.4.2, we train all models at a target resolution of 64 × 64 for 50,000 steps by default.

Effect of batch size. Here, we discuss the effect of different batch sizes for our method. As shown in Table 9.4, the performance of most of the tasks improves with the increase of the batch size. In particular, panoptic segmentation greatly benefits from the large batch size.

	Depth	**Sem. Seg.**	**Pan. Seg.**	**Denoise**	**Detrain**	**Enhance.**
	RMSE ↓	mIoU ↑	PQ ↑	SSIM ↑	SSIM ↑	SSIM ↑
	NYUv2	ADE-20K	COCO	SIDD	5 datasets	LoL
128	0.548	35.5%	26.2%	0.941	0.754	0.701
256	0.495	44.3%	30.0%	0.945	0.766	0.703
512	**0.491**	47.1%	33.5%	0.948	0.770	0.702
1024	0.511	**48.0%**	**35.5%**	**0.949**	**0.772**	**0.704**

Table 9.4: Large batch size improves the performance for all the tasks except depth estimation.

Effect of target resolution. Table 9.5 studies the effect of different target image resolutions. Since our method performs diffusion in the pixel space, increasing the target image resolution is important for good performance. Despite the increased memory cost, our method achieves its best performance at the resolution of 128 × 128 and can be further improved with even larger target images.

	Depth	**Sem. Seg.**	**Pan. Seg.**	**Denoise**	**Detrain**	**Enhance.**
	RMSE ↓	mIoU ↑	PQ ↑	SSIM ↑	SSIM ↑	SSIM ↑
	NYUv2	ADE-20K	COCO	SIDD	5 datasets	LoL
32x32	0.514	44.4%	32.1%	0.940	0.743	0.653
64x64	0.511	48.0%	35.5%	0.949	0.772	0.704
128x128	**0.467**	**49.2%**	**36.7%**	**0.953**	**0.810**	**0.762**

Table 9.5: Effect of output resolution. Increasing the target image resolution significantly improves the performance across tasks.

Importance of noise-signal ratio. In DDPM [HJA20], the forward diffusion process is defined as $x_t = \sqrt{\gamma_t} x_0 + \sqrt{1-\gamma_t}\epsilon$, where x_0 is the input image, ϵ is a Gaussian noise, and t is the number of diffusion step. As shown in [Che23], the denoising task at the same noise level (i.e. the same t) becomes simpler with the increase in the image size. In order to compensate for this, [Che23] proposed to scale the input with a constant b to explicitly control the noise-signal ratio, which results in the forward diffusion process as $x_t = \sqrt{\gamma_t} b x_0 + \sqrt{1-\gamma_t}\epsilon$. As we reduce b, it increases the noise levels. Table 9.6 shows the effect of the noise-signal ratio b where $b = 0.5$ gives the best performance.

	Depth	**Sem. Seg.**	**Pan. Seg.**	**Denoise**	**Detrain**	**Enhance.**
	RMSE ↓	mIoU ↑	PQ ↑	SSIM ↑	SSIM ↑	SSIM ↑
	NYUv2	ADE-20K	COCO	SIDD	5 datasets	LoL
0.1	**0.497**	46.9%	33.1%	0.948	0.770	0.702
0.3	0.511	48.0%	35.5%	**0.949**	0.772	0.704
0.5	0.514	**49.3%**	**35.9%**	**0.949**	**0.774**	**0.708**
0.7	0.533	48.2%	34.4%	**0.949**	0.773	0.707
1.0	0.572	40.3%	31.1%	0.948	0.770	0.706

Table 9.6: Importance of noise-signal ratio b in the forward diffusion process $x_t = \sqrt{\gamma_t} b x_0 + \sqrt{1-\gamma_t}\epsilon$.

9.5 Conclusion and Limitations

In this work, we explore a diffusion-based vision generalist, where different dense prediction tasks are unified as conditional image generation and we re-purpose pre-trained diffusion models for it. Furthermore, we analyze different design choices of diffusion-based generalists and provide a recipe for training such a model. In experiments, we demonstrate the model's versatility across six different dense prediction tasks and achieve competitive performance to the current state-of-the-art. This work, however, is also subject to limitations. For example, full fine-tuning of the pre-trained diffusion model at a larger target image resolution is memory intensive due to the pixel space diffusion. Thus, exploring parameter-efficient tuning for such a model would be an interesting future direction.

Conclusion and Future Work 10

Contents

In recent years, artificial intelligence has made remarkable strides, revolutionizing industries and transforming everyday life. At the core of this progress are two fundamental components: data and model design, both of which pose distinct challenges. While scaling data is an effective strategy for enhancing model performance, collecting large-scale labeled datasets is often expensive and time-consuming. This drives the development of semi-supervised learning, which seeks to leverage the vast amount of available unlabeled data. However, unlabeled data presents difficulties such as noise, imbalance, and domain shifts. How to extract robust representations from such imperfect unlabeled data remains an important research question. On the model side, there is a growing trend toward developing generalized models that can handle a diverse array of tasks. However, this pursuit comes with its own set of complexities. Effectively addressing these challenges is crucial for the future of AI advancements.

10.1 Key Insights and Conclusions

In this thesis, we aim to address critical challenges in advancing AI systems from both data and model perspectives. Our contributions span three major parts, each focusing on key aspects of improving semi-supervised learning and foundation models.

- In Part I, we delve into enhancing standard semi-supervised learning paradigms, specifically consistency regularization and pseudo-labeling. First, we revisit the concept of enforcing feature invariance in consistency regularization, and improve upon it with a technique called FeatDistLoss. This approach introduces a regularization term that constrains the distance between feature representations, leading to more robust features. Next, we introduce FreeMatch, which improves thresholding-based pseudo-labeling by incorporating a self-adaptive threshold. This threshold dynamically adjusts based on the model's learning status, resulting in more accurate pseudo-labels. Finally, SoftMatch addresses the quantity-quality trade-off in pseudo-labeling by effectively leveraging unconfident yet correct pseudo-labels, thus optimizing label utilization without sacrificing quality. Together, these contributions push the boundaries of standard semi-supervised learning by improving the learning dynamics and feature quality in both consistency regularization and pseudo-labeling approaches.

- In Part II, we shift our focus to semi-supervised learning in more realistic settings, where data challenges such as long-tail distributions, outliers, and domain shifts are prevalent. We propose CoSSL, a novel co-learning framework designed for imbalanced semi-supervised learning. This framework decouples representation learning from classifier learning while coupling them through a shared encoder and pseudo-label generation, providing better handling of imbalanced data. Additionally, we introduce a Simple but Strong Baseline (SSB) for open-set SSL. This approach highlights the importance of high-confidence pseudo-labels, regardless of whether a sample is an inlier or outlier, improving the overall utilization of unlabeled data and enhancing final model performance. We further contribute USB, a unified and challenging SSL benchmark, which offers consistent evaluation across 15 tasks in computer vision (CV), natural language processing (NLP), and audio. This benchmark serves as a fair testing ground to evaluate the generalization of SSL methods beyond CV tasks, marking a significant step forward in assessing SSL across diverse domains.

- In Part III, we explore the realm of vision generalist models and the scaling of foundation models. Here, we investigate diffusion-based vision generalists, where dense prediction tasks are unified as conditional image generation problems. By re-purposing pre-trained diffusion models, we enable the application of a single model across multiple dense prediction tasks, showcasing the versatility of this approach. Lastly, we introduce Tokenformer, a natively scalable architecture that not only employs the attention mechanism for token-to-token interactions but also for interactions between tokens and model parameters. This unique design enhances architectural flexibility and allows for progressive and efficient model scaling without the need for retraining from scratch, making it a powerful tool for large-scale applications.

In the following, we revisit the contributions of individual chapters in more detail, before discussing future work in Section 10.2.

Part 1. **Standard Semi-Supervised Learning:** In the first part, we focus on the standard setting of SSL, where the goal is to enhance the representation learning by leveraging a large amount of unlabeled data along with a small number of labeled data. Specifically, the thesis focuses on two popular SSL frameworks: consistency regularization and pseudo-labeling.

In Chapter 3, we revisit the idea of consistency regularization with data augmentation. Normally, consistency regularization enforces the model output to be invariant to data augmentations. However, when the data augmentation is too strong, it might generate images that diverge significantly from the original semantics. We argue that improving equivariance on such strongly augmented images can provide even better performance rather than making the model invariant to all kinds of augmentations. We formulate FeatDistLoss to explicitly encourage equivariance between features from different augmentations while enforcing the same semantic class label.

In Chapter 4, we tackle the challenge of quantity-quality trade-off in pseudo-labeling, where a high confidence threshold discards a significant number of potentially correct but low-confidence labels while a low threshold introduces noisy, incorrect labels that can mislead the model. To this end, we propose a parameter-free and self-adaptive thresholding scheme that changes thresholds according to the learning status of each class. To handle barely supervised settings more effectively, we further propose a class fairness objective to encourage the model to produce fair (i.e., diverse) predictions among all classes.

In Chapter 5, we further explore a different strategy to improve the threshold-based pseudo-labeling by replacing the threshold by fitting a weight function per sample. Specifically, we fit a truncated Gaussian function to the confidence distribution which assigns lower weights to possibly correct pseudo-labels with lower confidence scores. We further propose Uniform Alignment to resolve the imbalance issue of pseudo labels while maintaining their high quantity and quality.

Part 2. **Realistic Semi-Supervised Learning:** In the second part, we move beyond the standard setting of SSL, where data is clean and balanced. In particular, we consider more realistic settings where unlabeled data follow long-tailed distribution or contain outliers. In addition, we introduce a new SSL benchmark for classification as the existing SSL benchmarks have limitations in practical utility due to their constrained settings.

In Chapter 6, we investigate the challenging and realistic setting of imbalanced SSL where both the labeled and the unlabeled data are class-imbalanced. We address the problem by proposing a novel co-learning framework that closely couples representation and classifier while the training of them is decoupled. Moreover, we propose Tail-class Feature Enhancement (TFE) for improved classifier learning for imbalanced SSL, which utilizes unlabeled data as a source of augmentation to enhance the data diversity of tail classes, leading to a more robust classifier. In addition, we also propose new evaluation criteria for imbalanced SSL, and evaluate them over a large range of varying distributions, including those that are radically different from the training distribution.

In Chapter 7, we consider another realistic setting of open-set SSL, where unlabeled data contain out-of-distribution (OOD) samples from novel classes that do not appear in the labeled set. The goal is to correctly classify inliers, while identifying outliers seen during the training and, most importantly, unseen outliers that do not appear in the training set. To this end, we design SSB, which effectively separates the feature space of inlier classification and outlier detection via non-linear transformations and effectively leverages outliers via confidence-based filtering. In addition, we propose pseudo-negative mining to further improve outlier detector training by enhancing the data diversity of OOD data with pseudo-outliers.

In Chapter 8, we construct a Unified SSL Benchmark (USB) for classification, which selects 15 diverse, challenging, and comprehensive tasks from CV, natural language processing (NLP), and audio processing (Audio). We systematically evaluate the dominant SSL methods, and also open-source a modular and extensible codebase for fair evaluation of these SSL methods. We further provide the pre-trained versions of the state-of-the-art neural models for CV tasks to make the cost affordable for further tuning. USB enables the evaluation of a single SSL algorithm on more tasks from multiple domains but with less cost.

Part 3. **Building vision generalist model:** In the third part, we turn to the model perspective of representation learning and aim to build versatile vision models that are capable of handling many diverse types of vision tasks.

In Chapter 9, we address the challenge of the vision generalist model by exploring diffusion-based vision generalists, where we unify different types of dense prediction tasks as conditional image generation and re-purpose pre-trained diffusion models for it. Our investigation reveals a list of interesting findings as follows: 1. Diffusion-based generalists show superior performance over the non-diffusion-based generalists on tasks involving semantics or global understanding of the scene. 2. We find conditioning on the image feature extracted from powerful pre-trained image encoders results in better performance than directly conditioning on the raw image. 3. Pixel diffusion is better than latent diffusion as it does not have the quantization issue while upsampling. 4. We observe that text-to-image generation pre-training stabilizes the training and leads to better performance. In experiments, we evaluate our method on four different types of tasks and show competitive performance to the other vision generalists.

10.2 Future Directions

In the following, we provide a discussion on potential future directions within the scope of this thesis. The first direction is exploring more sophisticated ways to utilize unlabeled data. How can we harness inherent structures within unlabeled data, and what additional supervision signals can we leverage from them? Second, extending SSL methods beyond semi-supervised image classification to other tasks and modalities is essential, especially given the large performance differences in different domains as shown in Chapter 8. Tackling this generalization challenge would allow SSL advancements to impact a broader range of applications. Third, real-world data collection involves complexities beyond simple imbalances or outliers, as discussed in Chapters 6 and 7; identifying and addressing these nuanced pitfalls will be critical to SSL's applicability in real settings. Finally, with model design increasingly central to AI, exploring model-based solutions and enhancements remains vital to advancing the field.

Unsupervised and Self-Supervised Learning: Learning robust representations from unlabeled data has been a focal point of many recent research. Although this thesis primarily uses SSL frameworks, many self-supervised learning methods have been proposed to explore the internal structures of the unlabeled data as the supervision signals and have demonstrated strong generalization abilities[HFW+20, CKNH20, CTM+21]. Therefore, it is a promising direction to design methods that combine the strength of SSL and self-supervised techniques while managing potentially conflicting training signals. In fact, Chapter 3 can serve as a good starting point, where a self-supervised component is integrated into the SSL framework and further improves model performance.

Generalization beyond Image Classification: Much of the progress in standard semi-supervised learning has centered on image classification, potentially limiting its adaptability and effectiveness across domains. Expanding SSL research to encompass diverse data modalities and complex tasks is essential to broaden its real-world applicability.

- **SSL for diverse data modalities:** Real-world data often spans various modalities, such as images, sensor data, and text, particularly in fields like autonomous driving, healthcare, and multimedia. Current SSL methods, however, lack proven effectiveness with these mixed and complex modalities. For example, consistency regularization, a popular SSL approach, depends on strong data augmentations tailored for image data, which are challenging to design for text, audio, or medical images. Standard image augmentations, like horizontal or vertical flipping, are inappropriate for medical images, where anatomical structures may lose integrity if altered. Extending SSL to effectively handle such diverse data types is a complex but critical area for further research.

- **SSL for complex tasks**: Beyond classification, tasks like object detection and image segmentation require more sophisticated outputs, such as bounding boxes and pixel-level labels, introducing new challenges for SSL. For instance, object detection often involves multiple objects per image. Generating high-quality pseudo-labels for bounding boxes demands additional processes like non-maximum suppression (NMS) and faces issues with data imbalance, as smaller or infrequent objects may be underrepresented. Developing SSL methods that can effectively handle such complexities—or adapting existing techniques to accommodate them—remains an open challenge in advancing SSL capabilities.

Toward Realistic Semi-Supervised Learning: Real-world data collection often introduces various imperfections, as explored in Chapter 6 and 7, which focus on long-tailed distribution and outliers, respectively. However, real-world data issues extend beyond these cases. Below, we outline several promising directions toward more realistic SSL:

- **Open-world SSL:** In open-world SSL, the unlabeled dataset may contain classes that are absent from the labeled set. Given an unlabeled test set, the model must either assign instances to a previously seen class or form and assign instances to a novel class. This differs from open-set SSL as it operates in a transductive learning setting and emphasizes the discovery of novel classes, presenting unique challenges in class representation and generalization.

- **Semi-supervised domain adaptation (SSDA):** SSDA addresses the scenario where training and test data come from related yet distinct domains. This setup typically involves a source domain rich in labeled data and a target domain with few labeled instances but many unlabeled ones. The objective is to correctly classify new data from the target domain. Key challenges include developing domain-invariant representations and effectively extracting the underlying structure of the target domain.

- **Semi-supervised domain generalization (SSDG):** SSDG further extends SSDA by removing the assumption that test data originates from the same domain as the unlabeled data. SSDG generally involves multiple source and target domains and aims to develop a domain-generalizable model using a small amount of labeled data and abundant unlabeled data from each source domain. Since standard SSL methods assume matched training and test distributions, directly applying them to SSDG is suboptimal. The main challenge is creating representations that generalize across domains effectively.

- **Continual semi-supervised learning:** In many real-world settings, training data accumulates gradually, and storing all the old data may not be possible due to storage constraints or privacy concerns. Continual Semi-Supervised Learning addresses this issue by allowing both labeled and unlabeled data to arrive sequentially. In each task, the model receives both labeled and unlabeled data and is evaluated on a cumulative test set across tasks. The primary challenge here is mitigating catastrophic forgetting, ensuring that newly acquired knowledge does not erase previously learned information.

Foundation Model: Foundation models are large-scale, versatile neural networks pre-trained on massive datasets, designed to generalize across a wide range of downstream tasks with minimal task-specific fine-tuning. They have redefined the AI landscape and achieved promising performance across domains, from natural language processing to computer vision and beyond. Below, we outline three critical future directions for foundation models:

- **Model Scaling:** Expanding the scale of foundation models in terms of model size and training data continues to yield substantial performance gains across applications. Larger models often demonstrate enhanced generalization, capturing intricate patterns and complex representations. Yet, scaling poses significant cost and resource challenges. An promising direction is to leverage pre-trained smaller models to expedite the training of larger models, potentially reducing costs without compromising performance. Our work on TokenFormer [WFN+24] highlights one way toward more efficient scaling, serving as a foundational approach for further exploration in this area.

- **Efficient Model Inference:** Increasing the efficiency of model inference is another essential direction for foundation models. Enhancing inference efficiency not only enables faster processing times and reduces computational demands but also facilitates the deployment of these models in time-sensitive, resource-limited settings. Achieving efficient inference is crucial for broadening the accessibility of advanced AI across industries and devices, ensuring that even resource-constrained environments can benefit. In essence, prioritizing inference efficiency is key to making foundation models more sustainable and widely adopted in practical applications.

- **Generalist Models:** The unification of tasks and modalities stands out as one of the most transformative goals for foundation models. Unlike conventional models designed for single tasks, generalist foundation models can handle a broad spectrum of tasks (e.g., classification, segmentation, and detection) and even multi-modal inputs (e.g., visual question answering) with minimal adaptation. This shift towards task and modality unification unlocks new possibilities, paving the way for highly generalized AI systems capable of solving diverse challenges within a unified architecture.

List of Algorithms

List of Figures

List of Tables

Bibliography

[AAA+23] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. Cited on page 1.

[ADGF+09] Igor Aleksander, Massimo De Gregorio, Felipe Maia Galvao França, Priscila Machado Vieira Lima, and Helen Morton. A brief introduction to weightless neural systems. In *ESANN*, pages 299–305. Citeseer, 2009. Cited on page 21.

[AFIW18] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. *arXiv preprint arXiv:1806.05594*, 2018. Cited on pages 6 and 110.

[AK10] Ryan Gomes Andreas Krause, Pietro Perona. Discriminative clustering by regularized information maximization. In *Advances in neural information processing systems*, 2010. Cited on pages 53 and 61.

[ALB18] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. Cited on page 128.

[AOA+20a] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020. Cited on pages 27, 34, 36, 37, and 93.

[AOA+20b] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. Cited on pages 50, 55, 59, 60, 61, 63, 65, and 66.

[Asg16] Nabiha Asghar. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*, 2016. Cited on page 111.

[AY01] Charu C Aggarwal and Philip S Yu. Outlier detection for high dimensional data. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, 2001. Cited on page 92.

[AZKB24] Ibrahim M Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. Getting vit in shape: Scaling laws for compute-optimal model design. *Advances in Neural Information Processing Systems*, 36, 2024. Cited on page 1.

[Bai13] Eric Bair. Semi-supervised clustering methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(5):349–361, 2013. Cited on page 119.

[BAP14a] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in neural information processing systems*, 2014. Cited on pages 27, 29, 30, 34, 48, 78, and 79.

[BAP14b] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27:3365–3373, 2014. Cited on page 49.

[BB16] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016. Cited on page 20.

[BBM02] Sugato Basu, Arindam Banerjee, and Raymond Mooney. Semi-supervised clustering by seeding. In *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*. Citeseer, 2002. Cited on page 119.

[BBM04] Sugato Basu, Mikhail Bilenko, and Raymond J Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68, 2004. Cited on page 119.

[BCC$^+$20] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *8th International Conference on Learning Representations, ICLR*, 2020. Cited on pages 3, 7, 16, 18, 27, 33, 34, 36, 37, 46, 47, 57, 59, 60, 61, 66, 69, 70, 74, 78, 81, 84, 85, 86, 91, 93, 108, 112, 113, and 115.

[BCG$^+$19] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019. Cited on pages 3, 7, 16, 27, 33, 34, 35, 36, 37, 57, 66, 74, 78, 81, 84, 85, 91, 93, 108, 112, 115, and 145.

[BDK$^+$23] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. Cited on page 1.

[BDLR06] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 11 label propagation and quadratic criterion, 2006. Cited on pages 34 and 79.

[Bel66] Richard Bellman. Dynamic programming. *Science*, 153(3731), 1966. Cited on page 31.

[BGD$^+$22] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *NeurIPS*, 2022. Cited on page 125.

[BGVG14] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, 2014. Cited on page 86.

[BHB19] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, 2019. Cited on page 27.

[BMM18] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 2018. Cited on pages 7 and 77.

[BMR$^+$20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. Cited on page 1.

[BPL22] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. Cited on page 34.

[BRS$^+$22] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *International Conference on Learning Representations*, 2022. Cited on pages 3, 18, 50, 58, 59, 60, 64, 93, 112, and 113.

[BWA$^+$19] Murali Karthick Baskar, Shinji Watanabe, Ramon Astudillo, Takaaki Hori, Lukáš Burget, and Jan Černocký. Semi-supervised sequence-to-sequence asr using unpaired speech and text. *arXiv preprint arXiv:1905.01152*, 2019. Cited on pages 108 and 119.

[BZMA20] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020. Cited on page 113.

[CBHK02] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 2002. Cited on pages 7 and 77.

[CBL21] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. *arXiv preprint arXiv:2102.03526*, 2021. Cited on pages 21 and 93.

[CCTS20] Jeff Calder, Brendan Cook, Matthew Thorpe, and Dejan Slepcev. Poisson learning: Graph based semi-supervised learning at very low label rates. In *International Conference on Machine Learning*, pages 1306–1316. PMLR, 2020. Cited on page 17.

[CFG15] Douglas O Cardoso, Felipe França, and Joao Gama. A bounded neural network for open set recognition. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2015. Cited on page 21.

[CH21] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. Cited on pages 34 and 35.

[Che23] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint*, 2023. Cited on pages 128 and 131.

[CHL$^+$24] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. Cited on page 1.

[CJL$^+$19] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. Cited on pages 19, 74, and 83.

[CJW$^+$22] Baixu Chen, Junguang Jiang, Ximei Wang, Jianmin Wang, and Mingsheng Long. Debiased pseudo labeling in self-training. *arXiv preprint arXiv:2202.07136*, 2022. Cited on page 74.

[CKNH20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. Cited on pages 16, 18, 31, 33, 34, and 136.

[CKS$^+$20a] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. Cited on pages 18, 27, 31, 33, and 34.

[CKS$^+$20b] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020. Cited on page 108.

[CKUF17] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *CVPR*, 2017. Cited on page 125.

[CLH17] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017. Cited on page 85.

[CLML18] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc Le. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, 2018. Cited on page 119.

[CLP22] Léo Cances, Etienne Labbé, and Thomas Pellegrini. Comparison of semi-supervised deep learning algorithms for audio classification. *EURASIP Journal on Audio, Speech, and Music Processing*, 2022(1):1–16, 2022. Cited on pages 108 and 111.

[CMGS10] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 2010. Cited on page 85.

[CMK$^+$14] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. Cited on page 99.

[CMM$^+$20] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 2020. Cited on page 34.

[CMS$^+$20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. Cited on page 123.

[CMS$^+$22] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. Cited on page 123.

[CNL11a] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011. Cited on pages 22 and 35.

[CNL11b] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. Cited on pages 56, 70, and 110.

[Con18] MMCV Contributors. MMCV: OpenMMLab computer vision foundation. https://github.com/open-mmlab/mmcv, 2018. Cited on page 117.

[CRLL20] Luoxin Chen, Weitong Ruan, Xinyue Liu, and Jianhua Lu. Seqvat: Virtual adversarial training for semi-supervised sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8801–8811, 2020. Cited on page 119.

[CRRS08] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: Dataless classification. In *Aaai*, volume 2, pages 830–835, 2008. Cited on page 111.

[CSK21] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 2021. Cited on page 123.

[CSL$^+$21] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint*, 2021. Cited on pages 8, 23, 123, and 125.

[CSL$^+$22] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey E Hinton. A unified sequence interface for vision tasks. *NeurIPS*, 2022. Cited on pages 23, 123, and 125.

[CSZ06] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. The MIT Press, 2006. Cited on pages 50 and 63.

[CSZ09] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3), 2009. Cited on pages 4, 34, 79, and 112.

[CTF$^+$23] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. In *International Conference on Learning Representations*, 2023. Cited on pages 4, 10, 12, and 63.

[CTM$^+$21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. Cited on page 136.

[CW16a] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, 2016. Cited on page 35.

[CW16b] Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016. Cited on page 35.

[CWC$^+$22] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. Cited on page 3.

[CWG$^+$19] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019. Cited on pages 46, 78, 83, 84, and 86.

[CWKM20] Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. Self-training avoids using spurious features under domain shift. *Advances in Neural Information Processing Systems*, 33:21061–21071, 2020. Cited on page 60.

[CY21] Jiaao Chen and Diyi Yang. Simple conversational data augmentation for semi-supervised abstractive dialogue summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6605–6616, 2021. Cited on page 119.

[CYY20] Jiaao Chen, Zichao Yang, and Diyi Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, 2020. Cited on pages 108, 110, and 111.

[CYZW21] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021. Cited on page 63.

[CZL$^+$21] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 715–724, 2021. Cited on page 20.

[CZLG20] Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. Semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. Cited on pages 7, 20, 22, 92, 93, and 95.

[CZSL20] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE conference on computer vision and pattern recognition Workshops*, 2020. Cited on pages 33, 53, 74, 85, and 99.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. Cited on pages 109, 113, and 115.

[DCLT18a] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Cited on pages 2, 8, 23, 71, and 123.

[DCLT18b] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Cited on pages 108, 113, and 114.

[DDM+23] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023. Cited on page 1.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, 2009. Cited on pages 22, 37, 56, 70, 85, and 99.

[DG22] Shasvat Desai and Debasmita Ghose. Active learning for improved semi-supervised semantic segmentation in satellite images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 553–563, 2022. Cited on page 3.

[DGF16a] Emily Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430*, 2016. Cited on pages 34 and 79.

[DGF16b] Emily Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430*, 2016. Cited on pages 112 and 119.

[DGZ18] Qi Dong, Shaogang Gong, and Xiatian Zhu. Imbalanced deep learning by minority class incremental rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1367–1381, 2018. Cited on page 20.

[DJP+24] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. Cited on page 1.

[DL15] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28, 2015. Cited on page 119.

[DL19] Jinhao Dong and Tong Lin. Margingan: Adversarial training in semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. Cited on page 16.

[DT17] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. Cited on pages 33 and 85.

[DWF$^+$21] Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 402–411, 2021. Cited on page 119.

[DXX18] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888. IEEE, 2018. Cited on page 49.

[DYY$^+$17] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems*, 30, 2017. Cited on pages 16, 49, and 112.

[DZL$^+$17] Zhijie Deng, Hao Zhang, Xiaodan Liang, Luona Yang, Shizhen Xu, Jun Zhu, and Eric P Xing. Structured generative adversarial networks. *Advances in neural information processing systems*, 30, 2017. Cited on page 17.

[EADvdH17] M Ehsan Abbasnejad, Anthony Dick, and Anton van den Hengel. Infinite variational autoencoder for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5888–5897, 2017. Cited on page 17.

[EVGW$^+$10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 2010. Cited on page 77.

[FDKS22] Yue Fan, Dengxin Dai, Anna Kukleva, and Bernt Schiele. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. Cited on pages 45, 46, and 47.

[FDS22] Yue Fan, Dengxin Dai, and Bernt Schiele. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. Cited on pages 11, 12, 70, 77, 118, and 119.

[FKDS23] Yue Fan, Anna Kukleva, Dengxin Dai, and Bernt Schiele. Ssb: Simple but strong baseline for boosting performance of open-set semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16068–16078, 2023. Cited on pages 11, 12, and 91.

[FKS21a] Yue Fan, Anna Kukleva, and Bernt Schiele. Revisiting consistency regularization for semi-supervised learning. In *Pattern Recognition: 43nd DAGM German Conference, DAGM GCPR 2021*, 2021. Cited on pages 10 and 29.

[FKS21b] Yue Fan, Anna Kukleva, and Bernt Schiele. Revisiting consistency regularization for semi-supervised learning. In *DAGM German Conference on Pattern Recognition*, 2021. Cited on pages 10, 12, 27, 63, 74, 91, 112, and 113.

[FOS20] Geoff French, Avital Oliver, and Tim Salimans. Milking cowmask for semi-supervised image classification. *CoRR*, abs/2003.12022, 2020. Cited on page 27.

[FPE$^+$19] Eduardo Fonseca, Manoj Plakal, Daniel PW Ellis, Frederic Font, Xavier Favory, and Xavier Serra. Learning sound event classifiers from web audio with noisy labels. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25. IEEE, 2019. Cited on page 111.

[Fri37] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937. Cited on page 113.

[Fri40] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940. Cited on page 113.

[FXZ+24] Yue Fan, Yongqin Xian, Xiaohua Zhai, Alexander Kolesnikov, Muhammad Ferjad Naeem, Bernt Schiele, and Federico Tombari. Toward a diffusion-based generalist for dense vision tasks. In *Proceedings of the IEEE conference on computer vision and pattern recognition Workshops*, 2024. Cited on pages 12 and 123.

[FZD+20] Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural networks for semi-supervised learning on graphs. *Advances in neural information processing systems*, 33:22092–22103, 2020. Cited on page 119.

[GB04] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004. Cited on pages 61 and 74.

[GB05] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, 2005. Cited on pages 16, 96, and 98.

[GCB04] Nizar Grira, Michel Crucianu, and Nozha Boujemaa. Unsupervised and semi-supervised clustering: a brief survey. *A review of machine learning techniques for processing multimedia content*, 1:9–16, 2004. Cited on page 119.

[GDCG17] ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*, 2017. Cited on page 21.

[GHP+07] Gregory Griffin, Alex Holub, Pietro Perona, et al. Caltech-256 object category dataset. Technical report, Technical Report 7694, California Institute of Technology Pasadena, 2007. Cited on page 99.

[GIF+24] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024. Cited on page 1.

[GL22] Lan-Zhe Guo and Yu-Feng Li. Class-imbalanced semi-supervised learning with adaptive thresholding. In *International Conference on Machine Learning*, pages 8082–8094. PMLR, 2022. Cited on pages 19, 50, and 60.

[GLT+15] Chen Gong, Tongliang Liu, Dacheng Tao, Keren Fu, Enmei Tu, and Jie Yang. Deformed graph laplacian for semisupervised learning. *IEEE transactions on neural networks and learning systems*, 26(10):2261–2274, 2015. Cited on page 17.

[GPS+23] Yulu Gan, Sungwoo Park, Alexander Schubert, Anthony Philippakis, and Ahmed M Alaa. Instructcv: Instruction-tuned text-to-image diffusion models as vision generalists. *arXiv preprint*, 2023. Cited on pages 23 and 127.

[GPSW17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017. Cited on pages 51 and 53.

[GQC+20] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *InterSpeech*, 2020. Cited on page 107.

[GSA+20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 2020. Cited on page 34.

[GSK18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. Cited on pages 33 and 34.

[GTM+16] Chen Gong, Dacheng Tao, Stephen J Maybank, Wei Liu, Guoliang Kang, and Jie Yang. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, 25(7):3249–3260, 2016. Cited on page 49.

[GWD+19] Jiyang Gao, Jiang Wang, Shengyang Dai, Li-Jia Li, and Ram Nevatia. Note-rcnn: Noise tolerant ensemble rcnn for semi-supervised object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9508–9517, 2019. Cited on page 119.

[GWL21] Chengyue Gong, Dilin Wang, and Qiang Liu. Alphamatch: Improving consistency for semi-supervised learning with alpha-divergence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. Cited on pages 16 and 34.

[GYH+24] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12709–12720, 2024. Cited on page 23.

[GZH+16] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, 2016. Cited on page 77.

[GZJ+20a] Lan-Zhe Guo, Zhen-Yu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. Safe deep semi-supervised learning for unseen-class unlabeled data. In *International Conference on Machine Learning*, 2020. Cited on pages 7, 20, 22, 92, 93, and 95.

[GZJ+20b] Lan-Zhe Guo, Zhen-Yu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. Safe deep semi-supervised learning for unseen-class unlabeled data. In *International Conference on Machine Learning*, pages 3897–3906. PMLR, 2020. Cited on pages 118 and 119.

[GZQ+22] Maoguo Gong, Hui Zhou, AK Qin, Wenfeng Liu, and Zhongying Zhao. Self-paced co-training of graph neural networks for semi-supervised node classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. Cited on page 119.

[HA04] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 2004. Cited on page 92.

[HAE16] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. Cited on page 85.

[HBDB18] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 204–207. IEEE, 2018. Cited on page 110.

[HBDB19] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. Cited on page 110.

[HBT+21] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021. Cited on pages 108 and 113.

[HCX$^+$21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. Cited on pages 108 and 114.

[HFC$^+$21] Junkai Huang, Chaowei Fang, Weikai Chen, Zhenhua Chai, Xiaolin Wei, Pengxu Wei, Liang Lin, and Guanbin Li. Trash to treasure: Harvesting ood data with cross-modal matching for open-set semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. Cited on pages 6, 7, 20, 22, 92, 93, 95, 97, 103, and 105.

[HFW$^+$20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. Cited on pages 18, 34, and 136.

[HG09] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 2009. Cited on pages 7 and 77.

[HG16] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2016. Cited on page 99.

[HGSR19] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*, 2019. Cited on page 119.

[HHC$^+$21] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. Cited on pages 6, 19, 79, 87, and 147.

[HHLY22] Rundong He, Zhongyi Han, Xiankai Lu, and Yilong Yin. Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. Cited on pages 7, 20, 22, 92, 93, and 95.

[HHYY22] Rundong He, Zhongyi Han, Yang Yang, and Yilong Yin. Not all parameters should be treated equally: Deep safe semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. Cited on pages 7, 20, 22, 92, 93, and 95.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. Cited on pages 9 and 131.

[HJK20] Minsung Hyun, Jisoo Jeong, and Nojun Kwak. Class-imbalanced semi-supervised learning. *arXiv preprint arXiv:2002.06815*, 2020. Cited on pages 19, 35, 79, 118, and 119.

[HJT$^+$20] Xinting Hu, Yi Jiang, Kaihua Tang, Jingyuan Chen, Chunyan Miao, and Hanwang Zhang. Learning to segment the tail. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14045–14054, 2020. Cited on page 20.

[HKY$^+$21] Ju He, Adam Kortylewski, Shaokang Yang, Shuai Liu, Cheng Yang, Changhu Wang, and Alan Yuille. Rethinking re-sampling in imbalanced semi-supervised learning. *arXiv preprint arXiv:2106.00209*, 2021. Cited on pages 19, 35, and 80.

[HLLT16] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. Cited on pages 7, 19, and 77.

[HM13] Haibo He and Yunqian Ma. *Imbalanced learning: foundations, algorithms, and applications*. Wiley-IEEE Press, 2013. Cited on pages 7 and 77.

[HRCH21] Ting-I Hsieh, Esther Robb, Hwann-Tzong Chen, and Jia-Bin Huang. Droploss for long-tail instance segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 1549–1557, 2021. Cited on page 19.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. Cited on page 107.

[HSF18] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *International conference on learning representations*, 2018. Cited on page 35.

[HSX$^+$24] Haichuan Hu, Ye Shang, Guolin Xu, Congqing He, and Quanjun Zhang. Can gpt-o1 kill all bugs? *arXiv preprint arXiv:2409.10033*, 2024. Cited on page 1.

[HWC$^+$22] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 2022. Cited on page 115.

[HWH$^+$21] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. *Advances in Neural Information Processing Systems*, 34:22106–22118, 2021. Cited on pages 67 and 74.

[HWW21] Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. Distilling virtual examples for long-tailed recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 235–244, 2021. Cited on page 20.

[HXH$^+$21] Zhuo Huang, Chao Xue, Bo Han, Jian Yang, and Chen Gong. Universal semi-supervised learning. *Advances in Neural Information Processing Systems*, 34, 2021. Cited on pages 118 and 119.

[HYG22] Zhuo Huang, Jian Yang, and Chen Gong. They are not completely useless: Towards recycling transferable unlabeled data for class-mismatched semi-supervised learning. *IEEE Transactions on Multimedia*, 2022. Cited on pages 7, 20, 22, 92, 93, and 95.

[HZRS16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. Cited on pages 37, 85, 87, and 99.

[HZRS16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. Cited on pages 49, 56, 70, 107, 109, and 113.

[IPG$^+$18] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. Cited on page 16.

[ITAC19a] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019. Cited on pages 36 and 37.

[ITAC19b] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, 2019. Cited on page 74.

[Jap00] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proceedings of the International Conference on Artificial Intelligence*, 2000. Cited on pages 46, 84, and 86.

[JB91] David MacKay John Bridle, Anthony Heading. Unsupervised classifiers, mutual information and ′phantom targets. *Advances in neural information processing systems*, 1991. Cited on page 51.

[JEP+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021. Cited on page 1.

[Joa03] Thorsten Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003. Cited on pages 34 and 79.

[JST+20] Tom Joy, Sebastian M Schmon, Philip HS Torr, N Siddharth, and Tom Rainforth. Rethinking semi-supervised learning in vaes. *arXiv preprint arXiv:2006.10102*, 2020. Cited on page 17.

[JXE18] Neal Jean, Sang Michael Xie, and Stefano Ermon. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. *Advances in Neural Information Processing Systems*, 31, 2018. Cited on page 119.

[JZL+19] Bo Jiang, Ziyan Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-supervised learning with graph learning-convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11313–11320, 2019. Cited on page 17.

[KB15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International conference on learning representations*, 2015. Cited on pages 45, 71, 85, and 128.

[KDGBA19] Hoel Kervadec, Jose Dolz, Éric Granger, and Ismail Ben Ayed. Curriculum semi-supervised segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 568–576. Springer, 2019. Cited on page 49.

[KH+09a] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. Technical report. Cited on pages 6, 21, 35, 44, and 99.

[KH+09b] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. Technical report. Cited on pages 56, 70, and 110.

[KHP+20a] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems*, 33:14567–14579, 2020. Cited on pages 70, 118, and 119.

[KHP+20b] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, SungJu Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *Advances in neural information processing systems*, 2020. Cited on pages 7, 19, 35, 44, 45, 46, 47, 77, 78, 80, 82, 83, 84, 85, 86, and 146.

[KJS20] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13896–13905, 2020. Cited on page 19.

[KJYFF11] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, 2011. Cited on page 99.

[KKKR18] Georgios Kostopoulos, Stamatis Karlos, Sotiris Kotsiantis, and Omiros Ragos. Semi-supervised regression: A recent review. *Journal of Intelligent & Fuzzy Systems*, 35(2):1483–1500, 2018. Cited on page 119.

[KMHK20] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Featmatch: Feature-based augmentation for semi-supervised learning. In *Computer Vision – ECCV 2020*, pages 1–19, 2020. Cited on pages 16, 27, 29, 34, 36, 37, and 48.

[KMJRW14] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014. Cited on pages 17 and 119.

[KMK$^+$22] Jiwon Kim, Youngjo Min, Daehwan Kim, Gyuseong Lee, Junyoung Seo, Kwangrok Ryoo, and Seungryong Kim. Conmatch: Semi-supervised learning with confidence-guided consistency regularization. In *European Conference on Computer Vision*, 2022. Cited on pages 67 and 74.

[KML20] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pages 5468–5479. PMLR, 2020. Cited on page 60.

[KMRW14] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, 2014. Cited on pages 34 and 79.

[KSF17] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. *Advances in neural information processing systems*, 30, 2017. Cited on pages 16 and 112.

[KSPB$^+$22] Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. *NeurIPS*, 2022. Cited on pages 23, 125, and 127.

[KW52] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 1952. Cited on page 100.

[KWY$^+$19] Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson WH Lau. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6728–6736, 2019. Cited on page 16.

[KXR$^+$20] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International conference on learning representations*, 2020. Cited on pages 20, 22, 46, 78, 79, 81, 82, and 84.

[KZG$^+$17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 2017. Cited on page 77.

[LA17] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR*, 2017. Cited on pages 16, 27, 29, 30, 34, 48, 91, and 93.

[LBT$^+$18] Renjie Liao, Marc Brockschmidt, Daniel Tarlow, Alexander L Gaunt, Raquel Urtasun, and Richard Zemel. Graph partition neural networks for semi-supervised classification. *arXiv preprint arXiv:1803.06272*, 2018. Cited on page 17.

[LCG$^+$21] Huixiang Luo, Hao Cheng, Yuting Gao, Ke Li, Mengdan Zhang, Fanxu Meng, Xiaowei Guo, Feiyue Huang, and Xing Sun. On the consistency training for open-set semi-supervised learning. *arXiv preprint arXiv:2101.08237*, 3(6), 2021. Cited on pages 118 and 119.

[LCM+21] Huixiang Luo, Hao Cheng, Fanxu Meng, Yuting Gao, Ke Li, Mengdan Zhang, and Xing Sun. An empirical study and analysis on open-set semi-supervised learning. *arXiv preprint arXiv:2101.08237*, 2021. Cited on page 20.

[LCZ+22] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *ICLR*, 2022. Cited on pages 8, 23, 123, 125, and 127.

[Lee13] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. Cited on pages 4, 16, 27, 34, 49, 57, 63, 65, 66, 74, 78, 79, 91, 93, and 112.

[LH17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. Cited on page 71.

[LHW18] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. Cited on page 17.

[LLCD19] Qing Lian, Wen Li, Lin Chen, and Lixin Duan. Known-class aware self-ensemble for open set domain adaptation. *arXiv preprint arXiv:1905.01068*, 2019. Cited on page 21.

[LLK+21] Bo Liu, Haoxiang Li, Hao Kang, Gang Hua, and Nuno Vasconcelos. Gistnet: a geometric structure transfer network for long-tailed recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8209–8218, 2021. Cited on page 19.

[LLO21] Changchun Li, Ximing Li, and Jihong Ouyang. Semi-supervised text classification with balanced deep representation distributions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5044–5053, 2021. Cited on page 111.

[LLW19] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8577–8584, 2019. Cited on page 19.

[LLWL24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. Cited on page 1.

[LM05] Wei Li and Andrew McCallum. Semi-supervised sequence modeling with syntactic topic models. In *AAAI*, volume 5, pages 813–818, 2005. Cited on page 119.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, 2014. Cited on pages 27, 77, and 128.

[LMH+20] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *International Conference on Learning Representations*, 2020. Cited on page 119.

[LMW+22] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. Cited on pages 127 and 128.

[LMZ+19] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Cited on pages 20 and 77.

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. Cited on page 108.

[LPW+19] Yang Li, Quan Pan, Suhang Wang, Haiyun Peng, Tao Yang, and Erik Cambria. Disentangled variational auto-encoder for semi-supervised learning. *Information Sciences*, 482:73–85, 2019. Cited on page 17.

[LSH20a] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020. Cited on page 108.

[LSH+20b] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2970–2979, 2020. Cited on page 20.

[LSK21] Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *NeurIPS*, 2021. Cited on pages 19, 22, 35, 70, and 80.

[LWG+22] Zhengfeng Lai, Chao Wang, Henrry Gunawan, Sen-Ching S Cheung, and Chen-Nee Chuah. Smoothed adaptive weighting for imbalanced semi-supervised learning: Improve reliability against unknown distribution data. In *International Conference on Machine Learning*, pages 11828–11843. PMLR, 2022. Cited on page 74.

[LWHL19] Bin Liu, Zhirong Wu, Han Hu, and Stephen Lin. Deep metric transfer for label propagation with limited annotated data. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. Cited on pages 34 and 79.

[LWK+20] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. Cited on pages 20, 78, 79, and 81.

[LWLJ22] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint*, 2022. Cited on page 123.

[LWO22] Ao Liu, An Wang, and Naoaki Okazaki. Semi-supervised formality style transfer with consistency training. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4689–4701, 2022. Cited on page 119.

[LWZ08] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2008. Cited on page 19.

[LWZL11] Shoushan Li, Zhongqing Wang, Guodong Zhou, and Sophia Yat Mei Lee. Semi-supervised learning for imbalanced sentiment classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011. Cited on pages 118 and 119.

[LXH21] Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. Cited on pages 3, 18, 74, 93, and 112.

[LZ15] Yu-Feng Li and Zhi-Hua Zhou. Towards making unlabeled data never hurt. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):175–188, 2015. Cited on page 117.

[LZL+18] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. Cited on pages 34 and 79.

[LZZ17] Yu-Feng Li, Han-Wen Zha, and Zhi-Hua Zhou. Learning safe prediction for semi-supervised regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. Cited on page 119.

[McL75] Geoffrey J McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369, 1975. Cited on page 49.

[MDP$^+$11] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011. Cited on pages 71 and 111.

[MGR$^+$18] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018. Cited on page 19.

[MJR$^+$21] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International conference on learning representations*, 2021. Cited on pages 78, 79, and 81.

[ML13] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172, 2013. Cited on page 111.

[MMKI18a] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. Cited on pages 16, 57, 74, 110, and 112.

[MMKI18b] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8), 2018. Cited on pages 27, 91, and 93.

[MR05] Oded Maimon and Lior Rokach. *Data mining and knowledge discovery handbook*. Springer, 2005. Cited on page 92.

[MSSW16] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. In *International conference on machine learning*, pages 1445–1453. PMLR, 2016. Cited on page 17.

[Nes83] Yu Nesterov. A method of solving a convex programming problem with convergence rate $o(k^2)$. *Doklady Akademii Nauk*, 1983. Cited on pages 34, 78, and 79.

[NH10] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. Cited on page 100.

[Nob06] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006. Cited on page 117.

[NOF$^+$18] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 613–628, 2018. Cited on page 21.

[NVNL$^+$23] Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *arXiv preprint arXiv:2309.09400*, 2023. Cited on page 1.

[NWC+11a] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 2011. Cited on pages 56, 70, and 110.

[NWC+11b] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *In NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. Cited on pages 22, 35, and 99.

[NZ08] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008. Cited on page 99.

[Ode16a] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016. Cited on pages 34 and 79.

[Ode16b] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016. Cited on page 119.

[OHT20] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020. Cited on pages 108 and 112.

[OKK21a] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. *arXiv preprint arXiv:2106.05682*, 2021. Cited on pages 19, 35, 45, and 80.

[OKK21b] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. *arXiv preprint arXiv:2106.05682*, 2021. Cited on pages 118 and 119.

[OOR+18] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in neural information processing systems*, 2018. Cited on pages 5, 6, 20, 22, 33, 45, 46, 56, 63, 69, 84, 85, 91, 93, 108, 110, 115, and 146.

[OP19a] Poojan Oza and Vishal M Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2307–2316, 2019. Cited on page 21.

[OP19b] Poojan Oza and Vishal M Patel. Deep cnn-based multi-task learning for open-set recognition. *arXiv preprint arXiv:1903.03161*, 2019. Cited on page 21.

[PBS+20] Junran Peng, Xingyuan Bu, Ming Sun, Zhaoxiang Zhang, Tieniu Tan, and Junjie Yan. Large-scale object detection in the wild from imbalanced multi-labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9709–9718, 2020. Cited on page 19.

[PDXL21] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568, 2021. Cited on pages 50, 57, 63, and 108.

[Pic15] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press, 2015. Cited on page 111.

[PVDMD+17] Brooks Paige, Jan-Willem Van De Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, Philip Torr, et al. Learning disentangled representations with semi-supervised deep generative models. *Advances in neural information processing systems*, 30, 2017. Cited on page 17.

[PXDL20] Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V Le. Meta pseudo labels. *arXiv preprint arXiv:2003.10580*, 2020. Cited on pages 16, 27, 34, and 93.

[PYJS22] Jongjin Park, Sukmin Yun, Jongheon Jeong, and Jinwoo Shin. Opencos: Contrastive semi-supervised learning for handling open-set unlabeled data. In *European conference on computer vision Workshop*, 2022. Cited on pages 7, 20, 22, and 93.

[QL20] Guo-Jun Qi and Jiebo Luo. Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. Cited on page 108.

[RBH$^+$15a] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, 2015. Cited on pages 34, 78, and 79.

[RBH$^+$15b] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. *Advances in Neural Information Processing Systems*, 28:3546–3554, 2015. Cited on pages 57, 109, and 112.

[RBL$^+$22a] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. Cited on page 1.

[RBL$^+$22b] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. Cited on pages 9, 124, 127, 128, and 129.

[RD17] T-YLPG Ross and GKHP Dollár. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2980–2988, 2017. Cited on page 19.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. Cited on page 107.

[RDRS20] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2020. Cited on page 49.

[RDS$^+$15a] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 2015. Cited on page 27.

[RDS$^+$15b] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. Cited on pages 107 and 110.

[RDS$^+$15c] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. Cited on page 128.

[REH$^+$20] Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Semi-supervised learning with scarce annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020. Cited on pages 18, 33, and 34.

[RHS05a] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. *Semi-supervised self-training of object detection models*. Carnegie Mellon University, 2005. Cited on page 49.

[RHS05b] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. *Semi-supervised self-training of object detection models*. Carnegie Mellon University, 2005. Cited on page 79.

[RKH+21a] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. Cited on page 1.

[RKH+21b] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICLR*, 2021. Cited on page 128.

[RKK+22] Mamshad Nayeem Rizve, Navid Kardan, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Openldn: Learning to discover novel classes for open-world semi-supervised learning. In *European Conference on Computer Vision*, 2022. Cited on pages 21 and 93.

[RKS22] Mamshad Nayeem Rizve, Navid Kardan, and Mubarak Shah. Towards realistic semi-supervised learning. In *European Conference on Computer Vision*, 2022. Cited on pages 21 and 93.

[RL17] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR*, 2017. Cited on pages 35 and 37.

[RSR+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020. Cited on pages 2, 8, 23, and 123.

[RVR18] YCAP Reddy, P Viswanath, and B Eswara Reddy. Semi-supervised learning: A brief review. *Int. J. Eng. Technol*, 7(1.8):81, 2018. Cited on page 108.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. Cited on pages 2, 8, 23, and 123.

[RYS20] Zhongzheng Ren, Raymond A. Yeh, and Alexander G. Schwing. Not all unlabeled data are equal: Learning to weight data in semi-supervised learning. In *Neural Information Processing Systems (NeurIPS)*, 2020. Cited on pages 67 and 74.

[SBL+20] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020. Cited on pages 3, 4, 7, 8, 16, 18, 20, 27, 29, 31, 33, 34, 35, 36, 37, 38, 39, 45, 46, 47, 48, 49, 50, 51, 55, 56, 57, 58, 60, 63, 64, 65, 66, 68, 70, 71, 72, 74, 78, 81, 84, 85, 86, 87, 88, 91, 93, 95, 100, 108, 110, 111, 112, 113, 115, 145, 147, and 148.

[SBV+22] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. Cited on page 1.

[Sch90] Robert E Schapire. The strength of weak learnability. *Machine learning*, 1990. Cited on page 93.

[Scu65] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 1965. Cited on pages 34, 78, and 79.

[SFH17] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, 2017. Cited on page 35.

[SGG16] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. Cited on page 19.

[SHK$^+$14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, pages 1929–1958, 2014. Cited on pages 4, 16, and 34.

[SHKF12] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. Cited on page 128.

[SJB14] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044, 2014. Cited on page 111.

[SJT16a] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in neural information processing systems*, 2016. Cited on pages 16, 27, 29, 30, 34, 48, 78, and 79.

[SJT16b] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171, 2016. Cited on pages 49 and 74.

[SKS21] Kuniaki Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set semi-supervised learning with open-set consistency regularization. *Advances in Neural Information Processing Systems*, 2021. Cited on pages 6, 7, 20, 22, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 103, 105, 118, 119, and 143.

[SLH16] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 467–482. Springer, 2016. Cited on page 19.

[SM21a] Jong-Chyi Su and Subhransu Maji. The semi-supervised inaturalist-aves challenge at fgvc7 workshop, 2021. Cited on pages 44 and 45.

[SM21b] Jong-Chyi Su and Subhransu Maji. The semi-supervised inaturalist-aves challenge at fgvc7 workshop. *arXiv preprint arXiv:2103.06937*, 2021. Cited on page 110.

[Spr15] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015. Cited on pages 16, 112, and 119.

[SS21] Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. Cited on pages 20 and 93.

[ST17] Laine Samuli and Aila Timo. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2017. Cited on pages 49, 66, and 74.

[Stu13] Bob L Sturm. The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use. *arXiv preprint arXiv:1306.1461*, 2013. Cited on page 111.

[SVB+21] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. Cited on page 5.

[SYUH18] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 153–168, 2018. Cited on page 21.

[SZC19] Aravind Sankar, Xinyang Zhang, and Kevin Chen-Chuan Chang. Meta-gnn: Metagraph neural network for semi-supervised learning in attributed heterogeneous information networks. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 137–144, 2019. Cited on page 119.

[TBV21] Kai Sheng Tai, Peter Bailis, and Gregory Valiant. Sinkhorn label allocation: Semi-supervised classification via annealed self-training, 2021. Cited on page 74.

[TBW+24] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024. Cited on page 1.

[TCD+21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. Cited on page 115.

[TCLZ21] Yihe Tang, Weifeng Chen, Yijun Luo, and Yuting Zhang. Humble teachers teach better students for semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3132–3141, 2021. Cited on page 119.

[THZ20] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *Advances in Neural Information Processing Systems*, 2020. Cited on pages 78, 79, and 81.

[TL19] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. Cited on page 16.

[TLI+23] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. Cited on pages 2, 8, 23, and 123.

[TLZ+21] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1685–1694, 2021. Cited on page 19.

[TRW+21] Peng Tang, Chetan Ramaiah, Yan Wang, Ran Xu, and Caiming Xiong. Proposal learning for semi-supervised object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2291–2301, 2021. Cited on page 119.

[TSF+16] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. Cited on page 3.

[TV17a] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, 2017. Cited on pages 16, 27, 34, 36, 91, and 93.

[TV17b] Antti Tarvainen and Harri 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*, pages 1195–1204, 2017. Cited on pages 57, 66, 68, 74, 110, and 112.

[TWG+16] Yuxing Tang, Josiah Wang, Boyang Gao, Emmanuel Dellandréa, Robert Gaizauskas, and Liming Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2119–2128, 2016. Cited on page 119.

[TWL+20] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020. Cited on page 19.

[VAD+22] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022. Cited on page 1.

[Vap98] V Vapnik. Statistical learning theory. *NY: Wiley*, 1:2, 1998. Cited on page 3.

[VdMH08] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. Cited on page 58.

[VDOV+17] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 2017. Cited on page 125.

[VEH20] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. Cited on pages 108 and 112.

[VHP17] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017. Cited on page 77.

[VLK+19] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*. International Joint Conferences on Artificial Intelligence Organization, 2019. Cited on pages 16, 27, and 34.

[VQK+21] Vikas Verma, Meng Qu, Kenji Kawaguchi, Alex Lamb, Yoshua Bengio, Juho Kannala, and Jian Tang. Graphmix: Improved training of gnns for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10024–10032, 2021. Cited on pages 107 and 119.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. Cited on pages 8, 49, 107, 109, 113, and 123.

[VYBT19] Shikhar Vashishth, Prateek Yadav, Manik Bhandari, and Partha Talukdar. Confidence-based graph convolutional networks for semi-supervised learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1792–1801. PMLR, 2019. Cited on page 17.

[WBH19] Hongzhi Wang, Mohamed Jaward Bah, and Mohamed Hammad. Progress in outlier detection techniques: A survey. *Ieee Access*, 2019. Cited on page 92.

[WC19] Maurice Weiler and Gabriele Cesa. General e(2)-equivariant steerable cnns. In *Advances in Neural Information Processing Systems*, 2019. Cited on page 35.

[WCB⁺22] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, 2022. Cited on page 123.

[WCF⁺22a] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Wei Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki, Bernt Schiele, Jindong Wang, Xing Xie, and Yue Zhang. Usb: A unified semi-supervised learning benchmark for classification. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. Cited on pages 70, 91, and 107.

[WCF⁺22b] Yidong Wang, Hao Chen, Yue Fan, SUN Wang, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, et al. Usb: A unified semi-supervised learning benchmark for classification. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. Cited on pages 11 and 12.

[WCH⁺17] Jindong Wang, Yiqiang Chen, Shuji Hao, Wenjie Feng, and Zhiqi Shen. Balanced distribution adaptation for transfer learning. In *2017 IEEE international conference on data mining (ICDM)*, pages 1129–1134. IEEE, 2017. Cited on page 119.

[WCH⁺23] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. In *International Conference on Learning Representations*, 2023. Cited on pages 4, 10, 12, and 49.

[WD18] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. Cited on page 6.

[WFC⁺18] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 402–410, 2018. Cited on page 119.

[WFN⁺24] Haiyang Wang, Yue Fan, Muhammad Ferjad Naeem, Liwei Wang, Yongqin Xian, Jan Eric Lenssen, and Bernt and Tombari, Federico Schiele. Tokenformer: Rethinking transformer scaling with tokenized model parameters. *under review at ICLR*, 2024. Cited on pages 13 and 137.

[WGY⁺19] Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5017–5026, 2019. Cited on page 20.

[WHL⁺20] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 162–178. Springer, 2020. Cited on page 19.

[Wil72] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 408–421, 1972. Cited on page 19.

[WL07] Larry Wasserman and John Lafferty. Statistical analysis of semi-supervised regression. *Advances in Neural Information Processing Systems*, 20, 2007. Cited on page 119.

[WLK$^+$20] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *European Conference on computer vision*, 2020. Cited on pages 20, 22, 78, 79, and 81.

[WLL$^+$22] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022. Cited on page 119.

[WLM$^+$20] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020. Cited on page 20.

[WRH17] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. *Advances in neural information processing systems*, 30, 2017. Cited on page 20.

[WSKH24] Erik Wallin, Lennart Svensson, Fredrik Kahl, and Lars Hammarstrand. Improving open-set semi-supervised learning with self-supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2356–2365, 2024. Cited on page 21.

[WSM$^+$19] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*, 2019. Cited on page 107.

[WSM$^+$21] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan 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. Cited on pages 19, 35, 44, 45, 46, 47, 70, 77, 80, 82, 83, 84, 85, 86, 118, and 119.

[WSW$^+$20] Jialian Wu, Liangchen Song, Tiancai Wang, Qian Zhang, and Junsong Yuan. Forest r-cnn: Large-vocabulary long-tailed object detection and instance segmentation. In *Proceedings of the 28th ACM international conference on multimedia*, pages 1570–1578, 2020. Cited on page 20.

[WWC$^+$23] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, 2023. Cited on pages 23, 123, 125, 126, 127, 129, and 148.

[WWL$^+$22] Yidong Wang, Hao Wu, Ao Liu, Wenxin Hou, Zhen Wu, Jindong Wang, Takahiro Shinozaki, Manabu Okumura, and Yue Zhang. Exploiting unlabeled data for target-oriented opinion words extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, 2022. Cited on page 119.

[WWLY22] Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debiased learning from naturally imbalanced pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14647–14657, 2022. Cited on page 74.

[WWYL18] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *arXiv preprint*, 2018. Cited on page 128.

[WYM$^+$22] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *ICML*, 2022. Cited on pages 9, 23, and 125.

[WZC+23] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Segmenting everything in context. *arXiv preprint*, 2023. Cited on pages 23, 123, and 125.

[WZH+22] Yidong Wang, Bowen Zhang, Wenxin Hou, Zhen Wu, Jindong Wang, and Takahiro Shinozaki. Margin calibration for long-tailed visual recognition. In *The 14th Asian Conference on Machine Learning*, 2022. Cited on page 118.

[WZZ+21] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9695–9704, 2021. Cited on page 19.

[XCH+20] Chunyan Xu, Zhen Cui, Xiaobin Hong, Tong Zhang, Jian Yang, and Wei Liu. Graph inference learning for semi-supervised classification. *arXiv preprint arXiv:2001.06137*, 2020. Cited on page 17.

[XDH+19] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. Cited on pages 4, 16, 27, 29, 34, 36, 48, 78, 91, and 93.

[XDH20a] Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 247–263. Springer, 2020. Cited on page 20.

[XDH+20b] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33, 2020. Cited on pages 7, 50, 53, 57, 60, 63, 64, 65, 66, 71, 74, 108, 110, 111, 112, 113, 115, and 148.

[XLHL20a] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. Cited on pages 16 and 93.

[XLHL20b] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. Cited on page 49.

[XSY+21] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pages 11525–11536. PMLR, 2021. Cited on pages 7, 18, 50, 56, 57, 60, 63, 64, 65, 67, 74, 108, 110, and 112.

[XZH+21] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3060–3069, 2021. Cited on pages 63 and 119.

[YCC+21] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*, 2021. Cited on page 111.

[YD16] Dong Yu and Li Deng. *Automatic speech recognition*, volume 1. Springer, 2016. Cited on page 107.

[YHL$^+$19] Yang Yang, Chunping Hou, Yue Lang, Dai Guan, Danyang Huang, and Jinchen Xu. Open-set human activity recognition based on micro-doppler signatures. *Pattern Recognition*, 85:60–69, 2019. Cited on page 21.

[YIIA20a] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In *European Conference on Computer Vision*, 2020. Cited on pages 6, 7, 20, 22, 92, 93, 95, 97, 103, and 105.

[YIIA20b] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In *European Conference on Computer Vision*, pages 438–454. Springer, 2020. Cited on pages 118 and 119.

[YQLG17] Yang Yu, Wei-Yang Qu, Nan Li, and Zimin Guo. Open-category classification by adversarial sample generation. *arXiv preprint arXiv:1705.08722*, 2017. Cited on page 21.

[YSKX21] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *arXiv preprint arXiv:2103.00550*, 2021. Cited on page 112.

[YSN21] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021. Cited on page 110.

[YSW$^+$21] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification. *arXiv preprint arXiv:2110.14795*, 2021. Cited on page 110.

[YSZ$^+$15] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. Cited on page 99.

[YX20] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *NeurIPS*, 2020. Cited on pages 51, 118, and 119.

[YZC21] Han-Jia Ye, De-Chuan Zhan, and Wei-Lun Chao. Procrustean training for imbalanced deep learning. *ICCV*, 2021. Cited on pages 88 and 89.

[YZLL24] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, pages 1–28, 2024. Cited on page 5.

[ZAK$^+$22] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for fast image restoration and enhancement. *PAMI*, 2022. Cited on page 128.

[ZCDLP18] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR*, 2018. Cited on pages 16, 34, 57, 70, 88, and 93.

[ZCHC20] Xujiang Zhao, Feng Chen, Shu Hu, and Jin-Hee Cho. Uncertainty aware semi-supervised learning on graph data. *Advances in Neural Information Processing Systems*, 33:12827–12836, 2020. Cited on page 119.

[ZCHP23] Shaoyu Zhang, Chen Chen, Xiyuan Hu, and Silong Peng. Balanced knowledge distillation for long-tailed learning. *Neurocomputing*, 527:36–46, 2023. Cited on page 20.

[ZCLJ21] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16489–16498, 2021. Cited on pages 19, 20, and 22.

[ZCT$^+$21] Yan Zhao, Weicong Chen, Xu Tan, Kai Huang, Jin Xu, Changhu Wang, and Jihong Zhu. Improving long-tailed classification from instance level. *arXiv preprint arXiv:2104.06094*, 2021. Cited on page 20.

[ZCWC20] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, 2020. Cited on pages 20 and 79.

[ZDY$^+$23] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *CVPR*, 2023. Cited on pages 23 and 125.

[ZFW$^+$17] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE international conference on computer vision*, pages 5409–5418, 2017. Cited on page 20.

[ZG09a] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1), 2009. Cited on pages 34 and 79.

[ZG09b] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009. Cited on pages 49 and 108.

[ZHS05] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd international conference on Machine learning*, pages 1036–1043, 2005. Cited on page 17.

[Zhu05a] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. Cited on pages 34 and 79.

[Zhu05b] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005. Cited on pages 49 and 108.

[ZK16a] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016. Cited on pages 36, 37, 40, 45, 85, 88, 99, and 145.

[ZK16b] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016. Cited on pages 56 and 70.

[ZKH$^+$23] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10795–10816, 2023. Cited on page 5.

[ZKIC20] Xujiang Zhao, Killamsetty Krishnateja, Rishabh Iyer, and Feng Chen. Robust semi-supervised learning with out of distribution data. *arXiv preprint arXiv:2010.03658*, 2020. Cited on page 20.

[ZL$^+$05] Zhi-Hua Zhou, Ming Li, et al. Semi-supervised regression with co-training. In *IJCAI*, volume 5, pages 908–913, 2005. Cited on page 119.

[ZLY$^+$21] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021. Cited on pages 20 and 22.

[ZOKB19] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE international conference on computer vision*, 2019. Cited on pages 16, 18, 33, 34, and 108.

[ZPCU19] Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Ustebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5829–5836, 2019. Cited on page 17.

[ZWB20] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Time-consistent self-supervision for semi-supervised learning. In *International Conference on Machine Learning*, pages 11523–11533. PMLR, 2020. Cited on pages 56 and 70.

[ZWH+21] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 2021. Cited on pages 4, 6, 7, 18, 22, 29, 34, 36, 37, 38, 46, 47, 50, 56, 57, 58, 60, 63, 64, 65, 67, 69, 70, 71, 72, 73, 74, 91, 93, 108, 110, 111, 112, 113, 115, 117, 145, and 148.

[ZYH+22] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. Cited on pages 18, 74, 110, 112, and 113.

[ZYKW18] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018. Cited on pages 58 and 59.

[ZZL15] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657, 2015. Cited on pages 71 and 111.

[ZZL+22] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In *CVPR*, 2022. Cited on page 123.

[ZZP+17] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. Cited on page 128.